A Multi-Agent System Design and Implementation for Flexible Network Management

Minjiang Chen

A thesis submitted for the degree of Doctor of Philosophy
to the Department of Electronic and Electrical Engineering
University of Strathclyde

June 2018

This thesis is the result of the author’s original research. It has been composed by the author and has not been previously submitted for examination which has led to the award of a degree.

Abstract

With the introduction of renewable energy technologies to reduce greenhouse gas emissions, a significant amount of Distributed Generation (DG) has connected to distribution networks. Hence, the operation of electrical power distribution systems has become complicated and increasingly challenging due to the uncertainty of bi-directional power flow, voltage fluctuations, and frequency deviations. To help manage these issues, the system requires more intelligent functions and flexibility in order to support solutions for network management and operation. Moreover, distribution network operators are looking for an active approach to maximise the utilisation of network capacity while solving network issues for more DG connections. As a result, Active Network Management (ANM) has been proposed to facilitate DG connections without breaching network operation limits. This is achieved by managing network control functions in line with operational objectives and it is often seen as a way to avoiding the high costs of reinforcing the existing infrastructure. However, as the network continues to change, such as increasing DG connections, and control functions keep evolving overtime. ANM is considered to be part of the solution for at least the medium term. Therefore, ANM requires sufficient flexibility to adapt changes to its environment and extensibility to upgrade control functions overtime for future needs. A key aspect of this is interoperability to allow ANM to interact with different control devices, such as intelligent electronic devices, to collect data and realise control purposes as they also evolve.

Multi-agent Systems (MAS) is one of the most relevant technologies to address the above challenges as it provides autonomous and proactive behaviour in open and dynamic environments, which is analogous to new DG connections. In addition, MAS offers a flexible and extensible platform that is the advantage of other relevant technologies, such as a service-oriented architecture. Therefore, it is proposed as part of the work of this thesis that the requirement for flexibility and extensibility can be achieved through the development of control functions as intelligent agents with scalable capabilities brought about through the use of MAS technology. The novel solver agent developed as part of the work of this thesis is an essential component of the MAS architecture considered in this thesis incorporates an integrated control algorithm and negotiation capability to solve conflicts between various control solutions.

This thesis presents a fully integrated MAS architecture for ANM. It is developed by following a comprehensive design methodology and each stage of the proposed MAS architecture is detailed through specification to implementation. Selected control algorithms are developed as intelligent agents to achieve multiple ANM solutions. In order to provide a common understanding of terminology for agent communications, ontologies for power system control applications, including thermal overload and voltage violation, have been created. A novel IEC 61850 interface has been developed and embedded inside the agent to address interoperability issue between devices for data collection and control.

To evaluate the performance of the developed MAS architecture, along with the novel elements of this thesis a range of simulation case studies are explored. Studies are based on a closed-loop simulation environment with a power system simulator: one case study is based on an operational UK 11 kV radial distribution network to demonstrate the application of MAS for various power system controls; another examines the self-organising ability of the developed MAS architecture by using the ε decomposition algorithm for distributed voltage regulation. The results demonstrate the flexibility, extensibility, and self-organisation capabilities of the novel fully integrated MAS architecture. This is built upon and the prospects for the inclusion of the MAS technology within operational power systems is discussed and recommendations made as part of this thesis.

Acknowledgements

I would not have been able to complete my Ph.D without the generous help and support of numbers of people.

First of all, I would like to express my sincere thanks to my primary supervisor, Prof. Stephen McArthur, for his wise guidance, excellent support, and necessary challenges through the duration of the research. Also, my thanks go to my second supervisor, Dr. Ivana Kockar, for her professional advice and help.

Formal thanks go to the Engineering and Physical Research Council (EPSRC) and the HubNet research programme for providing financial support during my research years.

Special thanks to Prof. Jeremy Pitt, Dr. Didac Busquets-Font, Patricio Petruzzi from Imperial College, London, for providing me invaluable advice and support in the multi-agent programming of Presage2.

I also appreciate my colleagues, Michael Hosseini, Calum Mackinnon, and Anthony Florida-James, for their time to proofread my thesis.

I would like to acknowledge my friends for their constant help and support during my Ph.D research.

Last but not least, my greatest thanks go to my family and my wife Han Xu, for their unconditional support and encouragement over the past years. I thank Han for her patience, understanding, and always be there whenever good times and bad. Without her love and support, I would not have been possible to complete this work.

Contents

List of Figures

List of Tables

Glossary of Abbreviations

ADINE	Active DIstribution NEtwork
ADDRESS	Active Distribution network with full integration Demand and distribution energy RESourceS
ANM	Active Network Management
ACL	Agent Communication Language
ACE	Area Control Error
ANN	Artificial Neural Networks
AuRA-NMS	Autonomous Regional Active Network Management System
CBR	Case Based Reasoning
CHP	Combined Heat and Power
COMMAS	COndition Monitoring Multi-Agent System
CSP	Constraint Satisfaction Problem
CSPS	CSP Solver
CLNR	Customer-Led Network Revolution
DSM	Demand Side Management
DER	Distributed Energy Resources
DG	Distributed Generation
DGCG	Distributed Generation Coordination Group
DNO	Distributed Network Operator
EDLC	Electric Double-Layer Capacitors
EES	Electrical Energy Storage
ECS	Energy Capacitor System

FIPA	Foundation for Intelligent Physical Agents
GOOSE	Generic Object-Oriented Substation Event
GA	Genetic Algorithm
GUS	Grand Unified Scheme
GUI	Graphical User Interface
HMM	Hidden Markov Models
IED	Intelligent Electronic Device
KIF	Knowledge Interchange Format
KQML	Knowledge Query and Manipulation Language
LIFO	Last In First Off
LP	Linear Programming
LUA	Load Unit Agent
LCNF	Low Carbon Networks Fund
LPS	LP Solver
MPA	Market Player Agent
MCCA	Microgrid Central Controller Agent
MAS	Multi-Agent Systems
NVD	Network Violation Detection
NINES	Northern Isles New Energy Solutions
OLTC	On-Load Tap Changer
OPFS	OPF Solver
OPF	Optimal Power Flow
PSO	Particle Swarm Optimisation
PV	Photovoltaic
FPP	Flexible Plug and Play
PEV	Plug-in Electric Vehicle
PFC	Power Factor Control
PFM	Power Flow Management
PFSF	Power Flow Sensitivity Factor
PUA	Production Unit Agent
PID	Proportional Integral Derivative
PEDA	Protection Engineering Diagnostic Agent

RPZ	Registered Power Zone
RTU	Remote Terminal Unit
RES	Renewable Energy Sources
SV	Sampled-Value
SSE	Scottish & Southern Energy
SOA	Service-Oriented Architecture
SGS	Smarter Grid Solutions
SSSC	Static Synchronous Series Compensator
SCD	System Configuration Description
UK	United Kingdom
UPF	Unity Power Factor
VC	Voltage Control
WoC	Web of Cells

Chapter 1

Introduction

1.1. Introduction to the Research

Today, the electrical power system is experiencing significant challenges due to aging infrastructure, environmental problems, economic issues and increasing energy consumption. Moreover, it needs to enhance the security, capacity, and flexibility of the modern power system. In order to do that, the electric grid is undergoing restructuring by a gradual transition from centralised power generation to distributed power generation, in the form of renewable generation sources, including wind power, solar energy, hydro power, bioenergy etc. The main problems with these renewable sources are their cost and intermittency. This is due to the rapidly increasing connection costs when connecting to local power networks, where network constraints limit the capacity of new renewable generation. In addition, renewable generation is not always available where and when needed, these renewable sources are different from conventional sources of electric power in that they are less controllable and not always available for dispatch. Daily and seasonal effects and limited predictability result in intermittent generations. Connecting renewable generation to distribution networks means that networks are subjected to bidirectional power flows, which results in increased power flow magnitudes and adds to uncertainty in predicting the actual power flowing. A number of technical issues stand in the way of greater penetration of renewable generation. Distribution Network Operators (DNOs) must operate their power delivery systems within statutory voltage, thermal and frequency limits in a safe and secure way, and control is a key enabling technology for the development of a renewable energy system. As a result, operating and managing power systems has become more challenging. To deal with these challenges, Distribution Management System (DMS) [1] has been developed as a potential solution, such as the ones proposed by Schneider [2], and ABB [3]. However, the majority of DMSs have focused on the demand side management, which are based on the extension of the current and traditional tools [4]. Currently, DMS is in need of providing new solutions to effectively deal with network issues due to Distributed Energy Resources (DERs) connections (for example, distributed operation). In addition, DNOs are looking for technologies that can deliver higher levels of monitoring, management, and control for network in a cost-effective manner [5]. Therefore, Active Network Management (ANM) [6] technology has emerged as a possible solution, as it offers enhanced network monitoring that autonomously controls network within secure operating conditions by managing intelligent control functions. It can also facilitate more DER connections and maximise the utilisation of network capacity. Moreover, ANM reduces connection costs by limiting the amount of reinforcement needed for new DER connections and speeds up connection time. The concept of ANM has been widely accepted in both industry and academia to facilitate the integration of DERs at the distribution voltage level [7]. In addition, ANM has been developed as product in the power system, such as Smarter Grid Solutions developed ANM Strata and ANM Element [8], SIEMENS developed Spectrum Power ANM [9], and GE developed PowerOn Advantage [10].

However, the network keeps changing due to new devices connecting and control functions evolving overtime, therefore, ANM requires flexibility to adapt for new environment and extensibility to upgrade control functions for future needs. This has motivated the development of ANM to extend from just an automatic monitoring and control architecture, to additionally including flexibility and extensibility. To satisfy the industrial needs [11], this flexible and extensible architecture needs to consider interoperability operation within the network. To achieve this control architecture, modelling and design techniques based on artificial intelligence have to be exploited to provide flexible, extensible and efficient operations of networks.

As a result, several software architectures have been proposed to develop architecture with intelligence. One of the most relevant technologies is Service-Oriented Architecture (SOA) [12], which is based on providing and requesting services. A service is a component capable of performing a task. Services can communicate with each other by a communication protocol either to exchange data or coordinate some activities. Moreover, each service is reusable, scalable, and platform independent: making application development easier. A common way to develop an SOA-based solution is to use the web-service standard, such as web service description language [13]. However, SOA introduces security challenges as SOA is widely applied for web service implementation that could cause the use of wrong or unauthorised services [14]. In addition, the security issue of SOA could reduce the reliability of the SOA application because applying the wrong or unauthorised services could result in the failure of the application. Moreover, Zwiers in [15] stated that ‘Many of today’s SOA implementations are not that elaborate, this is due to the fact that either the SOA provides limited functions or that security arrangements are not sufficiently in place.’ As each service is reusable, selecting the best service from a crowded pool of similar services available in the market can be challenging to satisfy anticipated objectives [16].

Besides SOA, another relevant approach is Multi-Agent Systems (MAS) [17] which comprises multiple intelligent agents, each with some degree of autonomy and proactive to achieve control functions on aggregate from their interactions within an environment. Furthermore, a MAS offers a communication interface, as well as enabling autonomous intelligent control and decisions. Different problems could be solved through a comprehensive multi-agent platform by designing a multi-agent architecture relevant to the application. MAS can be applied for flexibility through designing agents to sense the environment and then accommodate their actions accordingly [18]. Moreover, MAS is extensible through easing the adding or deleting of agents that have various functions for different control purposes [18]. As a result, a MAS is able to combine various intelligent control algorithms, which can be changed dynamically, into a single framework thereby attaining a superior performance and MAS can be utilised to model competitive, cooperative or negotiating behaviours of the system. In addition, MAS technology can fill what the European SmartGrid

Technology Platform document calls ‘a toolbox of smart grid capabilities’ [19] to provide multiple control purposes.

Based on above information, SOA is not selected for ANM development due to the fact that either the SOA provides limited functions or the security arrangements are insufficient [15]. For MAS, it is autonomous and capable of effective proactive behaviour in dynamic environments and offers collaboration, flexibility, and extensibility. These features of MAS match the requirements of future ANM development including flexibility and extensibility and have the potential to develop a toolbox with different smart grid functionalities. Furthermore, MAS has the potential to be integrated with communication standards, such as IEC 61850, to address the interoperability issue of ANM in the practical application. Therefore, MAS is selected as an approach to developing ANM with greater flexibility and extensibility to control and management of distributed power system in this thesis.

In addition, these potential advantages motivate the use of MAS as a paradigm, which has been increasingly applied in smart grid development, for applications such as diagnostics [20]; condition monitoring [21]; microgrid control [22]; power system operation [23]; protection [24]; voltage control [25]; and demand response [26]. MAS technology has also attracted the attention of the industry, resulting in several simulation studies, pilot projects, and business-as-usual applications [27], [28]. However, most of these multi-agent systems only deal with one control technique each, and in some of them are not implemented with any industrial standards.

While MAS is seen as a promising approach toward a smart grid, applying MAS into complex power system control problems poses some key challenges that still need to be addressed, and highlights areas where novel advances are required. For instance, Vrba et al. [29] recently indicated that “An integrated solution addressing advanced automation and control functions is missing. Such an approach should be based on the MAS/SOA principles, guaranteeing interoperability, flexibility, and scalability.” Vrba et al. also pointed out that “The agents will be executing (running) on field devices, such as Intelligent Electronic Devices (IEDs), DERs, PEVs (plug-in electric vehicles) and others. It is not clear up to now from the related literature, how to implement widely used Java-based agents on such devices addressing the related deployment challenges.” Moreover, Strasser et al. [30] indicated that “Developments in the area of advanced automation solutions based on MAS, PSO (particle swarm optimisation), ANN (artificial neural networks), etc. for Smart Grids have to be aligned with domain standards”. In addition, as suggested by Catterson et al. [31], “Any agent systems developed for this arena must adhere to appropriate standards, to give open plug-and-play systems”. Catterson et al also mentioned that “The most practical course is to design the agents to be as flexible as possible in handling the current use cases”. More challenges include agent simulation platform selection to realise dedicated agent functions with flexibility, extensibility and interoperability features and developing the agent communication language and ontology for power system specific applications.

This thesis is focused on addressing the above challenges by proposing a fully integrated MAS architecture for ANM by implementation of the IEC 61850 standard to deliver the interoperability requirement for controlling industrial devices, as well as combining functions to provide flexible control solutions for power systems. It also fulfils the need for flexibility and extensibility in order to adapt and update the MAS architecture over time without completely re-engineering the whole system. In addition, the fully integrated MAS architecture can deal with multiple network events occurring simultaneously. Moreover, agents can adapt themselves if there is a network change, such as the connection of additional DG. To achieve these features, agent, simulation is implemented in the selected Presage2 [32] agent platform with flexible and effective agent communication. Furthermore, the performance of developed MAS architecture has been tested in the UK distribution network model that is an 11kV radial network. In addition, the developed MAS architecture is extended to include a self-organisation mechanism and is demonstrated for voltage regulation. Results show that this novel approach to network management with agent communication and negotiation capabilities can be applied to real-world power system control challenges, under different network conditions and scenarios.

1.2. Justification for Research

The main objective of this research is to develop a fully integrated control architecture based on MAS approach. The novel contribution of this research is that a flexible and extensible multi-agent architecture is presented for ANM, which addresses the challenges identified above. The proposed MAS architecture is tested through the integration of different control functions to address network issues such as voltage and thermal constraint violation. It provides an extensible environment where control functions can be adapted and altered over time, and are able to collaborate to reach a conclusion on control decisions. The interoperability issue is addressed using international standard IEC 61850 to interface agents with appropriate IEDs. The MAS architecture is evaluated by case studies which demonstrate the extensible deployment of different control agents, to explore flexibility and extensibility within the MAS architecture. Agent negotiation reaches a control solution agreement under different conditions, making use of the knowledge base and reasoning capability within the agents.

The novel aspects of the research and its contributions can be summarised as follows:

1. Design and implementation of **a fully integrated MAS architecture** to achieve **flexible** and **extensible** ANM with multiple control functions that can be adapted for new network environments and upgraded for future control needs over time without complete re-engineering of the system;
2. The creation of a **novel solver agent** which can embed control algorithms within a **self-organising** and **negotiating** agent system;
3. An implementation of an intelligent solver agent with its **own knowledge and reasoning logic to plan its activities**, and which has **communication and negotiation capabilities** to facilitate control decision consensus;
4. Design and implementation of **network violation detection agent** and **DG agents** that able to detect network violations and issue signal to control relevant DGs separately;
5. The **design of an ontology for power system control** application that enables agents to communicate with a shared lexicon within the agent message content;

6. The employment of the **IEC 61850 interface within the agent platform** to provide the interoperability required for devices in substations;
7. The development and implementation of the fully integrated MAS architecture using the **Presage2 agent platform**;
8. An investigation of **key challenges of MAS** application for complex control problems and smart grid applications;
9. Case studies to demonstrate the functionalities of fully integrated MAS, including:
    - A case study demonstrating the MAS architecture's **flexible** capability by adding various solver agents to solve various network violations when they become necessary;
    - A case study of how **agents negotiate** with one another to reach a control solution agreement to address network violations under different conditions;
    - A case study of solver agents dealing with **multiple simultaneous events**, including thermal violation and voltage violation;
    - A case study of agents adapting themselves to network changes through **a new DG connection**;
    - A case study of implementing **self-organising mechanism** within the developed MAS for voltage regulation based on the ε decomposition technique.

1.3. Thesis Overview

The organisation of the rest of the thesis is as follows.

- Chapter 2 provides background information and a literature review of active network management and projects where it has been applied. This chapter also discusses the challenges of future development of active network management for power system domain. Tools and control algorithms for control are presented in this chapter as well. Some recommendations and requirements for the development of an integrated control architecture are discussed at the end.

- Chapter 3 reviews multi-agent systems, and some reasons they are considered for power engineering applications. Moreover, it identifies MAS development and design based on MAS methodologies, agent communications, and agent platforms. Some research applications of MAS in the power system domain are presented as well. This chapter discusses the current technical problems and challenges of MAS development in power system field at the end.

- Chapter 4 presents the design and implementation details of the proposed MAS architecture by following the MAS design methodology. Each agent is implemented in term of its objectives, knowledge, and behaviours. The ontology design and interoperability development based on IEC 61850 standard are derived in this chapter. Furthermore, the selected Presage2 agent platform is also introduced with its characteristics and benefits in this chapter. Additionally, this chapter provides an illustrative example applying Presage2 within the power system domain. Finally, the control architecture of the MAS operation for network management applications is discussed.

- Chapter 5 takes the selected control algorithms and Presage2 agent platform to test the proposed MAS architecture within the UK distribution network model. Developed case studies demonstrate agent capabilities, which can successfully perform under different scenarios and conditions. These case studies also present agents' negotiation ability and extensible feature to facilitate the flexibility and extensibility development of MAS. This chapter also includes a case study showing how the proposed MAS is extended to achieve self-organising capabilities for voltage regulation.

- Chapter 6 summaries the work presented in this thesis and indicates where research questions remain for the future.

1.4. Associated Publications

1. M. Chen, S. D. J. McArthur, I. Kockar, and J. Pitt, “Evaluating a MAS Architecture for Flexible Distribution Power Flow Management,” in 18th International Conference on Intelligent Systems Application to Power Systems, September 2015.
2. E. Guillo Sansano, M .H. Syed, P. Dambrauskas, M. Chen, G. M. Burt, and S. D. J. McArthur, and T. Strasser, “Transitioning from Centralized to Distributed Control: Using SGAM to Support a Collaborative Development of Web of Cells Architecture for Real Time Control,” in 2016 CIRED Workshop, June 2016.
3. M. Chen, M. H. Syed, E. Guillo Sansano, S. D. J. McArthur, G. M. Burt, and I. Kockar, “Distributed Negotiation in Future Power Networks: Rapid Prototyping Using Multi-Agent System,” in IEEE PES Innovative Smart Grid Technologies Conference Europe, October 2016.
4. A. S. Zaher, V. M. Catterson, M. H. Syed, S. D. J. McArthur, G. M. Burt, and M. Chen, “Enhanced Situational Awareness and Decision Support for Operators of Future Distributed Power Network Architecture,” in IEEE PES Innovation Smart Grid Technologies Conference Europe, October 2016.
5. M. Chen, V. M. Catterson, M. H. Syed, S. D. J. McArthur, G. M. Burt, M. Marinelli, A. M. Prostejovsky, and K. Heussen, “Supporting Control Room Operators in Highly Automated Future Power Networks,” CIRED Proceedings Journal, vol.2017, no. 1, pp. 1492-1495, Jun. 2017.
6. M. Chen, D. Athanasiadis, B. Al Faiya, S. D. J. McArthur, I. Kockar, H. Lu, and F. De Leon, “Design of A Multi-Agent System for Distributed Voltage Regulation,” in 19th International Conference on Intelligent Systems Applications to Power Systems, September 2017.

Chapter 2

Review of Active Network Management and Control

2.1. Introduction

To reduce greenhouse gas emissions and limit climate change, low-carbon and energy efficient technologies are required. As a result, the large-scale penetration of Distributed Energy Resources (DERs) could be a promising approach. Renewable Energy Sources (RESs) are major providers of DER connected to the power distribution grids, such as wind, solar, biomass and hydro. As the nature of an RES is intermittent and stochastic, it can lead to occasions where voltage, power flow, or frequency is outside of statutory or operational limits within the power system. Intelligent technologies play an important role in network management and control to mitigate against this. Therefore, the electric power grid will become more intelligent with local autonomy, self-management and self-healing operation in the future to accommodate the increased penetration of RESs.

In addition, Active Network Management (ANM) has been proposed as a means of driving improvements in network operation and facilitating the connection of DER at the distribution voltage level. Moreover, ANM and control of distribution networks will need to adapt continually over time to accommodate new control needs (e.g. storage, microgeneration, electric vehicles, demand side control, etc.). For that reason, there is a drive to develop a comprehensive platform that contains different management/control algorithms, combine them and undertake decision between competing solutions. As a result, developing an integrated control architecture is a potential approach to enhance ANM's flexibility and extensibility.

This chapter will introduce ANM and its related projects to show various control challenges on the network that can be solved under ANM schemes. Moreover, it will highlight challenges and recommendations from previous ANM projects for the future development of ANM, such as automation, integration, and interoperability. These will be investigated by developing an integrated control architecture based on a MAS approach (detailed in Chapter 4). In addition, tools and control algorithms to solve network issues due to increasing penetration of Distributed Generation (DG) will be discussed, including Power Flow Management (PFM), Voltage Control (VC) and frequency management, and some of control algorithms discussed in this chapter are selected to be deployed for case studies to achieve ANM solutions in Chapter 5. Finally, discussions of requirements for developing an integrated control architecture with multiple control functions to realise flexible and extensible ANM will be provided in this chapter.

2.2. Active Network Management Introduction

Climate change has become the biggest issue in the world and necessitates new energy technology developments to reduce greenhouse gas emissions. For example, Germany aimed to increase the RESs contribution to primary energy consumption to 60% by 2050 [33]. United Kingdom (UK) targeted to achieve approximately 30% renewable generation by 2020 and increasing to 40% by 2030 [34]. However, with the increasing integration of DER for a low carbon energy system, the power system is heavily impacted, such as power balance between generation and demand, and operational limits, including power flow, voltage and frequency. Traditionally, electricity is delivered in one direction from a centrally large power plant over high voltage transmission system to end users, such as homes and businesses. Connecting the RESs into the distribution network results in the bidirectional power flows that can increase the uncertainty of the power flow direction and magnitude. Moreover, the voltage rises due to the power injected by the DG adds to the risk of network security. Furthermore, the fluctuation of RESs' generation can cause frequency deviation which has significant effects on the power system. Overcoming these network issues caused by DG connections requires extensive reinforcement work resulting in high costs for each connection.

In order to address these challenges and enable the smooth connection of DER within the power network, Distributed Network Operators (DNOs) are looking for new smart technology to minimise the extent of required network reinforcement and therefore ANM scheme is emerging. There is no formal definition of ANM and the core objective is to facilitate connection of DER to distribution networks by providing higher levels of automation, better network monitoring, more economical costs, and improved stability and reliability of electricity supply.

ANM first emerged in the work of the Embedded Generation Working Group [35], which became the Distributed Generation Coordination Group (DGCG) later, to look at the issues, barriers, and solutions around developing embedded generation in the UK. Consequently, the DGCG published solutions for the connection and operation of DG [36] and these solutions to voltage, thermal, and fault level violations are considered as ANM solutions. However, the greatest challenge of development of an ANM scheme is how to implement ANM solutions as a commercial deployment. This motivated an early assessment of ANM practice in the UK distribution networks to recommend best practices [37]. For example, monitoring and control capabilities need to be enhanced and extended to be able to control new connected generation within the network. Moreover, communication technologies need to be further exploited to establish a reliable communication link at the distribution level. From this ANM assessment practice, it also identifies the benefits of adopting ANM solutions to DER connection applications, which facilitated the DG connection without high network reinforcement expenditure. However, there are also limitations and barriers of processing ANM within existing power networks, such as regulatory and commercial issues, training and standards, communication protocols, lack of advanced control technologies, etc. [37].

With the continued development of ANM, it has been widely researched and evaluated in academia and industry in recent years. For example, the UK government initiated a range of projects to deploy ANM to enable greater flexibility in the energy system (e.g. Orkney Registered Power Zone [38-42], Flexible Plug and Play [43-46], and Customer-Led Network Revolution [47-50]). In addition, the UK Energy Networks Association has published an Active Network Management Good Practice Guide [51] that provides understanding, importance and benefits of deploying ANM schemes. All these activities facilitated greater DER penetration in existing distribution networks via ANM.

In the next section, some of ANM activities will be presented with their purposes and outcomes. In addition, further challenges and recommendations will be discussed in details for development of ANM.

2.3. Active Network Management Projects

With the increasing attention to the development of ANM, DNOs are working to deploy the ANM solutions into their business as usual practices and therefore a number of ANM projects, trials and research projects have been undertaken and demonstrated across the UK and EU. This section briefly discusses key findings and outcomes from some of the ANM activities. Key learnings from these projects for the future development of ANM are also presented.

2.3.1. Orkney Registered Power Zone

Orkney is an island located off the north coast of Scotland with abundant wind energy and the Orkney Registered Power Zone (RPZ) [38] was one of the incentives to deploy innovative solutions to connect DGs. This was introduced by UK energy regulator Ofgem in 2005. This work on the Orkney distribution network began with studies from the University of Strathclyde for Scottish & Southern Energy (SSE), who owned and operated the Orkney power network, to investigate the potential of connection of more DGs within the Orkney network. Then, a closed-loop trial of the ANM scheme in the Orkney network was completed in 2006 [39]. The first two wind generators connected to the Orkney distribution network came on-line under the ANM scheme in 2009 through cooperation between SSE and Smarter Grid Solutions (SGS), a company who developed the ANM platform and deployed it on the Orkney network [40]. Under the ANM scheme, there are about 19 MW of renewable generation that agreed to connect to the Orkney network by 2013 [41]. Further related works are still ongoing to enhance ANM system in the Orkney area. The key findings of the Orkney RPZ are summarised below and additional details can be found in the report [42].

- **Purpose**: Increase connection of RES to the existing Orkney network and control the output of multiple DGs to resolve violations on the distribution network through ANM solutions.

- **ANM Performance**: The ANM architecture deployed in the Orkney RPZ is shown in Figure 2.1 [40]. The ANM algorithms were running on numbers of distributed controllers located across various locations on the Orkney network to collect measurements, such as generator output, voltage and power flow. These measurements are sent to the central controller to monitor the operations at any location of the network to check whether they remain within limits or not. If there are any measured values exceeding the operational limit, the central controller calculates the control actions to manage the output of generators. The DG control order applied in the Orkney RPZ was the Last In First Off (LIFO) agreement, which is to curtail the last connected DG to the network first. Then, the central controller sends a set-point to the distributed controller to issue the control signal to the DG to limit its output.

- **Accomplishments**: Implemented the ANM scheme within the Orkney network, including network monitoring, providing a secure and reliable solution for dealing with network violations. The ANM scheme facilitated an extra 18.49 MW DG connected to the Orkney power system through the ANM scheme without the need to connect to the mainland of the UK, saving £30 million that would have been required for conventional network reinforcement [42].

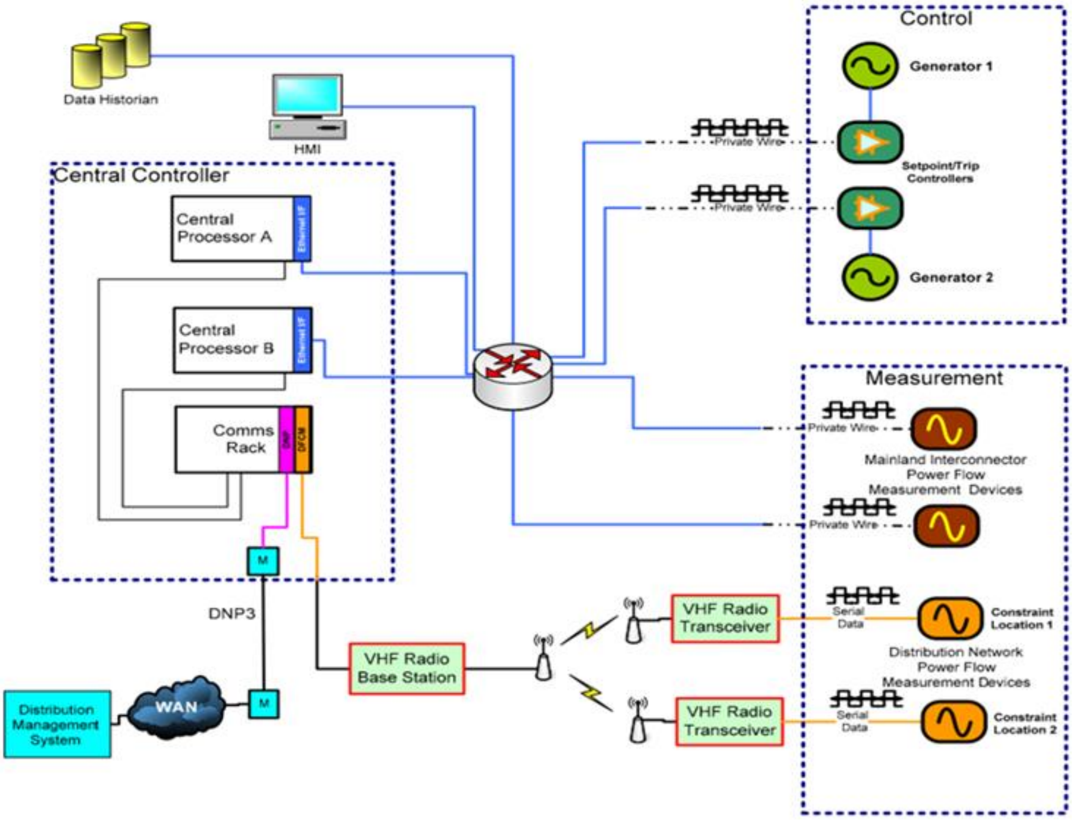


Figure 2.1 ANM Architecture for Orkney RPZ [40]

2.3.2. Flexible Plug and Play

Flexible Plug and Play (FPP) [43] was a three year project funded by Ofgem’s Low Carbon Networks Fund (LCNF) scheme from 2012 to 2014. This project selected the electricity network of the east of England as a trial area with rich renewable resources (wind and sun), which is operated by UK Power Networks [44, 45]. With increasing connection of renewable generation, this trial area has different technical constraints, such as thermal constraints, reverse power flows, limited generator control and lack of a commercial framework for flexible connections, and therefore was suitable for testing ANM solutions to manage these constraints. Under the FPP project, the flexible connection was proposed and that means interruptible connection for DG, allowing DG to connect to the distribution network without extensive reinforcement costs. Moreover, participating DG customers have a commercial agreement with the DNO (UK Power Networks) and therefore the DNO can manage the DGs’ outputs to maintain the network within operating limits. The FPP project was the first one to apply this flexible connection with its innovative commercial arrangements on mainland UK [46]. By 2015, there were 15 flexible connections that had been accepted which were planned to be connected throughout 2015 and in early 2016 at different distribution voltage levels, including 33kV, 11kV, and low voltage [46]. The key findings of FPP project are presented below and details of the FPP project can be found in its report [46].

- **Purpose**: Facilitate faster and cheaper DG connections into the distribution network. Actively manage network constraints through the application of ANM solutions, such as thermal and voltage violation, associated with DG connection.

- **ANM Performance**: The ANM architecture applied in the FPP project is illustrated in Figure 2.2 [44], which is based on the ANM platform and software developed by the SGS company. The ANM system was deployed on the central controller and monitors network constraints. From the ANM architecture, several ANM applications are implemented within the FPP project, including PFM, VC and thermal ratings to increase the capacity of the network [44]. Once a violation is detected, the ANM solution is calculated to control DG output to mitigate the violation. Each DG curtailment is determined based on the Pro-rata principle under FPP trial [45], which is to curtail DG output according to its proportional contribution to the violation. Furthermore, the Pro-rata principle is able to connect more generation by curtailing equally of all DGs [46]. Within ANM architecture, ANM was interfaced with devices by IEC 61850 for information exchange.

- **Accomplishments**: The ANM system has been deployed to monitor the network and control generators to mitigate violations by managing smart devices [44] from various locations across the trial network to collect data or send control commands. Under the FPP project, five flexible connections of a total of 6.75 MW generation were made, including Photovoltaic (PV), wind energy and Combined Heat and Power (CHP) [46].

The FPP project has successfully reduced the participating DG customers' upfront costs for connection by a total of approximately £44 million [46].

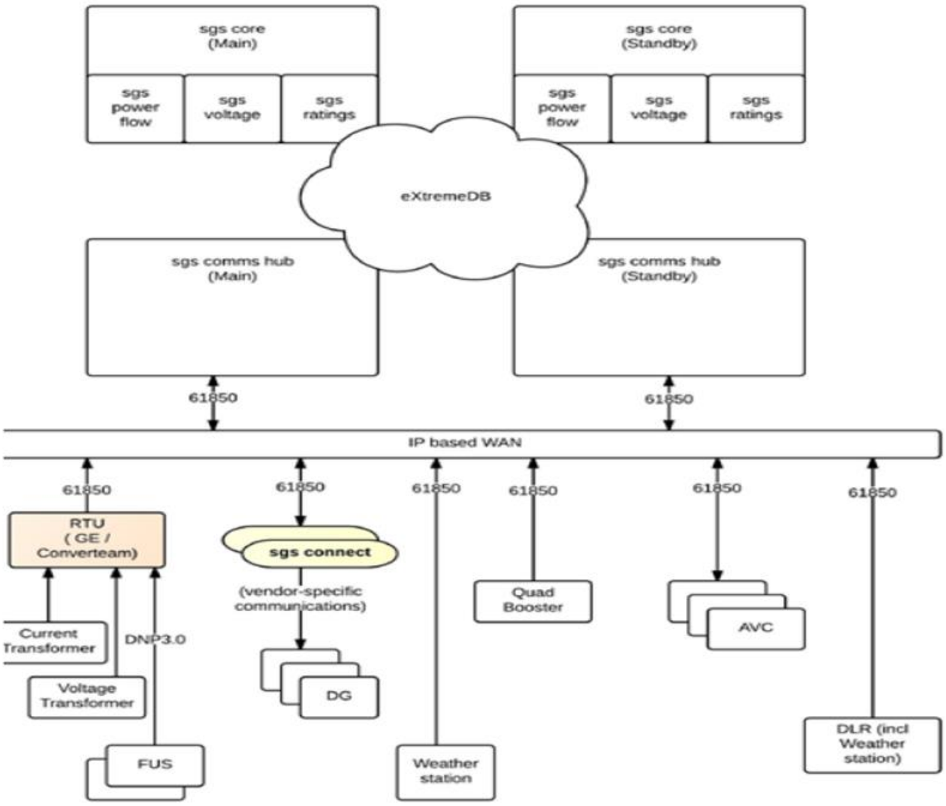


Figure 2.2 ANM Architecture for FPP Project [44]

2.3.3. Customer-Led Network Revolution

The Customer-Led Network Revolution (CLNR) was another Ofgem LCNF scheme funded project and involved five partners, including Durham University, Newcastle University, British Gas, EA Technology and Northern Powergrid [47]. This project lasted for four years between 2011 and 2014, and delivered customer-side solutions, such as novel tariffs and load control incentives, and network-side solutions, including voltage control, real time thermal rating and energy storage [48]. In order to achieve this, the CLNR project integrated customers, energy suppliers and DNOs to develop new commercial arrangements and innovative network solutions. In the CLNR project, Durham University and Newcastle University were responsible for developing advanced modelling techniques and simulations, and to validate physical trials within the laboratory emulation. British Gas as a UK energy retailer was to provide potential future commercial arrangements based on customers' needs and to communicate with trial participants to deliver a solution. EA Technology as a network consultant was involved to apply a cost benefit assessment model to assess the CLNR project. The innovative network solutions and new commercial arrangements were trialled on the Northern Powergrid electricity network located at the north east of England with a large connection of PV to deliver a more flexible choice of using and generating electricity for customers. In addition, CLNR published their trial network data and various network technologies with demonstrations so that researchers and other DNOs can do the same to get benefits from its learnings and findings [48]. Moreover, a guide for the development of smart grid technology and systems has been developed from the CLNR project [48]. The key findings of CLNR project are described below and details of the CLNR project can be found in its report [48].

- **Purpose**: Facilitate RESs integration in distribution networks by developing cost-effective solutions. Control and resolve network constraints by novel technical solutions by Electrical Energy Storage (EES) and coordinated control systems [49].

- **ANM Performance**: The CLNR project has developed an innovative ANM control system called Grand Unified Scheme (GUS) by Siemens and configuration of GUS is shown in Figure 2.3 [50]. GUS is a distributed control system with real-time monitoring and communications to provide VC and PFM by coordination with other technologies, including thermal rating, EES, voltage control devices and demand side management. Through the CLNR project, GUS has been demonstrated under various types of networks, such as rural and urban high voltage and low voltage networks with high DG penetrations. Based on the local monitoring of distribution controllers, GUS is able to detect the constraints on the network and to calculate solutions to mitigate these constraints. After that, real-time set points are sent out to local controllers of the distribution networks to take actions.

- **Accomplishments**: The GUS system has been demonstrated over 200 trial networks to test innovative technology solutions. In addition, there were

13000 electricity customers participating in the CLNR project and 17 MW of demand side response was provided from 16 industrial and commercial customers in trials for a large scale fast reserve [48]. Under the CLNR project, the time of use tariff was established based on customers usage behaviours and therefore could save up to 10% energy bills during the evening peak time for participated customers within the trial networks [48].

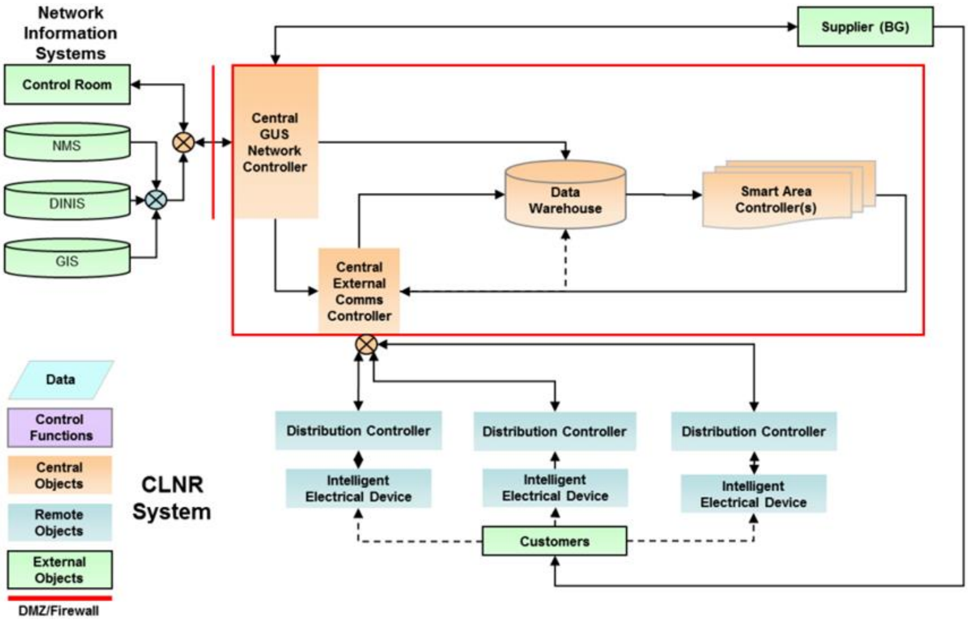


Figure 2.3 Configuration of GUS in CLNR project [50]

2.3.4. Northern Isles New Energy Solutions

The Northern Isles New Energy Solutions (NINES) [52, 53] project was to deliver a sustainable energy system for Shetland Isles, which are situated north of Scotland with massive renewable resources. This project lasted between November 2012 and December 2016 and was led by the owner and operator of the electricity network of Shetland, which is Scottish Hydro Electric Power Distribution [52]. There were two phases of the NINES project. The first phase was to deploy the necessary infrastructure for new technologies, including renewable generation, Demand Side Management (DSM), and energy storage, as most of the generation on Shetland is from fossil fuels and the electricity network on Shetland is isolated from the national electricity network. These innovative technologies were to connect more renewable generation to the Shetland network and to deliver active management of demand and generation to solve network constraints. The second phase was to upgrade or replace the current electricity infrastructure based on the learnings and findings from the first phase and extend the first phase technologies. In order to achieve these two phases, there were six partners involved in NINES project, including SGS (deliver ANM system), S&C Electric Co (battery system), Glen Dimplex and Hjaltland Housing Association (DSM), University of Strathclyde (network modelling and analysis), and Airwave Solutions Ltd (network communications) [53]. The developed ANM system associated with a battery system and DSM ensured that network constraints, including voltage, transient stability, thermal and frequency, can remain within operational limits. Moreover, the installed battery can provide 3 MWh during the peak demand to reduce the reliance on generation [53]. In addition, with the help of DSM, appliances in 234 homes on Shetland can be scheduled at peak demand [53]. The key findings of NINES project are described below and details of the NINES project can be found in the project report [53].

- **Purpose**: Enable the connection of more renewable generation to the existing electricity network on Shetland. Integrate smart grid technologies to monitor and control distribution networks to ensure the supply of secure and reliable electricity. Implement energy storage technology to improve system operation on Shetland. Demonstrate innovative technologies on the Shetland network and develop new commercial arrangement based on customers' behaviours.
- **ANM Performance**: The NINES project has developed and implemented an advanced ANM platform to achieve real-time monitoring and control associated with forecast information, such as generation of wind energy and demand on Shetland. The ANM system architecture is depicted in Figure 2.4 [53]. The ANM system is coordinated with multiple control devices, including DGs, DSM (up to 4 MW flexible demand) and energy storage (1 MW) [53]. The ANM system was deployed in various locations across Shetland network to collect real-time data and forecast information. Then, ANM system calculates the schedule (set-point) for each control devices by considering device constraint, network constraints, generation

and consumption balance, and system stability. If a violation is detected, the ANM system determines control solutions to mitigate the violation and sends set-points to the control devices. Once the network constraint has been solved, the ANM system can issue another set-point to release constrained devices back to unconstrained operation. The objective of the ANM system was to maximise connected renewable generation to reduce the reliance on fossil fuels whilst maintaining safe and reliable performance, such as voltage and frequency stability. The NINES project proved secure and stable network operation can be achieved under a high penetration of renewable generations with coordination using an ANM system.

- **Accomplishments**: By integration of an ANM system, storage technology and DSM, NINES has successfully connected 8.5 MW of renewable generation and 14.9 GWh generation from renewable energy to Shetland network [53]. In addition, 1.92 GWh and 1.45 GWh were imported and exported by the installed battery separately during NINES project [53]. Furthermore, storage heaters and hot water tanks were installed within 234 existing homes across Shetland to provide DSM capability [53].

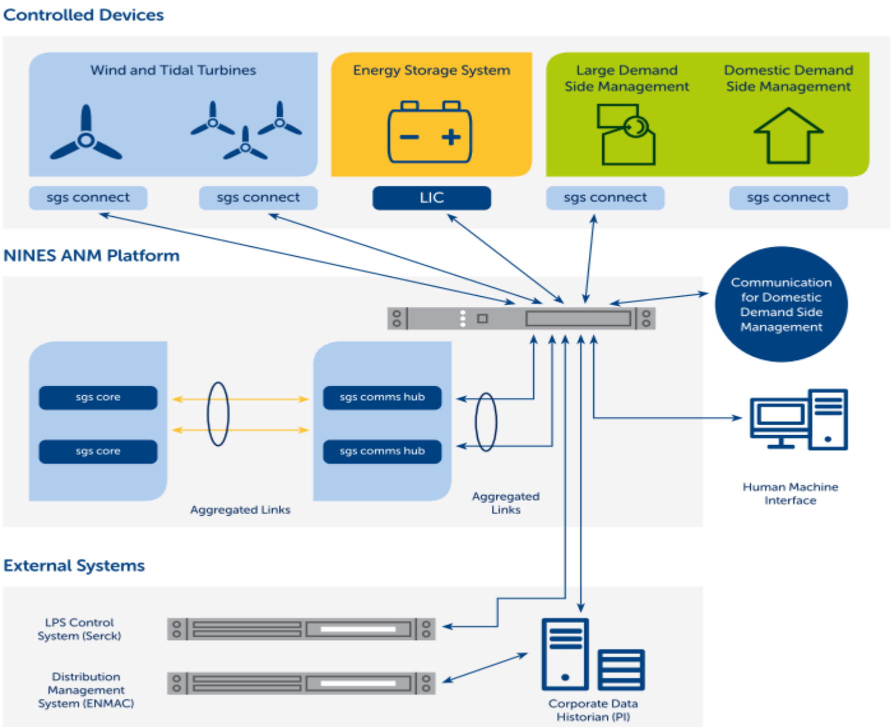


Figure 2.4 ANM System Overview for NINES Project [53]

2.3.5. Autonomous Regional Active Network Management System

The Autonomous Regional Active Network Management System (AuRA-NMS) [54] project was a collaborative research project funded by UK's Engineering and Physical Science Research Council. This project involved seven UK universities, two DNOs (Scottish Power and EDF Energy) and one manufacturer (ABB) to develop the different control solutions for active management of networks with connected DGs [54]. In order to face future network changes, AuRA-NMS aimed to develop a flexible and extensible ANM architecture. Flexibility in this context means that the system is capable of reconfiguration according to changes in the network, such as the connection of generation, energy storage or installation of control equipment. Extensible means that the system has the capability to add new control functionality once it has developed. In order to achieve that, AuRA-NMS integrated different control functions, including VC, PFM, automatic restoration, and proactive reduction of losses. Therefore, these control tasks were delivered by different universities through developing various control techniques. In order to integrate various developed control techniques, AuRA-NMS architecture was proposed to integrate developed control functionalities in a plug-and-play manner by means of Multi-Agent Systems (MAS) technology [55]. AuRA-NMS aimed not only to develop various control techniques, but also to implement these control techniques to different networks. The implementation networks were selected by DNOs (Scottish Power and EDF Energy) including an 11kV radial network with two DGs connected and a 33kV interconnection network with several connected DGs [54]. In addition, developed control techniques have been trialled on hardware platform called COM 600 provided by ABB. COM 600 is an industrial computer with substation communication protocol, such as DNP3, MODBUS and IEC 61850 [56], and therefore is suitable for developing monitoring and control systems using industry communication protocols. At the end of the AuRA-NMS project, it has developed control techniques for VC [57], and PFM were developed [58, 59]. However, it did not integrate these control techniques within the AuRA-NMS architecture to deliver flexible and extensible system. The key findings of AuRA-NMS project are described below.

- **Purpose**: Develop control solutions based on control functions to tackle various network challenges by increasing the number of DG connections, such as voltage and thermal violations. Integrate various control techniques in an ANM architecture with a degree of flexibility and extensibility by means of MAS technology to face the network changes in the future.
- **ANM Performance**: The AuRA-NMS architecture for ANM using MAS technology was shown in Figure 2.5 [60]. Within the architecture, required control functions were designed as intelligent agents, such as voltage control agent, power flow management agent and automatic restoration agent. These agents can automatically detect the network violations and offer solutions to the violations. However, different techniques to solve the same violation may result in conflicting solutions. As a result, the arbitration agent between different solutions is required in the architecture. This architecture offered plug-and-play manner to allow to add new control functionality or remove old control functions during system running.

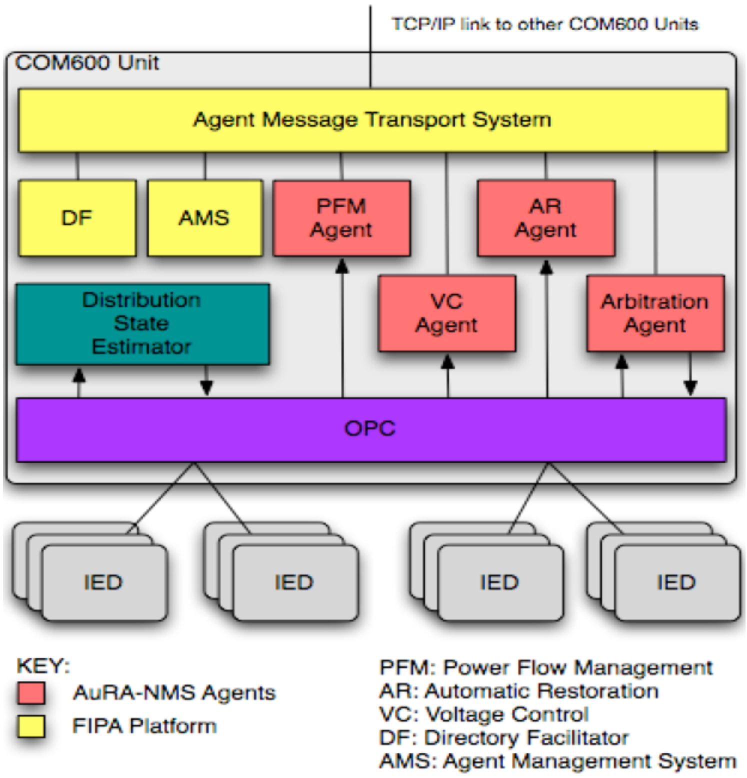


Figure 2.5 AuRA-NMS Architecture for ANM [60]

- **Accomplishments**: Within the AuRA-NMS project, several control approaches were selected and have been developed for voltage control and power flow management. For voltage control task, case based reasoning was investigated to provide a solution through retrieving and matching cases in the case base library [57]. The closest matching case will be retrieved and validated for success. For power flow management, AuRA-NMS has developed constraint programming [58] and optimal power flow to manage power flows [59]. These control approaches have been tested on the ABB's COM 600 hardware platform to realise control functions through several distribution networks.

2.3.6. Active Distribution Network

Active Distribution Network (ADINE) was an EU demonstration project co-funded by EU commission under FP6 [61]. The ADINE project started from October 2007 to November 2010 and involved six different partners across industry and academia [61]. This project was to develop, demonstrate and validate ANM methods in the distribution network with DGs. To achieve that, this project focused on developing solutions for protection and voltage control in networks with DG. In addition, developed ANM solutions were tested in a real network to analyse its performance via a demonstration in a real-time simulation environment. These ANM solutions were delivered by several partners. ABB and Tampere University of Technology conducted field tests for voltage control and protection solutions [62]. Compower AB and Lund University used a micro-turbine to improve voltage quality [62]. AREVA T&D Ltd developed a novel STATCOM controller to improve power quality by controlling wind farm [62]. Several control devices were used for real-time simulation, including protection relays, voltage controllers and STATCOM controllers. A power system simulator was developed based on a real-time digital simulator and the dSPACE [62] control system simulator. The developed ANM methodology can increase the distribution network security, improve stability and enhance network optimal management. Moreover, more renewable generation was connected at a European level during the ADINE project. The key findings of ADINE project are described below.

- **Purpose**: Facilitate DG integration into existing networks and ensure network constraints can be solved by applying ANM. Demonstrate developed ANM solutions in a real-time simulation environment to test their performances.

- **ANM Performance**: The overview of the ANM concept in the ADINE project is illustrated in Figure 2.6 [63]. For the protection of distribution network with DGs, each DG was installed with a Loss-of-Mains relay to prevent unintentional islanding [61]. Moreover, the DG influence was taken into account for fault location and communication based relays were implemented at the lowest level of the distribution network to provide fast tripping of the generator if the fault is at the generator's feeder. For voltage control, the ADINE project has developed droop control of micro-turbines. Moreover, a coordinated voltage control algorithm was designed to manage the substation voltage by adjusting the set value of an automatic voltage regulation controller and reactive power of DG. The STATCOM controller was used to improve the performance regarding harmonics and reactive power [64].

- **Accomplishments**: It demonstrated fast communication between IEDs for generator protection. A micro-turbine prototype with 5 kW was implemented for voltage control at low voltage and medium voltage levels [61]. In addition, voltage control algorithm was developed in Matlab and connected to SCADA system by OPC server [61]. The coordinated voltage control was realised in the field demonstration by applying it to a Finnish distribution network [61].

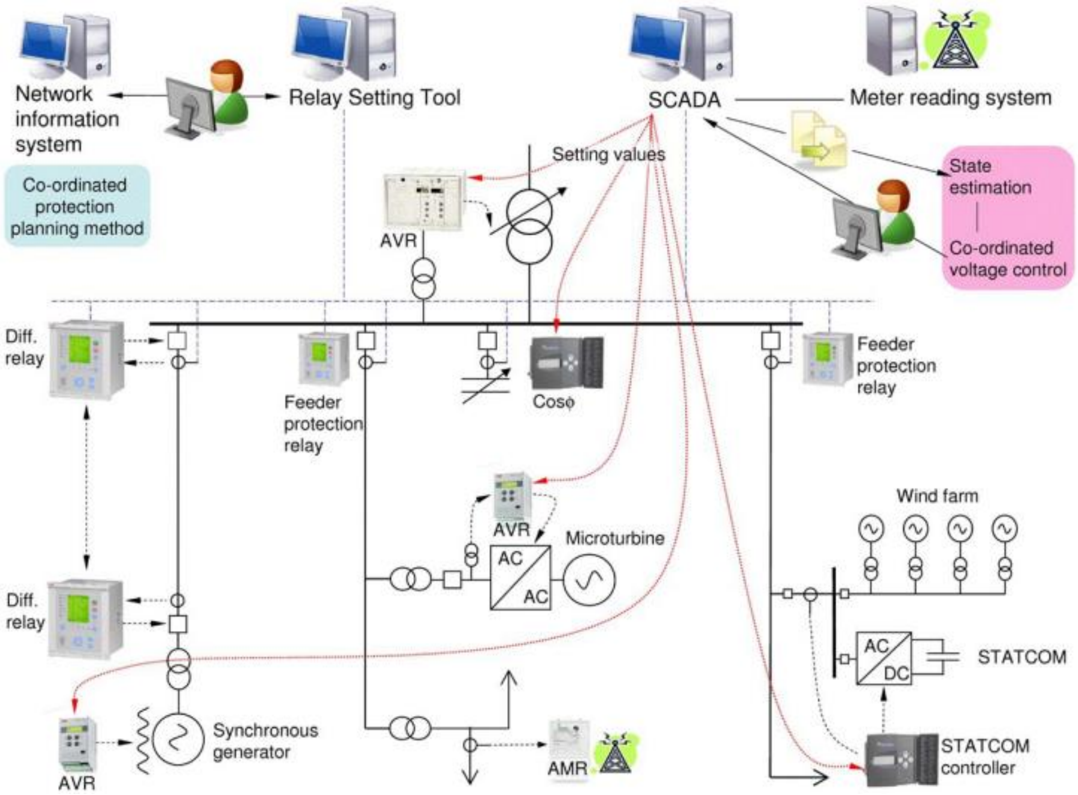


Figure 2.6 Overview of ANM Concept if ADINE project [63]

2.3.7. Discussion of Future Active Network Management Development

As the ANM projects presented above show, the ANM concept has been applied into both industry and academia to evolve distribution networks from traditional passive network management to active distribution network management. Each of these projects aimed to connect more renewable energy to existing distribution networks while developing new network solutions to tackle challenges due to the connection of DGs. In addition, each of these projects has its own innovation. For example, Orkney RPZ and FPP projects used the same ANM technologies and solutions developed by SGS to solve network constraints by controlling DG output, but Orkney RPZ was to constrain the DG output based on the LIFO agreement whereas the FPP project was to curtail by a Pro-rata principle to have greater generation access to the distribution network. For the NINES project, the ANM system was to collect real-time data and forecasting information to determine the set-point for each control device by integrating DSM and storage technology. The CLNR project was not only to solve issues at the network side by applying EES technology but also to develop a customer-side solution by proposing and implementing new commercial arrangements. The AuRA-NMS project aimed to develop a flexible ANM architecture by integrating various control functions into a MAS platform. The ADINE project achieved protection and voltage control by interacting with different network devices (protection relays, voltage regulation controller and STATCOM controller) in a distribution network.

There are other projects across the EU, US, Canada, and Japan to demonstrate innovative technology which is similar to ANM concept named Active Distribution Networks (ADNs) [65]. ADN was introduced by CIGRE C6.19 in 2008 it has systems in place to control a combination of DERs, including generators, loads, and storage [66]. For example, the ADDRESS (Active Distribution network with full integration Demand and distribution energy RESourceS) [67] project was an EU project that aimed to develop active demand to optimise loads and DERs. Additionally, the GRID4RU [68] project was another EU project to develop and test advanced smart grid solutions for integration of DERs, grid automation, energy efficiency, and demand management. Another example is the Fort Collins Demonstration Project [69] in the US which reduced peak loading on two feeders within Fort Collins electricity network by monitoring and coordinating the integrated DERs. One of ADN projects in Canada is the Advanced Distribution Automation [70] which was led by Hydro-Québec company to enable real-time monitoring, automatic reconfiguration, and optimise power delivery efficiency in distribution networks. Japan also has ADN project called Demonstrative Project on New Power Systems [71] to solve problems (e.g. voltage violation and thermal constraints) for DERs connections to the power grid. More ADN projects can be found in [72].

From the outcome of these ANM projects, DNOs have built confidence to develop a smart grid for the future power system. However, to achieve that, ANM will need to adapt continually over time as technology continues to evolve that can impact on the development of ANM. For instance, the Orkney RPZ project was the UK’s first practical implementation of ANM whose approach solved network violations by only controlling generator output, but this project could have also applied demand side management technology to solve the network violations by controlling demand. As a result, it is a challenge to develop enough flexibility in ANM to accommodate new control needs, such as energy storage, microgeneration, electric vehicles, and demand side control. In addition, there are other challenges for the future development of ANM, including automation, integration, and interoperability.

- **Automation**: With the increasing complexity of the network, the ANM system needs to manage and control numerous devices across the distribution network since a human operator cannot manually manage every task. Hence, the ANM system should have high-level automation to take control and respond accordingly. In addition, the ANM system should allow a human operator to override the control if it is necessary.

- **Integration**: To solve various network constraints, integration of various control functions is required. Integration is to ensure that the ANM system can support a range of solutions to different network violations. In addition, an ANM system should be extensible enough to integrate new control functions when it needs to adopt a new control approach.

- **Interoperability**: The increases in DG, energy storage and other power devices connected to the distribution network means the control of all this equipment to regulate voltage or power flow requires the development of interoperability. Interoperability enables the exchange of data and information between different control devices effectively and efficiently. The key to interoperability is communication standards, such as IEC 61850. In the ANM projects discussed above, not everyone applied the IEC 61850 standard to interface with control devices. As a result, ANM needs to develop interoperability to monitor and control local devices in the distribution network.

To address above challenges for ANM development MAS technology is a potential approach that uses intelligent agents with automatic, proactive and communication capabilities [18]. Furthermore, agents in this setting can automatically sense the environment by gathering information and proactively take the corresponding action based on its local/global knowledge. In addition, MAS technology offers a flexible and extensible architecture through integrating different control functions by developing them as autonomous intelligent agents. As a result, MAS is a good paradigm to develop such a control architecture, which meets ANM development requirements for multiple control functions and greater flexibility and extensibility. The details of designing a fully integrated MAS architecture for ANM by addressing above challenges are provided in Chapter 4.

2.4. Control Algorithms and Techniques

With the high penetration of DG in the distribution network, the complexity of the distribution network is increasing and can cause network disturbances, such as voltage, thermal or frequency violations. Hence, the development of control functions is needed to coordinate local DGs, local loads, and energy storage to mitigate network constraints. As discussed above regarding ANM development, to implement ANM requires a collection of technical solutions that are strongly based on the control techniques. Moreover, control technology continues to evolve and ANM is also expected to evolve considerably [73]. As a result, it is important to study existing and emerging control technologies as these technologies might be applied as ANM solutions in the future. In addition, some of the control algorithms are selected and deployed within the developed MAS system to provide ANM solutions under centralised control in Chapter 5.

This subsection focuses on the three main control aspects, including PFM, VC and frequency management, associated with a range of different tools and algorithms to achieve these control tasks.

2.4.1. Power Flow Management

Network power flow is influenced by power supply and consumption, as well as circuit impedance. With the increasing connection of DGs into the distribution network, traditional unidirectional power flow will transition toward bidirectional power flows, which will likely bring added uncertainty to the direction and magnitude of power flows in the distribution network. As a result, it may pose a significant risk to network security and it is very important to monitor the power flows to ensure that the thermal rating of associated network components is not exceeded. In order to avoid issues of overloading, reinforcement work will have to be undertaken and several control approaches have been proposed for PFM, including the use of Constraint Satisfaction Problem (CSP) [74]; Optimal Power Flow (OPF) [75]; and Power Flow Sensitivity Factors (PFSFs) [76]. Each of these control approaches is described below.

2.4.1.1. Constraint Satisfaction Problem

A CSP [77] is a general problem class defined by computer scientists working on artificial intelligence applications and it has been applied in several research areas, such as resource allocation, scheduling, and planning [78]. CSP defines a mathematical problem with a given set of variables each of which has a finite set of possible values, which can be assigned to each variable, and a list of appropriate constraints. Therefore, a finite domain CSP is composed of variables, domains, and constraints that can be expressed in the following form:

$$CSP = (V, D, C), \qquad (2.1)$$

where:

V is the finite set of variable, $V = [X_1, X_2, X_3, \dots, X_n]$; (2.2)

D is the domain of variable values, $D_1 = [v_1, v_2]$; (2.3)

$D_2 = [v_3, v_4, v_5]$; (2.4)

$D_3 = [v_6, v_7]$; (2.5)

…;

$D_n = [v_{n-2}, v_{n-1}, v_n]$. (2.6)

C is the constraint applied to the sets of variables,

The solution of the CSP will consist of values selected from given finite domain to be assigned to variables which meet all constraints.

An example based on the above variable, domain and constraint definition is shown below:

$$V = [X_1, X_2]; \qquad (2.9)$$

$$D_1 = [1, 2, 3]; \qquad (2.10)$$

$$D_2 = [4, 5, 6]; \qquad (2.11)$$

$$C: X_1 = 0.5 \times X_2. \qquad (2.12)$$

The solution is to find the value of each variable in its domain that respects the constraints. To do that, a value is selected from a given domain ($D_1$and D_2), to be assigned to each variable ($X_1$and X_2) in the problem, so that all constraints (C_1) relating to the variables are satisfied. If X_1 is assigned with value 1 from D_1, there is no value in D_2 that can be assigned to X_2 according to constraint C_1. If X_1 is assigned with value 2 from D_1, X_2 can be assigned with value 4 from D_2. If X_1 is assigned with value 3 from D_1, X_2can be assigned with value 6 from D_2. Therefore, the solutions can be summarised as:

$$S_1 =< (X_1, 2), (X_2, 4) > \qquad (2.13)$$

$$S_2 =< (X_1, 3), (X_2, 6) > \qquad (2.14)$$

In addition, CSP can be customised to find either a single solution, a finite number of solutions or all solutions. To do that, the user can design certain constraints within CSP to return either ranked optimal solutions or a user-defined number of solutions. However, considering too many constraints will result in heavy computational demands as the computation time to determine the required number of solutions depends on the number of variables, values in their domains, and constraints.

In [74], researchers developed PFM as a CSP in terms of a set of variables with finite discrete domain and related constraints, which are described below:

- **Variables**: Each controllable generator is selected as a variable and it is to control generator output.
- **Domains**: Values in the variable domain is output set-point of each generator, e.g. [1, 0.8, 0.5, 0]. 1 represents generator operation with rated output and 0 means generator is tripped with zero output.

- **Constraints**: For any potential solution (assign value from domain to each variable), it must solve thermal violations and follow DG connection order (e.g. LIFO). For multiple valid solutions, the solution curtailing the generator output the least is selected and applied.

2.4.1.2. Optimal Power Flow

An OPF [79] algorithm is a standard formulation that is used to optimise generation dispatch to meet demand subject to certain network constraints. For example, OPF could be represented as an objective cost function that minimises system operation costs subject to system operating constraints, including power balance; generation limits of generators; bus voltage limits; and thermal limits. As a result, using OPF has the potential to provide a solution to manage multiple DGs to prevent overload problems within a network [75, 80].

To formulate a power system control problem as an OPF, control variables and an objective function need to be defined. In general, the control variables of OPF are controllable DGs and the control parameter of each DG is active power output. The objective function of OPF is to minimise generation costs and the OPF formulation is presented as follows:

$$\min \sum_{i=1}^{N_g} C_i\left(P_{gi}\right) \qquad (2.15)$$

Subject to

- Power balance

$$P_{gi} - P_{di} = P_i(V,\delta) = \sum_{\substack{j=1 \\ j\neq i}}^{N_{lines}} P_{ij}(V,\delta) \qquad (2.16)$$

$$Q_{gi} - Q_{di} = Q_i(V,\delta) = \sum_{\substack{j=1 \\ j\neq i}}^{N_{lines}} Q_{ij}(V,\delta) \qquad (2.17)$$

- Generation limits

$$P_{gi}^{min} \leq P_{gi} \leq P_{gi}^{max} \qquad (2.18)$$

$$Q_{gi}^{min} \leq Q_{gi} \leq Q_{gi}^{max} \qquad (2.19)$$

- Thermal limits

$$|S_{ij}| \leq S_{ij}^{max} \qquad (2.20)$$

- Bus voltage limits

$$V_i^{min} \leq V_i \leq V_i^{max} \qquad (2.21)$$

Where at bus *i*:

- $C_i(P_{gi})$: Cost function of a generator at bus *i*
- P_{gi}, Q_{gi}: Active and reactive outputs of generator
- P_{di}, Q_{di}: Active and reactive demands
- V_i: Voltage magnitude at bus *i*
- P_{gi}^{min}, P_{gi}^{max}: Generator real power limits
- Q_{gi}^{min}, Q_{gi}^{max}: Generator reactive power limits
- V_i^{min}, V_i^{max}: Voltage magnitude limits at bus *i*

And at the line from buses *i* to *j*:

- P_{ij}, Q_{ij}: Active and reactive power flows in the line from buses *i* to *j*
- S_{ij}: Apparent power for the line from buses *i* to *j*
- S_{ij}^{max}: Apparent power thermal limit for the line from buses *i* to *j*

In [75], the developed OPF function was to minimise total operating costs and related constraints are the same as mentioned above from (2.15)-(2.21), including power balance between generation and demand, generator output limit, and power flow limitations. In addition, the DG connection principle (LIFO) was considered as an extra constraint into OPF function to seek a solution based on DG connection order. The solution determined by OPF is to curtail the only generator, which has an actual impact on the overload in the network [75]. Moreover, the OPF algorithm will be deployed as one of ANM solutions to solve voltage violation and thermal constraints in case studies (details are provided in Chapter 5).

2.4.1.3. Power Flow Sensitivity Factors

PFSFs are a matrix of sensitivity factors ($dL_{P,i}/dG_{P,j}$) that represent changes in real power at a branch i due to the change in real power injection by the generator j from AC power flow solution [76]. Then, the amount of real power of generator needed to be curtailed to resolve the overload based on the sensitivity factor that can be defined as $\Delta G_{P,j} = \Delta L_{p,i}/(dL_{P,i}/dG_{P,j})$. $\Delta L_{P,i}$ represents the amount of real power needed to be removed to mitigate overload at branch l and is calculated as $\Delta L_{P,i} = L_{P,i}^{max} - L_{P,i}$. $L_{P,i}^{max}$ and $L_{P,i}$ are the power flow statutory limit and real power flow along the branch i. In [76], LIFO was cast into the PFSFs function to control DG output by following the connection order. However, the PFSFs are evaluated offline through a network model and PFSFs are only calculated approximately for the actual power flow changes.

2.4.2. Voltage Control

Voltage is one of the most important parameters for the control of electric power systems. Voltage control in distribution systems is significant, and improper control can result in many problems for end users. Moreover, the connection of DG units on distribution feeders can significantly change the system voltage profile and the total system losses. One severe situation occurs during maximum power output from DG and minimum demand at the network where the voltage magnitude can exceed the upper voltage statutory limit. The voltage variation problem can be solved by several approaches, such as controlling the generator and load and voltage control devices. Traditionally, the On-Load Tap Changer (OLTC) transformer associated with automatic voltage control relays is an effective approach to regulate voltage [81]. In recent years, flexible AC transmission systems devices have become more popular, such as static synchronous compensator [82], voltage source converter [83] and thyristor-controlled series compensation [84]. Apart from using voltage control devices, some other novel approaches based on control algorithms to resolve voltage fluctuations from DG intermittency within distribution networks were developed, including Case Based Reasoning (CBR) [85], optimal voltage regulation based on ε decomposition [86], Genetic Algorithm (GA) [87] and coordinated voltage control by combination of Pareto optimisation and Fuzzy logic [88].

2.4.2.1. Case Based Reasoning

CBR is an artificial intelligence technique that retrieves similar cases with suggested solutions to solve the problem case. In [85], the researchers developed a CBR system for voltage control and this is illustrated in Figure 2.7. Once a voltage violation occurs, similar cases are retrieved based on the problem's characteristics. From retrieved cases, suggested solutions are obtained and reused to attempt to solve the current problem case. The adaption of retrieved cases might be needed to make it suitable for the current problem case. Moreover, the retrieved cases need to be verified and therefore online verification is used to apply the control solutions from retrieved cases to the current problem case to check network response. Verification is obtained if an applied solution can maintain the voltage within the operating limits. However, it might have several solutions passing the online verification. As a result, DNO's preference factor is taken into consideration to identify the best solution, such as the OLTC solution is the highest preference for a DNO [85]. Finally, the system is to implement the best solution that has the highest similarity and DNO's preference factor. The implemented case and its control solution are inserted into the case base as a new case.

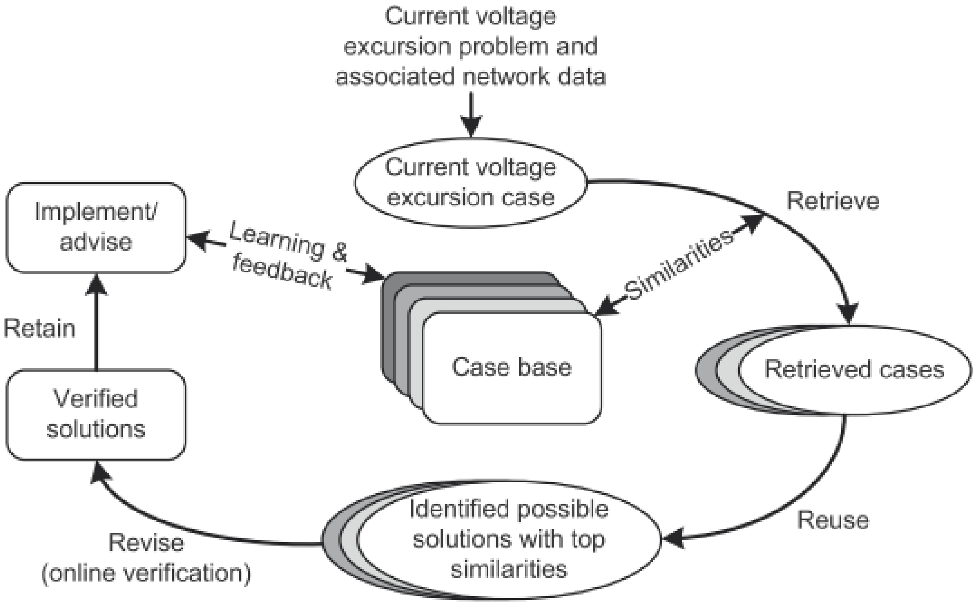


Figure 2.7 CBR System for Voltage Control [85]

2.4.2.2. Optimal Voltage Regulation Based on ε Decomposition

A novel optimisation technique for voltage regulation with DGs is proposed in [86]. This optimal voltage regulation is based on the sensitivity matrix obtained by running power flow analysis. From the sensitivity matrix, the relationship between changes of DG and voltage changes is determined. By performing ε decomposition, the strong coupling between the DGs are kept and weak coupling between DGs are abandoned and therefore a large distribution network can be divided into several sub-networks as shown in Figure [86]. Once voltage violation occurs, the nearest DG can be obtained and controlled to mitigate voltage violations. In addition, DGs can only communicate with other DGs within the same subnetwork to control the voltage which reduces large measurements and calculations. The numbers of subnetworks depends on the selected ε decomposition value and different ε decomposition values can result in a different percentage of covered nodes, the successful control rate for voltage control and power losses [86]. By applying ε decomposition, the system voltage control problem can be decomposed into several small subnetwork voltage control problems. In order to solve the voltage violation, the linear programming algorithm is used for the optimisation problem. There are two control modes that were developed to optimise DG's generation output. One is the unity power factor control mode to maximise the active power output of each involved DG. Another one is the power factor control mode to minimise the reactive power of each involved DG.

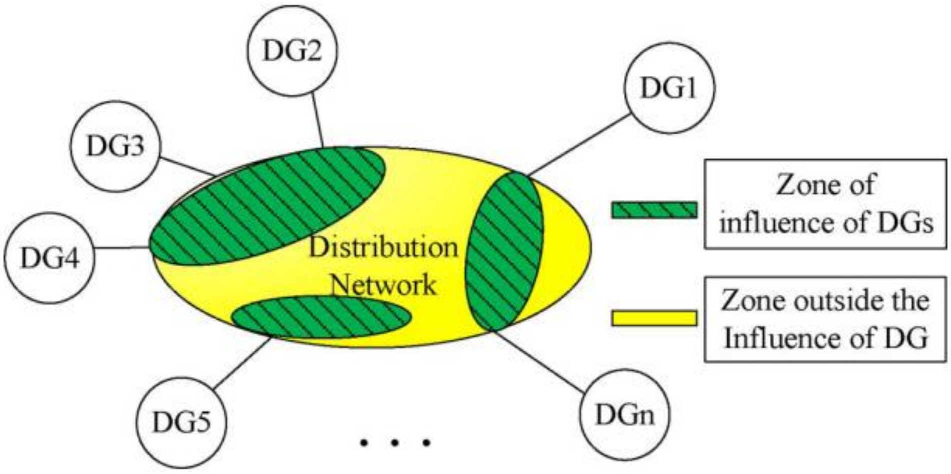


Figure 2.8 Distribution Network with ε Decomposition Result [86]

2.4.2.3. Genetic Algorithm

GA is an evolutionary algorithm that is used to solve optimisation problems or search problems based on the natural evolution process. Within GA, it keeps evolving a population of a set of solutions to the next generation until an optimal solution is found that can satisfy the end condition. During the process of evolution, each solution can be mutated or altered toward the optimal solution. In [87], it aimed to control voltage by coordinating between a load ratio control transformer, step voltage regulator, shunt capacitor, shunt reactor, and static var compensator. The GA algorithm is used to calculate the operation of each device, including tap location of the transformer, capacity of step voltage regulator, on or off of shunt capacitor and shunt reactor [87]. The objective function of the GA algorithm was set to solve the voltage violation within statutory limits and minimise network losses. However, this proposed approach would require reliable communication in order to coordinate between various devices. If one of the devices lost communication it will result in the failure of the voltage control.

2.4.2.4. Fuzzy Logic and Pareto Optimisation

A novel approach to combine fuzzy logic and Pareto optimisation to achieve coordinated voltage control was proposed in [88]. The approach developed aimed to solve a multi-objective voltage control problem that included minimising voltage variation at pilot buses, reactive power production ratio deviation, and generator voltage violation [88]. These objectives are solved by fuzzy logic and Pareto optimisation separately. Pareto optimisation was used to find the optimal voltage at the pilot buses. Pareto is an optimisation tool to find solutions for more than one objective function problem. In [88], Pareto optimisation calculated a set of solutions to optimise these control objectives, then, Pareto optimisation chose one solution from the set of solutions based on the decision maker by minimising the sum of the three objectives. For the fuzzy logic, it was used to determine the optimal reactive power of DG according to the optimal voltage value provided by Pareto optimisation. After that, fuzzy logic calculated the optimal reactive power of DG by analysing the voltage difference between the reference voltage and the optimal voltage of pilot bus [88].

2.4.3. Frequency Management

Frequency is a fundamental parameter of AC electrical power systems. With a nominal value of 50 Hz in the UK, it indicates the balance between electricity production and consumption. In order to reliably and economically deliver the adequate power to the load, a constant balancing of electricity supply and demand is required. The quality of the power supply is affected by repeated and random changes in load, as well as the intermittency of DG generation within the distribution network. If electricity supply is more than demand, the system frequency will increase. If electricity supply is less than the demand, the system frequency will decrease. In addition, if the difference between supply and demand is too great, it can result in generation curtailment or demand shedding. As a result, maintaining a continuous balance between power generation and demand to achieve constant frequency in the network is a challenge. Frequency control is usually divided into three levels, including primary, secondary and tertiary control [89]. In recent years, there are different optimisation techniques that are used to achieve frequency control, such as fuzzy logic based Proportional Integral Derivative (PID) controller [89], Particle Swarm Optimisation (PSO) [90], and Artificial Neural Network (ANN) [91]. Each of these control techniques for frequency control applications is presented below.

2.4.3.1. Fuzzy Logic Based PID Controller

As a power system is interconnected with others, each generator is equipped with a PID controller in each area to maintain the system frequency and power balance [89]. The objective of PID controller is to regulate the frequency of each area and input signal of the PID controller is Area Control Error (ACE). The application of fuzzy logic to calculate values for the parameters of PID controller is presented in [89]. To do that, the fuzzy logic controller was developed that contained four different elements, including a rule-base (i.e. if-then rules), an inference mechanism, fuzzification interface, and defuzzification interface [89]. The fuzzy logic controller calculated the values for PID controller parameters based on the offline control rules after receiving the inputs of ACE and change of ACE. The calculated PID parameters from fuzzy logic can suppress the system frequency and tie-line power flow by comparing with the classical Ziegeler-Nichols approach [89].

2.4.3.2. Particle Swarm Optimisation (PSO)

An enhanced load frequency control by using the PSO algorithm to find optimal parameters for PID controller and Static Synchronous Series Compensator (SSSC) was proposed in [90]. PSO is an optimisation technique based on a population called a swarm and a set of solutions named particles. PSO solves an optimisation problem by moving particles within a search space until finding a position with the optimal value. In [90], the objective function of PSO was to minimise ACE error and obtain an optimal value for the PID controller and SSSC device. The SSSC device was used to damp the overshoot and accelerate frequency by adding it to a tie-line between two areas of the power system. The application of PID controllers and SSSC with optimised parameters can improve frequency stability of the system [90].

2.4.3.3. Artificial Neural Network

ANN is an artificial intelligence method that emulates the way the human brain deals with information and learns [92]. An ANN is formed from multiple simple elements named neurons to perform specific tasks. Each neuron has inputs, a transfer function, an activation threshold, and one output. The ANN behaviour is based on the transfer functions and activation thresholds of its neurons, governed by the optimisation algorithm, and by the architecture itself [92]. The trained and tested neural network can provide predicted output based on new input information. A novel adaptive ANN based controller to control the Energy Capacitor System (ECS) to mitigate frequency deviation caused by a wind farm was developed in [91]. ECS is used to solve the power fluctuation issue of a wind farm that contains power electronic devices and Electric Double-Layer Capacitors (EDLC) [91]. As a result, ECS can be used to minimise the frequency deviation caused by wind generation through smoothing the wind farm line power output. Then, the developed ANN controller used the line power of the wind farm and reference line power as input to generate the output to control the EDLC by the adaptation algorithm called Widrow-Hoff [91]. In the simulation, the ANN based ECS can effectively solve the frequency deviation with wind generation connected.

2.4.4. Requirement for an Integrated Control Architecture

In the control functions discussed above, a complex control problem can be overcome by using a range of control techniques or algorithms, including those drawn from the area of artificial intelligence. By application of control techniques, power system control problems can be expressed as mathematical models or functions. For example, the PFM problem can be formulated as a CSP problem [74] or OPF problem [75] to be processed by the appropriate algorithms. In addition, control techniques can deal with multiple objectives. CSP technology can solve thermal violations while considering the DG contractual constraints and the Pareto optimisation approach is able to find a solution for multi-objective problems. Moreover, the developed CBR system [85] was not only to maintain the voltage within the statutory limit, but also to maximise the DG access to the network. However, there are also some drawbacks of applying control techniques. For instance, the OPF algorithm can only curtail the generator that has an impact on the thermal violation [75] and it might conflict with the DG connection order, such as LIFO. Moreover, PFSFs are only calculated approximately for the actual power flow changes. [76]. For ANN techniques, they might suffer from the convergence time and long training process [91]. Furthermore, although the ε decomposition approach [86] can divide the system voltage problem into several small subnetwork problems, selecting a suitable ε decomposition value is a challenge. Because a different ε value can result in a different percentage of covered nodes, in some situations, voltage violation nodes might not be covered which can require a revision to different ε value. One of the weaknesses of this approach is that no guidance is given on the best choice of ε. Therefore, selecting ε value depends on the purpose of the decomposition. If the objective of network decomposition is to have more subnetwork with less communication and computation, the ε decomposition value should start with the highest value that results in the largest number of subnetworks and therefore less communication and computation for control objective in each subnetwork.

These control techniques have the potential to be designed as part of ANM solutions and might need to be developed further and integrated together, and with additional solutions, to overcome the above problems and provide improved performance (for example an optimiser). As a result, an integrated control architecture is required that will provide a platform with multiple control techniques to achieve different power system control purposes. MAS is a potential technology to develop flexible, extensible and distributable architecture [55, 56] that satisfies future ANM development requirement. For flexibility, agents have the potential to be designed to take actions by sensing the changes in the network environment. For extensibility, MAS allows agents to be added or removed when it is necessary and therefore new control functions can be added by developing them as an intelligent agent. Furthermore, agents offer communication capabilities so that they can communicate with each other to exchange information or negotiate to achieve better control performance. In addition, agents can interface with control devices by utilising existing and emerging standards, such as IEC 61850, which can solve the interoperability for the ANM. As a result, MAS technology is a potential approach to develop an integrated control architecture to achieve ANM scheme associated with flexible and extensible capabilities.

2.5. Summary

In order to deliver low carbon energy and tackle the climate change problem, numerous RESs has been connected to the distribution network over the last few decades. However, the major disadvantage of the RESs is intermittent and stochastic generation by nature. Therefore, large RESs integration can bring significant challenges for power system operation and control, such as voltage deviation, frequency fluctuation, thermal violation, and protection issues. Moreover, distribution networks are not well designed to deal with large penetration of RESs and control challenges.

As a result, ANM has been proposed to facilitate the DG connection and solve network issues based on new technology and control strategies that can reduce the expensive costs of network reinforcement. There are many ANM applications that have been deployed within industry and academia to achieve more DG access to the existing distribution network. Under these ANM projects, technical solutions to solve network issues by connection and operation of DGs have been developed and have been applied in various networks, such as power flow management and voltage control. Furthermore, the new commercial arrangement has been implemented to provide the customer with cheaper electricity supply. Nevertheless, there are some challenges for the future development of ANM, including automation, integration and interoperability. With increasing complexity of power systems, automatic control plays an important role in the system operation and control and therefore ANM requires a high level of automation to reduce the operator burden. Moreover, more control techniques have been applied into the power system, the combination or integration of these new technologies are necessary to achieve an effective ANM. In order to effectively and quickly control the devices within the network, the interoperability issue needs to be addressed in the future development of ANM.

In addition, control techniques are a core part of developing ANM solutions and more new technologies have been applied into the power system for operation and control, especially those based on artificial intelligence. Difficult control problems can be formulated into mathematical models or objective functions and solved by the application of appropriate algorithms. These applications can provide better performance of control operation when compared to other traditional approaches. However, the control techniques also have limitations, such as the ANN technique potentially needing long training process, the OPF solution potentially conflicting with DG connection order, and PFSPs using approximate values for actual power flow changes.

Therefore, this research of this thesis will develop an integrated control architecture that contains appropriate control techniques to address the challenges of ANM, including automation, integration, interoperability, flexibility and extensibility. To achieve this control architecture, MAS technology is a suitable approach that can develop control techniques as intelligent agents to achieve extensibility. MAS supports communication interfaces such as IEC 61850 and this will be included in the work of this thesis to address the interoperability issues inherent with ANM. Further details about MAS technology, applications of MAS within power system and its challenges are discussed in the next chapter. The detailed design of such MAS architecture of this thesis will be provided in Chapter 4.

Chapter 3

Multi-Agent Systems

3.1. Introduction

Evolution toward smart grids requires new intelligent methodologies for control and management of power systems and with recent advances in information technology, artificial intelligence techniques have proliferated [93, 94]. Multi-Agent Systems (MAS) is a methodology which has recently generated excitement since it offers a new paradigm for conceptualising, designing, and implementing software systems [95]. Moreover, MAS is a suitable technology to develop an integrated control architecture for ANM as discussed in Chapter 2. In MAS, each single agent is limited by its knowledge, computing resources, and perspectives, so if a problem is large or complex, then it can instead be addressed through co-operation between multiple functionally specific agents, each specialised in solving a particular problem aspect. This approach allows each agent to use its own paradigm in contributing to an overall solution reached on aggregate. When a problem arises, a certain degree of collective intelligence can be achieved through interactions between agents to reach their goals.

This chapter introduces the background to MAS, including agent definition, agent characteristics and MAS advantages. Then, it presents how a MAS is designed and developed by considering MAS methodologies, agent communication and agent platforms. Moreover, it reviews MAS development in recent years with applications of MAS for power systems which have been reported in the literature, including a number of concepts and experiments used by researchers to apply the MAS approach. Finally, it summarises technical challenges and problems for the development of MAS in power systems engineering.

3.2. Multi-Agent System Introduction

In MAS, a key question is “what is an agent?” There is still an ongoing debate, and little consensus, about the definition of an agent. In its simplest sense, an agent is a software or hardware entity that is situated in some environment and is able to act autonomously within that environment [17]. An agent is able to monitor its environment, either directly by sensors, or through accessing data from other sources. This is a very broad definition, and an increasing number of researchers and industry practitioners have found that the following definition could be widely acceptable by Michael Wooldridge [17].

“*An agent is an encapsulated computational system that is situated in some environment and that is capable of flexible, autonomous action in that environment in order to meet its design objectives*. [17]”

The concept of an intelligent agent extends that of an agent that should exhibit proactive, reactive, and social behaviour. Thus, the following four key characteristics of an intelligent agent are required [96, 97]: autonomy; reactivity; pro-activeness; and social ability.

- **Autonomy**

Agents have a certain level of autonomy: they can take decisions without a central controller or commander. A simple example of agent autonomy is a thermostat which can be viewed as an agent: Action ‘heating on’ is taken when room temperature is too low. If the room temperature is okay, action ‘heating off’ is taken. Therefore, the autonomy of agents is determined based on their individual defined rules and goals.

- **Reactivity**

Agents are situated in a particular environment, and can perceive the state of their environment by sensors, and can often act in an environment to respond to changes, in line with their design objectives in a timely fashion. For example, an agent might control a generator to change its power output in order to maintain bus voltage within limits.

- **Pro-activeness**

Agents do not only simply act in response to their environment, but can also seek out ways of achieving its goals leading to goal-directed behaviours. This could result in an agent planning the course of action it will take, according to the total set of possible operations it can perform and the goal it is trying to fulfil. Agents can assess the usefulness of a particular action before effecting it; modify their behaviours in changing circumstances, and continuously ascertain if their goals are being met.

- **Social Ability**

Agents are able to interact with other agents in order to achieve their objectives. Social abilities are not simply the ability to pass data between agents, but also to negotiate and interact in a co-operative manner. This ability is normally underpinned by an Agent Communication Language (ACL), which structures agent messages around a common syntax.

These four properties combine to imbue intelligent agents with flexibility, which can be used to provide solutions to power engineering problems.

A MAS consists of more than one agent or intelligent agent and is able to perform multiple tasks based on the network situation. Each agent can be designed with individual goal, behaviours, and knowledge. In addition, coordination and communication between agents can be realised by the social ability of intelligent agents in order to achieve the desired outcomes.

3.3. Advantages of Multi-Agent System

Based on the features of an agent described above, there are several distinct advantages of MAS technology over traditional approaches for control and management of power systems. MAS technology has the potential to enhance overall system performance especially in terms of flexibility, reliability, and extensibility [11, 18, 29, 98] and some of the critical advantages of MAS technology are described as follows.

- **Autonomy**: The autonomy of an individual agent means that there is no central control of agents and each agent controls what it does. Other agents may request certain actions, however the agent is always free to refuse such requests, which is unlike a function being called by a program, and agents are truly peers requesting help from other agents.
- **Increase reliability of control system**: Failures of some agents in a MAS will not affect the whole activities and operations of MAS. In a MAS, agents can be designed to adapt to the situation and if one agent fails other agents can continue to complete system operations by programmed capability. Furthermore, if an agent loses communication with other agents it needs to fulfil its goal, it can be developed to search for alternative agents that can achieve the same goal.
- **Plug-and-play capability**: MAS is able to be scaled up by adding new agents or by dispersing them in a new environment with new resources and capacities. Hence, this capability can be applied to install future distributed energy resources and loads or programmable agents with a new control algorithm if they are needed.
- **Learning ability**: Agents can be built to update their rules based on the effectiveness of actions to provide better performance. Moreover, agents can integrate learning capabilities to be able to learn characteristic behaviours of an item (e.g. photovoltaic generation) and then update its knowledge based on its observations by using related learning algorithms, such as the Q-learning algorithm [99].
- **Flexibility**: An agent behaves as required in different situations, i.e. adding a new load, or generator. The flexibility of MAS is given by the fact that agents are independent and able to perceive their environment and then adapt their actions accordingly.

- **Extensibility**: MAS is extensible as an agent could be added or deleted and therefore specific functions can be designed as intelligent agents to be plugged in when they are available. For example, a MAS has the capability to add a dynamic line rating function agent or new optimisation algorithm agent when they become available without the need to re-engineer the complete MAS system.

As a result, MAS technology has the potential to be applied in a variety of research topics in power engineering, including system operation, monitoring, diagnostics, modelling, and simulation.

3.4. Multi-Agent System Development and Design

This section briefly reviews the literature on MAS design, including available design methodologies, MAS platforms, ACL, and ontology design within the ACL.

3.4.1. Multi-Agent System Design

The design of a MAS system is difficult because a MAS has all features of conventional distributed and concurrent systems, and exclusive difficulties due to autonomy, flexibility, and complex interactions of individual agents. To develop a MAS system, a range of design methodologies are available in the literature [100-108]. Some methodologies provide specifications for designing a MAS and some extend traditional software and knowledge engineering approaches. For example, MAS-CommonKADS [100] is a methodology that extends the CommonKADS knowledge engineering approach. The DESIRE [101], O-MaSE [102], Gaia [103], and PASSI [104], on the other hand, are developed from object-oriented approaches. In addition, the MAS-CommonKADS and TROPOS[105] can guide designers from initial specifications to system implementation to complete MAS system. The Cassiopeia [108] methodology emphasises agents' behaviours, which is useful for designing required behaviours for each agent in the MAS system. A list of methodologies for MAS development is illustrated in Table 3.1.

Table 3.1 List of Methodologies for MAS Design

Name	Descriptions
MAS-CommonKADS [100]	MAS-CommonKADS is an agent-oriented software engineering methodology that guides the process of analysing and designing MAS by defining a set of models: agent model; task model; expertise model, and coordination model.
DESIRE [101]	DESIRE is a compositional modelling framework for MAS development that provides models including: task decomposition, information exchange, task sequence, subtask delegation and knowledge structure.
O-MaSE [102]	O-MaSE builds off the MaSE methodology, which allows designers to create agent-oriented software processes from initial system requirements to final system implementations.
Gaia [103]	Gaia provides clear guidelines for analysis and design of complex and open software system by exploiting organisational abstractions, such as the environment, roles, interactions, rules, and structures.
PASSI [104]	PASSI integrates design models and concepts from both object-oriented software engineering and artificial intelligence approaches to design and develop multi-agent societies.
Tropos [105]	Tropos offers very early phases of requirements analysis and therefore provides a deeper understanding of the environment that software must operate within, and the interactions that should occur between human and software agents.
Prometheus [106]	Prometheus supports a range of activities from requirements specification through to detailed design according to goals and plans.

ROADMAP [107]	ROADMAP extends the Gaia methodology to provide support for modelling complex open systems by requirements gathering, a formal model for environment and knowledge, a role hierarchy, explicit modelling of social aspects between agents, and modelling dynamic changes.
Cassiopeia [108]	Cassiopeia is a bottom-up approach and emphasises agents' behaviours. It is composed of three levels of agent behaviours to design a MAS system: elementary, relational and organisational behaviours.

Although there are numbers of MAS design and development methodologies with different features, there is a lack of a proven methodology for designers to construct MAS for applications. Recently, a five-stage waterfall model for MAS development was proposed to summarise the basic development process of MAS [109]. The five-stage of multi-agent system design is presented in Figure 3.1 as follows.

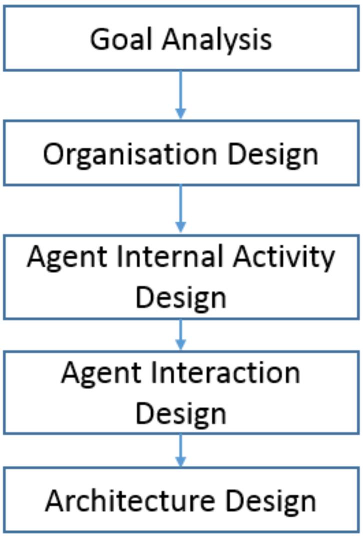


Figure 3.1 Five-Stage of MAS Design

- **Goal Analysis**: The first stage is to understand the problem and to identify functionalities that the MAS are required. Moreover, it needs to specify system tasks, and analyse conflicts between these tasks and therefore decompose these tasks into several small and easy-handled subtasks.

- **Organisation Design**: The second stage is the organisation structure of the MAS. To do the organisation design, a set of agents that comprise the MAS should be defined. After that, the organisational structure is created by defining a role for each agent and determining the relationship between these roles.
- **Agent Internal Activity Design**: In this stage, it aims to design the internal activity of each defined agent. The internal activities contain what goals an agent is required, what knowledge it has, and when and how to respond to an internal or external event. A goal of each agent is what an agent needs to achieve or satisfy. The knowledge of an agent, on the other hand, is an agent belief that refers to the information that an agent knows about. The responses of an agent are logic that an agent reacts according to the event. The reaction of an agent can be carried out based on the basic “if-then” rules or self-defined logic to define what actions can be taken.
- **Agent Interaction Design**: Interactions between agents and suitable interaction protocols for agent communications are designed at this stage. The interaction protocol should specify the agent message format and how communications are transmitted between agents. The interaction protocol should be the application-specific ontology so that agents can understand the terminology they operate on. In addition, it could define what information or data exchange between defined agents.
- **Architecture Design**: In the final stage, a suitable agent implementation platform for implementing agents for the proposed MAS is selected. The internal constructs of each agent, such as agent knowledge, goals, and behaviours, should be mapped into the architecture design during the MAS implementation and configuration.

This five-stage methodology covers all necessary development aspects for agents and can be used as a guide to achieve MAS design and development. It was followed for this research and the details are presented in Chapter 4.

3.4.2. Agent Communication

In order for agents to achieve their goals, and to facilitate coordination between agents, communication between agents is a necessity. The agent communications can be achieved by using communications common to IT networks, such as TCP/IP, SMTP, and HTTP. However, there are some other additional considerations:

- **Agent Communication Language**: it is required so that any agent receiving a message can understand its content and process it accordingly;
- **Ontology**: it is a common vocabulary to let agents understand the information contained within each message.

3.4.2.1. Agent Communication Language Introduction

One of the first ACLs is Knowledge Query and Manipulation Language (KQML), which was developed in the early 1990s as part of the US governments' DARPA knowledge sharing effort [110]. KQML is both a message format and a message handling protocol to support run-time knowledge sharing among agents for cooperative problem-solving. It uses a layered architecture of communication, where at the bottom the functionality for message transport or communication occurs and at the top, the contents are specified in the application layer. Another early ACL named Knowledge Interchange Format (KIF) [111] is largely for data transfer because it cannot represent agent actions. As a result, agents need to have an understanding and parse the content of the messages they received from the communication language [111].

In recent years, the Foundation for Intelligent Physical Agents (FIPA) developed an ACL called FIPA-ACL, aiming to set general standards for agent interoperability [112]. FIPA-ACL has its roots in speech-act theory and incorporates many aspects of KQML. FIPA-ACL provides 13 different parameters [112]: Performative; Sender; Receiver; Reply-to; Content; Language; Encoding; Ontology; Protocol; Conversion-id; Reply-with; In-reply-to; and Reply-by. A message can consist of a set of parameters to provide an effective communication between agents. The required parameters will depend on the situation, but the 'Performative' parameter is the only mandatory field in the message that indicates the type of communication or communicative act for the message. A list of most common performatives and the communicative act they represent in FIPA is portrayed in Table 3.2 [113].

Table 3.2 Some of the Most Commonly used ACL Message Performatives [113]

Performative	**Description**
Agree	Agreeing to perform some action, e.g., response to a Request
Cancel	Informing an agent that an action will no longer take place
Call for Proposal	Calling for proposals to perform some action
Failure	Informing an agent that an action was attempted, but failed
Inform	Informing an agent that a given proposition is true
Not Understood	An agent does not understand what another did
Propose	Submitting a proposal to perform an action
Refuse	Refusing to perform an action, with an explanation
Request	Requesting that an agent perform some action
Subscribe	Requesting persistent notification when object changes

3.4.2.2. Ontology Design

Agents coordinate and communicate with each other relying on a common understanding of terminology. A specification of terms providing a common basis for the understanding of a domain is called an *ontology*. A definition of ontology is shown as follows in [114]:

"*An ontology is a formal definition of a body of knowledge. The most typical type of ontology used in building agents involves a structural component. Essentially a taxonomy of class and subclass relations couple with definitions of the relationships between these things* [114]."

Ontologies generally from a tree structure based on the categorisation of entities. In an ontology, a class is a collection of things with similar properties. Ontologies provide a way to structure information for agents in MAS to understand the semantics of knowledge and agree on the terminologies used in agent communication. Within the FIPA standard, it requires to model three terms for the ontology, which are *Concept*, *Predicate*, and *AgentAction* [115]. *Concept* models the domain concept, including physical items, e.g. substation and transformer. *Predicate* specifies concept status, which can be evaluated as true or false. *AgentAction* is to specify the type of concept for communicative acts such as a request or inform, where an action of an agent can carry out. In [116, 117], researchers developed ontologies for Protection Engineering Diagnostic Agents (PEDA) and Condition Monitoring Multi-Agent System (COMMAS). Figure 3.2 [115] and Figure 3.3 [115] present both ontologies with *Concept*, *Predicate*, and *AgentAction* in power system domain for different applications. Additionally, the IEEE Power and Energy Society Multiagent Systems Working Group provides an upper ontology for power engineering applications based on common information model and also aims to promote openness of agent architectures within the power system domain.

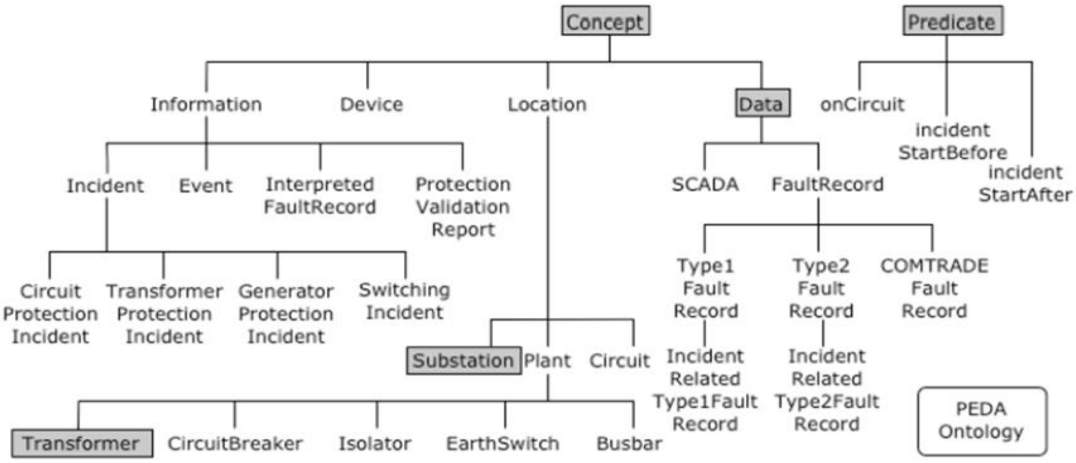


Figure 3.2 PEDA Ontology

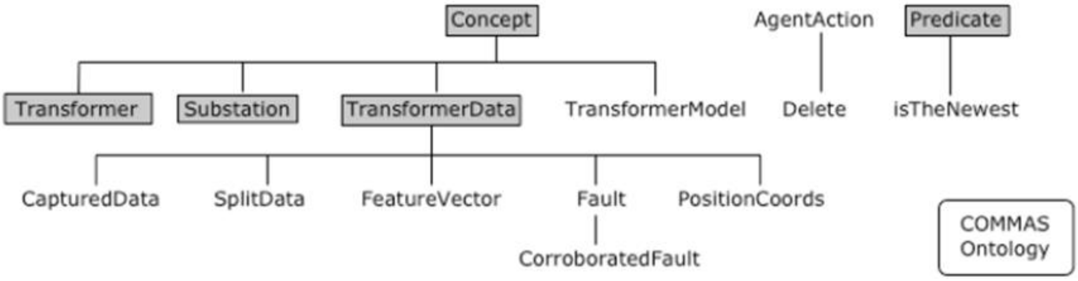


Figure 3.3 COMMAS Ontology

3.4.3. Agent Platform

When agents work together in a MAS, they need to be deployed on an agent platform. The agent platform provides inter-agent messaging capabilities for agent communication and agent management provisions for agent design and implementation. Since the late nineties, many agent platforms have been developed. There are number of multi-agent system platforms available on the literature for developing MAS, such as JANUS [118], JADE [119], ZEUS [120], Mad Kit [121], AgentScape [122], JACK [123], Jason [124], Presage2 [125], Repast [126], and Swarm [127]. A list of agent platforms is shown in Table 3.3.

Table 3.3 List of Agent Platforms

Name	Programming Language	Standard Capability	Communication	Open Source
JANUS	Java	FIPA	FIPA-ACL	Yes
JADE	Java	FIPA	FIPA-ACL	Yes
ZEUS	Java	FIPA	KQML	Yes
MaDKit	Java, C/C++, Python	UML	Peer-to-peer	Yes
Agent Scape	Java	None	ACL (platform defined syntax)	Yes
JACK	Java	FIPA	DCI network, TCP/IP	No
Jason	Java	Partially FIPA	Speech-act based, KQML	Yes
Presage 2	Java	None	ACL (platform defined syntax)	Yes
Repast	Java, C#, C++, Lisp, Prolog, Python	None	Peer-to-peer	Yes
Swarm	Java	None	ACL (platform defined syntax)	Yes

- **JANUS [118]**: An open-source multi-agent platform is fully implemented in Java, which offers three development modes for designers quickly to create multi-agent based applications for web, enterprise, and desktop. Moreover, it provides a comprehensive set of features to develop, run, display and monitor the MAS system. JANUS-based applications can be distributed across a network.

- **JADE [119]**: It is a framework developed in Java and implements a MAS via a middle-ware that claims to comply with the FIPA. JADE can be distributed across machines and configuration can be controlled through a remote Graphical User Interface (GUI) and offers a platform to run agents in real time. Moreover, JADE has a directory service that allows an agent to find other agents providing the service it requires to achieve its own goal. It provides a naming service called Agent Management System to ensure what agents in the platform have a unique name. In addition, the configuration can be changed at run-time by moving agents from one machine to another one when required. Therefore, JADE is supported by a wide range of platforms, including embedded controllers and mobile devices [128]. JADE is industry-driven and is currently the most popular FIPA-compliant agent platform in academia and industry communities. As a result, JADE has been widely adopted within the power system domain, such as power system restoration [129], power plant control [130], power system protection [131], microgrid control [132], condition monitoring [133], and demand side management [134].

- **ZEUS [120]**: It is a product of British Telecommunications Lab, which aims to facilitate the rapid development of new MAS. It provides a set of software components and tools, which can be used to design, develop and organise agent systems. In addition, ZEUS offers a runtime environment to enable applications to be observed and other assistant tools, including reports tool, statistic tool, agents and society viewer. It provides a GUI and a library to create co-ordination strategies and agent behaviours.

- **MaDKit [121]**: It is a multi-agent development kit that is an open source modular and scalable multi-agent platform. It provides general agent and communication models, which are based on the agent-group-role model to structure the agent domains by providing an organisational framework through groups and roles. MaDKit can be implemented using various programming and scripting languages: Java, C/C++, and Python.

- **AgentScape [122]**: It is an agent platform that supports developers to design and deploy large-scale, heterogeneous, secure, distributed agent systems. Agents are active entities that reside within the environment, communicate with each other and access services within AgentScape. The architecture of AgentScape consists of an operating system kernel, and middleware that includes a number of system services, among which agent servers, object servers, service access providers, location managers, and host managers.

- **JACK [123]**: It is a mature, cross-platform environment for building, running and integrating commercial-grade MAS. It is built based on a sound logical foundation: beliefs/desires/intentions. JACK is highly portable and runs on anything from smartphones to high-end, multi-CPU servers. Moreover, it can run with multiple threads across multiple CPUs and offers platform-independent GUIs. Furthermore, it supports to integrate third-party libraries.

- **Jason [124]**: It is a Java-based agent platform which provides user-customisable features, including: speech-act based inter-agent communication; annotations on plan labels; fully customisable selection functions; trust functions; and overall agent architecture, straightforward extensibility by user-defined internal actions, and a clear notion of a multi-agent environment.

- **Presage2 [125]**: It is a simulation platform for rapid prototyping agent societies implemented in Java. Moreover, it offers designers a flexible and generic set of Java classes, interfaces, and tools that can simulate agent societies. Furthermore, it can simulate large, heterogeneous populations, multiple different networks, inter-agent communication, policy modelling, the physical environment, event recognition, data logging, and visualisation.

- **Repast [126]**: It is a widely used free and open-source, cross-platform, agent-based modelling and simulation toolkit that provides multiple implementations in several programming languages and many built-in adaptive features. It includes a variety of agent templates and examples and gives users complete flexibility about how to specify the properties and behaviours of agents.

- **Swarm [127]**: It is the first re-usable software tool created for agent based modelling and simulation. It was specifically designed for artificial life applications and studies of complexity. In addition, it allows quick prototyping of agent-based social models, using an iterative simulation development cycle.

More agent platform examples can be found in [135]. Some of them have already been deprecated while others continue releasing new versions. However, a common problem of agent platforms is how people interested in using MAS are to choose from the vast amount of available agent platforms. As a result, selecting the right or most suitable platform to use in order to benefit from agent technology is a challenge for developers. A good agent platform should have a flexible, extensible and open-source architecture in which agents can interact with each other. Most of these platform offer functionalities that can easily program the agent and use of a MAS.

3.4.4. Summary of MAS Design and Development

For MAS design and development, MAS methodology provides a structure with specifications to realise the proposed MAS. However, different methodologies have different characteristics. For example, TROPOS focuses on the early requirements analysis and Cassiopeia emphasises agent behaviours design. As a result, selecting a suitable methodology is necessary for specifying the proposed MAS. In this research, the five-stage methodology is selected as it offers a complete-lifecycle methodology to cover from initial system requirements and knowledge to final system implementation. The five-stage methodology does not only provide the basic development process, such as agent behaviours and agent communication, but also provides ontology design within the agent interaction design stage. As a result, the five-stage MAS methodology has the potential to achieve the flexibility and extensibility of a MAS.

In addition, the ACL is another main aspect of MAS design and development. The ACL defines the essential message format and content for agent communications. There are several ACL available from the literature. For example, the FIPA standards form an important effort to align different agent platforms, aiming at making them interoperable without necessitating further coordination between the independent development teams. However, FIPA standards were formulated over ten years ago and have since not led to widespread adoption. In their current form, they are not easy to adopt and are not good for domain independent agent platforms. Hence, platforms that only support FIPA will be not be considered for agent implementations in this research. This includes: JADE, JANUS, Jack, ZEUS, and Jason, Ontology provides agent vocabularies to describe the content of the message. However, different agent developers try to develop an application-oriented ontology. This is because an agent can only meaningfully discuss concepts that it knows about, such as measurements it takes from the power system. Moreover, there is no need for an agent ontology to be a complete classification of concepts. Nevertheless, a structured representation of concepts for certain domains where agents operate will be necessary. As a result, the ontology is application-specific, and cannot be standardised. The agent designer needs to define a specialised ontology or extension of an existing upper ontology for a particular application at the system specification stage of development.

An agent platform provides the basic infrastructure for agent design, such as agent creation and communications. As a result, some agent platforms provide general libraries for users to extend and integrate existing ACL. To select a suitable agent platform, it requires an open architecture that allows the user to define ACL or extend existing ACL, such as Agent Scape, Presage2, and Swarm. In addition, an aspect that gains more and more attention by the research community is the trust in MAS. Trust models can help agents to decide who can be trusted, encouraging trustworthy behaviour and preventing dishonest participation by providing the means through which reputation and ultimately trust among the community can be qualified [135]. Hence, it is good to have trust mechanisms built into the agent platform. The Presage2 platform can be extended with Drools, a business logic integration platform, to be able to formalise such mechanisms called self-governing institutions in a rule-based environment. Therefore, the Presage2 platform has been selected in this research that allows the user to define the ACL or extend existing ACL for its own MAS application. Furthermore, Presage2 allows the developer to build self-organising agent environments. More details about the Presage2 platform and its characteristics are provided in Chapter 4.

3.5. Applications of Multi-Agent System in Power Systems

This purpose of this section is to provide a review of the applications of the MAS technique in the control and management of power systems. In particular, it covers key researchers and identifies applications and challenges of MAS in power systems. These applications include: power system operation; power system protection; condition monitoring and diagnostics; energy markets; and Demand Side Management (DSM).

3.5.1. Power System Operation

The operation of modern power systems requires novel and advanced control and management approaches, for which MAS could be beneficial. Although the overall power system is becoming more complex, it can be modelled by the interaction of simple entities. Recent MAS applications in power system operation are listed as below:

- Voltage control is one of the fields of MAS application for power system operation. Elkhatib, El-Shatshat, and Salama [136] developed novel coordinated voltage control in networks with Distributed Generation (DG) that regulated voltage efficiently at multiple feeders. The voltage regulation was achieved by a Remote Terminal Unit (RTU) device that was installed at each DG unit and line capacitor to measure the local data. By applying the MAS, these RTUs were developed as agents that coordinated together to determine the maximum and minimum voltage of the feeder based on their readings [136]. Once a voltage violation was detected, the RTU agent was responsible to execute calculations and send a command to the voltage regulator.

- Another application of MAS for power system operation is power flow management. Müller, Häger and Rehtanz [137] developed distributed power flow management by running controlling agents at the substation level. These controlling agents could measure the local data and update information from neighbouring agents. To mitigate overloads, agents were to manage power flow controlling devices by changing the set-points by calculation of sensitivities of transmission lines.
- A decentralised MAS based frequency control approach was investigated in [138]. Three different types of agents were created, including a DG agent, energy storage agent and load agent, where agents only communicate with their neighbouring agents. As a result, global information could only be determined by agent communications. The optimised average consensus algorithm was used to accurately share the important global information in a distributed way, such as total power deficiency, and total available power [138]. Based on the determined global information, the frequency control strategy could be achieved by controlling energy storage, DGs' output and controllable loads.
- For power system restoration, Solanki, Khushalani and Schulz [139] developed a MAS to restore a power system after a fault, including both full and partial restoration. There are three types of agent within the MAS, which are a switch agent, a load agent and a generator agent. Agents have the ability to measure the local data and can only communicate with neighbouring agents to maximise the supply of power for loads to achieve a feasible restoration solution for the system [139].
- Dimeas and Hatziargyriou [140] presented a microgrid operated by a MAS within a competitive market environment. The MAS was composed of one Microgrid Central Controller Agent (MCCA), and multiple Production Unit Agents (PUAs), Load Unit Agents (LUAs) and Market Player Agents (MPAs). As a result, these agents were to model relevant microgrid actors,

> such as a PUA representing a battery, LUAs were loads of the system and MPAs were sellers or buyers that participate in the market to send a bid to the MGCC according to their requirements. Agents interacted with each other in a market environment to determine the operation of the microgrid. The developed MAS was installed in the microgrid laboratory of the National Technical University of Athens.

As mentioned above, developing key components in a power network as agents provides them with communication and decision making abilities. Moreover, each agent is assigned objectives and can make individual decisions on local power management based on local information, and collective decisions by agent communication and cooperation. Different MAS modelling strategies have different objectives. As a result, modelling agents as components or devices of a power network can aid various power system operational functions, such as voltage control [25, 141, 142], and power flow management [80, 143, 144]. Another agent modelling strategy to improve stability and efficiency of distribution network proposed five types of agents: substation agent; bus agent; feeder agent; load agent; and generation agent [145]. In addition, three system objectives were formulated to guarantee system balance between power supply and demand, as well as to optimise system efficiency and to reduce power usage costs.

3.5.2. Power System Protection

Protection and fault management is an essential part of power system operation that requires the knowledge of failure modes and their causes and provides information regarding the presence of faults in order to take actions as soon as possible. Therefore, power system protection needs to detect abnormal events in a timely manner, to diagnose the causal origins of a fault and then to take appropriate supervisory control decisions and actions to bring the system back to a normal, safe and operational status. The challenges of power system protection are to identify fault situations precisely and operate quickly within in large and complex power systems, so that the rest of systems are not affected.

MAS has been investigated for power system protection by several research efforts. Some applications of MAS in power system protection are presented below.

- A novel MAS based protection system was proposed to provide protection, fault location, fault isolation, network reconfiguration, power supply restoration and diagnosis functions by designing an intelligent distributed terminal as an intelligent agent [146]. The main functions of the intelligent distributed terminal agents included device status monitoring, fault detection, fault diagnosis, information interaction and switching control. Moreover, it proposed an improved current differential protection approach to increase the reliability of protection with high speed action that had been integrated into the proposed MAS.

- Habib et al. [147] introduced a new distribution line protection scheme based on MAS. The proposed MAS consisted of distributed agents named ‘section agents’ that were located in the middle of two protection sections. A section agent detected a fault by comparison of current phase angle signals at both sides of a distribution line. Following this, the section agent has the ability to send trip commands to circuit breakers at both sides of the distribution line during abnormal operation. They presented a new protection methodology to locate and determine the fault types on a distribution line that could reduce the time required to isolate faults.

- Park and Lim [148] presented a MAS system for power system protection by integrating a supervisory control based discrete event system to achieve reliability and selectivity of power system protection. The proposed MAS contains several protection agents, including a diagnosis agent to search fault areas, a fault detection agent to measure fault current, and a relay agent to perform the functions of overcurrent relays. Once a fault occurs, the MAS aimed to trip the minimum number of circuit breakers to realise fault protection.

There are other MAS applications for power system protection that can be found in [149, 150]. In [149], a self-evolving MAS architecture for power system monitoring and protection is proposed that has the ability to find the presence of attacks and modify the agent module structure to prepare for similar future attacks. Furthermore, MAS based relay protection by modelling circuit breaker agents is presented in [150].

3.5.3. Condition Monitoring and Diagnostic Functions

In addition to power system protection, agents have been used for monitoring and diagnostic functions, such as condition monitoring and post-fault diagnosis. Two challenges associated with condition monitoring for power systems are: collection of data from a variety of sensors and interpretation of the data to extract meaningful information. A MAS can collect and manipulate information and knowledge for the management and interpretation of data, to solve such challenges. Some useful applications of MAS for condition monitoring and diagnostic are described below.

- Kenyon et al. [151] proposed a novel MAS for condition monitoring of a gas turbine by integrating Hidden Markov Models (HMMs). The proposed MAS contained three different agents, including a collector agent, thresholding agent, and HMM agent. The collector agent was to store the results from other agents in a database. The thresholding agent was to monitor gas turbine variables and the HMM agent was to implement the HMM algorithm for fault detection. Moreover, a custom web interface was developed to collate and present results of HMM and thresholding agent to users.
- Reppa, Polycarpou and Panayiotou [152] proposed an agent-based distributed methodology to detect and isolate multiple sensor faults in interconnected cyber-physical systems. The sensor fault detection and isolation were conducted in the cyber superstratum. The developed monitoring agent isolates multiple sensor faults by exchanging information between neighbouring monitoring agents. Moreover, the monitoring agent was able to isolate multiple sensor faults that could propagate in the cyber superstratum through a global decision logic.
- Liu et al. [153] developed a MAS for fault detection, diagnosis and prognosis on shipboard power systems. To do that, it had a monitoring agent to acquire data, a fault detection agent, and a diagnosis agent. The

diagnosis agent was to perform further diagnosis of the system via the fault detection agent, including fault modes, location and severity. The proposed MAS can reduce the manning requirements on the ship system.

As presented above, condition monitoring of equipment and plant items can be achieved by MAS approaches. MAS has the ability to deliver diagnostic information to appropriate engineers and automatically adjust device settings according to its condition. MAS techniques are capable of combining data from different devices by delegating the tasks of monitoring each device to intelligent agents. More examples of using MAS for condition monitoring applications can be found in [20, 117, 154]. Moreover, some other examples of MAS application for fault diagnosis can be found in [20, 155-157].

3.5.4. Energy Market

Another MAS application is within energy markets [158, 159]. MAS provides a platform to model autonomous decision making agents which allow developers to implement the market-based operation of a power system. Some applications of MAS for energy markets are described below.

- A MAS based virtual electricity market model was proposed to explore the behaviour of market participants from a regulator’s viewpoint [160]. The virtual electricity market model aimed to choose the optimal rule, regulation, and market structure [160]. As a result, each market participant was modelled as an agent to maximise its profit by forecasting the market prices and submitting the offers to the energy market. Each agent was integrated with a supply function equilibria model that was used to simulate the behaviour of participants to indicate its bid decisions to the energy market.

- Yu, Liu and Price [161] presented a novel method to evaluate the electricity market by using MAS. The modelled MAS contained three different agents, including a generation company agent to sell bulk power to the energy market, a load serving entity agent to purchase power from the market for the loads, and a market operator agent to determine the hourly dispatch schedules of generators. Moreover, the Q-learning algorithm was used to model the learning behaviour of the generation company agent to capture the dynamic interaction between strategic bidding market participants. This proposed MAS was simulated on a 225-bus system with real hourly time-varying load data.

- Foo, Gooi, and Chen [132] developed a MAS for a microgrid energy market that integrated both market operation and DERs implementation. The proposed MAS contained various intelligent agents to model different components in a microgrid, including DG agent, demand agent, market clearing engine agent, coordination agent, and utility grid agent. As a result, the microgrid market operation was achieved by delegating responsibilities to these various agents. For example, DG agents regulated and controlled DGs' power output and could participate in the market operation by negotiating with demand agents. The coordination agent was to coordinate the microgrid operation and monitored the violations in the network. The utility grid agent was to balance the power in the microgrid. Each agent could complete its own tasks by interacting with other agents. Simulation studies and results demonstrated that the proposed MAS could be extended to deal with a large network and corresponding energy market with numerous participants.

Other examples of MAS for energy market participation are discussed in [162-164]. For instance, building an agent-based power market simulator is presented in [162], which is able to balance power demand and supply in the electrical network. Moreover, an agent-based energy market model was proposed for microgrids to manage distributed energy storage systems, without the need for the day ahead forecast of energy generation or demand [163]. Another MAS based framework to implement a combinatorial auction algorithm for a smart grid energy market with a DSM approach was presented in [164] that aimed to reduce the average demand and max peak.

3.5.5. Demand Side Management

Smart grids encourage more customer participation; DSM could play an important role in energy efficiency for the smart grid. MAS has already been applied to achieve DSM as part of smart grid development. Some examples of MAS application for DSM are described below.

- Logenthiran et al. [165] developed a MAS for generation scheduling and DSM within a microgrid. To achieve that, a generation agent was designed to collect power scheduling information and calculate generator set-points. There were two levels of generation scheduling. One level was a day ahead scheduling and the other was real time scheduling. The day ahead scheduling adjusted DERs' power set-points hourly, and real-time scheduling then corrected these power set-points using a feedback system. The DSM agent was designed to take load shift by running a day in advance to optimise the controllable loads. The proposed MAS was simulated in a real time digital simulator for real time operation.
- Li et al. [166] proposed a unique DSM strategy to optimise home energy. To do that, a home energy management system was developed based on MAS. Moreover, household appliances, such as the fridge, air conditioner, heater and television, were developed as agents. The objective of the developed home energy management system was to optimise the energy usage and electricity bills by agent interactions and negotiations to control the appliances according to the priority of appliances. In addition, users could access the energy usage from the developed MAS system and make changes of controllable appliances if it is necessary.
- Ramchurn et al. [167] introduced novel decentralised DSM control based on an agent system. Researchers developed a decentralised DSM model to allow the agent to adapt deferrable loads, such as washing machines, fridges, and electrical heaters, based on electricity prices. Moreover, each agent was represented as a smart meter to optimise the deferrable loads by minimising the energy costs. To achieve that, agents used the Widrow-Hoff

learning mechanism to adapt deferrable loads according to the next day predicted market prices [167]. They investigated 5000 agents within the simulation by applying the load profiles from 26 million homes in the UK. The simulation results indicated that decentralised DSM could improve the performance of the grid by reducing peak demands.

Other examples of MAS for DSM can be found in [168, 169]. For example, [168] developed a MAS for active demand response to achieve minimum electricity cost by considering the electricity price and customer electricity habit. In addition, an electric auction platform for DSM was presented in [169] where agents shift energy demand from peak hours to off-peak hours to minimise operational costs of the system.

3.5.6. Discussion of MAS Applications in Power Systems

There is a wide range of MAS applications to address various problems within power systems. From the applications discussed above, MAS approaches exhibit the capability to solve a large complex problem using multiple agents with autonomy, interaction, and designed behaviours. Agents can be used to monitor and control devices within a power system and have the ability to communicate to achieve system objectives. As a result, the employment of MAS technology can enhance the functionalities of a power system, such as real-time monitoring, diagnosis, self-healing, energy market simulation, and negotiation.

However, there are some challenges and issues from the applications presented above for the adoption of MAS in the power system domain. For example, modelling and deploying agents as components or devices of a power system network [25, 80, 141-144] needs to consider how to deal with agent loss of communications or future changes in the network, such as adding a new DG or removing control devices. Furthermore, within the diagnostic applications of MAS [153, 155, 156], designing a single diagnostic agent to achieve component or whole system diagnosis is inadequate if the components/devices or system are large and complex. For the energy market operation by using MAS approach [160], generation start-up cost, emission cost and fuel costs of power plants are not considered within the MAS design.

In addition, most of MAS research focused on developing a single control function, such as voltage control, frequency management, condition monitoring, fault diagnosis, or DSM, and ignored key aspects of agent implementation design, such as agent communication, and the ontology. From the applications presented above, the developed MAS did not offer solutions for different network violations or enable integration of more control functions. As a result, these developed MAS lack extensibility in order to deal with multiple control purposes. Moreover, the interoperability issue is not considered within many MAS applications, in the context of data exchange between various devices. For example, Müller et al. [137] present a novel approach for a distributed real-time coordination of power flow by locating multi-agent systems at the substation level. However, researchers focused on developing an autonomous coordination system to compute set-points for power flow controllers and it is not clear how to receive data from devices and it lacks details of the MAS implementation at the substation level. Furthermore, the practical issue of interoperability between substation devices was not studied. In addition, implementing and installing agents at the distribution network level to control local devices rely on communications to share information or take control actions [25, 80, 141-144]. As a result, the proper interface between agent system and devices based on the industry standard, such as IEC 61850, are required to have efficient communication with different devices.

As a result, the research reported in this thesis aims to develop a MAS for multiple power system control applications by considering the above challenges and issues. To achieve such a MAS, the technical challenges and problems of MAS discussed in next section need to be addressed.

3.6. Technical Challenges and Problems of Multi-Agent System

It is important to identify the key technical challenges and problems in the development of multi-agent systems for power systems, and how to overcome them. The challenges include multi-agent platform selection, the design of intelligent agents, selection of agent communication languages, the design of ontologies in agent communication, the interoperability issue, development of flexibility and extensibility of MAS.

Selection of the multi-agent platform is a significant challenge because there are a large number and variety of MAS platforms presented in the literature. A good multi-agent platform should be flexible, extensible and an open-source architecture. Design of intelligent agents in a proper manner for power system applications is also a challenging task. This is because intelligent agents should be designed in such a way that they can demonstrate their unique characteristics, such as reactivity, pro-activeness, and social ability.

Agent communication languages and designing of ontologies are necessary for multi-agent development. It is a challenge to select a suitable ACL because there are several ACL available from the literature. Moreover, some agent platforms have integrated ACL that allows the user to define its own ACL or extend an existing ACL. For the ontology, it should be independent of the individual developer because it is difficult to define a standard ontology. However, designing of ontologies should be power system application based on the specific application, and extend any existing ontologies where available.

In the power system automation domain, a key aspect of MAS is to satisfy interoperability requirements as mentioned in [30]. In order to do this, MAS needs to integrate industrial standards for the power system domain. For example, IEC 61850 aims to standardise communication networks and systems in power system automation. Using this standard can help address the above requirements, but the applicability of this approach needs to be investigated further.

Applying MAS for management and control of power systems faces a challenge in terms of the flexibility and extensibility [29, 170]. Flexibility ensures that agents have the ability to adapt to a new environment, such as the addition of new DGs and control devices, without a complete re-engineering of the overall control scheme. Its extensibility makes sure that multiple control algorithms can be integrated and coordinated over time and not just at the initial commission to provide suitable solutions for network violations and beyond, such as network reconfiguration.

Additionally, as [29] recently indicates, “An integrated solution addressing advanced automation and control functions is missing.” Karnouskos et al. [170] also mentioned: “Agent intelligence may also be reached by embedding self-* algorithms, which properly regulate their behaviours, contributing for the emergence of the desired system characteristics.” As a result, developing a fully integrated MAS architecture that contains different control algorithms is needed for ANM as mentioned in Chapter 2. In addition, Catterson et al. [9] suggested that “Any agent systems developed for this arena (power engineering community) must adhere to appropriate standards, to give open plug-and-play systems.” Hence, flexibility and extensibility of MAS can be enhanced by extending various types of agents when they are needed, such as DG agents or control agents.

3.7. Summary

This chapter introduced multi-agent systems technology including agent definition, agent characteristics, and MAS advantages, such as autonomy, flexibility, and extensibility. The main aspects to complete MAS development and design are presented, including MAS methodologies, MAS platforms, ACLs and the ontology. From the literature reviews of MAS methodologies and platforms, a five-stage methodology and the Presage2 platform are selected in this research. The five-stage methodology provides a complete agent design structure, including goal analysis, organisation design, agent internal activity design, agent interaction design and architecture design. The Presage2 platform not only provides basic agent design infrastructure, such as agent creation and agent communication, but also allows a future self-organisation mechanism for the MAS.

In addition, applications of MAS in the power system domain are reviewed and include power system operation, protection, condition monitoring, fault diagnostics, energy markets and DSM. These applications show that MAS is a powerful and useful tool to solve complex problems. Moreover, MAS can be designed to process information, negotiate, and make decisions by embedding techniques like machine learning or problem-solving algorithms. However, most of the MAS applications targeted a single control objective and ignored agent implementation for agent communication and ontology designing. They also ignored the interoperability issue for data exchange between various devices. Furthermore, the interface allows the agent to control or collect data from devices and is needed if agents are deployed to interact with local devices. Other technical challenges related to the development and design of MAS, such as the design of intelligent agents, choosing agent platforms, selection of an ACL, and defining ontologies, are raised. The greatest challenge of a MAS is to create the flexibility and extensibility to deal with future changes in the network and provide fully integrated solutions by embedding multiple control algorithms. It is concluded that an appropriate fully integrated MAS architecture with associated flexibility and extensibility could enhance ANM to accommodate new devices/sources connections and could be upgraded for further control algorithms.

In the next chapter, a fully integrated MAS architecture is proposed to solve the above challenges and issues and to achieve greater flexible and extensible for ANM. The design and implementation details of such a MAS are also provided based on the selected five-stage methodology and Presage2 platform. Moreover, the customisation of the MAS operation and its control for network management applications are detailed.

Chapter 4

Design of a Fully Integrated Multi-Agent System Architecture for Flexible Network Management

4.1. Introduction

Developing a fully integrated system requires different control functions that provide benefit for the development of Active Network Management (ANM), including enhancing its flexible and extensible capabilities. As mentioned in Chapter 3, Multi-Agent Systems (MAS) technology has the potential to build such integrated systems that offer an extensible architecture by extending various agents to extend requirements for various control purposes.

As a result, this chapter proposes a fully integrated MAS architecture for flexible and extensible ANM and addresses the challenges discussed within Chapter 2, including: automation, integration, and interoperability. Detailed information about the proposed fully integrated architecture as well as the development of the MAS based on the five-stage methodology are presented. Ontology design for the Agent Communication Language (ACL) related to power system control applications is described in the agent interaction design stage. To address the interoperability issue of the control applications within the power system, an IEC

61850 standard based interface is developed inside the agent to facilitate the data exchange between agents and control devices. In addition, the selected agent platform Presage2 is introduced with its features, structure and own agent communication language. As a result, implementation of agents through the Presage2 platform is detailed, including an example of a Presage2 application for distributed control of frequency, based on agent negotiation capability. A control architecture for studying the performance of the proposed MAS for ANM is illustrated in this chapter. Furthermore, the design of a power system simulator is introduced to emulate the operation of a physical electric system and responses from the MAS. The details of the case studies of various power system control tasks based on the proposed MAS architecture are provided in Chapter 5.

4.2. Fully Integrated Flexible Multi-Agent System Architecture

Chapter 3 describes the five-stage methodology to specify the MAS design, including goal analysis, organisation design, agent internal activity design, agent interaction design and architecture design. To complete the fully integrated MAS design, the specifications of each stage of the five-stage methodology are presented in this chapter.

4.2.1. Goal Analysis: Needs of a Fully Integrated System

The first stage of MAS design is goal analysis to understand the problems and then identify the functions and knowledge that the MAS needs.

One of the Smart Grid visions is to create a toolbox of proven technical solutions for multiple control purposes [19], and therefore it is important to develop an ANM system that possesses critical control functions for the operation of power systems. Currently, a wide range of technologies and tools are available to provide different power control functionalities. However, each technology or tool focuses on specific control functions and have limitations as described in Chapter 2. For this reason, there is a benefit to have an ANM system that spans multiple software packages and tools for power system control. A general preference for such an ANM system is to build a fully integrated system that can be extended to meet various control requirements.

As a result, the proposed MAS should have the ability to provide multiple control functions. Moreover, agents should have the knowledge of the network operational conditions, such as voltage, power flow and frequency statutory limits, in order to take control actions when it is required. However, embedding different control functions might result in control solution conflicts between these control functions. Therefore, it requires agents to have the ability to solve the conflicts, such as negotiating with other agents, so that a feasible solution can be applied to manage network constraints. Moreover, upgrading existing functionality, adding better suited approaches, or updating the network model when new devices are installed or old devices are removed should be easily achieved without the need for redesigning the core system architecture.

4.2.2. Organisation Design: Fully Integrated Multi-Agent System

To achieve the above fully integrated MAS architecture for ANM, a set of agents and a role for each agent need to be determined within the fully integrated MAS in the organisation design stage.

To realise the multiple control purposes, specific functionality is “wrapped” as an intelligent agent, which is termed a “solver agent”. For each power system control function, it has detailed functional specifications that clearly define the required functionality. Then, these functional specifications will be used to identify blocks of functionality that will be developed into the solver agent imbued with the appropriate agent social, reactive and proactive abilities. In addition, this fully integrated MAS enables the deployment of various solver agents to ensure that it is flexible enough to handle current network management functions, but can also be extended when it is necessary in the future. The developed MAS allows upgrading of functionality to better or more suitable approaches, and these modifications of the MAS are easily achieved without redesigning the core system architecture. This approach ensures that the architecture is both flexible and extensible to support an ANM scheme. Thus, this fully integrated MAS architecture can be targeted at thermal and voltage violation control, as well as implemented for economic, demand management, and frequency control applications.

In addition to the solver agent, to monitor network constraints, such as voltage and thermal violations, the Network Violation Detection (NVD) agent is required. Moreover, the Distributed Generation (DG) agent is designed to take control actions once it receives commands from solver agents. The overview of the proposed MAS architecture that contains a set of agents is shown in Figure 4.1. Each agent has the ability and intelligence to communicate with each other through an agent message transport system (details are provided in Section 4.5.1) with certain data as shown in Figure 4.1.Agents work together to solve control and management problems by agent communications (details are provided in Section 4.5.2), decision-making, and cooperation in a power system (details are provided in each agent internal activity design in Section 4.3). Furthermore, the MAS architecture is scalable and allows agents to plug-and-play when they are available without re-engineering or restarting the system, such as the solver agent and DG agent. The details of each agent internal activity design are provided in Section 4.3. The proposed fully integrated MAS architecture concept is used for centralised control in this research and will be extended for distributed control in the future work.

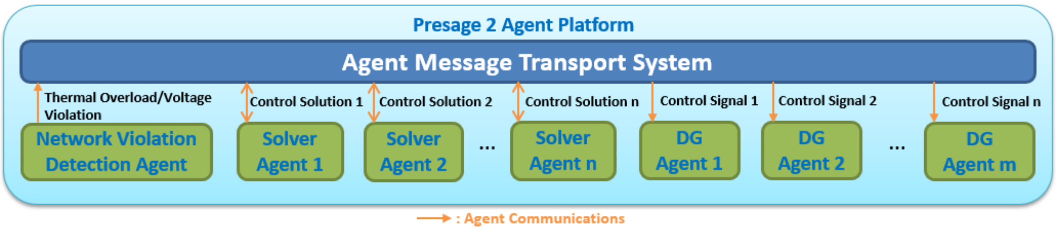


Figure 4.1 An Overall View of the Fully Integrated MAS Architecture

4.3. Agent Internal Activity Design

In the agent internal activity design stage, goals, knowledge and behaviour responses to an internal or external event need to be defined. Each intelligent agent could have a set of rules and tools associated with its actions and goals and can plan its own activities by using its knowledge. Although each agent has different behaviours, it can be placed into a common architecture. Therefore, a generic architecture for the intelligent agent in this research has been proposed as presented in Figure 4.2.

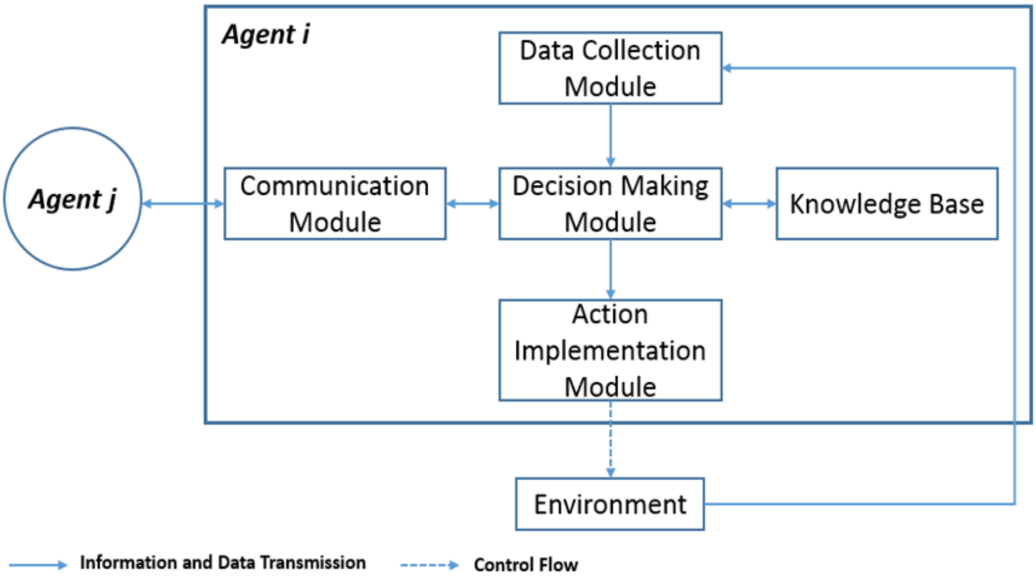


Figure 4.2 Proposed Generic Architecture of an Intelligent Agent

- **Data Collection Module**: the data collection module is to sense the environment (e.g. physical network) and measures current network data for decision making module analysis. In order to do that, it contains an interface to receive data from the environment.

- **Communication Module**: An agent needs to have communication protocols or mechanisms so it can communicate with other agents to enhance collaboration and be reactive. The communication module contains these functions to handle incoming and outgoing messages. The incoming messages are sent to the decision support module for evaluation and scheduling the next step for the agent. Once it receives a message signal from the decision support module it dispatches messages to respective agents using stored agents' addresses.

- **Decision Making Module**: Each agent has a pre-defined goal which is stored in the decision making module alongside a list of possible actions. As soon as the agent receives data from the environment by the data collection module, it analyses data via its the knowledge base and determines the feasible actions that the agent can execute if there is an event occurring in the network. In addition, it can determine a set of actions based on received messages from the communication module. These feasible actions could either trigger the communication module to send messages to other agents or send commands to the action implementation module to execute actions in the environment. Moreover, the decision making module can evaluate the goal through received data from the data collection module or messages in order to indicate whether the goal has been achieved or not.

- **Knowledge Base**: An agent has structured and defined knowledge for a specific domain and environment (e.g. power system control) to perform its tasks intelligently and successfully. The knowledge is formed by structured rules, which can be triggered and can lead to actions to achieve agent’s goals. These actions are determined by the decision making module. Furthermore, the knowledge base can be updated by the decision making module if it is needed.
- **Action Implementation Module**: Once the agent has recognised that a significant event has occurred and that the agent needs to take actions according to decision making module’s command in the environment, these commands are passed to the action implementation model. It will execute control actions through the certain interface to send signals to the certain device in the environment.

From the generic agent architecture, three of the proposed types of agents (NVD agent, DG agent, and solver agent) within the fully integrated MAS are presented in detail with their capabilities, behaviours and functions in the following sections.

4.3.1. NVD Agent Design

The NVD agent’s role is to check for violations in the network, such as voltage excursions and thermal constraints. The NVD agent architecture is illustrated in Figure 4.3.

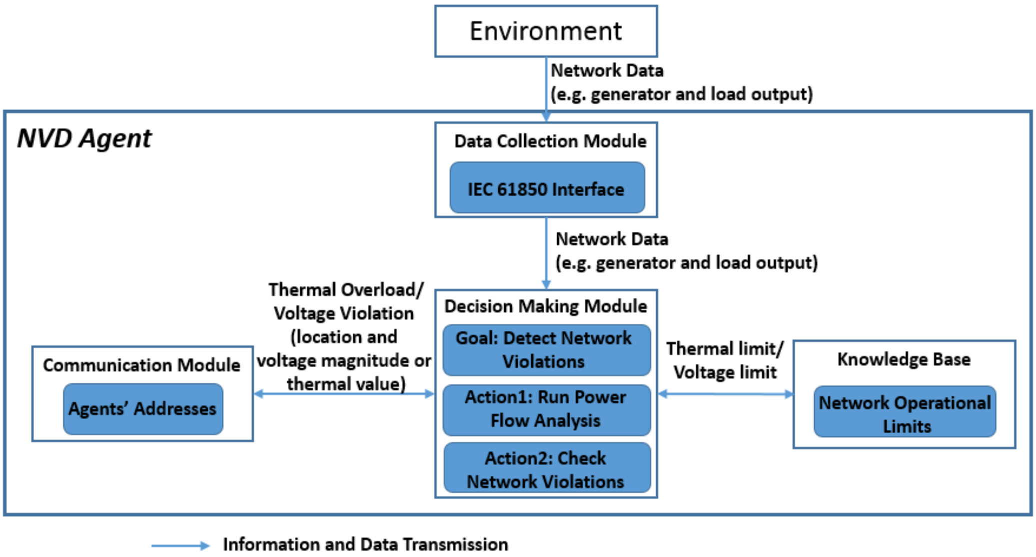


Figure 4.3 NVD Agent Architecture

- **Data Collection Module**: In order to receive the network data, the NVD agent has embedded within it the IEC 61850 interface (more details are provided in section 4.4.2) to receive network measurements from Intelligent Electronic Devices (IEDs) in the environment. This network data could include real power and reactive power of generators and loads. As soon as the NVD agent receives the network data, it runs a power analysis by applying the integrated power flow analysis tool developed by IPSA software [171] (details of IPSA software are presented in section 4.7).

- **Knowledge Base**: To capture network events, the NVD agent has knowledge about network operational limits (e.g. bus voltage operational limits is between 0.95 p.u. and 1.05 p.u).

- **Decision Making Module**: After receiving network data, the NVD agent has two behaviours, which are to run power flow analysis and check the power flow results to determine if there is a violation via the network operational limits in the knowledge base. Once the NVD agent has recognised that a violation event has occurred, it then informs the relevant solver agents with violation information (e.g. violation location and value) through the communication module.

- **Communication Module**: The NVD agent sends messages by the communication module to respective solver agents if a network violation is detected. As a result, the communication module stores other agents' addresses in the agent platform in order to send violation messages to solver agents.

4.3.2. DG Agent Design

The DG agent is responsible for controlling related DG to mitigate network constraints by sending a control command to the relevant IED. The architecture of DG agent is presented in Figure 4.4.

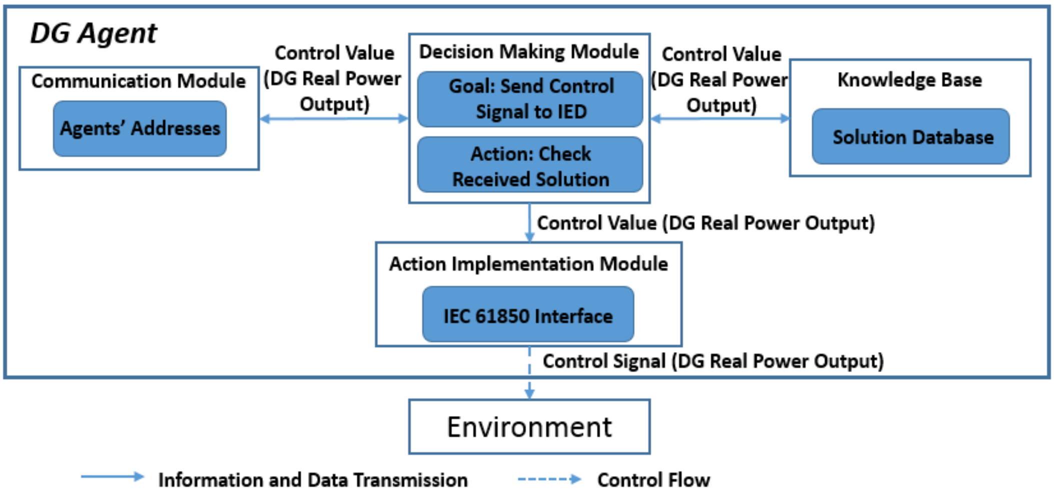


Figure 4.4 DG Agent Architecture

- **Decision Making Module**: Once the DG agent received a command from the solver agent, it aims to send a control signal to an IED. In unusual circumstances, the DG agent might receive multiple messages from different solver agents, e.g. one of the solver agents loses communication with other solver agents that result in multiple solutions sent to the DG agent. In order to resolve this issue, the DG agent checks every received solution before sending it to the IED to evaluate if it is better. If so, the DG agent updates the solution database to the received solution and sends it to the IED to control the corresponding DG. If not, the DG agent will not take any action. In addition, if the DG agent receives a message about lifting curtailment of DG that means the violation has been solved, it will send the lifting command to the IED and then empty its solution database in the knowledge module and wait for a solution message for another network violation.

- **Knowledge Base**: The knowledge base stores received solution if it is necessary. When the MAS starts, the solution (control signal) database in the knowledge base of the DG agent is empty and therefore it stores the first received solution into the database and sends it to the IED. If the DG agent receives another solution, it will compare it with the solution in the database. The DG agent sends another received solution to the IED and replaces the solution database with another received solution if another received solution is better. In order to determine the better solution, the DG agent aims to maximise DG's generation output, e.g. control solution curtails DG real power as less as possible.
- **Action Implementation Module**: To send a control signal to the IED, the DG agent is integrated with the IEC 61850 interface that allows it to connect with IEDs in the environment.

4.3.3. Solver Agent Design

The objective of the solver agent is not only to find the solution for the network violations but also to determine the best solution on the platform if there is more than one solver agent running. The architecture of a solver agent is shown in Figure 4.5.

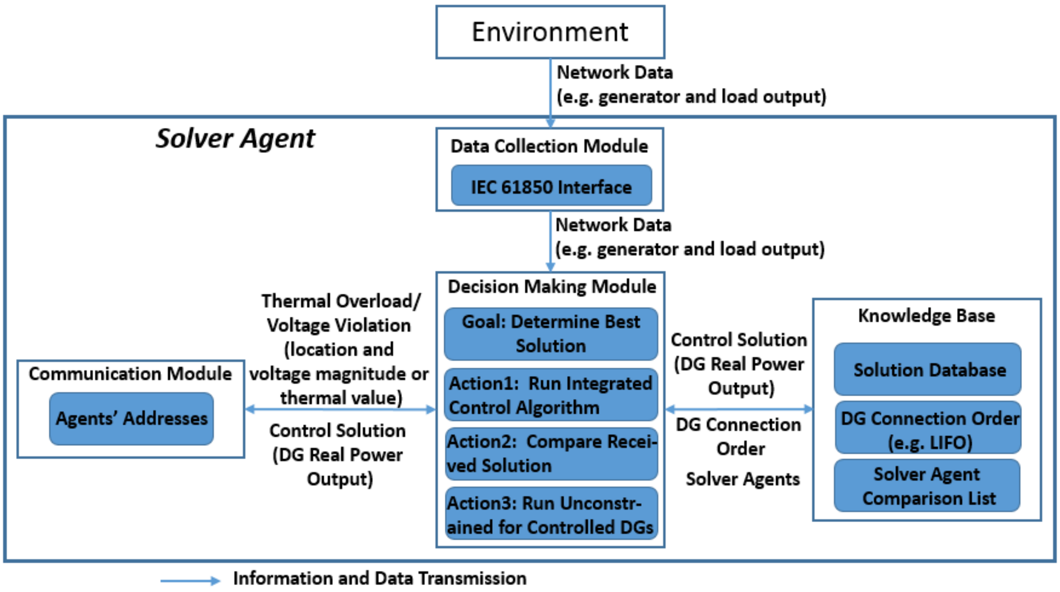


Figure 4.5 Solver Agent Architecture

- **Decision Making Module**: The goal of the solver agent is to find the best solution within the agent platform. There are three different actions that the solver agent can take. The first action is to run the embedded algorithm to determine the violation solution once solver agent receives violation message from the NVD agent. The second action is to negotiate with other solver agents to identify the best solution after its own solution has been determined. The third action is to determine when controlled DGs can be run unconstrained if the solver agent found the best solution among other solver agents.
    - *Run Integrated Control Algorithm*: Each solver agent is integrated with one control algorithm to search control solutions for network violations. After the solution has been determined, the solver agent stores it in the solution database for negotiation in the knowledge base module. In unusual circumstances, it might not find a solution, e.g. optimal power flow algorithm does not converge if the system load exceeds the steady-state loading limit [172], and therefore the solver agent needs to inform other solver agents about there being no solution found.

- *Compare Received Solution*: This action is triggered once the solution has been found and other solver agents are active on the platform. As long as the solution has been found, the solver agent checks if it received messages from other solver agents or not. If yes, it could receive either a *control solution* message or a *no solution found* message. If the solver agent receives a *control solution* message, it will compare it with its own solution to determine which one is better (e.g. maximise DG generation) and where applicable, replies with a *better solution message*. If it receives the *no solution found* message, it will remove this solver agent from comparison list in the knowledge base module. If the solver agent does not receive any solution message, it will send its own solution to other solver agents and wait for their responses, which could receive either a *better solution* message or a *no solution found* message. If the solver agent receives a *better solution* message, it checks with its own solution to decide whether it agrees. If yes, it replies with an *agree better solution* message. If not, it replies with its proposed *better solution* message. If the solver agent receives an *agree better solution* message, it will evaluate if it is the best solution in its solution database or not. If so, the solver agent updates its solution database. If not, the solver agent keeps the previous best solution in its solution database. As a result, there are four different types of solution messages (*control solution*; *no solution found*; *better solution*; and *agree better solution*) that each solver agent could receive. A flowchart of this is depicted in Figure 4.6. As soon as solver agent reaches an agreement with another solver agent, it removes this solver agent from the comparison list. Solver agent keeps negotiating with other solver agents until the comparison list is empty and therefore the best solution has been identified. At the end of agent negotiations, each solver agent knows the best solution within the platform. Moreover, the solver agent whoever found the best solution sends the control solution message to the DG agents to execute the control actions. This action is outlined in the pseudo-code illustrated in Figure 4.7.

- *Run Unconstrained for Controlled DGs*: If the solver agent found the best solution, it needs to keep monitoring when the controlled DGs can run unconstrained. To do that, the solver agent needs to run an integrated control algorithm. As soon as the controlled DGs can run unconstrained, it sends control signal messages to DGs to lift their curtailments.

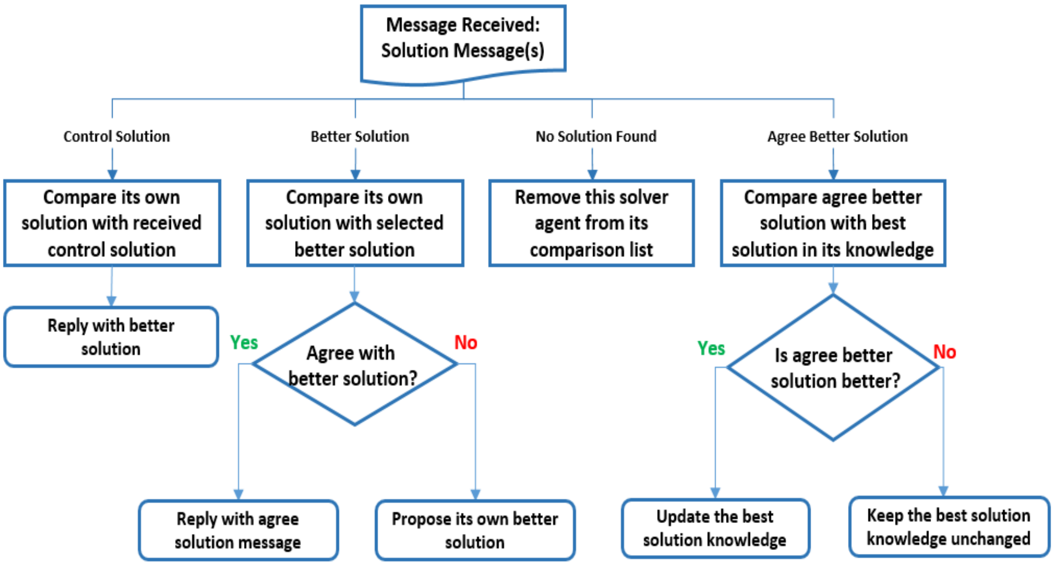
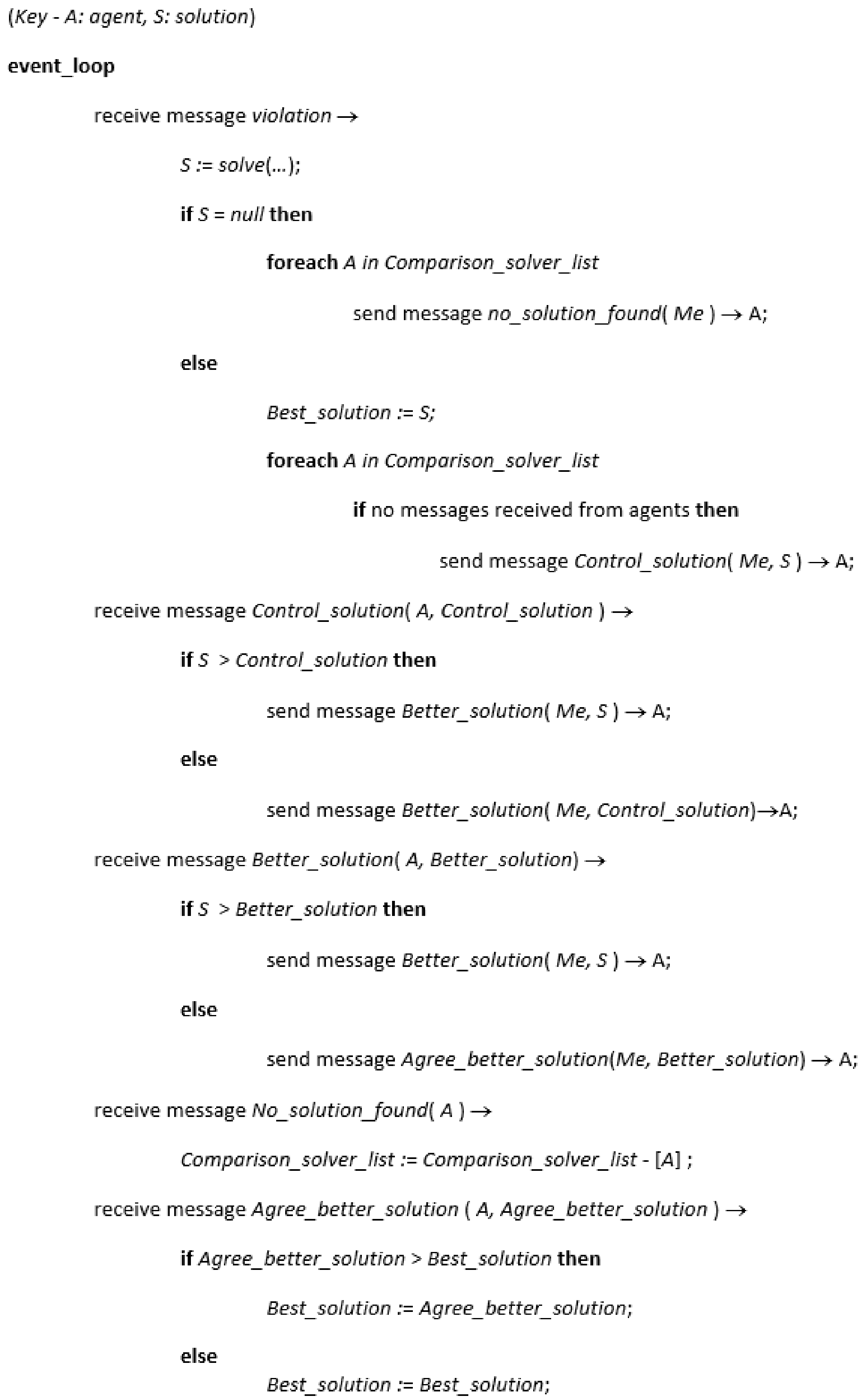


Figure 4.6 Flowchart of Four Different Type Solution Messages

- **Knowledge Base**: As the goal of the solver agent is to identify the best solution, it has a solution database in its knowledge and keeps it updated by negotiating with other solver agents. Furthermore, the solver agent has DG connection order knowledge to check control solutions, such as LIFO principle in the UK to curtail last connected DG first. To negotiate with all other solver agents on the platform, it has a comparison list to compare its own solution with other solver agents' solution.

- **Data Collection Module**: To determine the solution for a network violation, the data collection module obtains the required network data using the IEC 61850 interface.



```
(Key - A: agent, S: solution)
event_loop
    receive message violation →
        S := solve(…);
        if S = null then
            foreach A in Comparison_solver_list
                send message no_solution_found( Me ) → A;
        else
            Best_solution := S;
            foreach A in Comparison_solver_list
                if no messages received from agents then
                    send message Control_solution( Me, S ) → A;
    receive message Control_solution( A, Control_solution ) →
        if S  > Control_solution then
            send message Better_solution( Me, S ) → A;
        else
            send message Better_solution( Me, Control_solution)→A;
    receive message Better_solution( A, Better_solution) →
        if S  > Better_solution then
            send message Better_solution( Me, S ) → A;
        else
            send message Agree_better_solution(Me, Better_solution) → A;
    receive message No_solution_found( A ) →
        Comparison_solver_list := Comparison_solver_list - [A] ;
    receive message Agree_better_solution ( A, Agree_better_solution ) →
        if Agree_better_solution > Best_solution then
            Best_solution := Agree_better_solution;
        else
            Best_solution := Best_solution;
```

Figure 4.7 Pseudo-Code Design for Solver Agent Behaviours

4.4. Agent Interaction Design: Ontology and IEC 61850 Interface Design

In the agent interaction design stage, the suitable ontology of an agent message will be defined. The ontology should be application specific so that agents can understand the terminology in the agent communications. Moreover, an interface based on the IEC 61850 standard is designed to allow agents to interact with control devices. Details of the ontology and IEC 61850 interface design are provided as follows.

4.4.1. Ontology Design

An ontology is the data dictionary that supports co-operation and social ability and therefore is important for the operation of the MAS. For agent communication, an ontology provides agents with a shared vocabulary for use within the message content. Uschold et al [173] defined an ontology as "*an explicit formal specification of the terms in a domain and relations among them*". As a result, an ontology is application specific, and the designed ontology must suit the particular application that the agents operate on. In this research, a power system control application based ontology is proposed, which contains two classes. The first one is a *network violation* class and the second one is a *control action* class. These two classes are portrayed in Figure 4.8 and Figure 4.9 respectively.

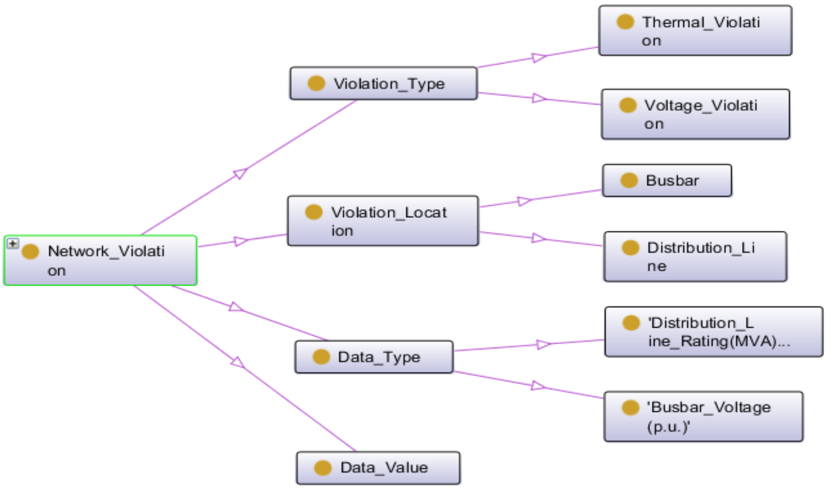


Figure 4.8 Network Violation Ontology

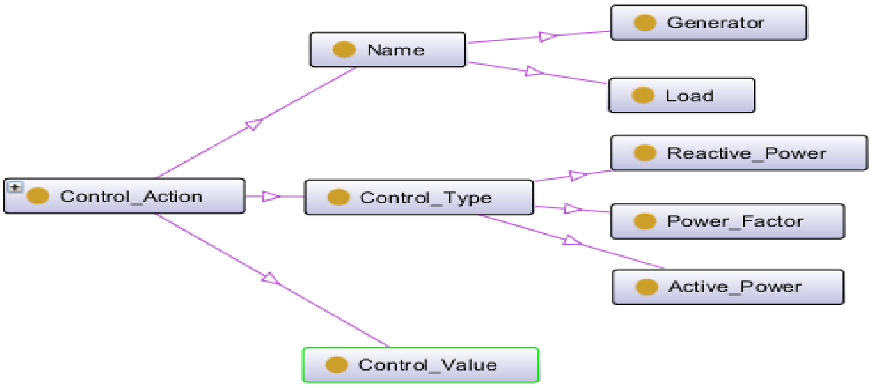


Figure 4.9 Control Action Ontology

The proposed ontology enables agents to provide particular types of data and information resources for network violation and control. The details of each class are shown below.

- **Network Violation**: This is used to describe the violation details on the network including the *violation type*, *violation location*, *data type* and *data value*. For *violation type*, it can distinguish between thermal and voltage violation. The *violation location* will be either the distribution line for thermal violation or busbar for voltage violation. Based on the violation location, the *data type* will be thermal line rating for distribution line and busbar voltage. Finally, the *data value* is the violation value measured in the network.

- **Control Action**: This defines the violation solution details determined by solver agent. It is comprised of *control name*, *control type*, and *control value*. The *control name* is the controllable device, such as generator, and load. The *control type*, on the other hand, is the control parameter, e.g. active power or reactive power of generator and load. The *control value* is the set point for control parameter.

4.4.2. IEC 61850 Interface Design

The interoperability between different grid monitoring, control, and supervisory systems is essential for viable and flexible power system operation and control. An open-standard interface, as mentioned in [29, 31, 170], such as IEC 61850 promotes interoperability between devices within substations, and can achieve system-wide interoperability. For the proposed fully integrated MAS platform, an IEC 61850 interface is designed to communicate between IEDs and the agent platform. To develop the IEC 61850 interface within the agent, an open-source platform for rapid-prototyping protection and control schemes with IEC 61850 [174] is used. This platform can generate the data model and communication code required for an IED to implement a publisher-subscriber Generic Object-Oriented Substation Event (GOOSE) and Sampled-Value (SV) messaging. GOOSE and SV messages perform the role of transferring the data sets according to defined logical nodes over a communication network between IEDs. This platform uses a System Configuration Description (SCD) file which defines all IEDs, data sets, and communications within a power system, to automatically generate C programming code to implement IEDs. In addition to generating C code to implement communications (GOOSE and SV), the platform also automatically generates Python or Java libraries from the C code. Therefore, this platform provides the option to implement IEDs in other high-level programming languages. More details about the code generation process can be found in [174]. As a result, agents can embed generated data model and communication codes to implement a GOOSE or SV messages with other IEC 61850 IEDs to either send or receive required data. With the help of IEC 61850 interface, the proposed fully integrated MAS architecture could be geographically distributed and therefore agents can run in a number of IEDs. An example of a control architecture to deploy MAS in the physical network associated with IEC 61850 interface is provided in section 4.6.

4.5. Architecture Design: Implementation of Fully Integrated MAS

In this research, a programming environment for the simulation of agent societies (Presage) [125] is used to implement the proposed fully integrated MAS. Presage is a simulation platform for rapid prototyping of societies of agents. Presage is a Java based programming environment that offers designers a flexible and generic set of Java classes, interfaces, and tools that can simulate agent societies. Presage2 is the second version of Presage which extends the original Presage platform by adding the support for declarative rule specifications and increasing modularity [32]. Moreover, Presage2 can simulate large, heterogeneous populations, multiple different networks, inter-agent communication, policy modelling, the physical environment, event recognition, data logging, and visualisation [175]. As a result, Presage2 is selected to simulate the proposed MAS in this research.

4.5.1. Presage2 Structure

The Presage2 platform is a discrete time-driven simulator (i.e. sequential steps). At each time step of the simulation, each agent can execute agent step function (e.g. submit actions). Presage2 platform treats agents as collections of behaviours. A behaviour is one unit of functionality that executes completely before another behaviour can run. There are no constraints on agent behaviours on the Presage2, and as such, it allows multiple actions per step and unbounded computation. The scheduling of behaviours is based on a simple queue. The agent developer must implement the behaviours to run the correct one at the correct time. This can be achieved by checking whether certain conditions have been met, and skipping the behaviour if they have not. The next behaviour would then run in the queue.

A Presage2 agent is created by extending the agent class (named *AbstractParticipant*), provided by the Presage2 library. By overriding the *execute()* method, an agent can be made to do a specific task. An example of a basic Presage2 agent which implements a random walk in the environment is illustrated in Figure 4.10. Behaviours generally deal with one sort of event that occurs, such as handling a particular type of message. A simple behaviour that waits for an *inform* type of message, which could be used as an agent location information, is shown in Figure 4.11. If the behaviour is run when no message has been received, it simply allows the next behaviour to run. It can be seen from this short example in Figure 4.11 that the Presage2 agent platform hides the low-level detail of messaging, by providing the *getMessage()* method to check received messages and the *sendMessage()* method to send out a message to other agents.

```
public class MyAgent extends AbstractParticipant{

    @Override
    protected void processInput(Input arg0){

    }
    Location myLoc; // The agent's location

    /*This method creates agent's id, name and other related parameters */
    MyAgent(UUID id, String name, Location myLoc){
        super(id, name);
        this.myLoc = myLoc;

        }

    /*Creates a share state */
    @Override
    protected Set<ParticipantSharedState> getSharedState(){
        Set<ParticipantSharedState> ss= super.getSharedState();
        ss.add(ParticipantLocationService.createSharedState(getID(),myLoc));
        return ss;
    }

    ParticipantLocationService locationService;

    /*Initialise agent */
    @Override
    public void initialise(){
        super.initialise();
        try {
            this.locationService =
getEnvironmentService(ParticipantLocationService.class);
        } catch (UnavailableServiceException e){
            logger.warn(e);
        }
    }

    /*Add behaviours into execute function to do something */
    @Override
    public void execute(){
        myLoc = locationService.getAgentLocation(getID()); // Get agent's
own location

        logger.info("My location is: "+ this.myLoc);  // Display agent's own
location in console

        int dx = Random.randomInt(2) - 1;
        int dy = Random.randomInt(2) - 1;
        Move move = new Move(dx, dy);   // Determine next location to move

        /*Act the move behaviour*/
        try {
            environment.act(move, getID(), authkey);
        } catch (ActionHandlingException e) {
            logger.warn ("Error trying to move", e);
        }

}
```

Figure 4.10 Agent Example in Java Code of Presage2

```
@Override
public void execute(){
    //Get its own location
    myLoc = locationService.getAgentLocation(getID());
    //Get other agents' addresses in platform
    Set<NetworkAddress> connected = network.getConnectedNodes();
    //Send messages to agents in the platform
    for (NetworkAddress addr : connected){
        UnicastMessage<Location> msg  =
                new UnicastMessage<Location>(Performative.INFORM, network.getAddress(), addr, getTime(), myLoc);
        this.network.sendMessage(msg);
    }

    //Check received messages
    List<Message<?>> lmsg = this.network.getMessages();
    //Get each message details
    for(int i = 0; i<lmsg.size();i++){
        MyUnicastMessage<?> msg = (MyUnicastMessage<?>) lmsg.get(i);
        if(msg.getData() instanceof Location){
            logger.info("Get location: " + msg.getData() + " from" + msg.getFrom());
        }
    }
```

Figure 4.11 Agent Communication Example in Presage2

The simulation environment of Presage2 is shown in Figure 4.12. In the platform environment, agents collaborate with each other through suitable agent communications through the agent message transport system. The agent message transport system is the central switchboard for messaging in the platform, rather than letting the agents communicate with each other directly. Designers can add constraints to allow the blocking of messages or modify the messages before delivery. The message transport system will either send the message via TCP/IP sockets or queue the message to the recipient's inbox queue, depending on whether the recipient is internal or external to the platform.

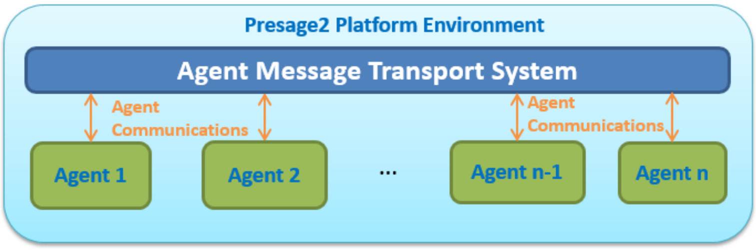


Figure 4.12 Simulation Environment of Presage2

4.5.2. Agent Communication Language of Presage2

For the Presage2 platform, it defines its own ACL. In order for agents to parse and interpret information exchanged between them, there must be an agreed-upon message syntax, where the message format in Presage2 is:

message (Performative, Sender, Receiver, Time, Content)

The *Performative* defines the type of communicative act being performed, such as *inform* or *request*, which is the same performative as FIPA-ACL standard. The *Sender* and *Receiver* denote the agents that are sending and receiving the message respectively. *Time* of the message is the time at which the message was sent. The *Content* is the information (e.g. network data) that is communicated between the agents. The message example is shown below:

message (Inform, Agent001, Agent002, 23, Location<2;3;0>)

Presage2 offers effective agent communication by allowing greater control for designers to develop autonomous behaviours and communication interactions. To do that, Presage2 allows the user to develop a protocol that agents can use for specific conversations, such as trade protocol to trade with each other. Hence, Presage2 provides greater flexibility for agent design, simulation, and communications from its structure and ACL.

4.6. Control Architecture for Fully Integrated Multi-Agent System

To study and demonstrate the fully integrated MAS for ANM within a power system, the following control architecture is proposed, which contains two layers, including the power system layer, and the multi-agent system layer, that is shown in Figure 4.13. The lower power system layer presents a physical electric network, and the upper multi-agent system layer handles the communication, decision-making, and cooperation between agents within the developed MAS.

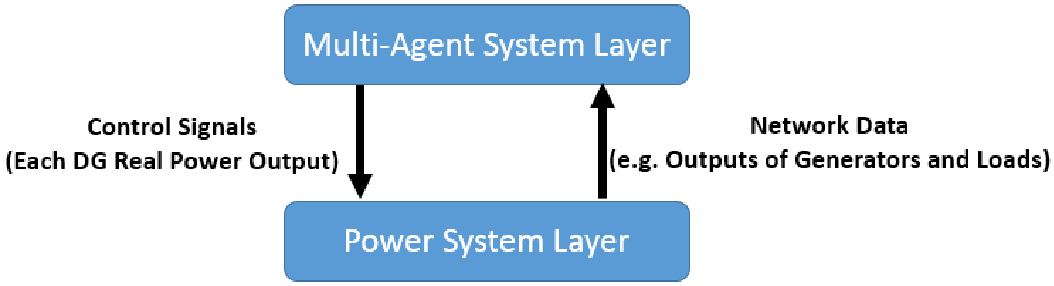


Figure 4.13 Control Architecture for Proposed Fully Integrated MAS

The power system layer is a physical environment that is interfaced with the multi-agent system layer. The power system environment could consist of multiple network components, including generators, transformers, lines, and loads. For network control and monitoring, IEDs can be used and interfaced with the multi-agent system layer to send network data or receive control signals. In order to do that, agents in the fully integrated MAS have been integrated with the IEC 61850 interface to facilitate data exchange between the power system layer and the multi-agent system layer.

The multi-agent system layer has been developed to fully integrate MAS to achieve centralised control, which aims to identify network events based on the received network data from the power system layer and therefore trigger solver agents to analyse network events and determine solutions. Each agent in the multi-agent system layer has been designed with certain communication capabilities. All communications among agents are implemented through a dedicated ACL and therefore agents can communicate with each other by exchanging information within the agent platform to solve control and management problems. Moreover, different solver agents with various functions are placed in the multi-agent system layer to negotiate the final solution using the designed knowledge, and actual control signals can be sent to the related IEDs in the power system layer by the DG agents.

An example of control architecture to deploy MAS in the physical network associated with an IEC 61850 interface is illustrated in Figure 4.14.

In the power system layer, it has an 11 kV radial network that contains six IEDs that connect with an agent platform through GOOSE or SV messaging (black line). In the multi-agent system layer, the fully integrated MAS contains one NVD agent, several solver agents and DG agents which are running at an agent platform and coordinating with each other via agent communications (green dashed line). The developed MAS can measure network data from six different IEDs by GOOSE or SV messaging (black line) and then the NVD agent monitors the network for violations. After that, the NVD agent informs the solver agents once there is a network violation to determine the control solution by agent communications within the agent platform. Each solver agent cooperates with other solver agents to decide the best solution within the agent platform and then send the best solution to DG agents. Eventually, the DG agents send commands through IEC 61850 interface to IEDs to control DG separately.

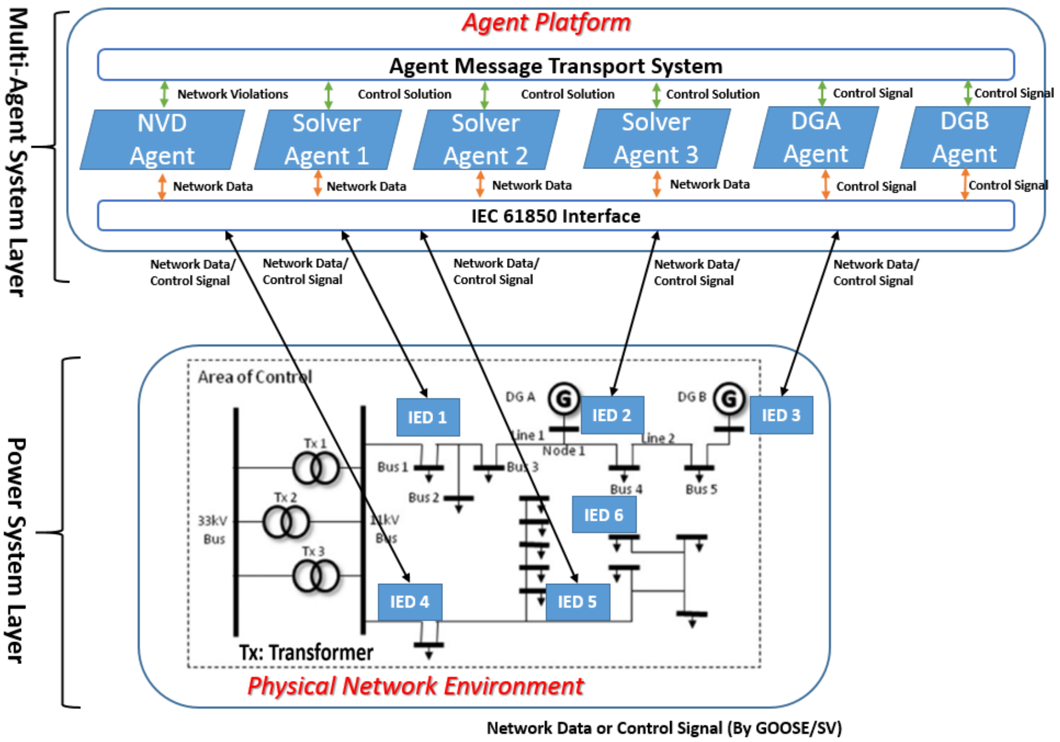


Figure 4.14 MAS Physical Network Deployment with IEC 61850 Interface

As a result, managing and controlling the power system for demonstrating the performance of the fully integrated MAS for ANM application is based on this control architecture and the details of the related case studies are presented in Chapter 5. However, the power system layer requires building. Therefore, a power system simulator is modelled to emulate the power system layer and details are provided in next section.

4.7. Power System Simulator Design

The power system simulator aims to provide the operation of power systems. In this research, the IPSA software [171] is used to build the power system simulator.

IPSA is a commercial power system simulation software that enables the simulation of detailed engineering analysis. IPSA software is a user-friendly and highly interactive software, and its interface is shown in Figure 4.15. From the user interface, it can be used to explain power system operation and status. In addition, IPSA provides power flow analysis, fault level analysis, harmonic analysis, transient stability analysis, and protection analysis.

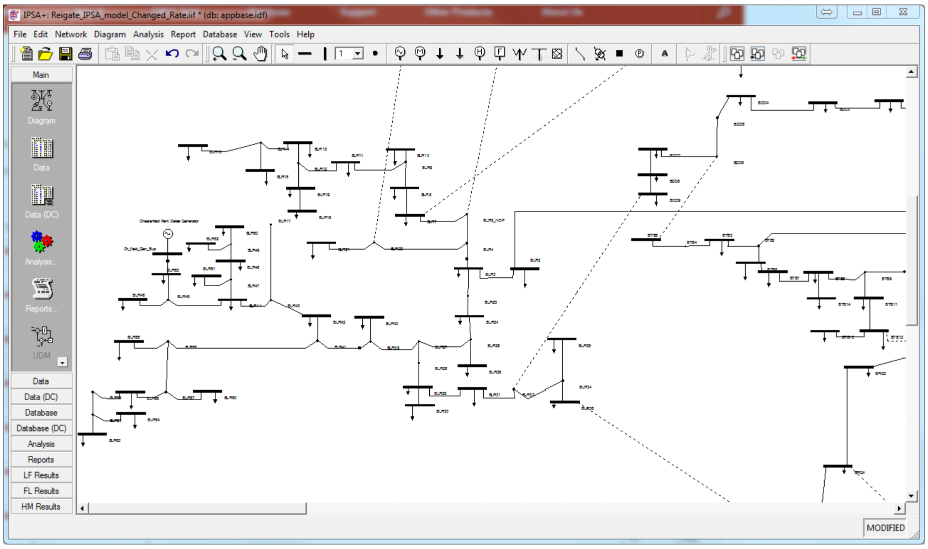


Figure 4.15 IPSA Software User Interface

Furthermore, IPSA has a scripted extension library for Python and therefore that enables it to be run without a user interface within a Python shell. As a result, the power system simulator is developed based on Python script to provide network data, such as real and reactive power of generator and load, as well as simulating the control response for network violations. In order to connect the power system simulator to the agent platform, the IEC 61850 interface has been applied, developed using the same open source platform [174] mentioned previously. As this IEC 61850 interface can be developed in Python code, it can be integrated into the Python shell and therefore has the capability to interface between the power system simulator and the agent platform. The power system simulator diagram is presented in Figure 4.16.

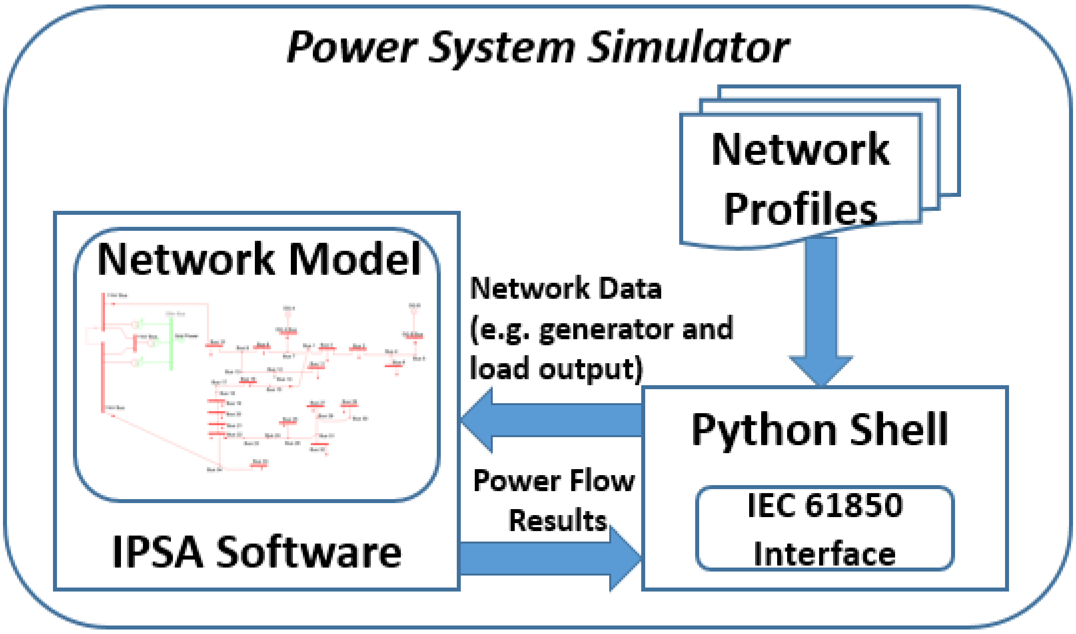


Figure 4.16 Power System Simulator Diagram

The power system simulator contains a python shell to run IPSA with the network model by importing the network profiles, such as generation and load data. Once the power system simulator is running, the Python shell can send network data to the agent platform and receive control signals from it to simulate control responses by the embedded IEC 61850 interface.

4.8. Summary

In this chapter, a fully integrated MAS architecture has been proposed for ANM to control and management of power system. It provides details of MAS design and development according to the five-stage methodology, including agent types, agent internal behaviours, agent interactions, and architecture. There are three types of agents that are defined, including the NVD agent, DG agent, and solver agent. As such, an advanced structure of each agent is provided with insight into the decision making, communication, action implementation, data collection, and the knowledge base. This chapter also exhibited the appropriate system integration and agent ontology design capabilities and shows that agents can achieve appropriate control decisions by interacting with other agents to negotiate and reach an agreement. Moreover, the fully integrated MAS provides advanced control functions alongside interoperability through the creation of an IEC 61850 interface. Together, this allows for the full specification of control within future ANM to change and adapt over time.

In addition, the agent simulation platform named Presage2 has been introduced with its structure, features and provides flexible ACL. Presage2 has been applied within the power system domain, such as distributed control of frequency by agent negotiation ability via agent communications. A control architecture for simulating the proposed MAS for ANM is presented with two different layers, including the multi-agent system layer and the power system layer. An example of implementing the proposed MAS in the power network based on the control architecture is demonstrated. Furthermore, the power system simulator is designed to execute control responses from the MAS based on the power system simulation software.

In the next chapter, case studies of fully integrated MAS for ANM application in power system control, including voltage control and power flow management, associated with different network conditions, will be provided in detail. Moreover, the developed MAS architecture will be extended for self-organising mechanism to enhance the capability of ANM to accommodate to changes in its environment.

Chapter 5

Case Studies and Results

5.1. Introduction

The previous chapter discussed how to develop a fully integrated Multi-Agent Systems (MAS) for Active Network Management (ANM) within different power system control applications, with associated flexibility, extensibility, and interoperability features. By implementing the Network Violation Detection (NVD) agent, Distributed Generation (DG) agent and solver agent, the fully integrated MAS is flexible and can therefore deploy various control functions. This flexibility enables the MAS to provide various control solutions following a network event.

In this chapter, two main case studies are presented. The first case study demonstrates the performance of the fully integrated MAS within various scenarios. Two different control algorithms are used in the first case study. Detailed functional design and configuration of each solver agents based on the control algorithms will be provided. The scenarios in the first case study will outline how the fully integrated MAS can be deployed for ANM within power system control applications, including power flow management and voltage control. The first scenario will show the extensible capability of a fully integrated MAS to be able to activate different solver agents to solve different network violations. Furthermore, the second scenario will demonstrate the negotiation capability between multiple solver agents on the agent platform to determine the best solution for a network violation. Moreover, the third scenario simulates how the fully integrated MAS handles a network violation due to the network changes, such as a new DG connection, without changing the core function of the agents. The fourth and final scenario for the first case study proves that a fully integrated MAS can deal with multiple events occurring simultaneously, including thermal overload and voltage violation. This first case study demonstrates the fully integrated MAS on an operational 11 kV radial distribution network model for centralise control. The second case study is an extension of the fully integrated MAS architecture for self-organisation. The scenario in this case study shows agents can self-organise and divide into several groups within the MAS to achieve voltage regulation for a large distribution network. The self-organising mechanism can enhance the ANM capability to accommodate changes to its enviorment.

5.2. Case Study One: Performance Testing of Fully Integrated MAS Architecture

The development details for first case study are discussed in this section. It introduces the selected control algorithms and specifies implementation details of these control algorithms within the solver agents. Moreover, it describes the network model and simulation environment for case study one. Finally, the results of each scenario are presented.

5.2.1. Control Algorithms Selection

To solve network violations, such as thermal overload and voltage excursion, a control algorithm is required to translate the problem into a mathematical model in order to solve problems under certain conditions. For example, an objective function of a mathematical model is to identify a control solution to mitigate network violations by minimising total curtailment of active power from controllable generation units. Besides the objective function, other constraints are also required to be considered. One constraint is the financial incentive based contracts between the Distribution Network Operator (DNO) and DG that clarifies the control priority list, such as Last In First Off (LIFO) in the UK. The aim of considering DG contractual agreements within the control algorithm is to seek solutions that closely follow this access principle for the majority of operating conditions. In addition, the application of access principle will allow for better use of DG resources and network infrastructure. Another constraint could be a DG control action constraint which is minimisation of active power curtailment of involved DGs. Furthermore, minimal cost for solving network violations might need to be considered as a cost constraint. In case study one, there are two control algorithms that are selected and adapted to compute solutions for power system control by applying the above constraints. These two algorithms are a Constraint Satisfaction Problem algorithm (CSP) [74] and an Optimal Power Flow algorithm (OPF) [75], and details of control algorithms are explained in Chapter 2. Details of modelling and configuring of CSP and OPF into solver agent for the fully integrated MAS applications are provided in Sections 5.2.2 and 5.2.3.

5.2.2. CSP Solver Agent Design Specifications

Based on the CSP algorithm described in Section 2.4.1.1, modelling thermal constraints and voltage violation problems as a CSP is explained. Furthermore, the implementation details of the developed CSP as a control function inside the solver agent are presented.

5.2.2.1. Power System Control with CSP Technique

In [74], it has developed a CSP for power flow management, including a set of variables with finite discrete domains and a set of constraints. In this case study, the developed CSP model [74]is applied to test for both power flow management and voltage control. The details of CSP modelling for power flow management and voltage control are provided as follows.

The variables selected for the CSP are the controllable DGs and a domain of discrete values is composed of active power output set-points [74]. DG can be controlled by discrete trim or trip signals i.e. the generator reduces its active power output to a given threshold or is tripped to mitigate network violations. Alternatively, the DG can operate without curtailment. Hence, the active power output set-points in the domain is the curtailment bandings for each generator, e.g. [1, 0.8, 0.5, 0], where 1 represents operation without curtailment and 0 represents the tripped state.

In addition to variables and their domains, there are constraints that need to be modelled. In this thesis, three constraints were considered:

- **Network Constraints**: Any potential generator set-point solution must solve network constraints (such as thermal overload and voltage constraint) under the current network conditions. To check each solution for the network constraints, a power flow engine can be used to run power flow analysis.
- **Generator Contractual Constraints**: Generator access rights is another constraint as DG units connected to a network is typically based on contracts with connection order in some countries. As a result, a valid control solution must respect the connection principle. In the case study one, the LIFO connection principle is used which means the last connected DG will be curtailed first if there is a violation in the network.
- **Maximise Generation Constraints**: In many situations, there can be many solutions that meet the two constraints above. For example, all generators can be tripped in order to solve network overloads which are not expected to occur within the network. As a result, the solutions that can maximise DG active power generation is required. This constraint is used to prioritise potential solutions based on maximising DG active power generation and determine the best solution.

From the above CSP modelling for network control problems with selected variables, finite discrete domains, and specific constraints, the CSP can be formulated as follows:

$$CSP = (V_{Gens}, D_{Gens}, C), \qquad (5.1)$$

Where:

$$V_{Gens} = [Gen_1, Gen_2, Gen_3, \ldots, Gen_n]; \qquad (5.2)$$

D_{Gens} is control set-point for variables: (5.3)

$$D_{Gen1} = [1, \ldots, 0];\ D_{Gen2} = [1, \ldots, 0];\ D_{Gen3} = [1, \ldots, 0];\ \ldots;\ D_{Genn} = [1, \ldots, 0]; \quad (5.4)$$

C is the constraint:

$$C_{Network\ Constraint} = [|S_{ij}| \leq S_{ij}^{max}\ ;\ V_i^{min} \leq V_i \leq V_i^{max}]; \quad (5.5)$$

$$C_{Generator\ Contratual} = [Gen_1 = 1, Gen_2 = 2, Gen_3 = 3, \ldots, Gen_n = n]; \quad (5.6)$$

$$C_{Maximise\ Generation} = [max \sum_{i=1}^{N_{Gen}} P_{Geni}]. \quad (5.7)$$

5.2.2.2. Implementing the CSP Function in a Solver Agent

To realise the CSP functionality designed above for power system control problems, a constraint solver [176] in combination with a power flow engine based on the IPSA software [171] is applied. The constraint solver is an open-source Python library named ‘Python Constraint’ that provides solver for the finite discrete domain CSP. As a result, the constraint solver allows the user to develop self-defined finite discrete domain CSP problems. The power flow engine is used to check the network constraints for any potential solution as discussed above. In order to interface the constraint solver with the power flow engine in IPSA, the Python script of IPSA is used to invoke the power flow engine from a Python program. Therefore, the constraint solver runs the power flow engine and receives power flow results. Figure 5.1 shows the developed CSP function with required inputs (variables, domains, and constraints), software interactions and the resultant output.

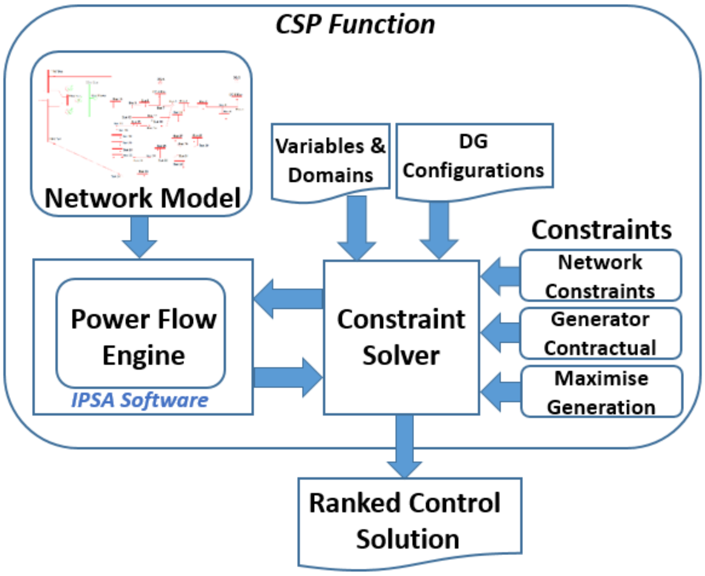


Figure 5.1 CSP Function Modelling

When a network violation occurs, the CSP solver searches a solution according to controllable DGs (variables & domains) and three constraints. To check network constraints, it interfaces with the power flow engine to receive power flow results to check whether the potential solutions solve the network violations or not. For the generator contractual constraints, DG configuration data is used which provides each DG's generation output and DG connection order within the network. The CSP algorithm is then able to check if the possible solutions follow the generator connection order or not. Finally, the best control solution is determined based on the maximise generation constraints.

To integrate the developed CSP function into a solver agent, it requires the solver agent to call the CSP function written in Python and capture its output in Java. To do that, it involves two Java classes, which are the *Runtime* class and the *Process* class. For the *Runtime* class, it uses *exec()* method to run a Python program and therefore a *Process* object is returned. Then, the *getInputSream()* methods of the *Process* class is used to read the output of the Python program. The code shown in Figure 5.2 provides a working example of using Java *Runtime* class and *Process* class to call a Python program in Java.

```
package runPython;

import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.io.LineNumberReader;

public class Test2 {
    public static void main(String[] args){
        try{
                System.out.println("start");
                Process pr = Runtime.getRuntime().exec("python test.py");
                BufferedReader in = new BufferedReader(new
                        InputStreamReader(pr.getInputStream()));
                LineNumberReader lineNumber = new LineNumberReader(new InputStreamReader(pr.getInputStream()));
                String line;
                while ((line = in.readLine()) != null) {
                    System.out.println(lineNumber.getLineNumber()+line);
                }
                in.close();
                pr.waitFor();
                System.out.println("end");
        } catch (Exception e){
                    e.printStackTrace();
                }
        }
}
```

Figure 5.2 Example of Run Python Script from Java

5.2.3. OPF Solver Agent Design Specifications

To deploy the OPF for power system control problems, the DG access right needs to be considered in order to provide a solution that stays consistent with operation in the distribution network. As a result, the OPF models the LIFO principles. The details of the integration of the OPF control function into the solver agent are presented in the following sections.

5.2.3.1. OPF Technique for Power Flow Management

Based on the general OPF formulation (2.15) – (2.21) discussed at Section 2.4.1.2, OPF has been used to achieve power flow management in [75]. In this thesis, the OPF functions has been extended to both power flow management and voltage control by adding the voltage constraints (2.21). In order to add the LIFO principle into OPF function, the only change within the definition and formulation of OPF is the cost term $C_i(P_{gi})$ in the objective function (2.15) [75]. The cost term $C_i(P_{gi})$ in the general OPF formulation represents generation costs associated with the electricity production [75]. In order to let the OPF formulation reflect generator connection order, each generator is allocated a constant cost value and cost term is defined as:

$$C_i(P_{gi}) = u_{gi}P_{gi} \qquad (5.8)$$

Where, u_{gi} is a constant assigned to each DG to reflect its connection order. For the LIFO connection principle, it means the last connected DG will be curtailed first. For the first connected DG, it will be allocated the lower value of u_{gi}. The last connected DG, on the other hand, will be allocated the highest value. However, the u_{gi} does not represent actual generation costs, which is used to state a connection order and would be considered to solve network constraints.

Once there is a thermal violation within the network, the curtailment level (P_{gi}^{OPF}) of each involved DG is calculated by solving the OPF. In order to send a control signal (CS_{gi}) to its related DG, the control signal is calculated according to the generator's curtailment output (P_{gi}^{OPF}) and actual rated output (P_{gi}^{max}) as shown below:

$$CS_{gi} = P_{gi}^{OPF} \div P_{gi}^{max} \qquad (5.9)$$

5.2.3.2. Implementing the OPF Function in a Solver Agent

To achieve the OPF function mentioned above, PowerWorld [177] power system simulation software is used. Within PowerWorld, an OPF solver is provided and this is used to model the minimum cost objective function and related constraints as shown in formulations (2.15) – (2.21). Moreover, PowerWorld supports Python script and therefore enables the invoking of the OPF solver in PowerWorld without running its user interface from a Python program. Hence, the OPF function diagram with required input (cost model, constraints and DG configurations) is illustrated in Figure 5.3.

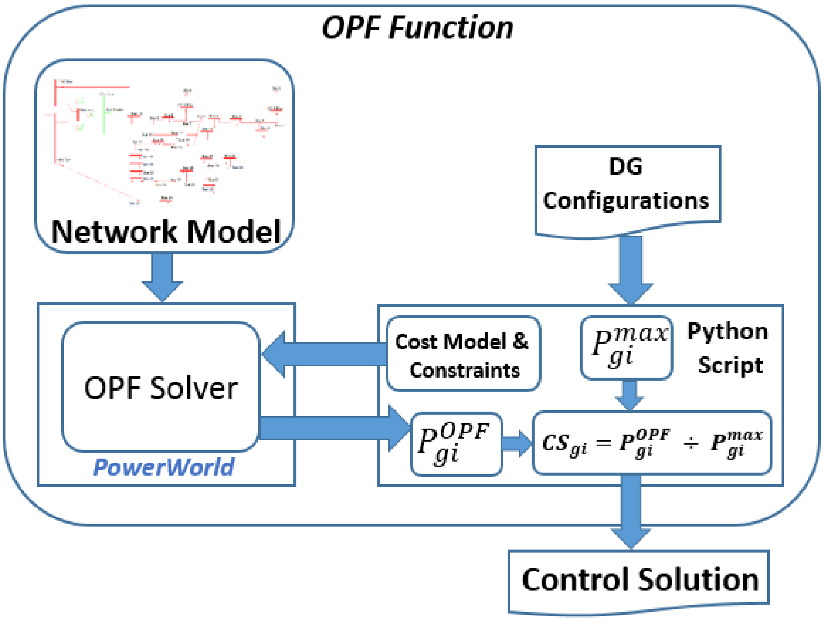


Figure 5.3 OPF Function Modelling

To integrate the developed OPF function into the solver agent, the same approach of applying the Java *exec()* method within the *Runtime* class to call OPF Python code from Java is used. Then, the output from OPF function can be captured by using the *getInputSream()*method of the *Process* class in Java.

5.2.4. Network Model and Simulation Environment

In order to apply the developed fully integrated MAS into a distribution network environment, an actual network model is selected. This is an 11 kV radial distribution network in the UK. This network provides DG penetration connected at different busbars and the simulation environment to test the MAS performance under this network model is presented below.

5.2.4.1. 11kV Radial Distribution Network

The 11kV radial distribution network with two DGs connected is shown in Figure 5.4 and DG capacity of this case study is presented in Table 5.1. Moreover, the associated network data for case study one simulation, such as line parameters, are presented in the Appendix A. For case study one, the system base power is 10 MVA and base voltage is 11 kV.

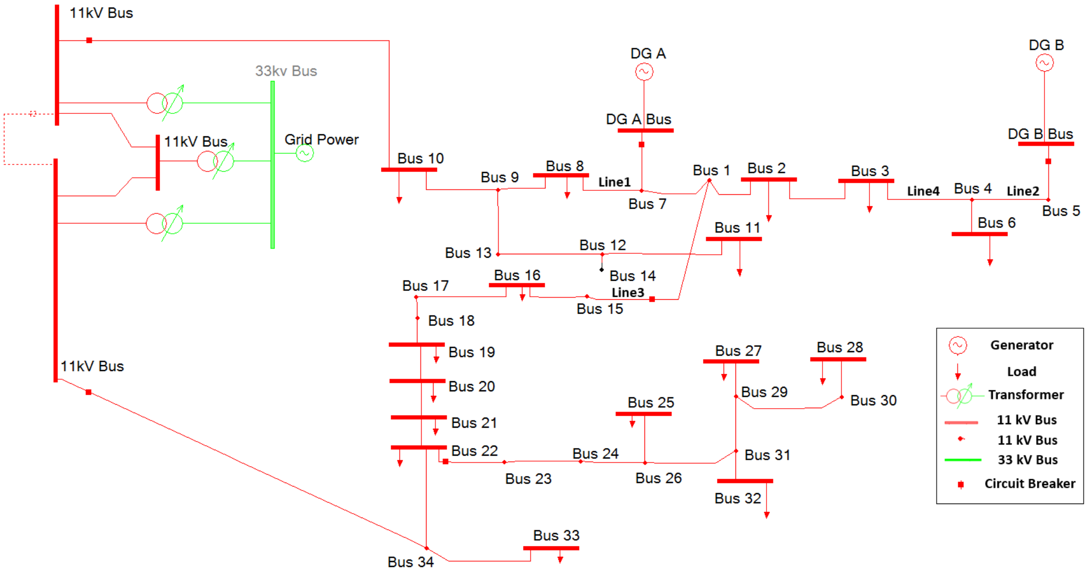


Figure 5.4 11 kV Radial Distribution Network

Table 5.1 Capacity of Each DG Connected to 11 kV Radial Network

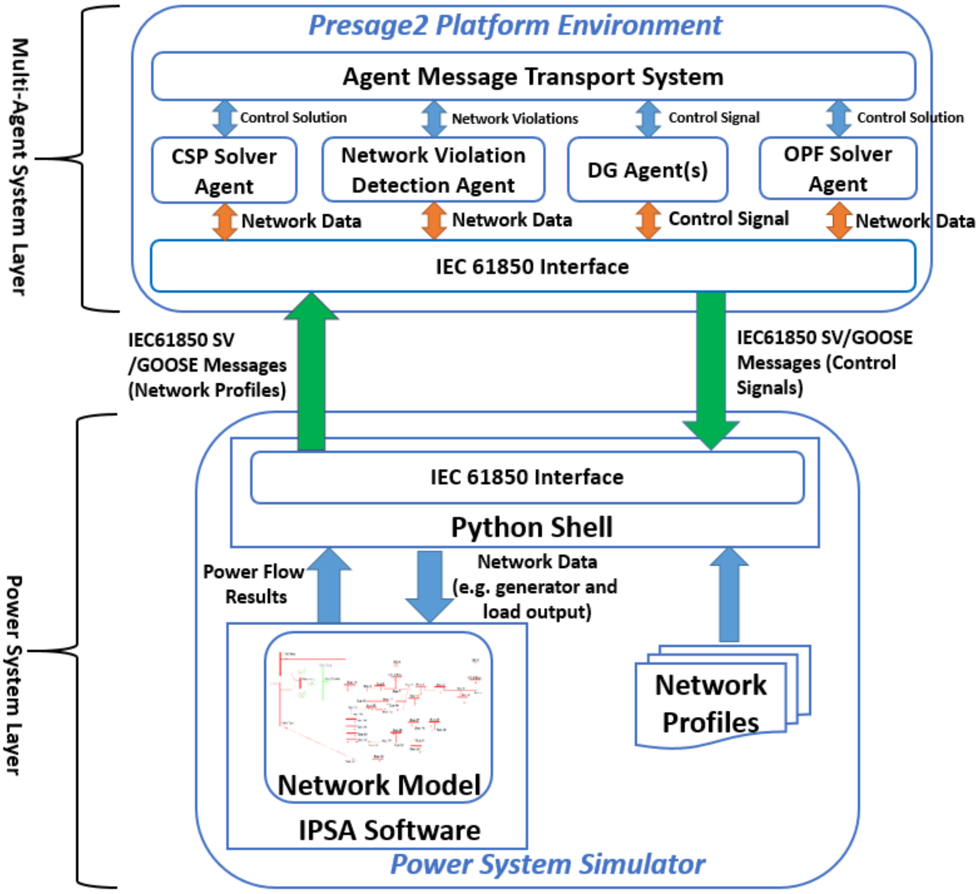
DG Name	Capacity (MW)
DG A	**1.6**
DG B	**2.0**

In the 11kV radial distribution network, it contains 40 buses and 16 loads. For each scenario in the case study one, network profiles, including DG generation output and loads, were developed that allow network violations to be manipulated. The voltage magnitude of this simulation network cannot exceed the normal operation limits by more than 3% and therefore the acceptable limit is between the lower bound 0.97 p.u. and upper bound 1.03 p.u..

As mentioned before, the CSPS and OPFS agent invoke power systems simulation tools (in this case IPSA and PowerWorld software) to achieve power flow analysis separately. The comparison of power flow results between IPSA and PowerWorld has been checked, based on the same network data (can be found within Appendix A). Both IPSA and PowerWorld give the same power flow results.

5.2.4.2. Simulation Environment

The simulation environment for case study one, based on the proposed control architecture in Chapter 4, is illustrated in Figure 5.5.



SV: Sampled Values; GOOSE: Generic Object Oriented Substation Event

Figure 5.5 Close-Loop Simulation Environment for Case Study One

At the multi-agent system layer, developed agents, including NVD agent, DG agents, CSP Solver (CSPS) agent, and OPF Solver (OPFS) agent, are running on the Presage2 platform. Within the agent platform, agents are able to be added when they are required. In addition, agents communicate with each other by the agent communication language of Presage2 and defined ontologies to achieve control goals.

For the power system layer, a sequential steady-state power system simulator calls network profiles every second, including loads and each generator output, to represent the system steady-state conditions during the simulation. The results of the following scenarios are based on the different network profiles. Moreover, this simulator emulates measurement that would be provided by IEDs in the field and simulates responses to the control signals issued by the agent platform. To do that, the IEC 61850 interface has been embedded into the agent platform and power system simulator to send/receive network measurements and control signals.

5.2.5. Case Study One Results

There are four scenarios in case study one to demonstrate the fully integrated MAS architecture performance for ANM under centralise control:

- Scenario 1: Connect different solver agents to deal with various network violations, including voltage violation and thermal overload;
- Scenario 2: Negotiation between multiple solver agents;
    - Scenario 2-A: Control solutions respect the LIFO agreement;
    - Scenario 2-B: One of the control solutions does not respect the LIFO agreement.
- Scenario 3: MAS response to a new DG connection;
- Scenario 4: Multiple network violations occurring simultaneously.

For each scenario, the DG connection order is presented in Table 5.2. In Table 5.2, ‘1’ means the DG is the first connected and ‘2’ means the DG is the second connected so that this DG is required to first respond to the network violations if there are only two DGs connected in the network. For scenario 3, ‘3’ means the DG is the last connected and this DG needs to be curtailed first when there is a network violation.

Table 5.2 Distributed Generator Connection Order

DG Unit	**Scenario 1**	**Scenario 2-A**	**Scenario 2-B**	**Scenario 3**	**Scenario 4**
DG A	1	1	2	1	1
DG B	2	2	1	2	2
DG C	-	-	-	3	-

5.2.5.1. CSP Algorithm Control Specification

In each of the following scenarios, the variables, domains, and constraints of the CSP algorithm are set as follows:

$V_{Gens} = [DG_A = 1.6\,MW, DG_B = 2MW]$; (5.10)

$D_{DG\,A} = [1, 0.8, 0.5, 0.2, 0]$; $D_{DG\,B} = [1, 0.8, 0.5, 0.2, 0]$; (5.11)

For example, DG$_A$ can operate either uncurtailed at 1.6 MW with value 1, or curtailed at (1.6 MW * 0.8), or (1.6 MW * 0.5), or (1.6 MW * 0.2) or tripped at (1.6 MW * 0). DG$_B$ can also operate the same domain with its rated output 2 MW.

The solutions must meet the following constraints:

$C_{Network\ Constraint} = [|S_{ij}| \leq S_{ij}^{max}; 0.97\,p.u. \leq V_i \leq 1.03\,p.u.]$; (5.12)

$C_{Generator\ Contratual} = [DG_A = 1, DG_B = 2]$; (5.13) or

$C_{Generator\ Contratual} = [DG_A = 2, DG_B = 1]$; (5.14) or

$C_{Generator\ Contratual} = [DG_A = 1, DG_B = 2, DG_C = 3]$; (5.15)

$C_{Maximise\ Generation} = [max \sum_{i=1}^{N_{Gen}} P_{Geni}]$. (5.16)

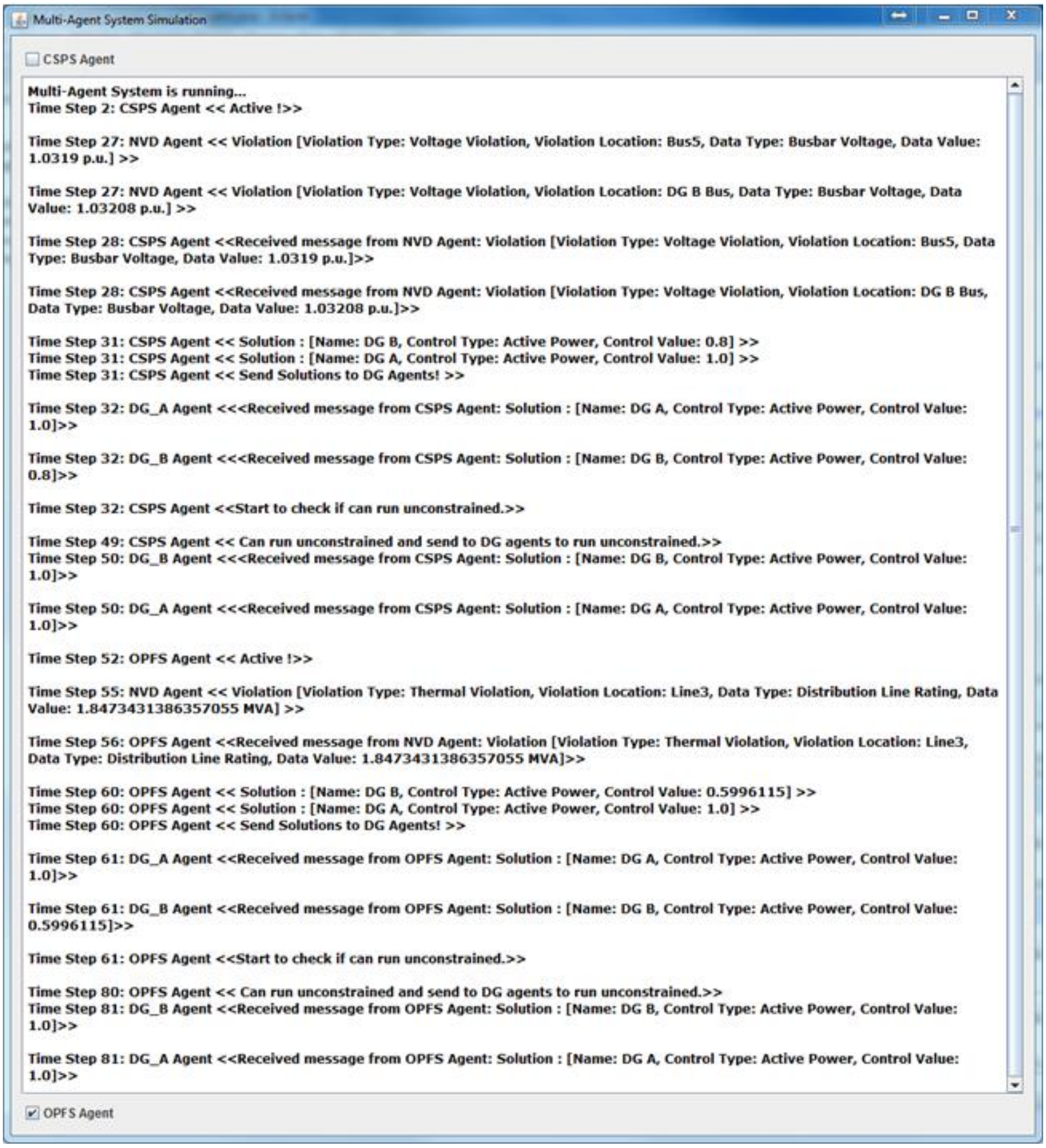
5.2.5.2. OPF Algorithm Control Specification

In the following scenarios, the OPF cost model for each DG has been developed based on the different u_{gi} as discussed in Section 5.2.3.1. If DG$_A$ has network access over DG$_B$, DG$_B$ will have the highest cost value u_{DGB} and is deemed to be the last connected DG. In addition, the cost values (u_{gi}) associated with DG$_A$ and DG$_B$ is reversed when DG$_A$ is deemed to be the last connected DG unit.

5.2.5.3. Scenario 1: Connecting of Different Solver Agents

Scenario 1 demonstrates the extensible capability of the developed MAS system by connecting the CSPS agent and the OPFS agent separately to handle different network violations. To simulate network violations, network profiles are developed to set DG_A with a constant 1.6 MW generation output and DG_B with varying generation output to represent a small wind farm. In such a system, network violations, including thermal overload and voltage excursion, could occur. For example, when the output of DG_B is high while loads connected at associated buses are low at the same time.

To manipulate agents of the MAS and display agent behaviours and interactions, a MAS simulation interface was developed and results are shown in Figure 5.6 for scenario 1. From the MAS simulation interface, the user can add and remove solver agents by clicking the relevant solver agent checkbox. In this scenario, the CSPS agent is connected in first to solve voltage violations that occur at the “DG_B Bus” and “Bus 5” as shown in Figure 5.4. Then, the CSPS agent is removed and then the OPFS agent is connected to solve a thermal overload which occurs at “Line 3” as shown in Figure 5.4. Following the DG connection order indicated in Table 5.2, DG_B is the last connected DG and therefore DG_B is the first DG to be curtailed under a network violation condition in this scenario. The DG_B active power output associated with a control signal is presented in Figure 5.7. Simulation results of voltage profiles of violation buses and the “Line 3” loading, including before and after control, are depicted in Figure 5.8 and Figure 5.9 separately.



```
Multi-Agent System is running...
Time Step 2: CSPS Agent << Active !>>

Time Step 27: NVD Agent << Violation [Violation Type: Voltage Violation, Violation Location: Bus5, Data Type: Busbar Voltage, Data Value:
1.0319 p.u.] >>

Time Step 27: NVD Agent << Violation [Violation Type: Voltage Violation, Violation Location: DG B Bus, Data Type: Busbar Voltage, Data
Value: 1.03208 p.u.] >>

Time Step 28: CSPS Agent <<Received message from NVD Agent: Violation [Violation Type: Voltage Violation, Violation Location: Bus5, Data
Type: Busbar Voltage, Data Value: 1.0319 p.u.]>>

Time Step 28: CSPS Agent <<Received message from NVD Agent: Violation [Violation Type: Voltage Violation, Violation Location: DG B Bus,
Data Type: Busbar Voltage, Data Value: 1.03208 p.u.]>>

Time Step 31: CSPS Agent << Solution : [Name: DG B, Control Type: Active Power, Control Value: 0.8] >>
Time Step 31: CSPS Agent << Solution : [Name: DG A, Control Type: Active Power, Control Value: 1.0] >>
Time Step 31: CSPS Agent << Send Solutions to DG Agents! >>

Time Step 32: DG_A Agent <<<Received message from CSPS Agent: Solution : [Name: DG A, Control Type: Active Power, Control Value:
1.0]>>

Time Step 32: DG_B Agent <<<Received message from CSPS Agent: Solution : [Name: DG B, Control Type: Active Power, Control Value:
0.8]>>

Time Step 32: CSPS Agent <<Start to check if can run unconstrained.>>

Time Step 49: CSPS Agent << Can run unconstrained and send to DG agents to run unconstrained.>>
Time Step 50: DG_B Agent <<<Received message from CSPS Agent: Solution : [Name: DG B, Control Type: Active Power, Control Value:
1.0]>>

Time Step 50: DG_A Agent <<<Received message from CSPS Agent: Solution : [Name: DG A, Control Type: Active Power, Control Value:
1.0]>>

Time Step 52: OPFS Agent << Active !>>

Time Step 55: NVD Agent << Violation [Violation Type: Thermal Violation, Violation Location: Line3, Data Type: Distribution Line Rating, Data
Value: 1.8473431386357055 MVA] >>

Time Step 56: OPFS Agent <<Received message from NVD Agent: Violation [Violation Type: Thermal Violation, Violation Location: Line3,
Data Type: Distribution Line Rating, Data Value: 1.8473431386357055 MVA]>>

Time Step 60: OPFS Agent << Solution : [Name: DG B, Control Type: Active Power, Control Value: 0.5996115] >>
Time Step 60: OPFS Agent << Solution : [Name: DG A, Control Type: Active Power, Control Value: 1.0] >>
Time Step 60: OPFS Agent << Send Solutions to DG Agents! >>

Time Step 61: DG_A Agent <<Received message from OPFS Agent: Solution : [Name: DG A, Control Type: Active Power, Control Value:
1.0]>>

Time Step 61: DG_B Agent <<Received message from OPFS Agent: Solution : [Name: DG B, Control Type: Active Power, Control Value:
0.5996115]>>

Time Step 61: OPFS Agent <<Start to check if can run unconstrained.>>

Time Step 80: OPFS Agent << Can run unconstrained and send to DG agents to run unconstrained.>>
Time Step 81: DG_B Agent <<Received message from OPFS Agent: Solution : [Name: DG B, Control Type: Active Power, Control Value:
1.0]>>

Time Step 81: DG_A Agent <<Received message from OPFS Agent: Solution : [Name: DG A, Control Type: Active Power, Control Value:
1.0]>>
```

Figure 5.6 MAS Simulation Interface and Results for Scenario 1

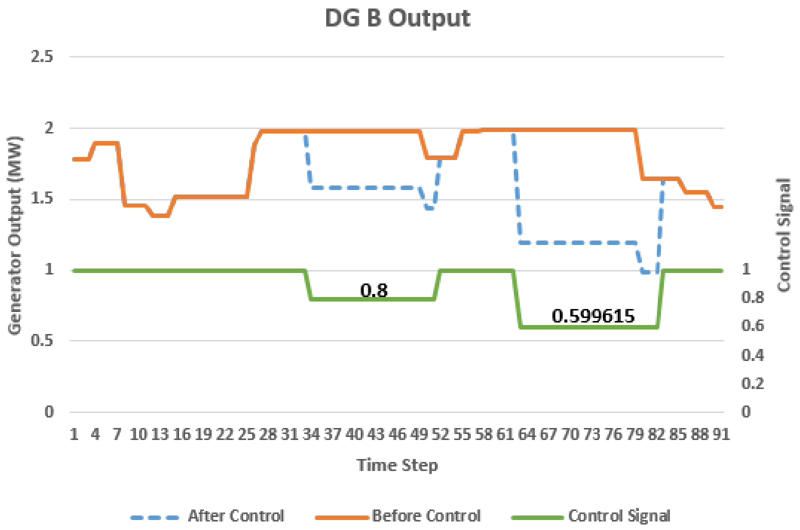


Figure 5.7 DG_B Output and Control Signal for Scenario 1

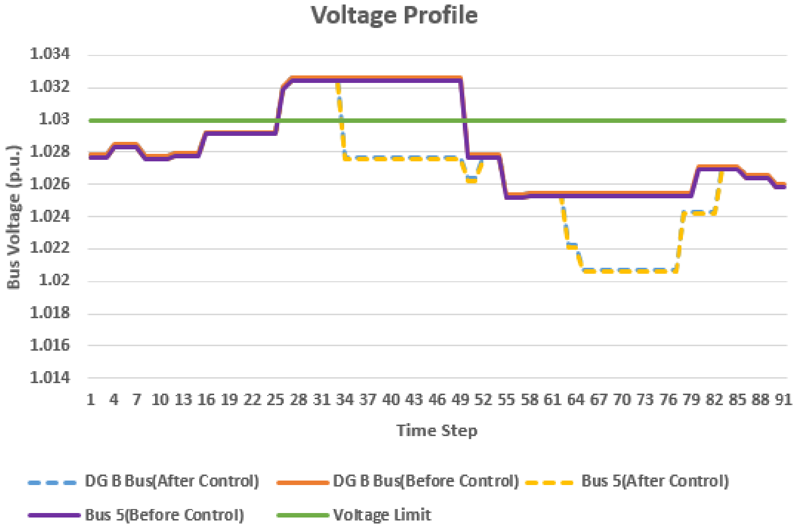


Figure 5.8 Voltage Profile of Violation Buses for Scenario 1

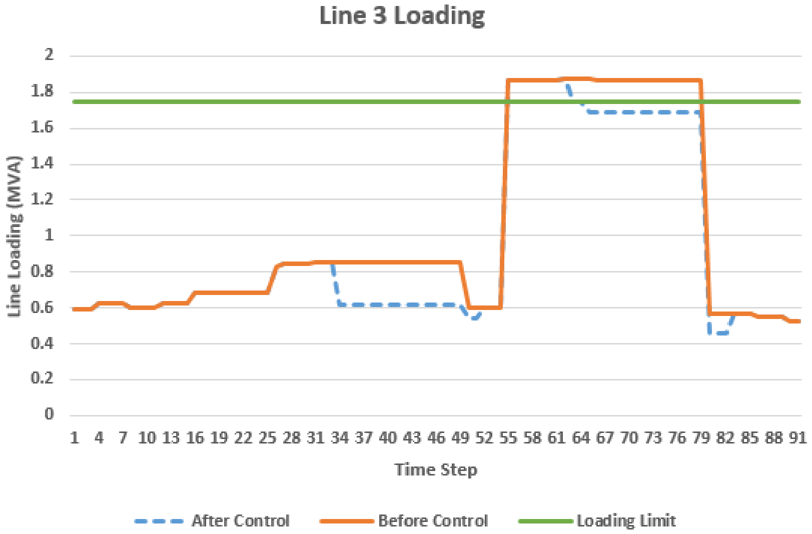


Figure 5.9 Line 3 Loading for Scenario 1

From the simulation results, when voltage violations occurred at "time step 27", the CSPS agent determined a solution that required the DG_B to curtail its output to 0.8 of maximum rated output and the control action was carried out to alleviate the voltage violations at the "time step 32". In Figure 5.8, it is clear that voltages could return to the acceptable limit after curtailed DG_B output. The curtailment could be relaxed at the "time step 50" with a control signal of 1.0 issued to the DG_B. For the thermal overload, it was detected at the "time step 59" and the OPFS agent found a solution which restricted the DG_B to output at a maximum of 0.6 of rated output. Then, the control signal was sent to the DG_B at the "time step 61". From Figure 5.9, the line rating could be brought back within the thermal limit after curtailing DG_B output. When DG_B was allowed to run unconstrained, the control signal of value 1 was issued at the "time step 81".

An agent communication diagram for scenario 1 is illustrated in Figure 5.10 that shows agent communication details associated with a timeline and indicates different operation stage of MAS.

- ***Agent Initialisation***: Each agent advertises its abilities when launched in the agent platform (e.g. the ability of NVD agent is violation detection), and the others use this mechanism to build their knowledge about other agents' abilities and status available to them. An agent can plan its own activities by using its knowledge and behaviours.

- ***CSPS Agent Plug In***: The CSPS agent was plugged into the agent platform and sent its own ability to the other agents and also received other agents' responses about their abilities.

- ***Voltage Violation***: Once voltage violations were detected, the NVD agent sent a message with voltage violation information including location and measured voltage values to the CSPS agent. Then, the CSPS agent was triggered to find a solution and sent a solution message to the DG_B agent to take action. The CSPS agent sent another message to the DG_B agent when DG_B could run unconstrained. After that, the CSPS agent was plugged-out and messages were sent to the other agents to inform that the CSPS agent was not running on the agent platform. This ensures that there is no dependency on the CSPS agent's operation being maintained within the community of agents.

- ***OPFS Agent Plug In***: The OPFS agent was plugged into the agent platform and operated the same operations as the CSPS agent in the "*CSPS Agent Plug In*" stage.
- ***Thermal Violation:*** The NVD agents sent the detected thermal overload details to the OPFS agent. Then, the OPFS agent determined a solution and sent a solution message to the DG_B agent to mitigate the thermal constraints. Once the DG_B was allowed to relax its curtailed output, the OPFS agent sent another message to the DG_B agent.

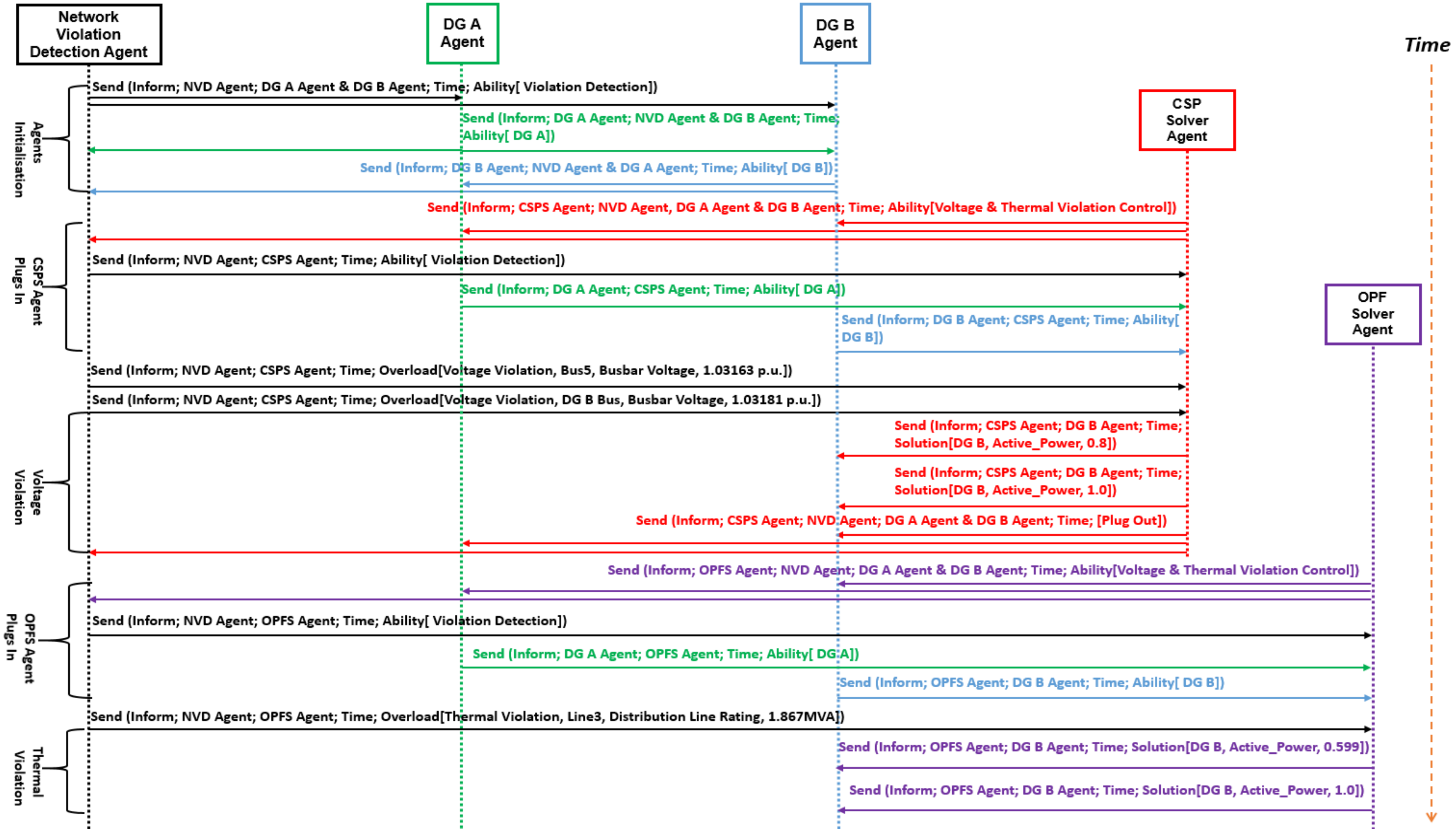


Figure 5.10 Agent Communication Diagram for Scenario 1

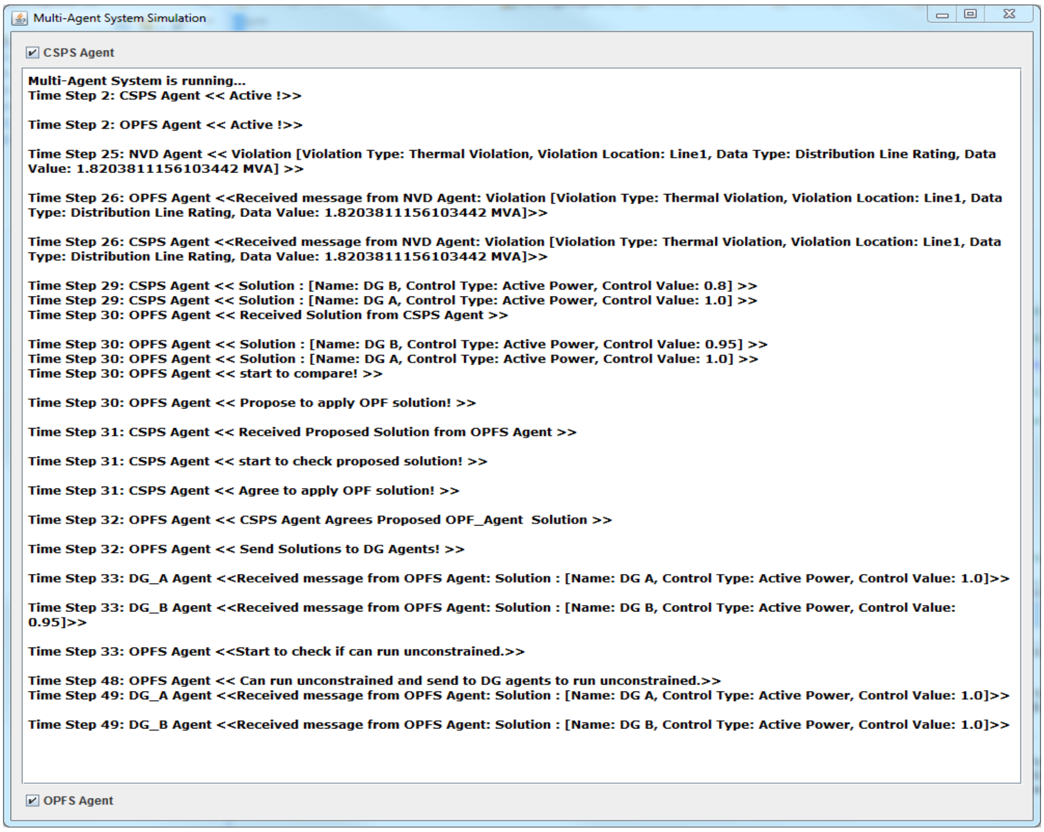
5.2.5.4. Scenario 2: Solver Agent Negotiation

Scenario 2 demonstrates agent negotiation capability based on agent knowledge and reasoning to achieve an agreement on the solution under different conditions when multiple solver agents are running on the agent platform. There are two sub-scenarios that are developed to simulate different conditions. Scenario 2-A presents all control solutions found by solver agents respecting the LIFO agreement and therefore solver agents negotiate to find the best solution that maximises the DGs' outputs. For scenario 2-B, the solution found by the OPFS agent could not follow the LIFO order so the other solution from the CSPS agent had to be applied to the network. Details of each scenario are described below.

5.2.5.4.1. Scenario 2-A

Scenario 2-A emulates a thermal overload occurring at "Line 1" as shown in Figure 5.4 when DG_B is increased to meet the high demand at loads. From Table 5.2, the DG_B unit is the last connected DG and therefore will be the first DG to be curtailed once the overload occurs in the network. The MAS simulation results of this scenario are displayed in the interface as shown in Figure 5.11. The DG_B output associated with its control signal is illustrated in Figure 5.12 and the "Line 1" loading before and after control is presented in Figure 5.13.

According to the simulation results, the thermal overload detection occurred at "time step 25". After that, the CSPS agent determined that DG_B output required to be curtailed to 0.8 of the rated output first. However, the OPFS agent found a solution that constrained the DG_B output to 0.95 of rated output. Then, the OPFS agent compared its own solution with the solution of the CSPS agent based on its knowledge, including maximising DG output and LIFO connection order. The OPFS agent proposed its own solution (0.95) to the CSPS agent, which curtails DG_B output less than the solution of the CSPS agent (0.8). The CSPS agent, on the other hand, agreed to the OPFS agent's proposal and therefore the OPFS agent was responsible for sending a message to DG_B agent about the control action. Then, the DG_B agent issued the control signal to curtail the DG_B output at "time step 33". From Figure 5.13, the control solution could successfully bring the "Line 1" loading back to the operational limit. Once DG_B can operate unconstrained, the OPFS agent sent another message to the DG_B agent to issue the control signal of value 1 at the “time step 49”.



```
Multi-Agent System is running...
Time Step 2: CSPS Agent << Active !>>

Time Step 2: OPFS Agent << Active !>>

Time Step 25: NVD Agent << Violation [Violation Type: Thermal Violation, Violation Location: Line1, Data Type: Distribution Line Rating, Data
Value: 1.8203811156103442 MVA] >>

Time Step 26: OPFS Agent <<Received message from NVD Agent: Violation [Violation Type: Thermal Violation, Violation Location: Line1, Data
Type: Distribution Line Rating, Data Value: 1.8203811156103442 MVA]>>

Time Step 26: CSPS Agent <<Received message from NVD Agent: Violation [Violation Type: Thermal Violation, Violation Location: Line1, Data
Type: Distribution Line Rating, Data Value: 1.8203811156103442 MVA]>>

Time Step 29: CSPS Agent << Solution : [Name: DG B, Control Type: Active Power, Control Value: 0.8] >>
Time Step 29: CSPS Agent << Solution : [Name: DG A, Control Type: Active Power, Control Value: 1.0] >>
Time Step 30: OPFS Agent << Received Solution from CSPS Agent >>

Time Step 30: OPFS Agent << Solution : [Name: DG B, Control Type: Active Power, Control Value: 0.95] >>
Time Step 30: OPFS Agent << Solution : [Name: DG A, Control Type: Active Power, Control Value: 1.0] >>
Time Step 30: OPFS Agent << start to compare! >>

Time Step 30: OPFS Agent << Propose to apply OPF solution! >>

Time Step 31: CSPS Agent << Received Proposed Solution from OPFS Agent >>

Time Step 31: CSPS Agent << start to check proposed solution! >>

Time Step 31: CSPS Agent << Agree to apply OPF solution! >>

Time Step 32: OPFS Agent << CSPS Agent Agrees Proposed OPF_Agent  Solution >>

Time Step 32: OPFS Agent << Send Solutions to DG Agents! >>

Time Step 33: DG_A Agent <<Received message from OPFS Agent: Solution : [Name: DG A, Control Type: Active Power, Control Value: 1.0]>>

Time Step 33: DG_B Agent <<Received message from OPFS Agent: Solution : [Name: DG B, Control Type: Active Power, Control Value:
0.95]>>

Time Step 33: OPFS Agent <<Start to check if can run unconstrained.>>

Time Step 48: OPFS Agent << Can run unconstrained and send to DG agents to run unconstrained.>>
Time Step 49: DG_A Agent <<Received message from OPFS Agent: Solution : [Name: DG A, Control Type: Active Power, Control Value: 1.0]>>

Time Step 49: DG_B Agent <<Received message from OPFS Agent: Solution : [Name: DG B, Control Type: Active Power, Control Value: 1.0]>>
```

Figure 5.11 MAS Simulation Interface and Results for Scenario 2-A

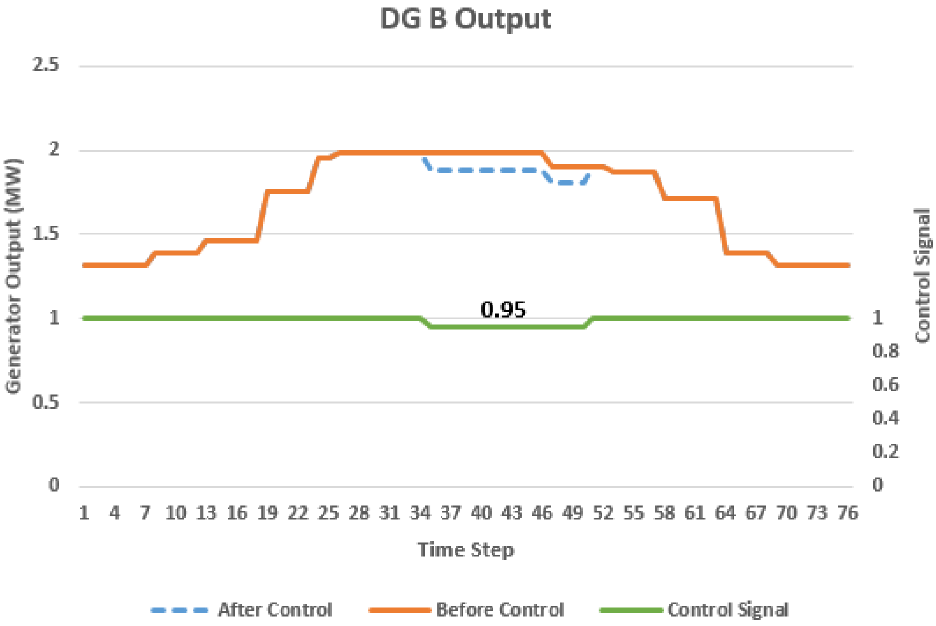


Figure 5.12 DG_B Output and Control Signal for Scenario 2-A

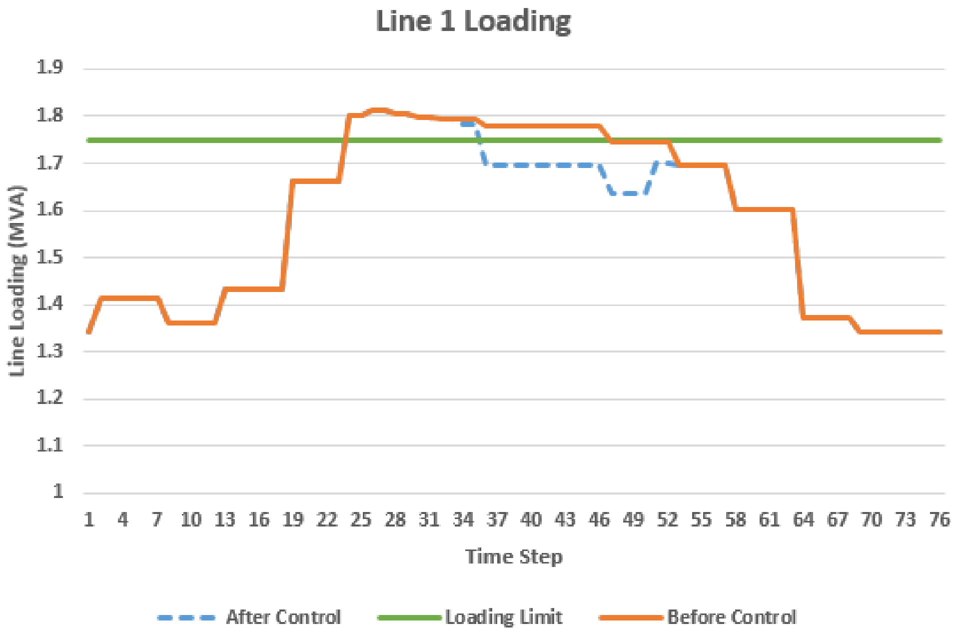


Figure 5.13 Line 1 Loading for Scenario 2-A

The agent communication diagram for scenario 2-A is shown in Figure 5.14. There are two different stages of operation, including the agent initialisation stage and the violation negotiation stage. For the agent initialisation stage, it is the same operation as for scenario 1.

- ***Violation Negotiation***: Once the thermal overload was detected by the NVD agent, overload messages were sent to the CSPS agent and OPFS agent with details, including overload location and measured thermal value. Then, the CSPS and the OPFS agent were triggered to solve the thermal overload. The CSPS agent was the first one to find a solution and therefore the CSPS agent sent it to the OPFS agent. The OPFS agent, on the other hand, determined its own solution and compared with the received solution from the CSPS agent. After that, the OPFS agent sent a message to propose its own solution which curtailed the DG_B output less than the CSPS agent solution. The CSPS agent replied to the OPFS agent about accepting the solution proposal. Then, the OPFS agent sent a message to DG_B agent to take action based on the best solution. Once the DG_B could run unconstrained, the OPFS agent sent another message to the DG_B agent to resume the output at full rating.

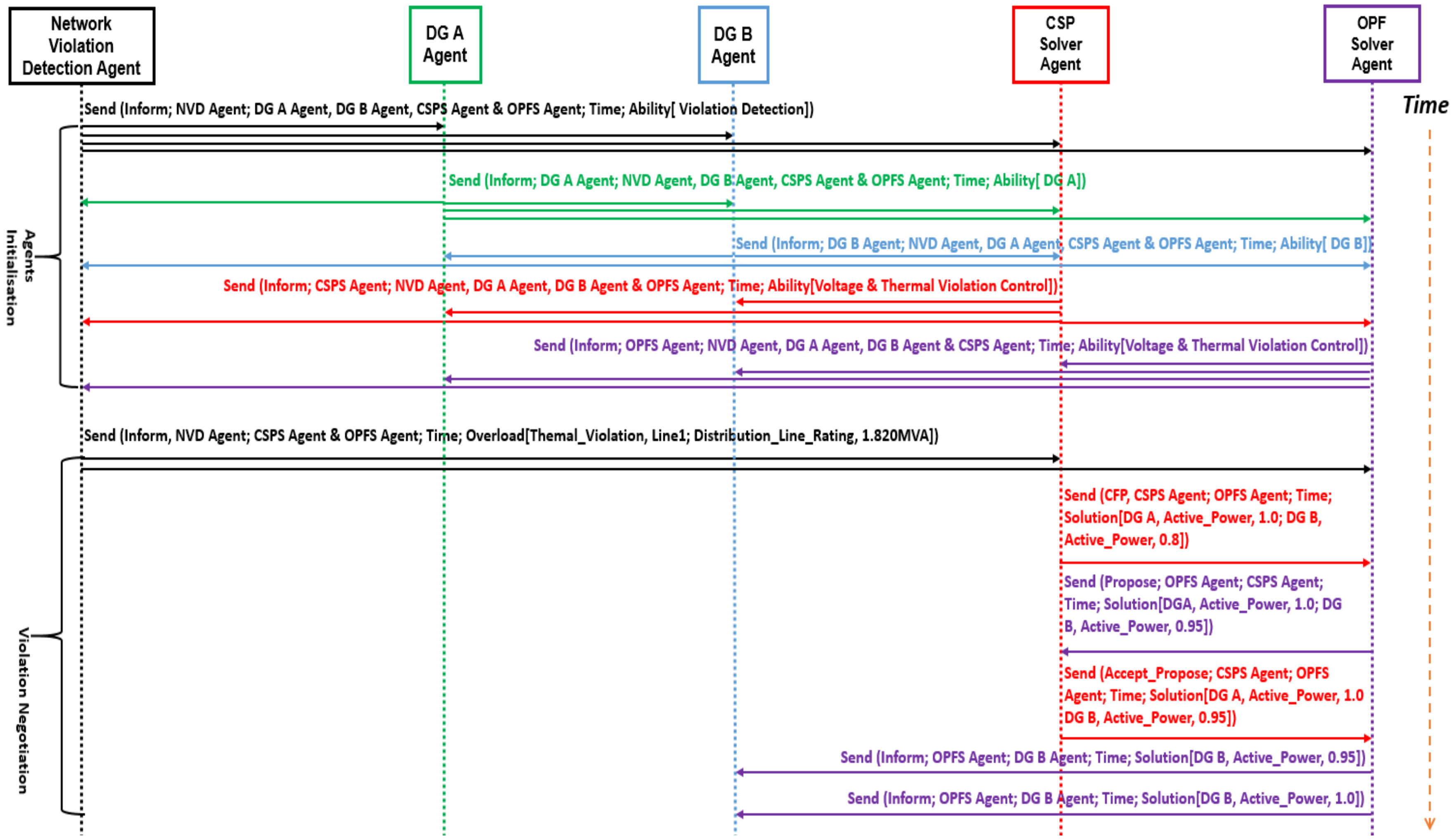
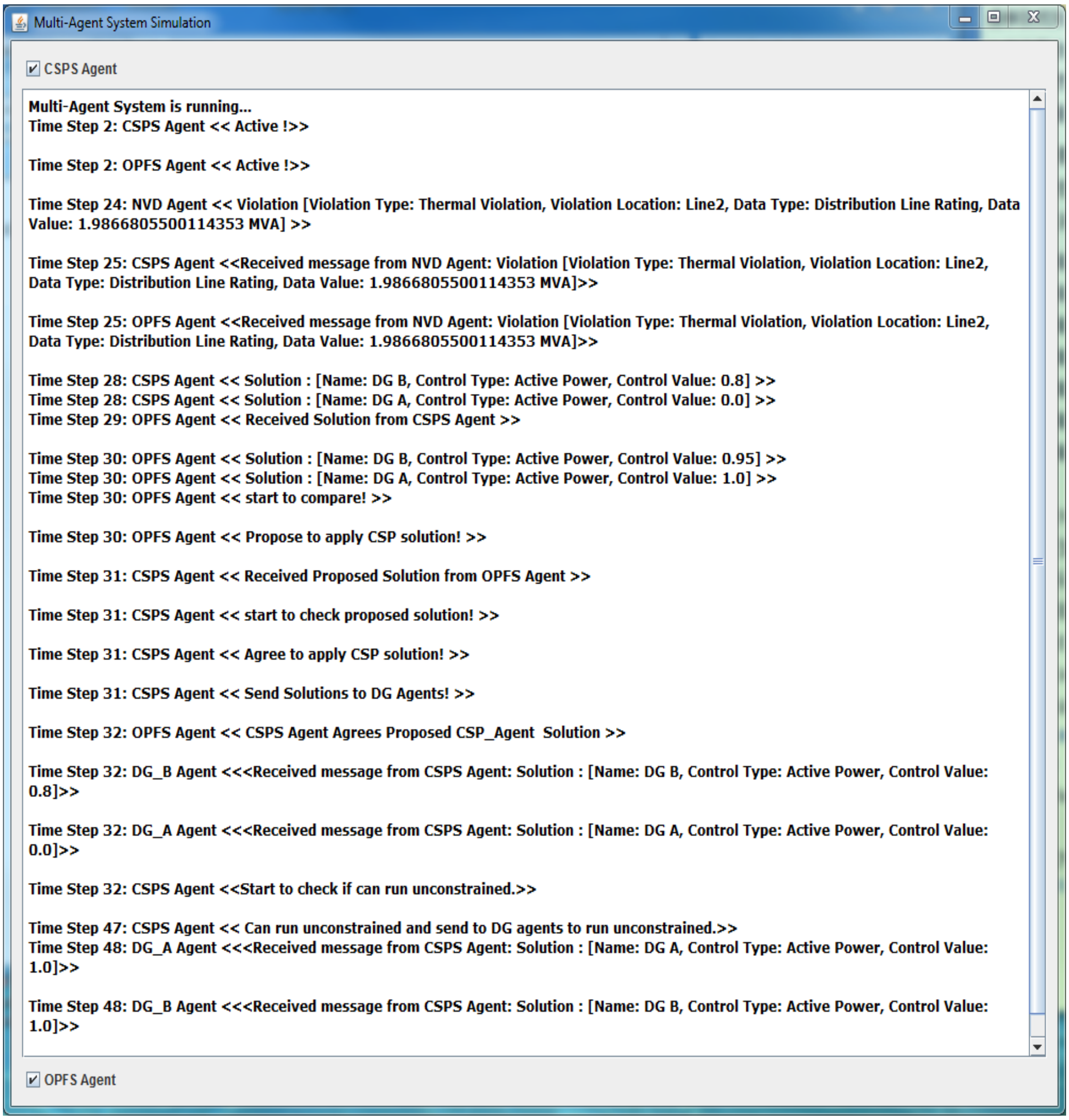


Figure 5.14 Agent Communication Diagram for Scenario 2-A

5.2.5.4.2. Scenario 2-B

For scenario 2-B, it used the same network profiles as for scenario 2-A. However, the thermal limit of “Line 1” was increased and “Line 2” thermal limit was re-rated to emulate an overload. In addition, the DG_A unit is the last connected DG and therefore will be the first DG to be curtailed when there is an overload in the network.

Figure 5.15 shows MAS simulation results of this scenario in the developed interface.The DG_A and the DG_B outputs and their control signals are provided in Figure 5.16 and Figure 5.17 separately. The simulation results of the “Line 2” loading before and after control are illustrated in Figure 5.18.



```
Multi-Agent System is running...
Time Step 2: CSPS Agent << Active !>>

Time Step 2: OPFS Agent << Active !>>

Time Step 24: NVD Agent << Violation [Violation Type: Thermal Violation, Violation Location: Line2, Data Type: Distribution Line Rating, Data
Value: 1.9866805500114353 MVA] >>

Time Step 25: CSPS Agent <<Received message from NVD Agent: Violation [Violation Type: Thermal Violation, Violation Location: Line2,
Data Type: Distribution Line Rating, Data Value: 1.9866805500114353 MVA]>>

Time Step 25: OPFS Agent <<Received message from NVD Agent: Violation [Violation Type: Thermal Violation, Violation Location: Line2,
Data Type: Distribution Line Rating, Data Value: 1.9866805500114353 MVA]>>

Time Step 28: CSPS Agent << Solution : [Name: DG B, Control Type: Active Power, Control Value: 0.8] >>
Time Step 28: CSPS Agent << Solution : [Name: DG A, Control Type: Active Power, Control Value: 0.0] >>
Time Step 29: OPFS Agent << Received Solution from CSPS Agent >>

Time Step 30: OPFS Agent << Solution : [Name: DG B, Control Type: Active Power, Control Value: 0.95] >>
Time Step 30: OPFS Agent << Solution : [Name: DG A, Control Type: Active Power, Control Value: 1.0] >>
Time Step 30: OPFS Agent << start to compare! >>

Time Step 30: OPFS Agent << Propose to apply CSP solution! >>

Time Step 31: CSPS Agent << Received Proposed Solution from OPFS Agent >>

Time Step 31: CSPS Agent << start to check proposed solution! >>

Time Step 31: CSPS Agent << Agree to apply CSP solution! >>

Time Step 31: CSPS Agent << Send Solutions to DG Agents! >>

Time Step 32: OPFS Agent << CSPS Agent Agrees Proposed CSP_Agent  Solution >>

Time Step 32: DG_B Agent <<<Received message from CSPS Agent: Solution : [Name: DG B, Control Type: Active Power, Control Value:
0.8]>>

Time Step 32: DG_A Agent <<<Received message from CSPS Agent: Solution : [Name: DG A, Control Type: Active Power, Control Value:
0.0]>>

Time Step 32: CSPS Agent <<Start to check if can run unconstrained.>>

Time Step 47: CSPS Agent << Can run unconstrained and send to DG agents to run unconstrained.>>
Time Step 48: DG_A Agent <<<Received message from CSPS Agent: Solution : [Name: DG A, Control Type: Active Power, Control Value:
1.0]>>

Time Step 48: DG_B Agent <<<Received message from CSPS Agent: Solution : [Name: DG B, Control Type: Active Power, Control Value:
1.0]>>
```

Figure 5.15 MAS Simulation Interface and Results for Scenario 2-B

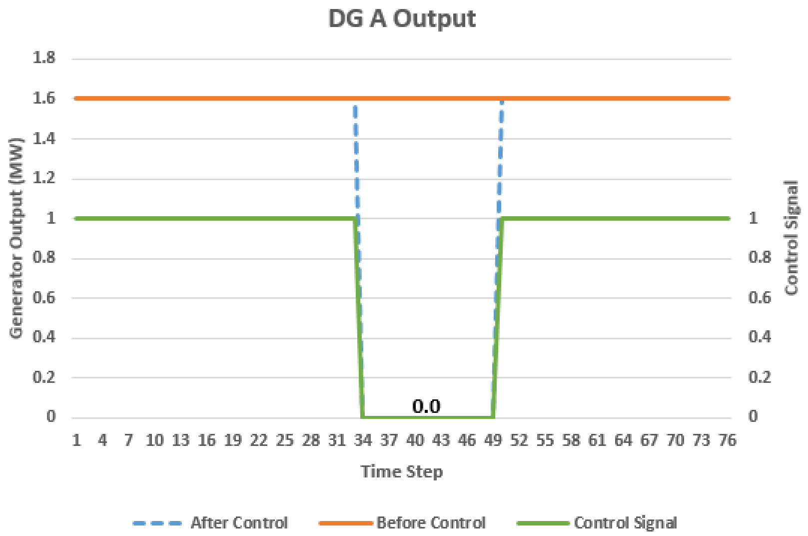


Figure 5.16 DG$_A$ Output and Control Signal for Scenario 2-B

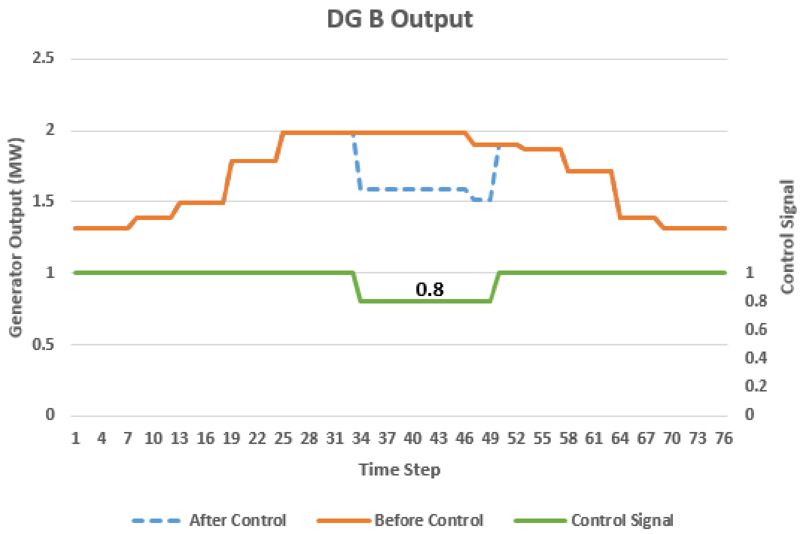


Figure 5.17 DG$_B$ Output and Control Signal for Scenario 2-B

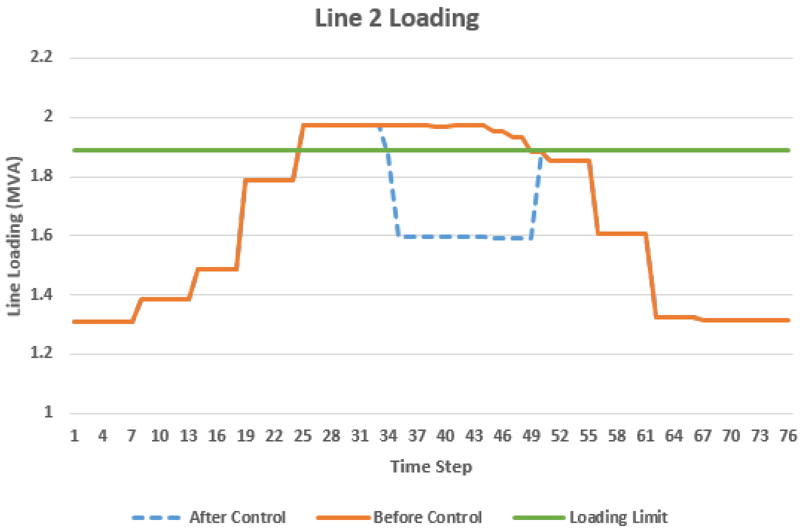


Figure 5.18 Line 2 Loading for Scenario 2-B

As shown in the simulation results, the thermal overload occurred at the “time step 25”. The CSPS agent determined a solution that tripped DG_A and constrained the DG_B output to 0.8 of rated output. For the OPFS agent, it found a solution that allowed the DG_A to operate without curtailment and curtailed the DG_B output to 0.95 of rated output. However, the OPFS agent solution did not respect the LIFO arrangement as DG_A needs to be curtailed first. This is because if the curtailment of the particular generator (DG_A) does not address line overloads (“Line 2”), the OPF function will leave the output of this generator (DG_A) unmodified. Thus, the OPFS agent curtails only the generator (in this case DG_B) that can solve the constraint in the overloaded “Line 2”. Then, the OPFS agent compared its own solution with the solution received from the CSPS agent based on its knowledge, including maximising the DG output and LIFO agreement. As a result, the OPFS agent proposed the CSPS agent solution, which respected the LIFO arrangement. The CSPS agent, on the other hand, agreed the OPFS agent’s proposal and then the CSPS agent sent messages to DG_A agent and DG_B agent to take actions. The control signals were issued by the DG agents at “time step 32”. The “Line 2” loading could be returned to the thermal limit as presented in Figure 5.18. Furthermore, the CSPS agent kept monitoring the network condition and control signals of value 1 were sent to both DGs to let them run unconstrained at “time step 48”. Figure 5.19 shows agent communication diagram for scenario 2-B. The agent operation stages are the same as the previous scenario 2-A. The violation negotiation stage is described as follows:

- ***Violation Negotiation***: Once the thermal overload occurred, the NVD agent informed the CSPS agent and OPFS agent of the violation details, including the overload location and measured thermal value. Then, the CSPS agent and OPFS agent started to calculate overload solutions. The CSPS agent was the first one to find a solution and therefore sent it to the OPFS agent. After the OPFS agent found its own solution, it compared it with the received CSPS agent solution and proposed the CSPS agent solution. The CSPS agent replied to the OPFS agent with a message accepting the proposed solution. Then, the CSPS agent sent messages to the DG agents to take actions. Once the DGs could run unconstrained, CSPS agent sent another messages to the DG agents to increase their outputs up to full rating.

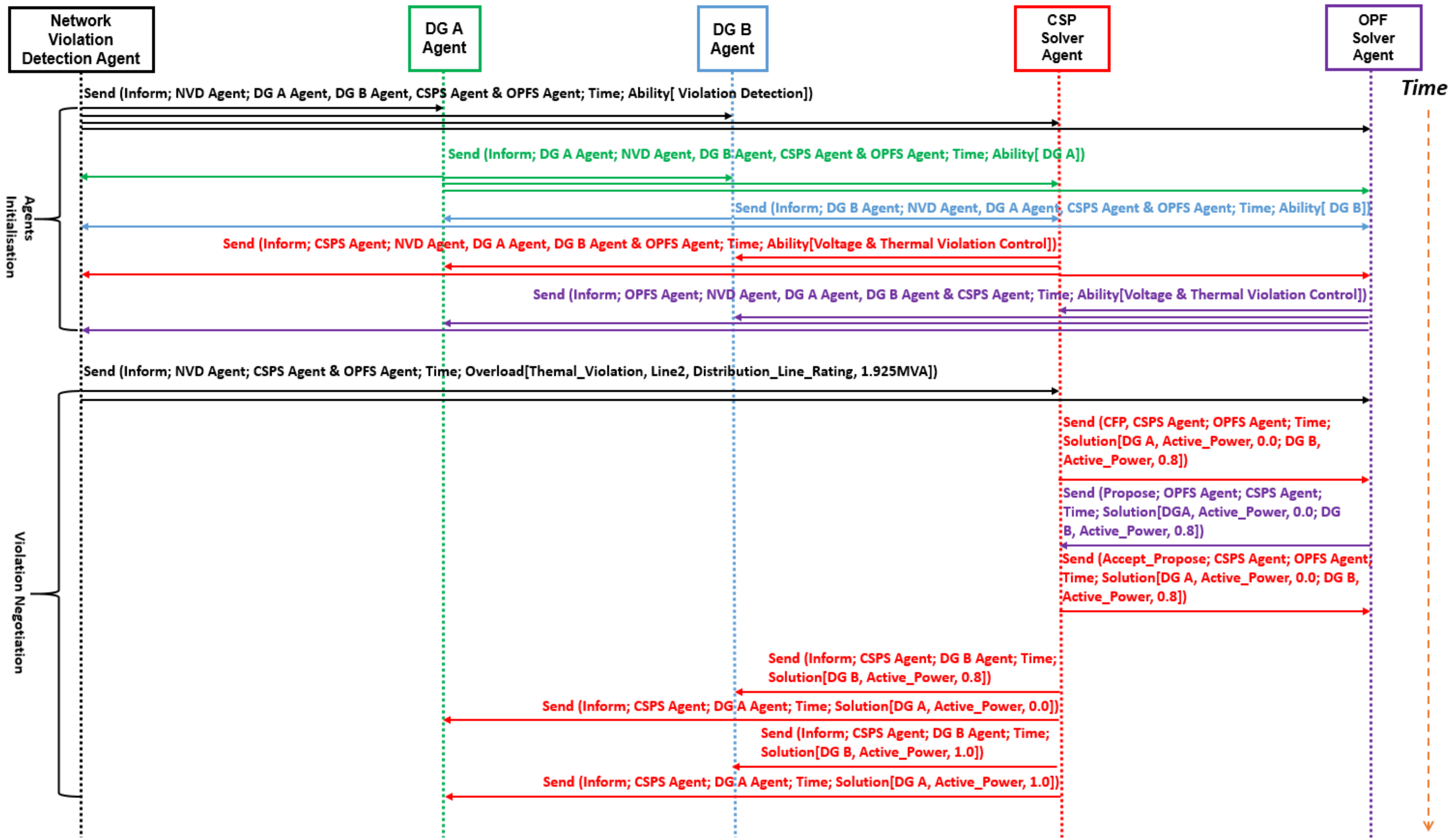


Figure 5.19 Agent Communication Diagram for Scenario 2-B

5.2.5.5. Scenario 3: MAS Response to a New DG Connection

Scenario 3 connected a new DG during the simulation to demonstrate the flexibility of the fully integrated MAS architecture under a network change situation. The new DG_C with 1 MW capacity is connected to “Bus 6” as shown in Figure 5.20 and therefore is deemed to be the last connected DG. As a result, the DG_C will be curtailed first when there is a violation in the network. In addition, with the connection of $DG_{C,}$ a thermal overload will occur on “Line 4” due to load increasing at “Bus 3” as shown in Figure 5.20. In this scenario, the CSPS agent and OPFS agent were both activated at the beginning of the simulation. However, the DG_C was activated during the simulation once the DG_C was connected to the network.

The MAS simulation results of this scenario are presented in Figure 5.21. The DG_C output with associated control signals is depicted in Figure 5.22 and the “Line 4” loading before and after control is illustrated in Figure 5.23.

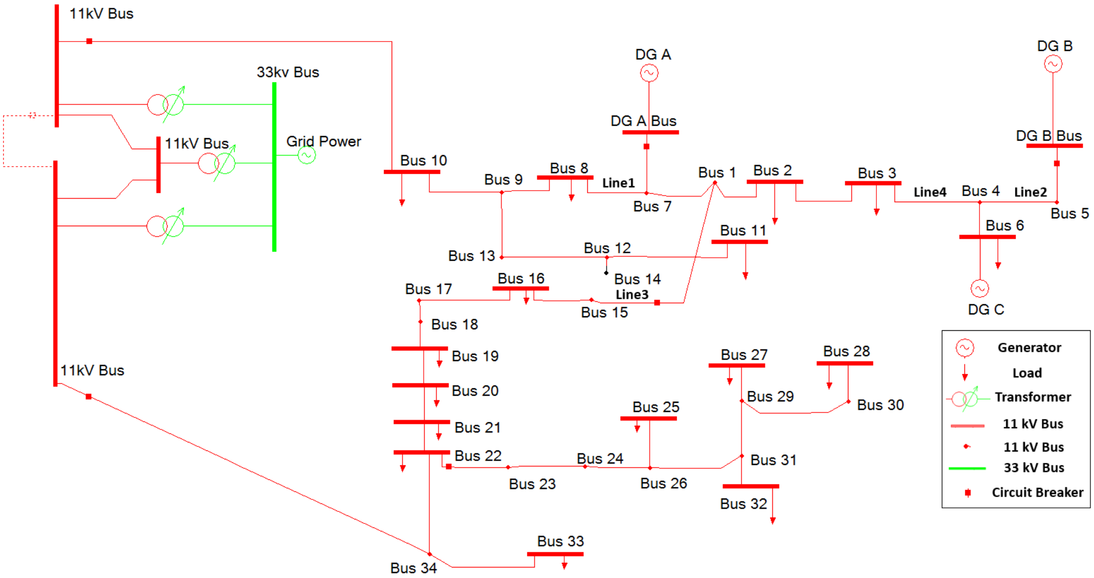
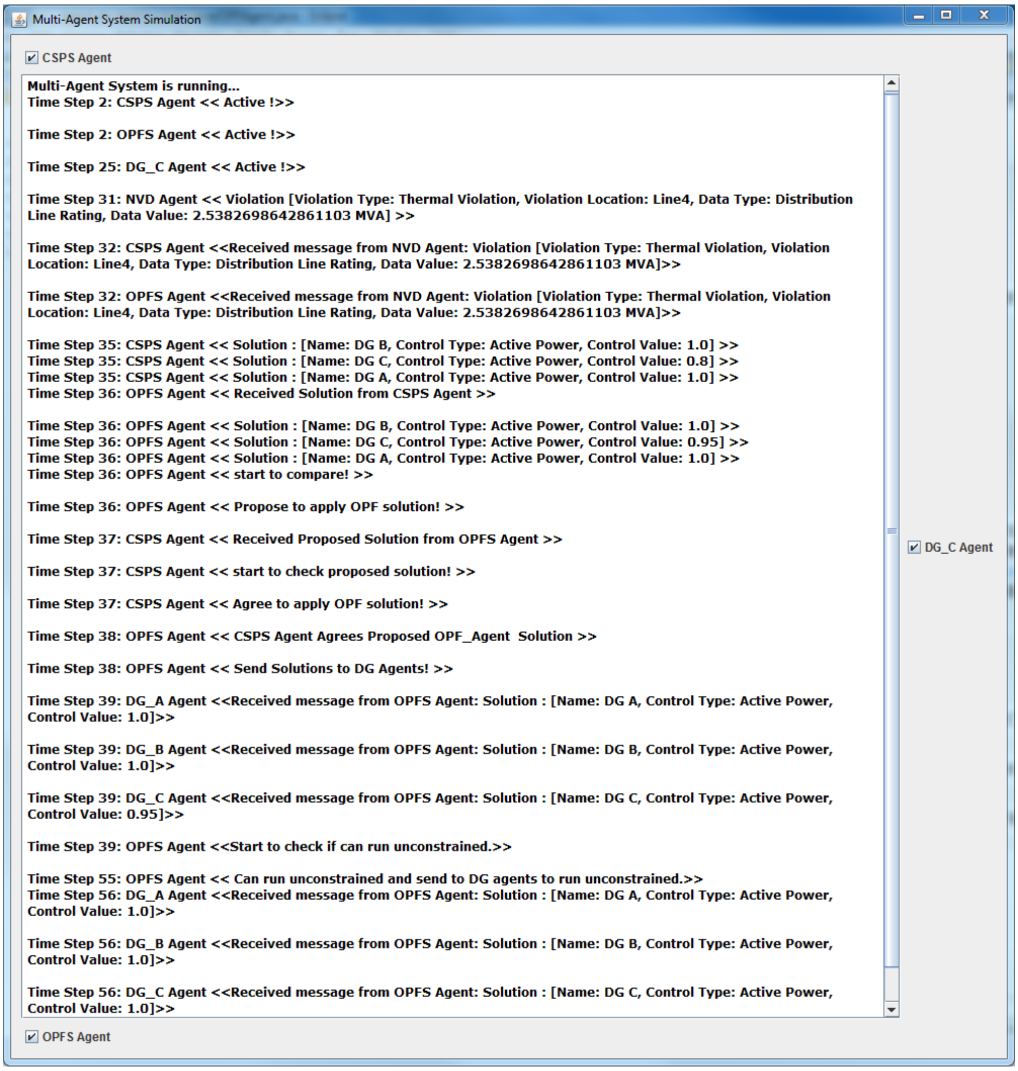


Figure 5.20 11 kV Radial Distribution Network with New DG Connected



```
Multi-Agent System is running...
Time Step 2: CSPS Agent << Active !>>

Time Step 2: OPFS Agent << Active !>>

Time Step 25: DG_C Agent << Active !>>

Time Step 31: NVD Agent << Violation [Violation Type: Thermal Violation, Violation Location: Line4, Data Type: Distribution
Line Rating, Data Value: 2.5382698642861103 MVA] >>

Time Step 32: CSPS Agent <<Received message from NVD Agent: Violation [Violation Type: Thermal Violation, Violation
Location: Line4, Data Type: Distribution Line Rating, Data Value: 2.5382698642861103 MVA]>>

Time Step 32: OPFS Agent <<Received message from NVD Agent: Violation [Violation Type: Thermal Violation, Violation
Location: Line4, Data Type: Distribution Line Rating, Data Value: 2.5382698642861103 MVA]>>

Time Step 35: CSPS Agent << Solution : [Name: DG B, Control Type: Active Power, Control Value: 1.0] >>
Time Step 35: CSPS Agent << Solution : [Name: DG C, Control Type: Active Power, Control Value: 0.8] >>
Time Step 35: CSPS Agent << Solution : [Name: DG A, Control Type: Active Power, Control Value: 1.0] >>
Time Step 36: OPFS Agent << Received Solution from CSPS Agent >>

Time Step 36: OPFS Agent << Solution : [Name: DG B, Control Type: Active Power, Control Value: 1.0] >>
Time Step 36: OPFS Agent << Solution : [Name: DG C, Control Type: Active Power, Control Value: 0.95] >>
Time Step 36: OPFS Agent << Solution : [Name: DG A, Control Type: Active Power, Control Value: 1.0] >>
Time Step 36: OPFS Agent << start to compare! >>

Time Step 36: OPFS Agent << Propose to apply OPF solution! >>

Time Step 37: CSPS Agent << Received Proposed Solution from OPFS Agent >>

Time Step 37: CSPS Agent << start to check proposed solution! >>

Time Step 37: CSPS Agent << Agree to apply OPF solution! >>

Time Step 38: OPFS Agent << CSPS Agent Agrees Proposed OPF_Agent  Solution >>

Time Step 38: OPFS Agent << Send Solutions to DG Agents! >>

Time Step 39: DG_A Agent <<Received message from OPFS Agent: Solution : [Name: DG A, Control Type: Active Power,
Control Value: 1.0]>>

Time Step 39: DG_B Agent <<Received message from OPFS Agent: Solution : [Name: DG B, Control Type: Active Power,
Control Value: 1.0]>>

Time Step 39: DG_C Agent <<Received message from OPFS Agent: Solution : [Name: DG C, Control Type: Active Power,
Control Value: 0.95]>>

Time Step 39: OPFS Agent <<Start to check if can run unconstrained.>>

Time Step 55: OPFS Agent << Can run unconstrained and send to DG agents to run unconstrained.>>
Time Step 56: DG_A Agent <<Received message from OPFS Agent: Solution : [Name: DG A, Control Type: Active Power,
Control Value: 1.0]>>

Time Step 56: DG_B Agent <<Received message from OPFS Agent: Solution : [Name: DG B, Control Type: Active Power,
Control Value: 1.0]>>

Time Step 56: DG_C Agent <<Received message from OPFS Agent: Solution : [Name: DG C, Control Type: Active Power,
Control Value: 1.0]>>
```

Figure 5.21 MAS Simulation Interface and Results for Scenario 3

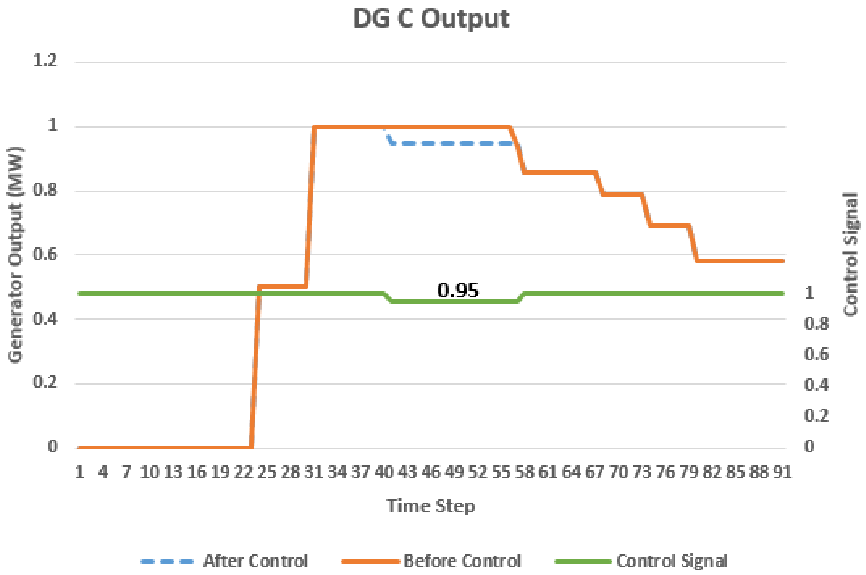


Figure 5.22 DG$_C$ Output and Control Signal for Scenario 3

Figure 5.23 Line 4 Loading for Scenario 3

From the simulation results, the DG_C was connected to the network at “time step 23”. As a result, the thermal overload occurred at “time step 31”. The CSPS agent determined a solution that constrained the DG_C output to 0.8 of its maximum output. However, the OPFS agent found a solution that curtailed DG_C output to 0.95 of rated output. Then, the OPFS agent compared its own solution with the solution received from the CSPS agent based on its knowledge, including maximising DG output and LIFO DG connection order. Hence, the OPFS agent proposed its own solution (0.95) that curtailed the DG_C output less than the CSPS agent solution. The CSPS agent accepted the proposal from the OPFS agent and then the OPFS agent was responsible for sending the solution message to the DG_C agent. The DG_C agent issued the control signal to the DG_C at “time step 39”. The control solution could mitigate the overload on “Line 4” as shown in Figure 5.23. Furthermore, the CSPS agent continued to monitor the network operation until DG_C can operate unconstrained. Finally, a further control signal is sent to DG_C to relax its curtailment operation at “time step 56”.

Figure 5.24 shows the agent communication diagram for scenario 3. There are three different stages of agent interactions, including agent initialisation, DG_C agent plug in, and violation negotiation. The agent initialisation stage includes the same operations as in previous scenarios. The DG_C agent plug-in and violation negotiation stages are described below.

- ***DG_C Agent Plug In***: The DG_C agent plugged into the agent platform once DG_C was connected to the network. DG_C agent sent its own ability to the other agents and also received other agents' responses about their abilities.

- ***Violation Negotiation***: When the NVD agent detected the thermal overload, the NVD agent informed the CSPS agent and OPFS agent about the violation details, including overload location and measured value. Then, the CSPS agent and OPFS agent initiated their algorithms to solve the overload. The CSPS agent found a solution first and then sent it to the OPFS agent. After that, the OPFS agent determined its own solution and compared it with the received solution from the CSPS agent. As the OPFS agent solution curtailed DG_C output less than the CSPS agent solution, the OPFS agent proposed its own solution. The CSPS agent replied with an agreement message to the OPFS agent and then the OPFS agent sent a solution message to the DG_C agent to take action. Once the DG_C could run unconstrained, the OPFS agent sent another message to the DG_C agent to lift the DG_C output to rated value.

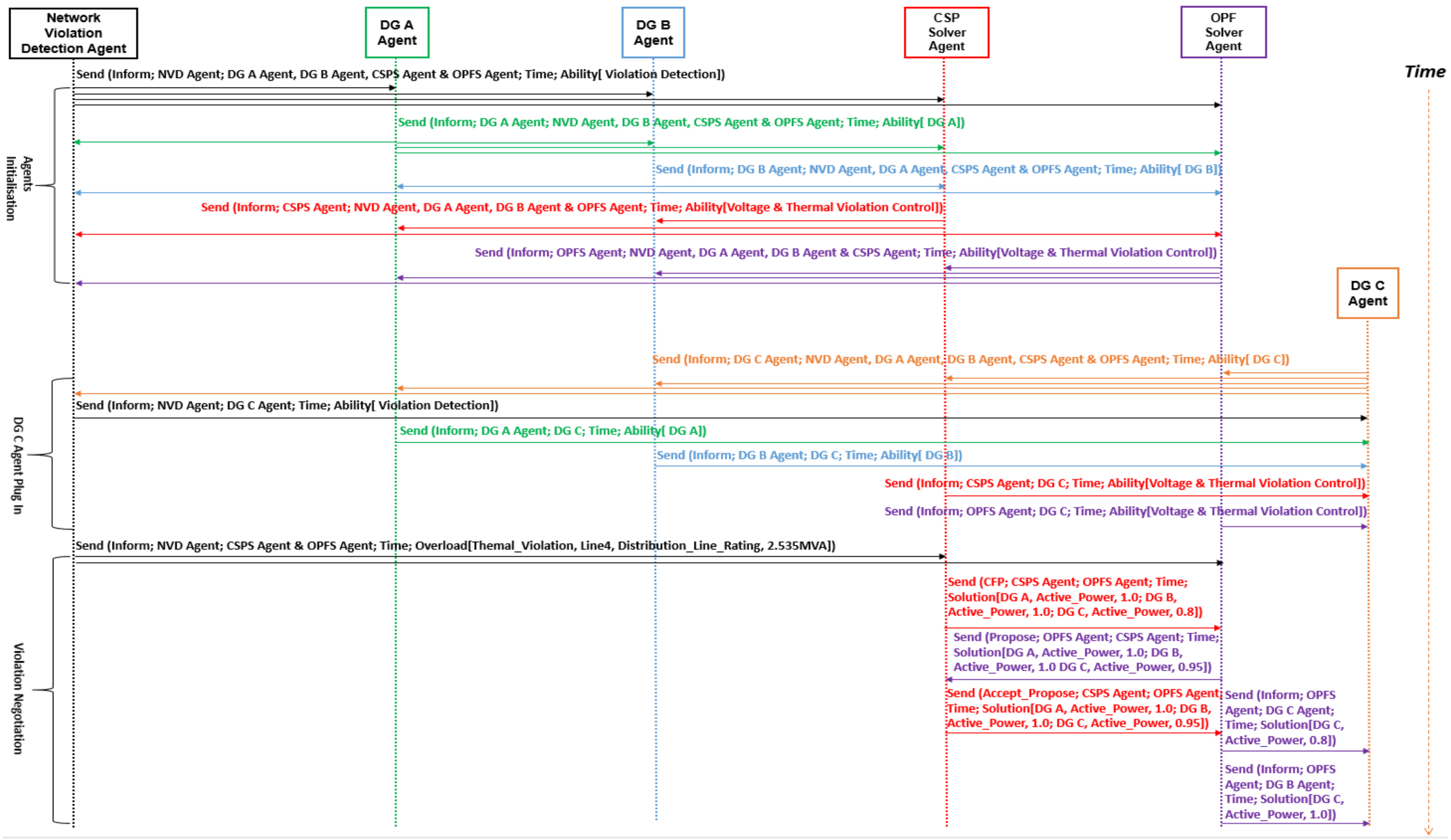


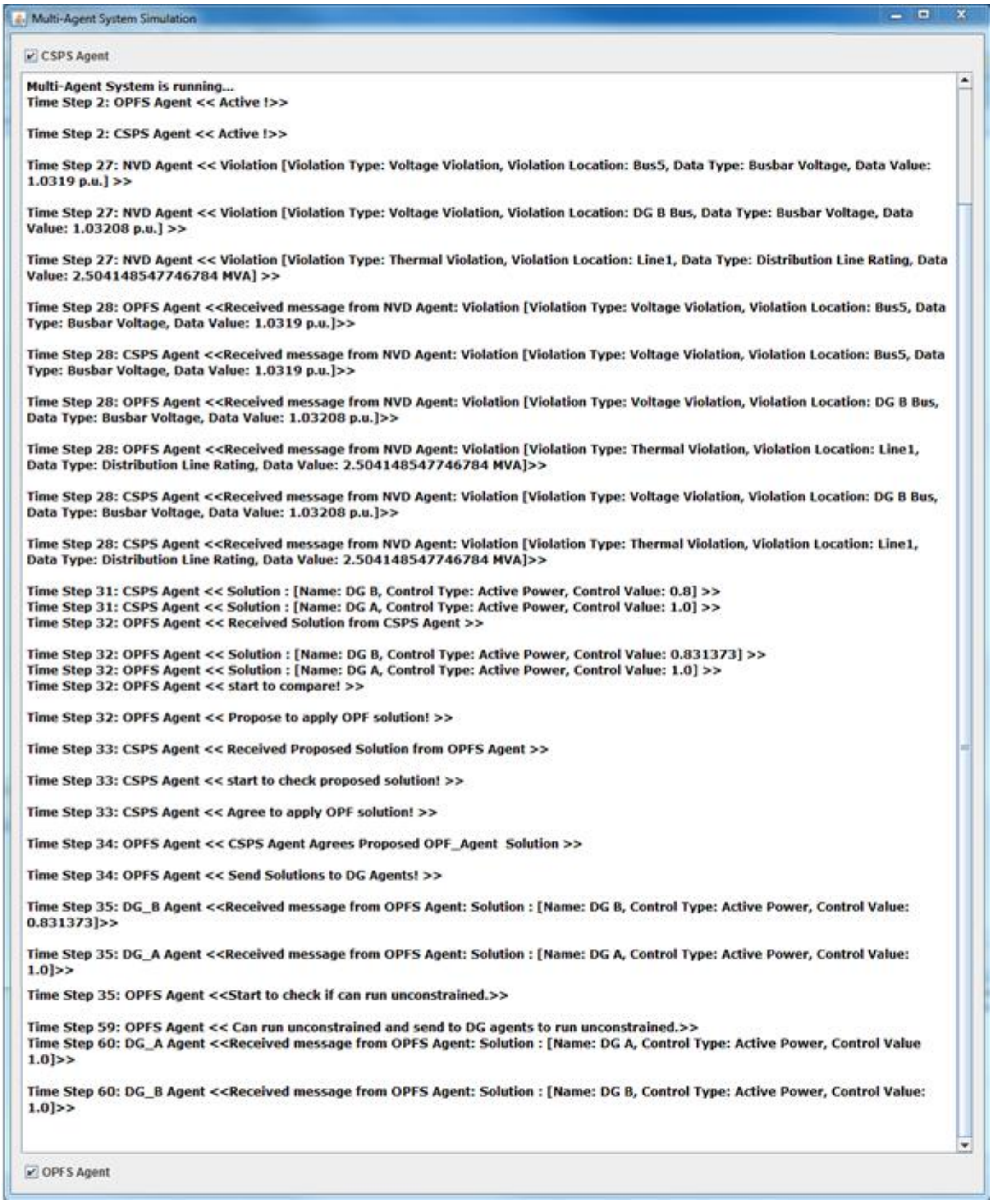
Figure 5.24 Agent Communication Diagram for Scenario 3

5.2.5.6. Scenario 4: Multiple Violations Occurring Simultaneously

Scenario 4 proves that the fully integrated MAS can deal with multiple violations occurring simultaneously, including voltage violation and thermal overload. The network is connected with two DGs as shown in Figure 5.4 and varying DG_B output and the load profile forced a thermal overload on “Line 1” and voltage excursions on “Bus 5” and “DG_B Bus” as shown in Figure 5.4. From Table 5.2, DG_B is the last connected DG and therefore will be the first to attempt to alleviate the network violations.

The MAS simulation results of this scenario are illustrated in Figure 5.25. In this scenario, the CSPS agent and OPFS agent were both activated and therefore two solver agents needed to negotiate to find the best solution for the multiple network violations. The DG output with control signal is shown in Figure 5.26. “Line 1” loading and the voltage profile at violation buses before and after control are presented in Figure 5.27 and Figure 5.28 separately.

Based on the simulation results, the multiple network violations happened at “time step 27”. The CSPS agent determined the DG_B output which was required to curtail to 0.8 of rated output. However, the OPFS agent calculated a solution that constrained the DG_B output to 0.83 of rated output. Then, the OPFS agent compared its own solution with the solution received from the CSPS agent along with its knowledge, including maximising DG output and the LIFO agreement. After that, the OPFS agent proposed its own solution (0.83) to the CSPS agent, which curtailed the DG_B output less. The CSPS agent, on the other hand, agreed to the OPFS agent’s proposal and then the OPFS agent took responsibility to send a message to the DG_B agent to execute the solution. The DG_B agent issued the control signal to constrain the DG_B at “time step 35”. The “Line 1” loading and violating bus voltage can be successfully controlled within the normal operation limit by applying the solution as presented in Figure 5.27 and Figure 5.28. Moreover, the OPFS agent kept monitoring the network conditions to check when DG_B can run unconstrained. Finally, the control signal was sent to the DG_B to allow it to operate at rated output at “time step 60”.



```
Multi-Agent System is running...
Time Step 2: OPFS Agent << Active !>>

Time Step 2: CSPS Agent << Active !>>

Time Step 27: NVD Agent << Violation [Violation Type: Voltage Violation, Violation Location: Bus5, Data Type: Busbar Voltage, Data Value:
1.0319 p.u.] >>

Time Step 27: NVD Agent << Violation [Violation Type: Voltage Violation, Violation Location: DG B Bus, Data Type: Busbar Voltage, Data
Value: 1.03208 p.u.] >>

Time Step 27: NVD Agent << Violation [Violation Type: Thermal Violation, Violation Location: Line1, Data Type: Distribution Line Rating, Data
Value: 2.504148547746784 MVA] >>

Time Step 28: OPFS Agent <<Received message from NVD Agent: Violation [Violation Type: Voltage Violation, Violation Location: Bus5, Data
Type: Busbar Voltage, Data Value: 1.0319 p.u.]>>

Time Step 28: CSPS Agent <<Received message from NVD Agent: Violation [Violation Type: Voltage Violation, Violation Location: Bus5, Data
Type: Busbar Voltage, Data Value: 1.0319 p.u.]>>

Time Step 28: OPFS Agent <<Received message from NVD Agent: Violation [Violation Type: Voltage Violation, Violation Location: DG B Bus,
Data Type: Busbar Voltage, Data Value: 1.03208 p.u.]>>

Time Step 28: OPFS Agent <<Received message from NVD Agent: Violation [Violation Type: Thermal Violation, Violation Location: Line1,
Data Type: Distribution Line Rating, Data Value: 2.504148547746784 MVA]>>

Time Step 28: CSPS Agent <<Received message from NVD Agent: Violation [Violation Type: Voltage Violation, Violation Location: DG B Bus,
Data Type: Busbar Voltage, Data Value: 1.03208 p.u.]>>

Time Step 28: CSPS Agent <<Received message from NVD Agent: Violation [Violation Type: Thermal Violation, Violation Location: Line1,
Data Type: Distribution Line Rating, Data Value: 2.504148547746784 MVA]>>

Time Step 31: CSPS Agent << Solution : [Name: DG B, Control Type: Active Power, Control Value: 0.8] >>
Time Step 31: CSPS Agent << Solution : [Name: DG A, Control Type: Active Power, Control Value: 1.0] >>
Time Step 32: OPFS Agent << Received Solution from CSPS Agent >>

Time Step 32: OPFS Agent << Solution : [Name: DG B, Control Type: Active Power, Control Value: 0.831373] >>
Time Step 32: OPFS Agent << Solution : [Name: DG A, Control Type: Active Power, Control Value: 1.0] >>
Time Step 32: OPFS Agent << start to compare! >>

Time Step 32: OPFS Agent << Propose to apply OPF solution! >>

Time Step 33: CSPS Agent << Received Proposed Solution from OPFS Agent >>

Time Step 33: CSPS Agent << start to check proposed solution! >>

Time Step 33: CSPS Agent << Agree to apply OPF solution! >>

Time Step 34: OPFS Agent << CSPS Agent Agrees Proposed OPF_Agent  Solution >>

Time Step 34: OPFS Agent << Send Solutions to DG Agents! >>

Time Step 35: DG_B Agent <<Received message from OPFS Agent: Solution : [Name: DG B, Control Type: Active Power, Control Value:
0.831373]>>

Time Step 35: DG_A Agent <<Received message from OPFS Agent: Solution : [Name: DG A, Control Type: Active Power, Control Value:
1.0]>>
Time Step 35: OPFS Agent <<Start to check if can run unconstrained.>>

Time Step 59: OPFS Agent << Can run unconstrained and send to DG agents to run unconstrained.>>
Time Step 60: DG_A Agent <<Received message from OPFS Agent: Solution : [Name: DG A, Control Type: Active Power, Control Value
1.0]>>

Time Step 60: DG_B Agent <<Received message from OPFS Agent: Solution : [Name: DG B, Control Type: Active Power, Control Value:
1.0]>>
```

Figure 5.25 MAS Simulation Interface and Results for Scenario 4

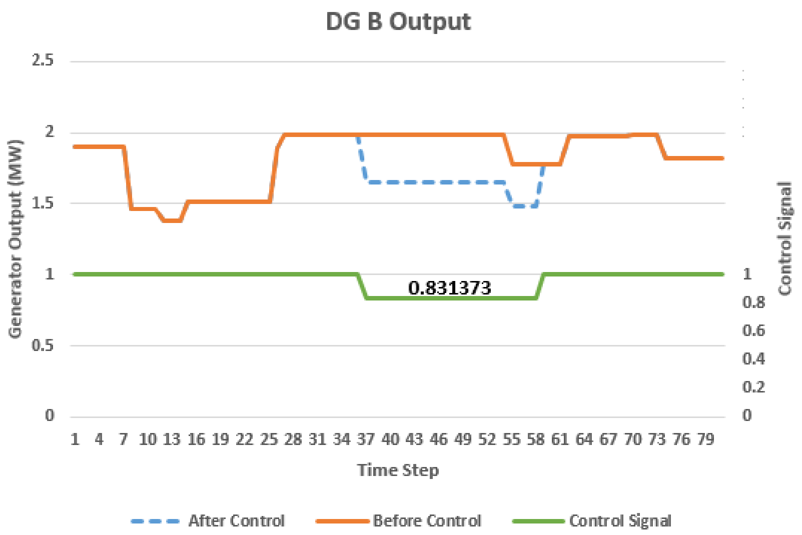


Figure 5.26 DG$_B$ Output and Control Signal for Scenario 4

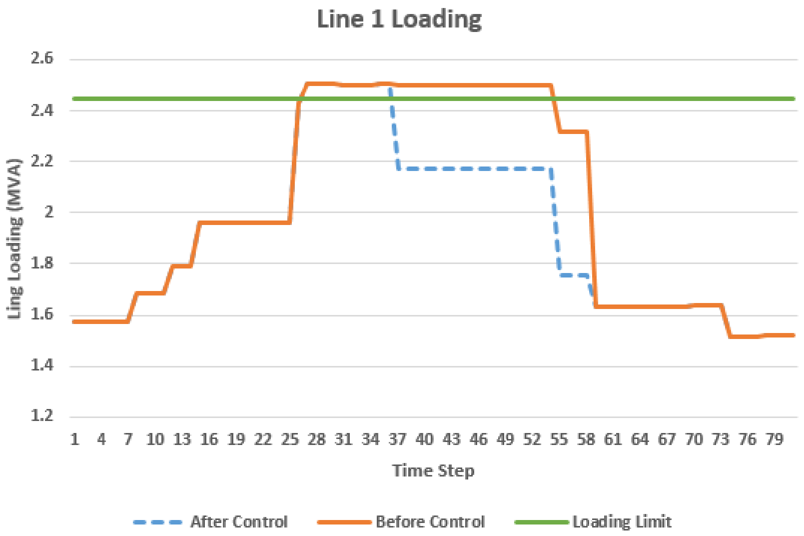


Figure 5.27 Line 1 Loading for Scenario 4

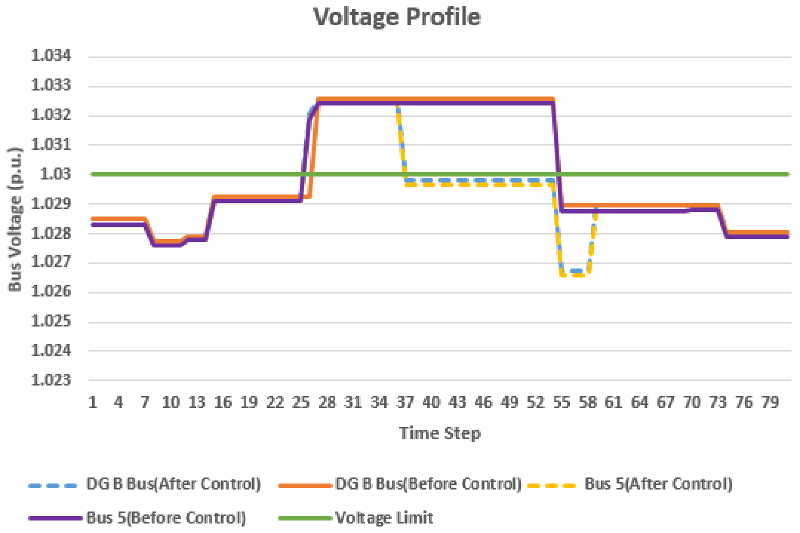


Figure 5.28 Voltage Profile of Violation Bus for Scenario 4

The agent communication diagram for scenario 4 is presented in Figure 5.29, and contains two agent operation stages. For the agent initialisation stage, it is the same as in previous scenarios. The details of the multiple network violations stage are discussed below.

- ***Multiple Network Violations***: Once network violations occurred, the NVD agent informed the CSPS agent and OPFS agent about details of violations. Then, the CSPS agent and OPFS agent started to solve the violation through their integrated control functions. After that, the CSPS agent found a solution first and then sent it to the OPFS agent. When the OPFS agent determined its own solution, the OPFS agent compared it with the solution received from the CSPS agent. As the OPFS agent solution curtailed the DG_B output less than the CSPS agent solution, the OPFS agent proposed its own solution to the CSPS agent. The CSPS agent accepted the proposal and replied with an agreement message to the OPFS agent. As a result, the OPFS agent sent a solution message to the DG_B agent to take control in the network. Once the DG_B can run unconstrained, the OPFS agent sent another message to the DG_B agent to let the DG_B run at rated output.

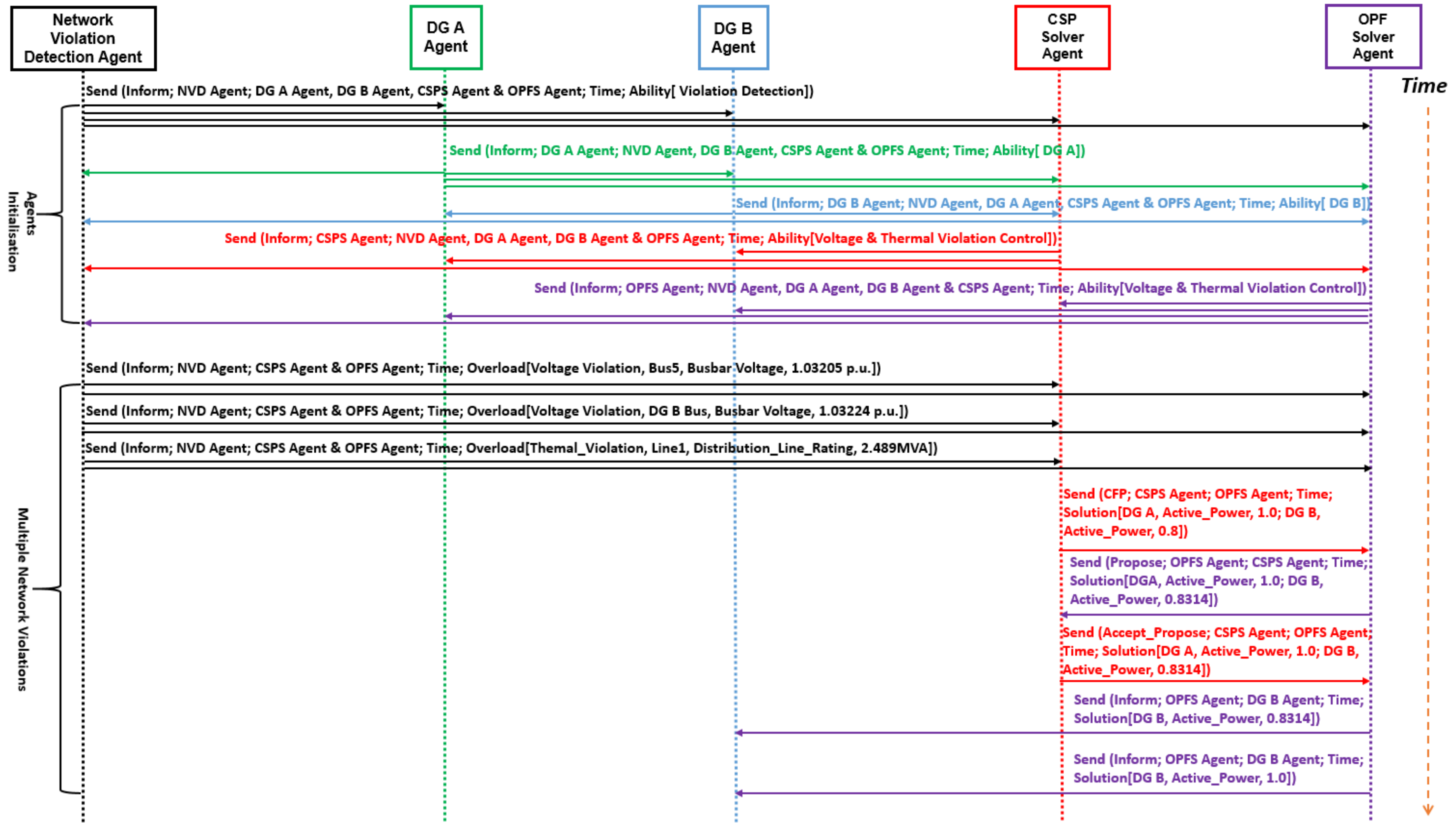


Figure 5.29 Agent Communication Diagram for Scenario 4

5.2.6. Discussion

Case study one has demonstrated the fully integrated MAS architecture for ANM to realise multiple power system control applications through various scenarios, including connection of different solver agents, negotiation between multiple solver agents to determine the best solution for network violations, response to new DG connection, and solving multiple violations occurring simultaneously. From this case study, the agent capabilities and performance of the MAS architecture have also been evaluated. The CSP and OPF control algorithms have been implemented as control functions inside the solver agents through the addition of toolkits, including Python Constraint and PowerWorld, which can provide solutions during network violations for each solver agent in the MAS architecture. Although the OPF function found a solution that did not respect the LIFO order in scenario 2-B, the OPF approach could help in increasing the level of renewable generation outputs while seeking to maintain LIFO rights in scenarios 1, 2-A, 3, and 4. In addition, each agent can follow the designed logic to use relevant behaviours in order to achieve their own objectives. Moreover, agents can collaborate and work collectively to monitor for violations on the network and generate the solution to control relevant DGs to mitigate violations. From the simulation results, network violations can be successfully controlled to bring them back to the normal operational limits by applying control solutions. As a result, the fully integrated MAS is a significant advance for the ANM development for a number of reasons:

- Connecting of various control functions through existing solver agents, thereby increasing the extensibility of the MAS architecture to provide solutions to several critical network violations by implementing the available control algorithms into the solver agent shell.
- Conflict issues from different violation solutions can be solved by developing relevant rules/knowledge within the solver agents so that solver agents can negotiate with each other to determine the best solution.

- The MAS architecture can adapt over time to changes in the network, such as connecting new DG, without the need for redesigning the core system architecture.
- The developed MAS has the ability to deal with multiple violations occurring simultaneously and can eliminate these violations at the same time.

Moreover, the prototype user interface gathers and provides agent interactions, violation information, and control solutions that can make engineers aware of violations and allows them to deploy different agents when they are required. However, engineers can only manage the agents manually through the user interface. A level of self-organisation could greatly enhance the ability of the ANM to manage various control functions and adapt to network changes through agents adjusting themselves. Hence, case study two extends the developed fully integrated MAS architecture to consider an initial self-organising mechanism.

5.3. Case Study Two: Self-Organisation for Network Management

The second case study demonstrates an initial element of self-organising within the developed MAS architecture by distributing multiple instances of the developed MAS architecture within a large distribution network. To achieve this, the ε decomposition and Linear Programming (LP) algorithm are used. This concept was developed by New York University to regulate voltage in the distribution network [86].

Initial research has been undertaken with New York University to prove that their research concept can be deployed within a self-organising agent architecture. This uses the same MAS architecture developed within this thesis, but extends some of the available agents and functions. It is designed to self-configure and launch a MAS control solution within multiple subnetworks of control. Each subnetwork is controlled by the local MAS, not globally. Under certain conditions, the system will "self-organise" into new subnetworks [178].

5.3.1. The ε Decomposition

The ε decomposition algorithm is used to eliminate weak coupling elements in a matrix [179]. For any given matrix, the element of the matrix equal or smaller than the ε is set to zero. To divide a large network into a number of small subnetworks, the ε decomposition of the sensitivity matrix is required. The sensitivity matrix is calculated from the solution of the Newton-Raphson power flow problem that provides the linear relationship between adjustment of DG outputs (active power and reactive power) and voltage changes. After applying the ε decomposition into the sensitivity matrix, the new matrix keeps strong couplings between DGs and nodes while abandoning weak couplings. As a result, a large system is functionally divided into several subnetworks. From the ε decomposition of the sensitivity matrix, the voltage influence range of each DG in the entire network is determined. Once voltage violations occur at particular nodes, the relevant DGs can be located and controlled to solve the voltage issue. In addition, a different ε value can result in different decomposition outcomes, such as numbers of subnetworks and numbers of DGs and nodes covered in each subnetwork.

5.3.2. Linear Programming Algorithm

As the ε decomposition of the sensitivity matrix provides the involved DGs that have influence on the voltage violation nodes, adjustment of these DGs output for solving the voltage violation needs to be calculated. As a result, the objective function of controlling every involved DG outputs can be modelled as linear functions and therefore LP algorithm is used. There are two LP objective functions based on the different control modes, including Power Factor Control (PFC) mode and Unity Power Factor (UPF) mode [86]. For the PFC mode, it is to minimise the reactive power output of every involved DG and the objective function is [86]:

For voltage higher than 1.05 p.u.

$$Max: Min\{x_i\}; \qquad (5.17)$$

For voltage lower than 0.95 p.u.

$$Min: Max\{x_i\}; \qquad (5.18)$$

Subject to the following constraints:

$$\begin{cases} V_l \leq V_0 + \Lambda_{VQ} \cdot x \leq V_u \\ x \leq Q_{surplus} \end{cases} \quad (5.19)$$

Where x_i is the i th adjustment of reactive power for the i th involved DG and x is the vector of all x_i. Λ_{VQ} is the element within the sensitivity matrix.

For UPC control mode, it aims to maximise the active power generation of every involved DG and the objective function is [86]:

$$Max: Min\{x_i\} \quad (5.20)$$

Subject to the following constraints:

$$\begin{cases} V_l \leq V_0 + \Lambda_{VP} \cdot x \leq V_u \\ x \leq P_{surplus} \end{cases} \quad (5.21)$$

Where x_i is the i th adjustment of active power for the i th involved DG and x is the vector of all x_i. Λ_{VP} is the element within the sensitivity matrix.

For the above two LP objective functions, V_0 is the initial voltage and V_l and V_u are the lower bound and upper bound of voltage separately. $Q_{surplus}$ and $P_{surplus}$ are surplus capacities of DG for reactive and active power separately.

5.3.3. Extension of MAS Architecture for Self-Organisation

To implement the ε decomposition and the LP algorithm into the proposed MAS architecture, the ε decomposition agent and the LP Solver (LPS) agent are developed. Moreover, each subnetwork has one LPS agent to regulate voltage at its own subnetwork. The NVD agent is located at each bus to monitor its bus voltage magnitude and report to the LPS agent at the same subnetwork. Furthermore, the LPS agent, NVD agent, and DG agent can only communicate with each other within the same subnetwork. The details of the ε decomposition agent and the LPS agent are described below.

- *ε Decomposition Agent*: It is to provide results of decomposition of the sensitivity matrix based on the selected ε value. As mentioned previously, each subnetwork has one LPS agent. As a result, the numbers of LPS agents varies from different ε decomposition results. In order to update the numbers of LPS agents according to different ε value, the ε decomposition agent is designed to launch or kill the LPS agents based on the ε decomposition results. Moreover, the ε decomposition agent has the ability to change the ε value according to various conditions, such as connecting new DG or failing to regulate the voltage violation at the subnetwork. If the ε decomposition agent needs to change the ε value, it will select a new ε value and calculate new results of ε decomposition from the sensitivity matrix. As a result, the number of subnetworks and the range of influence of each DG on voltage are changed according to the new ε decomposition. After that, the ε decomposition agent will launch new numbers of LPS agents and inform new sensitivity matrix to each LPS agent so that LPS agent can determine the involved agents for voltage violations. The ε decomposition agent can re-group LPS agents without a complete re-engineering of the overall MAS framework. As a result, the MAS platform and other agents do not need to be stopped or restarted, and this can happen at any point.

- *LPS Agent*: The LPS agent is responsible for regulating the voltage violation at its own subnetwork by means of an integrated LP algorithm. Once the LPS agent receives a voltage violation message from the NVD agent, the LPS agent determines the DGs which have the voltage influence on these violating buses according to the sensitivity matrix and calculates adjustment of every involved DG output based on the selected control mode (UPF or PFC). After solutions have been determined, the LPS agent sends the control values by agent messages to these DG agents to take action. If the LPS agent cannot find a solution for voltage violations, it will send a message to the ε decomposition agent about failing to find a solution. Then, the ε decomposition agent is triggered to change the ε value and regroup the DGs and nodes. After that, new LPS agent could adjust outputs of a new set of DGs to solve the voltage violation.

5.3.4. Case Study Network and Simulation Environment

The simulation network model is a real heavily-meshed distribution network that is illustrated in Figure 5.30 [86]. This network contains 2083 nodes (including 1043 nodes at primary feeders at 13.8 kV and remaining1040 nodes are at 480 V or 216 V of the secondary network), 311 PQ loads, and 224 network transformers (13.4 kV to 216 V or 480 V). Moreover, there are 311 DGs installed at the load buses and located at the secondary network. Only the secondary network with 1040 nodes is used for ε decomposition in this case study. The voltage magnitude of this simulation network cannot exceed the normal operation limits by more than 5%, so that should be between the lower bound of 0.95 p.u. and upper bound of 1.05 p.u.. In addition, the system base power is 1 MVA and voltage levels vary from 13.4 kV, 208 V to 460 V in this case study. For case study two, the network model and LP algorithms are both developed based on Matlab by NewYork University. During the case study simulation, the developed MAS architecture is interfacing with the Matlab to receive network data and send control signal by using the Matlab Engine application programming interface for Java [180].

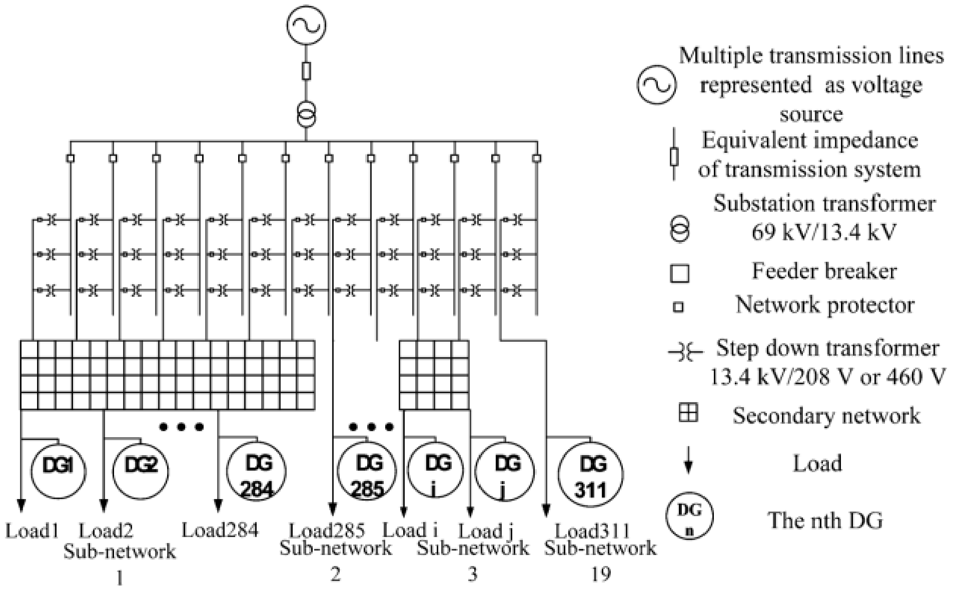


Figure 5.30 Simulation Network Model for Case Study Two [86]

5.3.5. Case Study Two Results

In this case study, it aims to achieve the large subnetwork to have less communication and calculation for MAS and LP algorithm. As a result, the large $\varepsilon = 0.01$ is selected and PFC model for LP algorithm is used for demonstrating the self-organising ability of the developed MAS architecture. From the ε decomposition, the secondary network was divided into 57 subnetworks and therefore the ε decomposition agent generated 57 LPS agents. Then, the NVD agents started to monitor voltage magnitudes of buses and detected a voltage violation at bus 1909 with value 0.9296 p.u. within subnetwork 10 that exceeded the lower bound of the normal voltage operation limit. After that, NVD agent 1909 sent a voltage violation message with violation location and measured voltage value to the LPS agent 10. Based on the violation information, LPS agent 10 determined that the DG_{1379} had the influence on the violation bus. However, the LPS agent 10 cannot find a feasible solution to bring the voltage back to the acceptable limit by adjusting the reactive power of the DG_{1379}. Hence, LPS agent 10 informed the ε decomposition agent about failing to find a solution. The ε decomposition agent was triggered to change the ε value to 0.008 and the network was regrouped into 34 subnetworks. The voltage violation was now detected at subnetwork 4. At that point, the new LPS agent 4 found three DGs had influence on the violation bus, including DG_{1372}, DG_{1373}, and DG_{1379}. In addition, LPS agent 4 determined a solution that constrained reactive power of these DGs to 0.622 p.u., 0.63 p.u. and 0.93 p.u. separately. The voltage profile for this case study before control and after control is illustrated in Figure 5.31.

It is clear from Figure 5.31, that voltage violation could be successfully restored to the normal operation limit between 0.95 p.u. and 1.05 p.u. by increasing reactive power of the involved DGs.

The agent communication diagram of this case study is shown in Figure 5.32. As there are a large number of agents running in case study two, the agent communication diagram only presents those agents who were involved with the voltage violation event. There are two agent operation stages, including the first $\varepsilon = 0.01$ decomposition and the second $\varepsilon = 0.008$ decomposition.

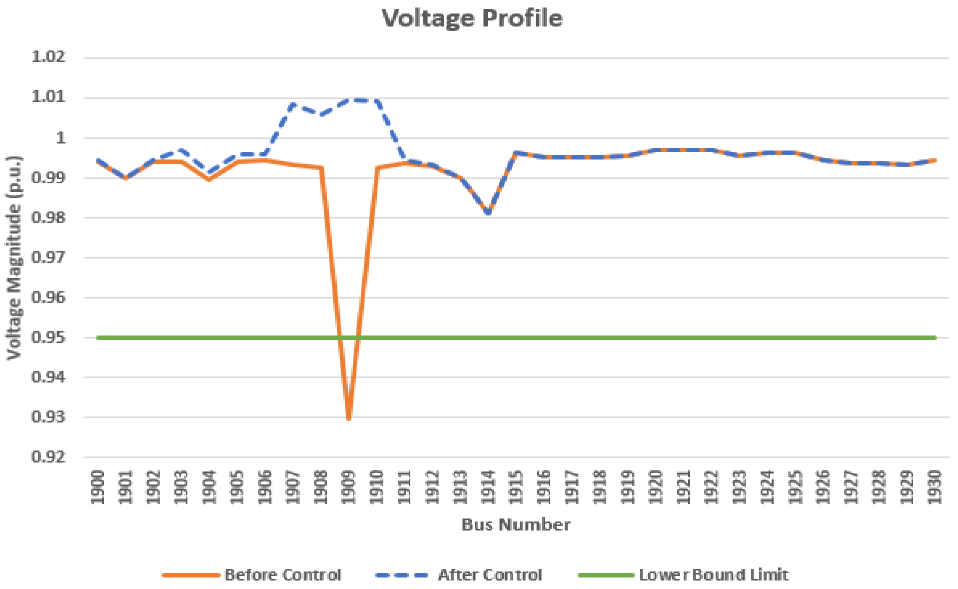


Figure 5.31 Selected Buses Voltage Profiles for Case Study Two

- ***ε = 0.01 Decomposition***: After the first ε decomposition, the ε decomposition agent sent the decomposed sensitivity matrix to the LPS agents to build their knowledge of the sensitivity matrix to determine the involved DGs for the buses. Once a voltage violation occurred, NVD agent 1909 sent violation information to the relevant LPS agent 10. However, the LPS agent could not find a feasible solution and therefore sent a message to the ε decomposition agent about failing to find a solution.

- ***ε = 0.008 Decomposition***: After the ε decomposition agent selected the ε = 0.008, new numbers of LPS agents were generated. Then, the new sensitivity matrix was sent to each LPS agent to build its knowledge of the sensitivity matrix to solve the voltage violation. When the NVD agent 1909 detected the voltage violation, the LPS solver agent 4 was informed and started to find a solution. Finally, the LPS agent 4 determined a feasible solution and sent control messages to the involved DGs to adjust their reactive power output.

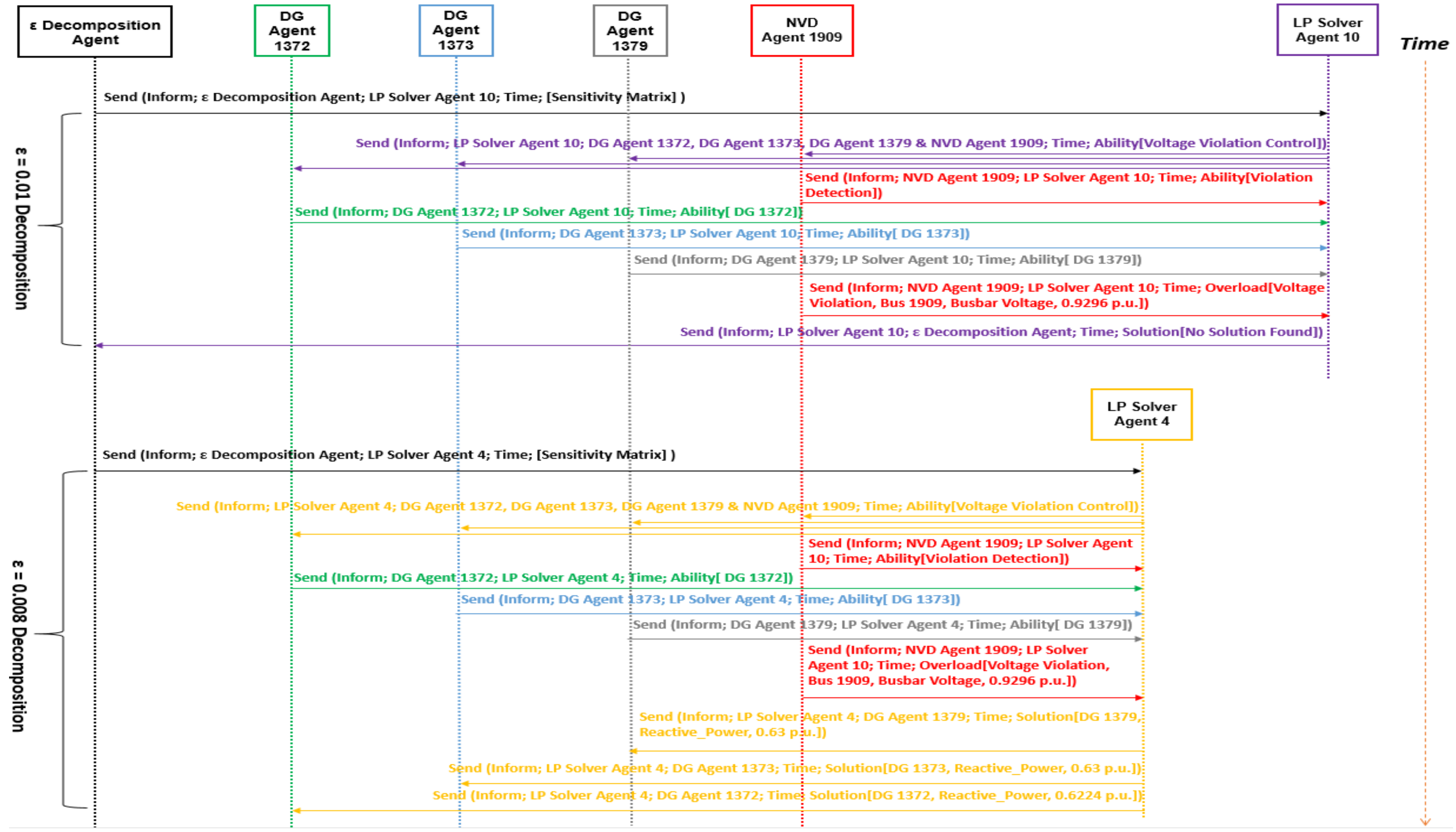


Figure 5.32 Agent Communication Diagram for Case Study Two

5.3.6. Discussion

Within this case study, the fully integrated MAS architecture is extended to provide an initial self-organising ability based on ongoing research with New York University. This focused on distributed voltage regulation based on ε decomposition and the LP algorithm. The implemented ε decomposition agent can generate relevant LPS agents according to the ε decomposition results. Moreover, the ε decomposition agent can select a new ε value if the voltage violation cannot be solved. To mitigate the voltage violation, the ε decomposition agent can re-generate and regroup LPS agents based on the new ε decomposition results without a complete re-engineering of the whole MAS framework. Under the new ε decomposition results, the LPS agent can find a feasible solution that successfully controls the voltage violation and returns the network to the normal operational limit. As a result, self-organisation facilitates automatic decision making for the user by gathering information from the other agents through interactions.

However, this case study is only based on the PFC control mode. Once a voltage violation cannot be solved, the LPS agent could switch to another control mode to search for a solution. Moreover, the ε decomposition agent and the LPS agent can use learning capability to gain experience from failure cases to increase successfully their control rate in the future. In addition, from the self-organising case study, new DG agents could be automatically added to the MAS architecture if there are new DGs connected to the network to enhance the flexibility of the ANM. The research and development of such a MAS is seen as one of the next major steps in self-organisation development.

5.4. Conclusion

The capabilities of the fully integrated MAS architecture have been evaluated through two case studies. In the first case study, the closed-loop simulation results demonstrated that the MAS can successfully integrate different control approaches to achieve ANM solutions for different types of network violations, including thermal overload and voltage violation. In addition, when there are multiple solver agents running on the agent platform, they can negotiate with each other to determine the best solution using their own knowledge and reasoning under different conditions. Moreover, a solver agent can recognise when the DG curtailment signal could be relaxed and removed.

Through implementation and testing different scenarios within case study one, the integrated MAS architecture for ANM has been proven to operate effectively and robustly. Moreover, the fully integrated MAS architecture can support extensible functionality to extend the solver agents and control algorithms. These can also be configured to ensure that they adhere to the LIFO connection agreement.

Testing of a scenario where there is connection of new DG to the network was important to ensure that the agents can adapt to such updates over time. The fully integrated MAS could still resolve the thermal overload without changing the core functions of agents.

Under the multiple violations scenario, it was proven that the MAS architecture can achieve an appropriate control solution by negotiation between solver agents to reach an agreement and mitigate multiple network violations at the same time.

Case study two showed a step towards extending the self-organisation capabilities of the MAS. Agents can self-organise to generate and group LPS solver agents based on the ε decomposition results, and agents can interact with each other locally to regulate voltage in their own subnetwork. This can be extended in the future to probe the opportunities around self-organisation for ANM with more complex network control and operation.

Chapter 6

Conclusions and Future Work

6.1. Conclusions

As more Renewable Energy Sources (RESs) are connected to the grid to reduce greenhouse gas emissions and limit climate change, they can result in significant challenges for power system operation and control, including voltage violation, frequency deviation, and thermal constraints. Active Network Management (ANM) emerged to facilitate RESs' integration whilst solving the aforementioned challenges. As described in chapter two, ANM has been widely applied in both industry and academia by a number of projects, trials and research projects, to enable more Distributed Generation (DG) connection to the existing network and enhance the performance of the network. Moreover, ANM can reduce the expensive costs of network reinforcement for solving the network issues that are caused by numerous DG connections. However, ANM needs to adapt continually to face the future network changes and evolving control functions. As a result, a flexible and extensible control architecture is required to support the future needs of ANM.

Within this thesis, Multi-Agent Systems (MAS) technology has been advocated as a potential approach to achieve an integrated flexible control architecture for ANM by exploiting the benefits of agent technology. MAS technology has numerous benefits:

- Key components and devices in the power system can be modelled as agents, such as DG agents, load agents, bus agents and substation agents, to realise different power system operational purposes with individual objectives, knowledge, and communication and decision making ability. In addition, control algorithms are able to be wrapped as intelligent agents to provide various control functions.
- A MAS architecture offers flexibility to face the network changes, such as adding a new DG, or control device. Agents have the potential to develop such flexibility by sensing the environment and then accommodating the environment changes.
- A MAS provides the extensible capability to add agents when available to enhance the extensibility. As control functions can be developed as intelligent agents, it allows the MAS system to add or delete control functions in various situations to deal with multiple control problems. Moreover, it is possible to add or delete agents in the MAS without the need of re-engineering or relaunching the whole system.
- Agents can be deployed across a range of platforms. As a result, the MAS can operate locally by implementing numbers of agents at distributed local devices.

To realise the benefits of adopting such a MAS approach, a suitable MAS design and development methodology is required. Chapter three described a five-stage methodology that not only provides basic MAS design, such as the initial MAS system goal analysis, requirements, and task decomposition, but also offers agent internal activities design, an interaction specification, ontology design, and agent implementation details and best practice.

Based on the five-stage methodology, a novel fully integrated MAS architecture has been established for ANM. The agent behaviour functions and social ability of each agent have been presented. By following the methodology, the flexibility and extensibility of the MAS architecture can be realised through several key agents, including the Network Violation Detection (NVD) agent, the DG agent, and the

Solver agent. The NVD agent can monitor violations of the network and informs Solver agents about the violations. The DG agent's role is to send a command to the related IED to control the relevant DG. In addition, the DG agent can check a received solution message and evaluate whether it is suitable to apply in cases where it has received multiple solution messages. The solver agent performs the following functions: calculating the violation solution using its integrated control algorithm, determining the best solution through negotiating with other solver agents, and allowing DGs to run unconstrained when DGs can be released from their curtailments. In the agent interaction design, power system control application based ontologies were developed to support the agent communication. Interoperability is enhanced through an IEC 61850 interface inside the agent. To implement the proposed MAS, the Presage2 agent platform was selected as it provides not only the basic agent design infrastructure, such as agent behaviour creation and inter-agent communication development, but also offers the ability to develop self-organisation mechanisms within the MAS. In order to demonstrate the proposed MAS architecture within the power system domain, a control architecture containing two layers was developed, including a MAS layer and a power system layer.

The fully integrated MAS architecture has been implemented in a closed-loop simulation environment and assessed by running a range of different network situations for ANM application. The case studies presented in chapter five demonstrated that the novel MAS provides the control functionality, the flexibility and the extensibility required to tackle the evolving control needs within a power system. These also demonstrated it can be extended to offer the first stages of a self-organising system, proven through its application to voltage regulation. The results show that the agents can cooperate and work collectively to monitor the network violations and can successfully control relevant DGs' outputs to solve network violations.

In summary, the significant contributions of this thesis are as follows:

- A fully integrated MAS control architecture capable of providing multiple control functions to provide the flexibility and extensibility required for ANM. This architecture can integrate various control functions by developing control algorithms as intelligent agents and agents can sense the environment and accommodate network changes.

- Following an investigation and assessment of existing MAS design methodologies, a five-stage methodology has been selected. By following this methodology, a user can fully specify the appropriate MAS from initial requirements and knowledge to final implementation of the flexible and extensible MAS architecture.
- The design of the fully integrated MAS architecture for flexible network management by decomposing it into key sub-tasks that are performed by NVD agents, DG agents and Solver agents. A specification of each agent within the MAS architecture has been created. Based on the requirements of agent communication, an ontology for power system control applications has been developed. Using the IEC 61850 standard, an IEC 61850 interface has been implemented within each agent to facilitate the interoperability between control devices.
- The fully integrated MAS architecture has been developed in the Presage2 platform. The developed MAS is interfaced with a power system simulator that is based on IPSA, which emulates the network and simulates control responses within the network. Trial algorithms were selected (constraint satisfaction problem, optimal power flow, ε decomposition and linear programming) and have been implemented as control functions within the agents.
- The fully integrated MAS architecture has been evaluated using two case studies. One case study has demonstrated the benefits of a flexible and extensible control architecture for ANM through several scenarios, including the connecting of different solver agents, agent negotiations to realise the best control solution, response to network changes caused by adding a new DG, and multiple network violations occurring simultaneously. Another case study extended the proposed MAS architecture to achieve self-organisation through using the ε

decomposition algorithm. These case studies demonstrated that agents can follow the designed logic, use relevant behaviours, collaborate and work collectively to detect network violations and successfully control violations by managing the appropriate DGs' output (active power and reactive power).

6.2. Future Work

Within the UK, several Distribution Network Operators (DNOs) have deployed the ANM (i.e. ANM system has been live on Orkney Islands [181]) or have planned to roll-out the ANM in the near future, such as the Scottish Power Energy Networks will initiate the ANM trial areas at Dumfries and Galloway and North and Mid Wales between 2019-2020 [182]. The Orkney ANM provides real-time control of wind and marine generating units based on measures and control logic. In addition, the voltage issue is solved by reactive compensation equipment. For the power flow constraints, ANM adjusts generator outputs by issuing control signals.

As a result, the fully integrated MAS architecture for ANM developed in this thesis could be used by DNOs to provide enhanced monitoring and solutions for multiple network operation purposes, such as generation/demand scheduling and ancillary service. Moreover, this MAS architecture can provide solutions to utilities and generators to improve the performance of their assets by collecting their data (as case study one shows network violation detection agent can measure network data and detect network violations with its knowledge), such as fault diagnostic [116] and condition monitoring [151], through developing relevant functions within the existing solver agent architecture. As all these applications are running under the real-time environment, the developed MAS architecture of this thesis still needs to be demonstrated and validated within the real-time environment. This need not be on a fully operational systems a physical laboratory environment with real-time digital simulator would be the first logical step.

To achieve remote network monitoring and controlling for efficient network operation, the MAS architecture has to be scalable and deployable across different hardware platforms to collect data or achieve control purposes. A suitable user interface is needed to allow to access the current data for monitoring, take actions to operate the network, and add or delete relevant agents when it is necessary. As the MAS architecture supports various communication interfaces, it has the potential to link with other existing systems or other ANM systems, such as SCADA, distribution management systems, and network management systems, to share the data and create the opportunity for maximum operational benefits.

As a result, the developed MAS architecture still has a number of areas for further work to realise above benefits. The potential future of this research is presented as follows.

- **Adding More Solver Agents with Different Control Functions**

  As the developed MAS architecture is flexible and extensible, it could be expanded for further power system application studies. As a result, research in the area of solving power system problems, such as demand side management, network reconfiguration, frequency control, etc., could be undertaken by creating and implementing relevant Solver agents in the MAS architecture.

- **Learning Capability of Agents**

  Adding learning capability is one of the potential further benefits that could be introduced to the MAS. Learning ability could be introduced into the agents. Learning means that agents can acquire knowledge from the situations and scenarios they encounter. For example, learning can be used to gain experience from successful cases and then can be directly applied to the same case in the future. Moreover, agents can learn from failure cases to increase the successful control rate by improving their control performance. In the self-organising case study, the initial $\varepsilon = 0.01$ could not solve the voltage violation as there was not enough DGs influencing the violation buses in the subnetwork. So, the ε value had to change to 0.008. The ε decomposition agent could learn from this case and build it into its knowledge to apply the suitable ε value for the same case in the future. Furthermore, a solver agent could apply the same solution to the same network violation without negotiating with other solver agents in the near future as learning is applied on that situation and it builds the relevant knowledge within itself.

- **Future Development of Self-Organisation**

  Although this research has proved that the fully integrated MAS architecture can be extended to offer a self-organising mechanism, this self-organising example depends on the ε decomposition algorithm to re-organise the relevant solver agents to regulate voltage in the subnetwork. The developed MAS architecture still needs to enhance its self-organisation capabilities and reduce the reliance on human interventions. For example, in the self-organising case study, the solver agent could switch between different control modes (power factor control and unity power factor) itself if one of the control modes does not find a feasible solution. Moreover, when a new DG connects to the network, the MAS architecture can self-organise to add a relevant DG agent without manual addition through the interface. Furthermore, the MAS architecture could self-organise to add a new Solver agent when made aware that a new control algorithm is available.

- **Deploying on Multiple Agent Platforms**

  One of the major benefits of agent technology is that an agent can be reused and agents are able to run on multiple agent platforms. Within this research, the developed MAS architecture is only running on one agent platform. With the massive deployment of distributed devices across power systems, as distributed an agent deployment as possible would be preferred, and this would involve deploying the agent-based system on multiple platforms. This would require setting up a communication link to transfer data/information between multiple agent platforms, such as TCP/IP.

- **Physical Laboratory Testing**

  In this research, the fully integrated MAS architecture is implemented through simulations and has not yet been tested in a physical environment. In order for agent technology to be employed for practical applications, the next step of this research is to migrate the MAS architecture from simulation into a physical laboratory environment, such as the University of Strathclyde's Microgrid Laboratory. Through the laboratory experience, it can improve the agent technology by addressing the uncovered implementation issues, such as deploying agents within field devices and meeting real-time requirements. As a result, implementing the agent-based system in the laboratory could facilitate agent technology to prove it is practical and suitable for real world power system applications.

Appendix A

11 kV Radial Distribution Network Model Data

A.1. Network Model Overview

The 11 kV radial distribution network model is shown in Figure A.1 that contains 40 buses and 16 loads. The system power base value is 10 MVA and voltage base value is 11 kV. The details of network data are presented in the following sections, including, circuit data, bus data, generator data, and load data.

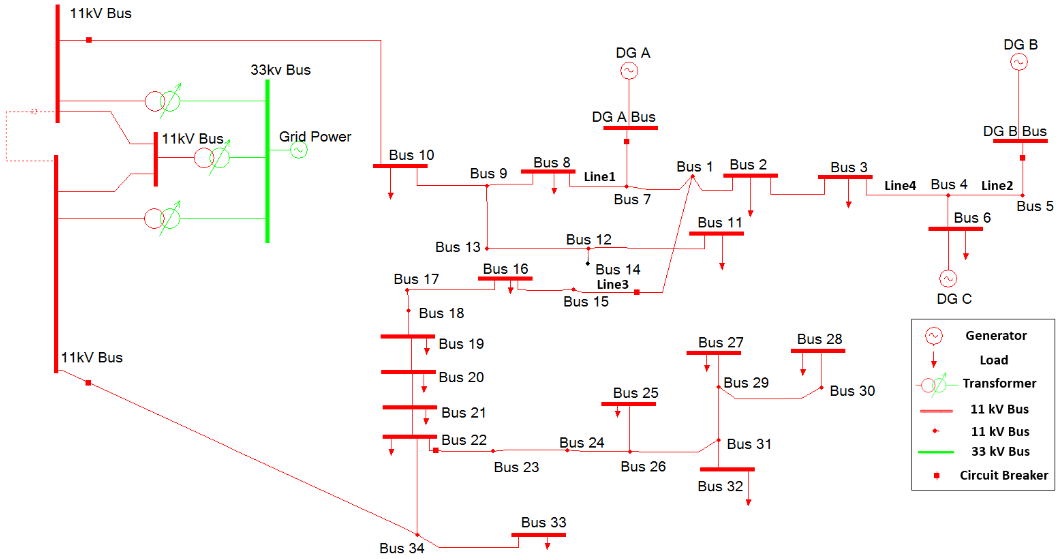


Figure A.1 11 kV Radial Distribution Network Diagram

A.2. Circuit Data

The circuit data of the 11 kV radial distribution network is shown in Table A.1, including resistance and reactance of circuit.

Table A.1 Circuit Data of 11 kV Radial Distribution Network

From Busbar	**To Busbar**	**Resistance (p.u.)**	**Reactance (p.u.)**
11kV Bus	Bus 10	0.017385	0.008455
Bus 13	Bus 12	0.001058	0.000288
Bus 12	Bus 11	0.001322	0.00036
Bus 2	Bus 3	0.000068	0.000033
Bus 3	Bus 4	0.001612	0.000784
Bus 4	Bus 5	0.02993	0.008139
Bus 4	Bus 6	0.000664	0.000323
Bus 10	Bus 9	0.000097	0.000052
Bus 9	Bus 13	0.002538	0.00069
Bus 9	Bus 8	0.00519	0.002782
Bus 8	Bus 7	0.000339	0.000165
Bus 7	Bus 1	0.001707	0.00083
Bus 1	Bus 2	0.002209	0.001074
Bus 26	Bus 25	0.007375	0.001677
Bus 34	Bus 33	0.00456	0.001237
Bus 30	Bus 28	0.001348	0.000367
Bus 29	Bus 30	0.002598	0.000946
Bus 29	Bus 27	0.004428	0.001201
Bus 31	Bus 29	0.000848	0.00023
Bus 31	Bus 32	0.002033	0.000989
Bus 26	Bus 31	0.001206	0.00055
Bus 24	Bus 26	0.001231	0.00066
Bus 23	Bus 24	0.004204	0.003207
Bus 22	Bus 23	0.003645	0.001773
Bus 15	Bus 1	0.000108	0.000053

Bus 16	Bus 15	0.010153	0.002754
Bus 17	Bus 16	0.004759	0.001294
Bus 18	Bus 17	0.012084	0.001934
Bus 19	Bus 18	0.00119	0.000326
Bus 20	Bus 19	0.019209	0.003074
Bus 21	Bus 20	0.031459	0.007153
Bus 22	Bus 21	0.008813	0.002509
Bus 34	Bus 22	0.005921	0.00288
11kV Bus	Bus 34	0.003657	0.001904
33kv Bus	11kV Bus	0	0.1
33kv Bus	11kV Bus	0	0.1
33kv Bus	11kV Bus	0	0.1
11kV Bus	11kV Bus	0.0001	0.0001
11kV Bus	11kV Bus	0.0001	0.0001
11kV Bus	11kV Bus	0	0
DG A Bus	Bus 7	0.001	0.001
DG B Bus	Bus 5	0.001	0.001
Bus 12	Bus 14	0.003437	0.009347

A.3. Bus Data

The bus data of the 11 kV radial distribution network is shown in Table A.2, including nominal voltage, voltage magnitude, and voltage angle.

Table A.2 Bus Data of 11 kV Radial Distribution Network

Bus Name	**Nominal Voltage (kV)**	**Voltage Magnitude (p.u.)**	**Voltage Angle (deg)**
11kV Bus	11	1.022	-4.69
11kV Bus	11	1.022	-4.69
11kV Bus	11	1.021	-4.7

33kv Bus	33	1	0
DG A Bus	11	1.02	-4.58
DG B Bus	11	1.026	-4.47
Bus 1	11	1.02	-4.59
Bus 2	11	1.02	-4.58
Bus 3	11	1.02	-4.58
Bus 4	11	1.02	-4.57
Bus 5	11	1.026	-4.48
Bus 6	11	1.02	-4.57
Bus 7	11	1.02	-4.59
Bus 8	11	1.02	-4.59
Bus 9	11	1.02	-4.62
Bus 10	11	1.02	-4.62
Bus 11	11	1.02	-4.61
Bus 12	11	1.02	-4.61
Bus 13	11	1.02	-4.61
Bus 14	11	1.02	-4.61
Bus 15	11	1.02	-4.59
Bus 16	11	1.018	-4.62
Bus 17	11	1.018	-4.62
Bus 18	11	1.018	-4.63
Bus 19	11	1.018	-4.63
Bus 20	11	1.018	-4.65
Bus 21	11	1.019	-4.68
Bus 22	11	1.02	-4.69
Bus 23	11	1.019	-4.69
Bus 24	11	1.019	-4.69
Bus 25	11	1.018	-4.69
Bus 26	11	1.018	-4.7
Bus 27	11	1.018	-4.69
Bus 28	11	1.018	-4.69

Bus 29	11	1.018	-4.69
Bus 30	11	1.018	-4.69
Bus 31	11	1.018	-4.69
Bus 32	11	1.018	-4.69
Bus 33	11	1.021	-4.69
Bus 34	11	1.021	-4.69

A.4. Generator Data

The generator data of the 11 kV radial distribution network is shown in Table A.3, including real power output and reactive power output.

Table A.3 Generator Data of 11 kV Radial Distribution Network

Busbar	**Generator Name**	**Real Power Output (MW)**	**Reactive Power Output (MVAr)**
33kv Bus	Grid Power	26.686	14.557
DG A Bus	DG A	1.6	0
DG B Bus	DG B	2	0
Bus 6	DG C	1	0

A.5. Load Data

The load data of the 11 kV radial distribution network is shown in Table A.4, including real power output and reactive power output.

Table A.4 Load Data of 11 kV Radial Distribution Network

Busbar	**Real Power Output (MW)**	**Reactive Power Output (MVAr)**
Bus 2	0.81	0.392
Bus 3	0.81	0.392

Bus 6	0.405	0.196
Bus 8	0.32	0.174
Bus 10	0.284	0.137
Bus 11	0.518	0.251
Bus 16	2.5	0.098
Bus 19	0.293	0.142
Bus 20	0.173	0.084
Bus 21	0.241	0.117
Bus 22	0.067	0.033
Bus 25	0.207	0.1
Bus 27	0.675	0.327
Bus 28	0.09	0.044
Bus 32	0.675	0.327
Bus 33	0.158	0.076

Reference

[1]. A. S. Meliopoulos, E. Polymeneas, Z. Tan, R. Huang, and D. Zhao, *Advanced distribution management system*. IEEE Transactions on Smart Grid, 2013. **4**(4): p. 2109-2117.

[2]. *Schneider: Advanced Distribution Management System (ADMS)*. Available from: https://www.schneider-electric.com/en/product-range-presentation/61751-ecostruxure%E2%84%A2-adms?parent-category-id=1900&parent-subcategory-id=1995.

[3]. *ABB: MicroSCADA Pro.*

[4]. F. Campos, et al., *ADMS4LV–advanced distribution management system for active management of LV grids*. CIRED-Open Access Proceedings Journal, 2017. **2017**(1): p. 920-923.

[5]. D. Roberts, *Network management systems for active distribution networks: a feasibility study*. 2004: DTI.

[6]. R. Macdonald, et al., *Challenges of Active Network Management Business As Usual Roll-Out*, in *CIRED Workshop*. 2014: Rome, Itlay. p. 5.

[7]. M. Chen, S. D. McArthur, I. Kockar, and J. Pitt. *Evaluating a MAS architecture for flexible distribution power flow management*. in *Intelligent System Application to Power Systems (ISAP), 2015 18th International Conference on*. 2015.

[8]. *Smarter Grid Solutions: ANM Strata and ANM Element*. Available from: https://www.smartergridsolutions.com/products/.

[9]. *SIEMENS: Spectrum Power ANM*. Available from: https://www.siemens.com/global/en/home/products/energy/energy-

automation-and-smart-grid/spectrum-power-management-systems/spectrum-power-anm.html.

[10]. *GE: PowerOn Advantage*. Available from: https://www.gegridsolutions.com/UOS/catalog/poweron_advantage.htm.

[11]. P. Leitão, V. Mařík, and P. Vrba, *Past, present, and future of industrial agent applications*. IEEE Transactions on Industrial Informatics, 2013. **9**(4): p. 2360-2372.

[12]. E. Newcomer and G. Lomow, *Understanding SOA with Web services*. 2005: Addison-Wesley.

[13]. A. Ankolekar, et al. *DAML-S: Web service description for the semantic web*. in *International Semantic Web Conference*. 2002.

[14]. A. Shashwat, D. Kumar, and L. Chanana. *An end to end security framework for service oriented architecture*. in *Infocom Technologies and Unmanned Systems (Trends and Future Directions)(ICTUS), 2017 International Conference on*. 2017.

[15]. T. Zwiers, *Comparison between information system architectures*. 2011.

[16]. Z. A. Siddiqui and K. Tyagi, *Study on service selection effort estimation in service oriented architecture-based applications powered by information entropy weight fuzzy comprehensive evaluation model*. IET Software, 2017.

[17]. M. Wooldridge and N. R. Jennings, *Intelligent agents: Theory and practice*. The knowledge engineering review, 1995. **10**(2): p. 115-152.

[18]. S. D. McArthur, et al., *Multi-agent systems for power engineering applications—Part I: Concepts, approaches, and technical challenges*. IEEE Transactions on Power systems, 2007. **22**(4): p. 1743-1752.

[19]. *European Technology Platform SmartGrids: Vision and Strategy for Europe's Electricity Networks of the Future*. 2006: Office for Official Publications of the European Communities.

[20]. S. McArthur and E. Davidson. *Multi-agent systems for diagnostic and condition monitoring applications*. in *IEEE Power Engineering Society General Meeting*. 2004.

[21]. C. J. Wallace, G. Jajn, and S. D. McArthur. *Multi-agent system for nuclear condition monitoring*. in *Proc. 2nd International Workshop on Agent*

Technologies for Energy System (ATES-11), a workshop of the 10th International Conference of Agent and Multi-agent System (AAMAS-11), Taipei, Taiwan, 2nd of May. 2011.

[22]. A. Dimeas and N. Hatziargyriou. *Design of a MAS for an Island System*. in *International Conference on Intelligent Systems Applications to Power Systems*. 2007.

[23]. T. Yokoyama and T. Nagata, *A multi-agent restoration method for distribution network.* Energy Procedia, 2012. **14**: p. 726-731.

[24]. H. Wan, K. Li, and K. Wong. *Multi-agent application of substation protection coordination with distributed generators*. in *International Conference on Future Power Systems*. 2005.

[25]. V. Loia, A. Vaccaro, and K. Vaisakh, *A self-organizing architecture based on cooperative fuzzy agents for smart grid voltage control.* IEEE Transactions on Industrial Informatics, 2013. **9**(3): p. 1415-1422.

[26]. H. K. Nunna, et al., *Multi-agent based Demand Response management system for combined operation of smart microgrids.* Sustainable Energy, Grids and Networks, 2016. **6**: p. 25-34.

[27]. M. Pěchouček and V. Mařík, *Industrial deployment of multi-agent technologies: review and selected case studies.* Autonomous Agents and Multi-Agent Systems, 2008. **17**(3): p. 397-431.

[28]. P. Vrba, et al., *Rockwell automation's holonic and multiagent control systems compendium.* IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 2011. **41**(1): p. 14-30.

[29]. P. Vrba, et al., *A review of agent and service-oriented concepts applied to intelligent energy systems.* IEEE Transactions on Industrial Informatics, 2014. **10**(3): p. 1890-1903.

[30]. T. Strasser, F. Andrén, M. Merdan, and A. Prostejovsky. *Review of trends and challenges in smart grids: An automation point of view*. in *International Conference on Industrial Applications of Holonic and Multi-Agent Systems*. 2013.

[31]. V. M. Catterson, E. M. Davidson, and S. D. McArthur, *Practical applications of multi‐agent systems in electric power systems.*

International Transactions on Electrical Energy Systems, 2012. **22**(2): p. 235-252.

[32]. S. Macbeth, D. Busquets, and J. V. Pitt, *System modeling: Principled operationalization of social systems using Presage2*. 2014.

[33]. J.-K. He, *Global low-carbon transition and China's response strategies.* Advances in Climate Change Research, 2016. **7**(4): p. 204-212.

[34]. *The UK Low Carbon Transition Plan: National strategy for climate and energy*: The Stationery Office. 2009.

[35]. E. G. W. Group, *Future network design, management and business environment.* rapporteur contribution, DTI/OFGEM, London, UK, 2000.

[36]. A. Collinson, F. Dai, A. Beddoes, and J. Crabtree, *Solutions for the connection and operation of distributed generation.* EA Technology Ltd, 2003: p. 8-21.

[37]. *A Technical Review and Assessment of Active Network Management Infrastructures and Practices*. 2006: Great Britain. Department of Trade Industry, DTI.

[38]. L. Kane, G. Ault, and S. Gill*, An assessment of principles of access for wind generation curtailment in active network management schemes.* 2013.

[39]. R. A. Currie, et al., *Actively managing wind farm power output.* IEEE Transactions on Power Systems, 2008. **23**(3): p. 1523-1524.

[40]. R. Currie, G. Ault, C. Foote, N. McNeill, and A. Gooding. *Smarter ways to provide grid connections for renewable generators*. in *IEEE Power and Energy Society General Meeting*. 2010.

[41]. C. Foote, et al., *Second generation active network management on Orkney.* 2013.

[42]. *Innovation Funding Incentive and Registered Power Zone Report*: Scottish and Southern Energy Power Distribution. 2013.

[43]. S. Georgiopoulos and G. Ault, *Flexible plug and play project: key considerations for network wide roll out of active network management for distributed generation connections*, in *CIRED Workshop*. 2014: Rome, Italy.

[44]. S. Georgiopoulos, E. Cerqueira, R. Johnston, R. Currie, and C. Marantes, *Flexible plug and play low carbon networks: An open and scalable active*

network management solution for a faster and cheaper distributed generation connection. 2013.

[45]. A. Laguna-Estopier and S. Georgiopoulos, *Active Network Management for integrating renewable energy to the distribution network.* 2013.

[46]. *Flexible plug and play closedown report*: UK Power Networks. 2015.

[47]. P. Wang, et al., *Customer led network revolution-integrating renewable energy into LV networks using energy storage.* 2012.

[48]. L. Sidebotham and N. Powergrid, *Customer-led network revolution project closedown report.* Customer Led Network Revolution. Newcastle upon Tyne, 2015.

[49]. P. Wang, et al., *Integrating electrical energy storage into coordinated voltage control schemes for distribution networks.* IEEE Transactions on Smart Grid, 2014. **5**(2): p. 1018-1032.

[50]. D. Hollingworth, I. Lloyd, and R. Hetherington, *Lessons Learned Report Grand Unified Scheme Active Network Management*. 2014.

[51]. E. N. Association, *Active Network Management Good Practice Guide*. 2015.

[52]. B. O'Neill, et al. *Active network management on islanded systems: The shetland experience*. in *4th IEEE/PES Innovative Smart Grid Technologies Europe (ISGT EUROPE)*. 2013.

[53]. A. H. Stevie Adams, Stewart Reid, *NINES Project Closedown Report*: Scottish & Souther Electricity Networks. 2017.

[54]. E. Davidson, M. Dolan, G. Ault, and S. McArthur. *AuRA-NMS: An autonomous regional active network management system for EDF energy and SP energy networks*. in *IEEE Power and Energy Society General Meeting*. 2010.

[55]. E. M. Davidson, S. D. McArthur, M. J. Dolan, and J. R. McDonald. *Exploiting intelligent systems techniques within an autonomous regional active network management system*. in *IEEE Power & Energy Society General Meeting*. 2009.

[56]. E. M. Davidson, S. D. McArthur, C. Yuen, and M. Larsson. *AuRA-NMS: Towards the delivery of smarter distribution networks through the application of multi-agent systems technology*. in *IEEE Power and Energy*

Society General Meeting-Conversion and Delivery of Electrical Energy in the 21st Century. 2008.

[57]. P. Taylor, et al. *Distributed voltage control in AuRA-NMS*. in *IEEE Power and Energy Society General Meeting*. 2010.

[58]. E. Davidson, M. Dolan, S. McArthur, and G. Ault. *The use of constraint programming for the autonomous management of power flows*. in *15th International Conference on Intelligent System Applications to Power Systems*. 2009.

[59]. M. Dolan, et al. *Using optimal power flow for management of power flows in active distribution networks within thermal constraints*. in *Proceedings of the 44th International Universities Power Engineering Conference (UPEC)*. 2009.

[60]. E. M. Davidson, S. D. McArthur, J. R. MdDonald, and P. Taylor. *An architecture for flexible and autonomous network management systems*. in *20th International Conference and Exhibition on Electricity Distribution*. 2009.

[61]. O. Samuelsson, et al. *Active distribution network—Demonstration project ADINE*. in *IEEE PESInnovative Smart Grid Technologies Conference Europe (ISGT Europe)*. 2010.

[62]. S. Repo, K. Maki, P. Jarventausta, and O. Samuelsson, *ADINE-EU demonstration project of active distribution network.* 2008.

[63]. A. Kulmala, K. Mäki, S. Repo, and P. Järventausta, *Network interconnection studies of distributed generation.* IFAC Proceedings Volumes, 2009. **42**(9): p. 434-439.

[64]. P. Järventausta, S. Repo, A. Rautiainen, and J. Partanen, *Smart grid power system control in distributed generation environment.* Annual Reviews in Control, 2010. **34**(2): p. 277-286.

[65]. C. D'Adamo, et al., *Active distribution networks: General features, present status of implementation and operational practices.* Electra, 2009. **246**: p. 22-29.

[66]. F. Pilo, et al., *Planning and optimisation of active distribution systems-An overview of CIGRE Working Group C6. 19 activities.* 2012.

[67]. E. Peeters, et al. *ADDRESS: Scenarios and architecture for active demand development in the smart grids of the future*. in *20th International Conference and Exhibition on Electricity Distribution*. 2009.

[68]. D. Stein, L. Consiglio, and J. Stromsather. *Enel's large scale demonstration project inside GRID4EU: the challenge of RES integration in the MV network*. in *Electricity Distribution (CIRED 2013), 22nd International Conference and Exhibition on*. 2013.

[69]. D. Sumner, et al., *Research, Development and Demonstration of Peak Load Reduction on Distribution Feeders Using Distributed Energy Resources for the City of Fort Collins*. 2015.

[70]. F. Zavoda. *Advanced distribution automation (ADA) applications and power quality in Smart Grids*. in *Electricity Distribution (CICED), 2010 China International Conference on*. 2010.

[71]. S. Morozumi, et al. *Distribution technology development and demonstration projects in Japan*. in *Power and Energy Society General Meeting-Conversion and Delivery of Electrical Energy in the 21st Century, 2008 IEEE*. 2008.

[72]. C. D'Adamo, et al. *Development and operation of active distribution networks: Results of CIGRE C6. 11 working group*. in *Proceedings of the CIRED 21st international conference on electricity distribution, Frankfurt, Germany*. 2011.

[73]. L. Kane and G. W. Ault, *Evaluation of wind power curtailment in active network management schemes.* IEEE Transactions on Power Systems, 2015. **30**(2): p. 672-679.

[74]. M. J. Dolan, E. M. Davidson, G. W. Ault, K. R. Bell, and S. D. McArthur, *Distribution power flow management utilizing an online constraint programming method.* IEEE transactions on Smart Grid, 2013. **4**(2): p. 798-805.

[75]. M. J. Dolan, E. M. Davidson, I. Kockar, G. W. Ault, and S. D. McArthur, *Distribution power flow management utilizing an online optimal power flow technique.* IEEE Transactions on Power Systems, 2012. **27**(2): p. 790-799.

[76]. S. C. Jupe, P. C. Taylor, and A. Michiorri, *Coordinated output control of multiple distributed generation schemes.* IET Renewable Power Generation, 2010. **4**(3): p. 283-297.

[77]. K. Apt, *Principles of constraint programming*. 2003: Cambridge University Press.

[78]. K.-Y. Huang, H.-T. Yang, and C.-L. Huang. *A new thermal unit commitment approach using constraint logic programming*. in *20th International Conference on Power Industry Computer Applications.,*. 1997.

[79]. H. W. Dommel and W. F. Tinney, *Optimal power flow solutions.* IEEE Transactions on power apparatus and systems, 1968(10): p. 1866-1876.

[80]. Y. Liu, Z. Qu, H. Xin, and D. Gan, *Distributed real-time optimal power flow control in smart grid.* IEEE Transactions on Power Systems, 2016(99): p. 12.

[81]. Z. Dong, F. Zhengcai, Y. Du, and Z. Liuchun. *Capacitor switching and network reconfiguration for loss reduction in distribution system*. in *IEEE Power Engineering Society General Meeting*. 2006.

[82]. Y. Xu and F. Li, *Adaptive PI control of STATCOM for voltage regulation.* IEEE transactions on power delivery, 2014. **29**(3): p. 1002-1011.

[83]. T. C. Y. Ho, L. Ran, A. Garcia-Cerrada, J. L. Zamora-Macho, and P. Brogan. *Voltage source converter AC voltage controller design and dynamic response for a large offshore wind farm network*. in *International Conference on Renewable Energy Research and Applications (ICRERA)*. 2013.

[84]. V. Yarlagadda, B. S. Ram, and K. Rao, *Automatic Control of Thyristor Controlled Series Capacitor (TCSC).* International Journal of Engineering Research and Applications (IJERA), 2012. **2**(3): p. 444-449.

[85]. T. Xu, et al., *Case-based reasoning for coordinated voltage control on distribution networks.* Electric Power Systems Research, 2011. **81**(12): p. 2088-2098.

[86]. L. Yu, D. Czarkowski, and F. De León, *Optimal distributed voltage regulation for secondary networks with DGs.* IEEE Transactions on Smart Grid, 2012. **3**(2): p. 959-967.

[87]. T. Senjyu, Y. Miyazato, A. Yona, N. Urasaki, and T. Funabashi, *Optimal distribution voltage control and coordination with distributed generation.* IEEE Transactions on power delivery, 2008. **23**(2): p. 1236-1242.

[88]. J. R. Castro, M. Saad, S. Lefebvre, D. Asber, and L. Lenoir, *Coordinated voltage control in distribution network with the presence of DGs and variable loads using pareto and fuzzy logic.* Energies, 2016. **9**(2): p. 107.

[89]. N. E. Y. Kouba, M. Menaa, M. Hasni, and M. Boudour. *Load frequency control in multi-area power system based on fuzzy logic-PID controller*. in *IEEE International Conference on Smart Energy Grid Engineering (SEGE)*. 2015.

[90]. D. Lastomo, H. S. Widodo, and M. Djalal. *Enabling PID and SSSC for Load Frequency Control using Particle Swarm Optimization*. in *3nd International Conference on Science in Information Technology (ICSITech)*. 2017.

[91]. S. Muyeen, H. M. Hasanien, and J. Tamura, *Reduction of frequency fluctuation for wind farm connected power systems by an adaptive artificial neural network controlled energy capacitor system.* IET Renewable Power Generation, 2012. **6**(4): p. 226-235.

[92]. S. Agatonovic-Kustrin and R. Beresford, *Basic concepts of artificial neural network (ANN) modeling and its application in pharmaceutical research.* Journal of pharmaceutical and biomedical analysis, 2000. **22**(5): p. 717-727.

[93]. N. Bassiliades and G. Chalkiadakis, *Artificial Intelligence Techniques for the Smart Grid*. 2018, Taylor & Francis.

[94]. B. K. Bose, *Artificial Intelligence Techniques in Smart Grid and Renewable Energy Systems—Some Example Applications.* Proceedings of the IEEE, 2017. **105**(11): p. 2262-2273.

[95]. D. Ye, M. Zhang, and A. V. Vasilakos, *A survey of self-organization mechanisms in multiagent systems.* IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2017. **47**(3): p. 441-461.

[96]. M. Wooldridge, *Agent-based software engineering.* IEE Proceedings-software, 1997. **144**(1): p. 26-37.

[97]. M. Wooldridge, *Multiagent systems: a modern approach to distributed artificial intelligence.* Gerhard Weiss, 1999.

[98]. S. D. McArthur, et al., *Multi-agent systems for power engineering applications—Part II: Technologies, standards, and tools for building multi-agent systems.* IEEE Transactions on Power Systems, 2007. **22**(4): p. 1753-1759.

[99]. C. J. Watkins and P. Dayan, *Q-learning.* Machine learning, 1992. **8**(3-4): p. 279-292.

[100]. C. Iglesias, M. Garijo, J. González, and J. Velasco, *Analysis and design of multiagent systems using MAS-CommonKADS.* Intelligent Agents IV Agent Theories, Architectures, and Languages, 1998: p. 313-327.

[101]. F. M. Brazier, B. M. Dunin-Keplicz, N. R. Jennings, and J. Treur, *Desire: Modelling multi-agent systems in a compositional formal framework.* International Journal of Cooperative Information Systems, 1997. **6**(01): p. 67-94.

[102]. J. C. Garcia-Ojeda, S. A. DeLoach, W. H. Oyenan, and J. Valenzuela, *O-MaSE: a customizable approach to developing multiagent development processes*, in *Agent-oriented software engineering VIII.* 2008, Springer. p. 1-15.

[103]. M. Wooldridge, N. R. Jennings, and D. Kinny, *The Gaia methodology for agent-oriented analysis and design.* Autonomous Agents and multi-agent systems, 2000. **3**(3): p. 285-312.

[104]. M. Cossentino and C. Potts, *PASSI: A process for specifying and implementing multi-agent systems using UML.* Citeseer, 2002.

[105]. P. Bresciani, A. Perini, P. Giorgini, F. Giunchiglia, and J. Mylopoulos, *Tropos: An agent-oriented software development methodology.* Autonomous Agents and Multi-Agent Systems, 2004. **8**(3): p. 203-236.

[106]. L. Padgham and M. Winikoff. *Prometheus: A methodology for developing intelligent agents*. in *Proceedings of the first international joint conference on Autonomous agents and multiagent systems: part 1*. 2002.

[107]. T. Juan, A. Pearce, and L. Sterling. *ROADMAP: extending the gaia methodology for complex open systems*. in *Proceedings of the first*

international joint conference on Autonomous agents and multiagent systems: part 1. 2002.

[108]. A. Collinot, A. Drogoul, and P. Benhamou. *Agent oriented design of a soccer robot team*. in *Proceedings of the Second International Conference on Multi-Agent Systems (ICMAS-96)*. 1996.

[109]. Q.-N. N. Tran and G. Low, *MOBMAS: A methodology for ontology-based multi-agent systems development.* Information and Software Technology, 2008. **50**(7): p. 697-722.

[110]. K. INTERCHANGE, *The DARPA knowledge sharing effort: Progress report.* Readings in agents, 1998: p. 243.

[111]. M. R. Genesereth and R. E. Fikes, *Knowledge interchange format-version 3.0: reference manual.* 1992.

[112]. A. Fipa, *Fipa acl message structure specification.* Foundation for Intelligent Physical Agents, http://www. fipa. org/specs/fipa00061/SC00061G. html (30.6. 2004), 2002.

[113]. T. FIPA, *Fipa communicative act library specification.* Foundation for Intelligent Physical Agents, http://www. fipa. org/specs/fipa00037/SC00037J. html (30.6. 2004), 2008.

[114]. M. Wooldridge, *An introduction to multiagent systems*. 2009: John Wiley & Sons.

[115]. V. M. Catterson, E. M. Davidson, and S. D. McArthur. *Issues in integrating existing multi-agent systems for power engineering applications*. in *Proceedings of the 13th International Conference on Intelligent Systems Application to Power Systems*. 2005.

[116]. J. A. Hossack, J. Menal, S. D. McArthur, and J. R. McDonald, *A multiagent architecture for protection engineering diagnostic assistance.* IEEE Transactions on Power Systems, 2003. **18**(2): p. 639-647.

[117]. S. D. McArthur, S. M. Strachan, and G. Jahn, *The design of a multi-agent transformer condition monitoring system.* IEEE Transactions on Power Systems, 2004. **19**(4): p. 1845-1852.

[118]. N. Gaud, S. Galland, V. Hilaire, and A. Koukam. *An Organisational Platform for Holonic and Multiagent Systems*. in *ProMAS*. 2008.

[119]. F. Bellifemine, G. Caire, A. Poggi, and G. Rimassa, *JADE–A White Paper. exp–In Search of Innovation 3 (2003) 3*. S.

[120]. H. Nwana, D. Ndumu, L. Lee, and J. Collins. *Zeus: A collaborative agents toolkit*. in *Proceedings of the 2nd uk workshop on foundations of multi-agent systems*. 1997.

[121]. O. Gutknecht and J. Ferber. *Madkit: a generic multi-agent platform*. in *Proceedings of the fourth international conference on Autonomous agents*. 2000.

[122]. M. Oey, S. van Splunter, E. Ogston, M. Warnier, and F. M. Brazier. *A framework for developing agent-based distributed applications*. in *International Conference on Web Intelligence and Intelligent Agent Technology*. 2010.

[123]. M. Winikoff, *JACK™ intelligent agents: an industrial strength platform.* Multi-Agent Programming, 2005: p. 175-193.

[124]. R. H. Bordini, J. F. Hübner, and M. Wooldridge, *Programming multi-agent systems in AgentSpeak using Jason*. Vol. 8. 2007: John Wiley & Sons.

[125]. B. Neville and J. Pitt. *PRESAGE: A Programming Environment for the Simulation of Agent Societies*. in *ProMAS*. 2008.

[126]. N. Collier, *Repast: An extensible framework for agent simulation.* The University of Chicago’s Social Science Research, 2003. **36**.

[127]. M. J. North, *Agent-based modeling of complex infrastructures*. 2001.

[128]. F. Perkonigg, D. Brujic, and M. Ristic, *Platform for multiagent application development incorporating accurate communications modeling.* IEEE Transactions on Industrial Informatics, 2015. **11**(3): p. 728-736.

[129]. K. Anisha and M. R. Kumar, *Jade Implementation of Power Restoration in Automated Distributed System.* Indian Journal of Science and Technology, 2015. **8**(19).

[130]. D. F. Estrada and K. Y. Lee. *Multi-Agent System implementation in JADE environment for power plant control*. in *Power and Energy Society General Meeting (PES), 2013 IEEE*. 2013.

[131]. P. Peidaee, A. Kalam, and M. H. Moghaddam. *Developing a simulation framework for integrating multi-agent protection system into smart grids*.

in *Universities Power Engineering Conference (AUPEC), 2017 Australasian*. 2017.

[132]. Y. F. Eddy, H. B. Gooi, and S. X. Chen, *Multi-agent system for distributed management of microgrids.* IEEE Transactions on power systems, 2015. **30**(1): p. 24-34.

[133]. P. Baker, V. Catterson, and S. McArthur. *Integrating an agent-based wireless sensor network within an existing multi-agent condition monitoring system*. in *Intelligent System Applications to Power Systems, 2009. ISAP'09. 15th International Conference on*. 2009.

[134]. K. H. Nunna and S. Doolla, *Responsive end-user-based demand side management in multimicrogrid environment.* IEEE Transactions on Industrial Informatics, 2014. **10**(2): p. 1262-1272.

[135]. K. Kravari and N. Bassiliades, *A survey of agent platforms.* Journal of Artificial Societies and Social Simulation, 2015. **18**(1): p. 11.

[136]. M. E. Elkhatib, R. El-Shatshat, and M. M. Salama, *Novel coordinated voltage control for smart distribution networks with DG.* IEEE transactions on smart grid, 2011. **2**(4): p. 598-605.

[137]. S. C. Müller, U. Häger, and C. Rehtanz, *A multiagent system for adaptive power flow control in electrical transmission systems.* IEEE Transactions on Industrial Informatics, 2014. **10**(4): p. 2290-2299.

[138]. W. Liu, et al., *Decentralized multi-agent system-based cooperative frequency control for autonomous microgrids with communication constraints.* IEEE Transactions on Sustainable Energy, 2014. **5**(2): p. 446-456.

[139]. J. M. Solanki, S. Khushalani, and N. N. Schulz, *A multi-agent solution to distribution systems restoration.* IEEE Transactions on Power systems, 2007. **22**(3): p. 1026-1034.

[140]. A. L. Dimeas and N. D. Hatziargyriou, *Operation of a multiagent system for microgrid control.* IEEE Transactions on Power systems, 2005. **20**(3): p. 1447-1455.

[141]. N. Yorino, Y. Zoka, M. Watanabe, and T. Kurushima, *An optimal autonomous decentralized control method for voltage control devices by*

using a multi-agent system. IEEE Transactions on Power Systems, 2015. **30**(5): p. 2225-2233.

[142]. A. Ashrafi and S. Shahrtash, *Dynamic wide area voltage control strategy based on organized multi-agent system.* IEEE Transactions on Power Systems, 2014. **29**(6): p. 2590-2601.

[143]. U. Häger, et al. *Applicability of coordinated power flow control based on multi-agent systems*. in *IREP Symposium Bulk Power System Dynamics and Control*. 2010.

[144]. P. H. Nguyen, W. L. Kling, G. Georgiadis, M. Papatriantafilou, and L. Bertling. *Distributed routing algorithms to manage power flow in agent-based active distribution network.* in *IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT Europe)*. 2010.

[145]. F. Ren, M. Zhang, and D. Sutanto, *A multi-agent solution to distribution system management by considering distributed generators.* IEEE Transactions on Power Systems, 2013. **28**(2): p. 1442-1451.

[146]. H. Gao, Q. Pang, and Y. An, *New type of protection and control method for smart distribution grid.* 2012.

[147]. H. F. Habib, T. Youssef, M. H. Cintuglu, and O. A. Mohammed, *Multi-Agent-Based Technique for Fault Location, Isolation, and Service Restoration.* IEEE Transactions on Industry Applications, 2017. **53**(3): p. 1841-1851.

[148]. S.-J. Park and J.-T. Lim, *Modelling and control of agent-based power protection systems using supervisors.* IEE Proceedings-Control Theory and Applications, 2006. **153**(1): p. 92-98.

[149]. A. Manickam, S. Kamalasadan, D. Edwards, and S. Simmons, *A novel self-evolving intelligent multiagent framework for power system control and protection.* IEEE Systems Journal, 2014. **8**(4): p. 1086-1095.

[150]. Q. Zhao and S. Liu. *Relay protection based on multi-agent system*. in *International Conference on Electrical and Control Engineering (ICECE)*. 2011.

[151]. A. D. Kenyon, V. M. Catterson, S. D. McArthur, and J. Twiddle, *An agent-based implementation of hidden Markov models for gas turbine condition*

monitoring. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2014. **44**(2): p. 186-195.

[152]. V. Reppa, M. M. Polycarpou, and C. G. Panayiotou, *Distributed sensor fault diagnosis for a network of interconnected cyberphysical systems.* IEEE Transactions on Control of Network Systems, 2015. **2**(1): p. 11-23.

[153]. L. Liu, K. P. Logan, D. A. Cartes, and S. K. Srivastava, *Fault detection, diagnostics, and prognostics: software agent solutions.* IEEE Transactions on Vehicular Technology, 2007. **56**(4): p. 1613-1622.

[154]. Y. Biçen and F. Aras. *Intelligent condition monitoring platform combined with multi-agent approach for complex systems*. in *IEEE Workshop on Environmental Energy and Structural Monitoring Systems (EESMS)*. 2014.

[155]. R. K. Pandey and S. V. Kumar. *Multi agent system based intelligent fault diagnosis with fault current limiter for microgrid*. in *National Power Systems Conference (NPSC)*. 2016.

[156]. P. Bhasin, H. K. Nunna, S. Doolla, and S. Kulkarni. *Multi-agent based diagnosis framework for transformers in a smart distribution system*. in *IEEE International Conference on Power Electronics, Drives and Energy Systems (PEDES)*. 2014.

[157]. F. Tian, F. Wen, X. Wang, Y. Xue, and M. A. Salam. *A multi-agent system based fault diagnosis for active distribution systems*. in *IEEE Innovative Smart Grid Technologies-Asia (ISGT-Asia)*. 2016.

[158]. A. Somani and L. Tesfatsion, *An agent-based test bed study of wholesale power market performance measures.* IEEE Computational Intelligence Magazine, 2008. **3**(4).

[159]. H. Li and L. Tesfatsion. *The AMES wholesale power market test bed: A computational laboratory for research, teaching, and training*. in *IEEE Power & Energy Society General Meeting*. 2009.

[160]. M. Shafie-khah, M. P. Moghaddam, and M. K. Sheikh-El-Eslami, *Ex-ante evaluation and optimal mitigation of market power in electricity markets including renewable energy resources.* IET Generation, Transmission & Distribution, 2016. **10**(8): p. 1842-1852.

[161]. N.-P. Yu, C.-C. Liu, and J. Price, *Evaluation of market rules using a multi-agent system method.* IEEE Transactions on Power Systems, 2010. **25**(1): p. 470-479.

[162]. J. K. Kok, C. J. Warmer, and I. Kamphuis. *PowerMatcher: multiagent control in the electricity infrastructure*. in *Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems*. 2005.

[163]. H. K. Nunna and D. Srinivasan. *An agent based energy market model for microgrids with Distributed Energy Storage Systems*. in *IEEE International Conference on Power Electronics, Drives and Energy Systems (PEDES)*. 2016.

[164]. R. F. Brena, C. W. Handlin, and P. Angulo. *A smart grid electricity market with multiagents, smart appliances and combinatorial auctions*. in *IEEE First International Smart Cities Conference (ISC2)*. 2015.

[165]. T. Logenthiran, D. Srinivasan, A. M. Khambadkone, and H. N. Aung, *Multiagent system for real-time operation of a microgrid in real-time digital simulator.* IEEE Transactions on smart grid, 2012. **3**(2): p. 925-933.

[166]. W. Li, T. Logenthiran, W. Woo, V.-T. Phan, and D. Srinivasan. *Implementation of demand side management of a smart home using multi-agent system*. in *IEEE Congress on Evolutionary Computation (CEC)*. 2016.

[167]. S. D. Ramchurn, P. Vytelingum, A. Rogers, and N. Jennings. *Agent-based control for decentralised demand side management in the smart grid*. in *The 10th International Conference on Autonomous Agents and Multiagent Systems-Volume 1*. 2011.

[168]. N. Yu and L. Yu. *Research on active demand response based on multi-agent simulation*. in *5th International Conference on Electric Utility Deregulation and Restructuring and Power Technologies (DRPT)*. 2015.

[169]. T. Logenthiran, D. Srinivasan, and T. Z. Shun. *Multi-agent system for demand side management in smart grid*. in *IEEE Ninth International Conference on Power Electronics and Drive Systems (PEDS)* 2011.

[170]. S. Karnouskos and P. Leitão, *Key Contributing Factors to the Acceptance of Agents in Industrial Environments.* IEEE Transactions on Industrial Informatics, 2017. **13**(2): p. 696-703.

[171]. *IPSA Power*. 2017; Available from: https://www.ipsa-power.com/.

[172]. *MATPOWER*. 2017; Available from: http://www.pserc.cornell.edu/matpower/.

[173]. M. Uschold and M. Gruninger, *Ontologies: Principles, methods and applications.* The knowledge engineering review, 1996. **11**(2): p. 93-136.

[174]. S. M. Blair, F. Coffele, C. D. Booth, and G. M. Burt, *An open platform for rapid-prototyping protection and control schemes with IEC 61850.* IEEE Transactions on Power Delivery, 2013. **28**(2): p. 1103-1110.

[175]. S. Macbeth, J. Pitt, J. Schaumeier, and D. Busquets. *Animation of self-organising resource allocation using presage2*. in *IEEE Sixth International Conference on Self-Adaptive and Self-Organizing Systems (SASO)*. 2012.

[176]. G. Niemeyer. *Python Constraint*. Available from: https://labix.org/python-constraint.

[177]. *PowerWorld Simulator*. Available from: https://www.powerworld.com/.

[178]. M. Chen, et al. *Design of a multi-agent system for distributed voltage regulation*. in *19th International Conference on Intelligent System Application to Power Systems (ISAP)*. 2017.

[179]. M. Amano, A. Zecevic, and D. Siljak, *An improved block-parallel Newton method via epsilon decompositions for load-flow calculations.* IEEE transactions on power systems, 1996. **11**(3): p. 1519-1527.

[180]. *Matlab Engine API for Java*. Available from: https://uk.mathworks.com/help/matlab/matlab-engine-api-for-java.html.

[181]. M. Plecas, S. Gill, I. Kockar, and R. Anderson. *Evaluation of new voltage operating strategies for integration of distributed generation into distribution networks*. in *PES Innovative Smart Grid Technologies Conference Europe (ISGT-Europe), 2016 IEEE*. 2016.

[182]. G. Boyd, *SPEN–DSO Vision.* CIRED-Open Access Proceedings Journal, 2017. **2017**(1): p. 2007-2010.