IMPLEMENTATION OF A MULTI-AGENT CONTROL SCHEME FOR AUTONOMOUS MICROGRIDS

By

Khaleel O Khadedah

A THESIS

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

Electrical Engineering - Master of Science

2015
ABSTRACT

IMPLEMENTATION OF A MULTI-AGENT CONTROL SCHEME FOR AUTONOMOUS MICROGRIDS

By

Khaleel O Khadedah

This thesis describes the implementation of a Multi-Agent control scheme for autonomous microgrids. This is realized by a Multi-Agent system, which executes a three-stage algorithm to control the voltage at the nodes and the power balance of the microgrid. The power system is simulated by using a Real Time Digital Simulator (RTDS). The Gauss method is used for distributed calculation of power flows. Additionally, using a Multi-Agent system in a low-voltage distribution network to fulfill voltage constraint, power balance and frequency droop for low voltage are also discussed. In the next step a script feature of RSCAD and file sharing protocol over the local area network (LAN) is adopted for establishing of necessary communication between the agents and the RTDS. Numerous hardware configurations for the interface were examined during the research work. This interfacing shows a flexible communication structure to exchange the data. The link between the agents and the RTDS has been examined and the results are given. The final stages of the project an interface using the DAQ toolbox was investigated. The communication speed among the agents, the RTDS, and the hardware available were a constraint that needed to be considered for constructing the framework. However, the communication between the agents and the system has been established with regard to the communication speed.
To every soul who has fallen fighting terrorism
ACKNOWLEDGMENTS

I would like to express my deepest gratitude to my advisor Professor Joydeep Mitra at Michigan State University for his excellent guidance, caring, patience, and providing me with an excellent atmosphere through the work on my first research thesis and giving me valuable advices. The door to Professor Mitra’s office was always open whenever I ran into a trouble spot or had a question about my research or writing. He consistently steered me in the right the direction whenever he thought I needed it. Thanks for giving me the opportunity to be part of your research group. I would never have been able to finish my thesis without the guidance of Professor Mitra, help from friends, and support from my family.

I would like to thank Stefan Lang, Dr. Niannian Cai, and Venkateswaran Lakshminaryanan who as a good friends, were always willing to work with me and give their best suggestions. It would have been a lonely lab without them.

I would like to thank my fellow labmates the members of Professor Mitra’s research group, especially, Dr. Mohammed Ben-Idris, Yuting Tian, Nga Nguyen, and Samer A. Sulaeman who supported me in the various sections of my work.

I would like to thank my loved ones, specially my wife, my children, and Patricia Walters who have supported me throughout the time of my study, both by keeping me harmonious and helping me putting pieces together. I will be grateful forever for your love.

Finally, and most importantly, I would like to express my most profound gratitude to my parents for their endless love, consistent support and encouragement throughout this research.
# TABLE OF CONTENTS

LIST OF TABLES ........................................................................ vii

LIST OF FIGURES ...................................................................... viii

Chapter 1 Introduction .............................................................. 1
  1.1 Problem Statement ......................................................... 4
  1.2 Project Objective .......................................................... 5

Chapter 2 Background .............................................................. 6
  2.1 Microgrid (MG) ............................................................ 6
  2.2 Multi-Agent System ...................................................... 8
  2.3 Agent Characteristics .................................................. 9
    2.3.1 Agent Communication ........................................... 11
  2.4 Hardware ................................................................. 12
    2.4.1 Raspberry Pi ......................................................... 13
      2.4.1.1 Programmable interface ................................. 13
    2.4.2 Real-Time Digital Simulator: RTDS ....................... 14
  2.5 Interfacing Software .................................................... 16
    2.5.1 Operating System ................................................. 16
      2.5.1.1 Remote Desktop Connection ......................... 17
    2.5.2 RSCAD ............................................................. 19
      2.5.2.1 Draft Sheet .................................................. 19
      2.5.3 Runtime Sheet .................................................. 20
        2.5.3.1 Script File ................................................ 20

Chapter 3 Distributed Power Balance Control and Voltage and Frequency Regulation ......................................................... 22
  3.1 Distribution Network Model and Power Balance ................... 24
  3.2 Frequency Droop Control in a Microgrid ............................. 25
  3.3 Voltage Regulated Distribution Transformer (VRDT) ............ 29
    3.3.1 Cause of Different Voltage Potentials ....................... 34
  3.4 Steps of Voltage Regulation .......................................... 39
    3.4.1 Communication Structure: ................................... 44
    3.4.2 Agent Based Power Flow ..................................... 48
  3.5 Five-Bus Test System .................................................. 51
    3.5.1 Example Calculation: .......................................... 51
      3.5.1.1 \( Y_{bus} \) Construction: .................................. 52
      3.5.1.2 Multi-Agent Based Power Flow Calculation .......... 52
3.5.1.3 Net Power Calculation

Chapter 4 Interfacing Methods

4.1 Method 1

4.2 Method 2

4.3 Method 3

4.4 Method 4

4.5 Simulation and Results

4.6 Power System Models

4.6.1 Runtime

4.7 Script feature of RSCAD

Chapter 5

5.1 Conclusion

5.2 Future Work

BIBLIOGRAPHY
## LIST OF TABLES

<table>
<thead>
<tr>
<th>Table 3.1</th>
<th>Power Value at Each Bus of the System</th>
<th>42</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table 3.2</td>
<td>Parent Child Relationship Diagram</td>
<td>46</td>
</tr>
<tr>
<td>Table 3.3</td>
<td>Bus Input Data for Five-Bus System</td>
<td>51</td>
</tr>
<tr>
<td>Table 3.4</td>
<td>Line Input Data for Five-Bus System</td>
<td>52</td>
</tr>
<tr>
<td>Table 3.5</td>
<td>Local $Y_i$ Vectors</td>
<td>52</td>
</tr>
<tr>
<td>Table 3.6</td>
<td>Results for the First Five Iteration</td>
<td>54</td>
</tr>
</tbody>
</table>
# LIST OF FIGURES

| Figure 2.1 | Raspberry Pi Model B+ 512MB RAM, 2-USB-Ports und Etherne (www.vesalia.de) | 13 |
| Figure 2.2 | a) Full view of the RTDS rack; b) 3 GPC processor units, 1 WIF ethernet connection card, digital I/O-port; c) Screenshot of RTDS software "RSCAD" | 15 |
| Figure 2.3 | Remote connection from a windows operating system to Ubuntu. The registered hostname was used to address the required machine | 18 |
| Figure 3.1 | Hierarchical control structure of microgrids | 25 |
| Figure 3.2 | Power Flowing through a Line | 26 |
| Figure 3.3 | Droop Control Characteristic Plots | 28 |
| Figure 3.4 | Example for a grid and the position of agents | 30 |
| Figure 3.5 | Microgrid scheme with VRDT | 33 |
| Figure 3.6 | Complete representation of a line | 34 |
| Figure 3.7 | Process describing the development of the vector $V_{diff}$ | 39 |
| Figure 3.8 | Example of the information flow, which transports the highest and lowest voltage of a node to the first agent | 40 |
| Figure 3.9 | Configuration of the grid | 43 |
| Figure 3.10 | The diagram shows the influence of the several steps on the voltage at each node | 44 |
| Figure 3.11 | (a) Token transmission route and removed redundancy between agent 3 and 5. (b) Minimal spanning tree constructed. (c) Information flow path for stage 2 | 46 |
| Figure 3.12 | Tested five-bus system | 51 |
a) A fibre optic cable connects the RTDS internal communication system to an FPGA board. The board converts the data and builds an interface with the Ethernet network; b) With the help of an AD/DA converter the outputs of several channels can be converted in digital values and send to a computer or directly in a Ethernet network.

Interfacing using the script feature of RSCAD over the LAN (Method 3).

Interfacing using DAQ (Method 4).

System consisting three generating and three load units.

System consisting three generating and three load units.

System with one wind turbine, one solar PV unit, one DG and four load units.

Shows the plots of simulated system consisting of six buses, four load units, with three generating units.

Sliders to adjust the generation and load.

Meters showing values of power generated and bus voltage and Slider for adjust the wind turbine generation and the push buttons.

Script feature in Runtime sheet of RSCAD.

Flow chart for the programming of the script.

shows the script_in.csv outputted by the system which is the input to the agents.

shows the GUI of the agents.

the output script_out.csv file given by the agents then this file is sent to RTDS.
Chapter 1

Introduction

The rapid evolution of electricity supply will have major implications for reliability, transmission, distribution, consumer engagement, security, and integration. Regardless of the ultimate generation mix, the electric grid will play a critical role in future electricity infrastructure [1].

The integration of renewable energy sources, and the liberalization of energy markets, therefore, require elementary operations in the still predominantly classical structure of the existing power grids. The partial automation of the middle and low-voltage distribution grid is a major challenge. This is mainly due to the efforts to enable energy to be generated primarily from renewable sources. This contributes highly to the reduction of CO₂ emissions of the energy sector, but it will not be enough for complete elimination. An increase in the efficiency of energy applications is also required [2]. Thoughtful debate and planning are needed today in order to address tomorrow’s challenges and seize tomorrow’s opportunities. With this in mind, the concept of a microgrid will fit these aspirations.

According to [3] “A MicroGrid is a group of interconnected loads and Distributed Energy Resources (DER) with clearly defined electrical boundaries that acts as a single controllable entity with respect to the grid [and can] connect and disconnect from the grid to enable it to operate in both grid connected or island mode.” The microgrid is an independent section of the electrical distribution grid that can transmit, produce, and distribute power within a localized area. The primary components of a microgrid consist of Distributed Energy
Resources (DERs), loads, and controllable interconnection ties between the microgrid and external power sources, in which all three must coordinate effectively to behave as designed. Implementation of the microgrid increases the reliability and quality of the power supplied to the microgrid loads. For areas where the loss of electricity may cause severe economic damage or loss of human life, the implementation of a microgrid is invaluable. However, microgrids have been historically developed as closed, tightly controlled systems for a limited number of applications. Moreover, the control strategies for microgrids have traditionally been developed for systems consisting of few control variables, including the types and capabilities of DERs, the microgrid loads, and the power flow at the interconnection ties. The result is a relatively simple control strategy which can manage anticipated events before the microgrid is implemented. Furthermore, as the concept of the microgrid expands to include loads and DERs beyond the capabilities of a single, centralized controller, such as residential subdivisions or several blocks in a central business district, the number of control variables the control strategy is required to deal with increases substantially [4, 5, 6, 11, 13].

This applied research thesis is a continuation of a research developed in Energy Reliability and Security Research Laboratory (ERISE). The use of a regulated transformer in cooperation with a Multi-Agent system is a useful solution to maintain the voltage stability in a grid. Several agents can take the voltage measurement across the grid and induce a control intervention of the agents installed on Voltage Regulated Distributed Transformer (VRDT) in excess of the prescribed limits, so the correction algorithm control steps are shown later in chapter 3. A software which currently controls the communication flow is described in section 2.3.1. Also mathematical algorithms and functions can be integrated into the software to test their influences. For the simulation of real-time values, a possible connection to a Real Time Power System simulator (RTDS) is described in section 2.4.2.
Reference [7] used the RTDS, and the signals had been taken out from the output ports either (analog or digital) through an optic fiber cable to a Field Programmable Gate Array (FPGA) which acts as the interfacing medium between the RTDS and the agents. This method would be suitable for large systems where there are many GTAO or GTIO. Due to hardware constraints where only one GTAO port is available, the signals had to be sent sequentially. The agents were neither programmed to take in signals sequentially nor to give signals sequentially. However, this method gave an idea to find a better interface with the agents. While working with sending signals sequentially, RSCAD’s script features were considered. While working with the script file, a method of interfacing the agents with RTDS using no extra hardware was found; the connection was over the Ethernet itself. This method does not require any special or external hardware and interfacing could be done using the Ethernet over the LAN. The script file in RSCAD was programmed for interfacing in this method. The steps of how it works are explained in section 4.3.

In chapter 3 the voltage and frequency regulation in a low voltage microgrid is briefly discussed with mathematical representation of a typical transmission line.

In later chapters, methods of interfacing that describe the practical contribution to the project running in ERISE are discussed. In a microgrid, integrating and interfacing sensing and control devices is challenging because it involves different communication protocols. the final stages of the project an interface using the DAQ toolbox was started implemented.

Finally, simulation and results are given in chapter 4. The main part of this project was the development of the required hardware structure and the establishment of operating systems. In addition, the previously mentioned software was developed to simulate the behavior of a Multi-Agent system.
1.1 Problem Statement

The ongoing development in the growth of renewable energy technology has brought in the subject of microgrids. This present state of microgrids is volatile, as it not only involves inefficient power sources (e.g. wind, solar), but also has very heavy demand with heat pumps, air conditioners, electric vehicles, etc. To maintain the required guidelines for voltage stability and frequency stability, operation and control, a human operator is required. However; it is difficult because there are some restrictions related to operation costs and privacy issues. To overcome these restrictions and for autonomous operation of the microgrids, an intelligent agent system for microgrids operation has been studied as a potential solution which mostly performs the required processes. In this project the implementation of a Multi-Agent control scheme for autonomous microgrids is described. The intelligence in these networks consists of many different components. It must be equipped with hardware components which include sensors and actuators. These systems are additionally connected to other agents over a communication medium. The control within this interactive network must be implemented using specialized software.

The main aim of this project was to build a proper communication link between the agents and the system. This part is proved to be a major challenge because of the difference in the communication speed. The system simulated in the RTDS runs at real time hence the communication speed was about 100,000 samples per second, on the other hand the agents were established in the Raspberry Pi which is a smallest microprocessor and had a communication speed of a few cycles per second. The second constrain that was faced during this interfacing was the hardware that was available.
1.2 Project Objective

It can be seen in this summary that many different elements are a part of this project. In this project the implementation of a Multi-Agent control scheme for autonomous microgrids is described. With its help, the interaction of several agents can be simulated and tested. The software that used to control the communication flow is described in section 3.4.1.

The way using Multi-Agent system to calculate the difference in data between nodes in the microgrids is the objective of this applied research. The integration of mechanisms for automating the distribution network presents a particular challenge because they have to be integrated into existing structures. The use of a regulated transformer in cooperation with a Multi-Agent system is, therefore, a useful solution to maintain the voltage stability in a grid. A case study with simulation and calculation is shown. Also, the communication structure of the agents and information flow path for this structure are explained with definition and mathematical representation. For power balance control, a method using a Multi-Agent system in a low-voltage distribution network to fulfill voltage constraints is discussed and results are given. Notably renewable generation facilities, such as wind power, solar and biomass are found in rural areas, in contrast to the low consumption of the population. In return, the regenerative power generated in cities is rather low, compared to the grid size. Thus, it is difficult to ensure a balanced decentralized system of producers and consumers. In some areas, the installed renewable energy substantially exceeds the installed consumer loads. In order to maintain grid stability, power flow control and voltage stability are required. These are also needed to obtain the protection of the network and the consumer and to prevent the over-load of the grid. Benefits from MAS include accomplishing attributes, avoiding single point failures, and realizing distributed control.
Chapter 2

Background

This chapter will explain some important background information; the following sections describe the main idea of microgrids. This leads to the concept of Multi-Agent systems, which are based on a variable number of agents. The microgrid are considered important for the infrastructure of smart grid, however fluctuation and intermittence resulted from unstable micro-sources and nonlinear loads will execute considerable impacts on normal operation of the microgrid. Energy storage (ES) technology presents a preferable solution to the above issue.

2.1 Microgrid (MG)

Due to the increasing shortage of fossil fuels and the compelling pressures from environmental protection, new generation sources of high efficiency such as fuel cell and micro gas turbine, as well as Renewable Energy Sources (RES) such as wind and solar power, are becoming important DERs. The distributed and renewable energy sources will see a remarkably increasing portion in the whole electric power generation. For example, in California, USA, 20% of energy generated must be from RES by 2017, while 15% will be achieved in China by 2020, according to a government report [8]. A microgrid combined with renewable energy sources and distributed generation sources can be a preferable solution to the raised energy crises as well as a complement to centralized modern power grids. However, due to
relatively small capacity, the normal operation of microgrids may be vulnerable to random power exchange between the supplier and the loads. For instance, when a microgrid works at the islanded mode, possible malfunction of some micro-source or DG will cause immediate active and reactive power shortage. Though it might eventually be remedied by load shedding, the interim power shortfall must be instantly compensated from somewhere else, for which case energy storage technology is critical and necessary. More importantly, many RES and DGs behave intermittently, due to their strong dependency on climatic and meteorological conditions. As a result, the RES energy output will fluctuate. Energy storage is presently the important equipment in a microgrid as the smart way to restrain probable power fluctuation and deal with arduous imbalance challenges between the demand side and the supply side [9]. Microgrids are modern, small-scale versions of the centralized electricity system. They achieve specific local goals established by the community being served, such as reliability, carbon emission reduction, lower greenhouse gas emission, diversification of energy sources, reduce the stress on the transmission and distribution system and cost reduction. Like the bulk power grid, smart microgrids generate, distribute, and regulate the flow of electricity to consumers, but do so locally. Smart microgrids are an ideal way to integrate renewable resources on the community level and allow for customer participation in the electricity enterprise. A microgrid is capable of operating in parallel with, or independently from, the main grid. The primary purpose is to ensure reliable, affordable energy security for private households, commercial, industrial and federal government consumers. The core of a microgrid will be one or more small conventional generation assets (e.g. engines or turbines) fuelled by natural gas, biomass or landfill methane. When connected to the main grid, microgrids will rely on a mix of power generation sources depending on the metric to be optimized. Specialized hardware and software systems control the integration
and management of the microgrids components and the connection to the utility.

A microgrid includes generation, a distribution system, consumption and storage, and manages them with advanced monitoring, control and automation systems. The critical first step of pursuing a microgrid solution is a permanent reduction in consumption (electricity, water and gas). This will give the consumer near-term cost savings driven by measured and verifiable conservation measures. A fully-developed microgrid has the capability of automatically disconnecting and operating independently from the main grid. For example, if a storm disrupts energy service from the main grid, automated controls will reduce non-critical loads (selected lighting, HVAC systems, etc.) and the microgrid will distribute power from on-site generation and storage for an extended period of time. When the main grid is back online, the microgrid will automatically reconnect, recharge energy storage, and ramp down on-site generation as appropriate [10], [12].

To fulfill all these functions, very often Multi-Agent Systems (MAS) are proposed for the autonomous control of microgrids. These systems can usually adapt the flexible structure of the grids. New algorithms and mathematical models can be developed and tested for the control of a microgrid.

**2.2 Multi-Agent System**

The terminology of Multi-Agent system (MAS) was originally modeled in the information technology and extended to automation. A Multi-Agent system is a system comprising two or more agents or intelligent agents. The system designer’s intentions for the system can only be realized by including multiple intelligent agents, with local goals corresponding to sub-parts of that intention. Such an agent is generally a computer system embedded to
the physical system, which is capable thereby to perform autonomous actions and tasks to affect its condition itself. In our case, these automation capabilities can be useful for the energy sector. The advantage is that the desired intelligence are able to control the power flow or the frequency and voltage and should be able to work independently. Thus, the Multi-Agent system consists of a variety of distributed software and hardware units that follow a defined goal in a working environment. Furthermore, they should be provided with a communication structure that allows them communicate and collaborate with each other. The agent captures information through the use of various sensors, which measure for example, current or voltage at a node. A host of different actuators are also required to obtain the desired result. For example, these actuators could influence power electronic devices or control inverters or voltage regulated transformers [14, 31].

2.3 Agent Characteristics

To understand the behavior of a Multi-Agent system, therefore, the status of each agent must be considered. Multi-Agent system has found its way into a number of technologies and has been widely used. For example, in artificial intelligence, databases, operating systems and computer networks literature. Although there is no single definition of an agent, all definitions agree that an agent is essentially a special software component. The agent has autonomy that provides an interoperable interface to an arbitrary system and/or behaves like a human agent, working for some clients in pursuit of its own agenda. In the special case of a power network, these interoperable interfaces are sensors and actuators or power electronics, as well as communication bindings. Even if an agent system can be based on a solitary agent working within an environment and if necessary interacting with its users, usually
they consist of multiple agents. The agents may interact with each other both indirectly (by acting on the environment) or directly (via communication and negotiation). Agents may decide to cooperate for mutual benefit or may compete to serve their own interests [15]. Some of the required qualities of the agents are:

- **Autonomous**, because it operates without the direct intervention of humans or others and has control over its actions and internal state.

- **Social**, because it cooperates with other agents in order to achieve its tasks.

- **Reactive**, because it perceives its environment and responds in a timely fashion to changes that occur in the environment.

- **Proactive**, because it does not simply act in response to its environment but is able to exhibit goal-directed behavior by taking initiative.

- **Mobile**, with the ability to travel between different nodes in a computer network.

- **Truthful**, providing the certainty that it will not deliberately communicate false information.

- **Benevolent**, always trying to perform what is asked of it.

- **Rational**, always acting in order to achieve its goals and never to prevent its goals from being achieved, and

- **Learning**, adapting itself to fit its environment and to the desires of its users [16].

The points autonomous and social are an important part of this project. Only with functioning communication can an agent behave socially and interact with other agents in a
system. Each of the agents in the laboratory for this project is independent. Nevertheless, all agents should be able to communicate with each other, while preserving maximum flexibility. When creating the software, the principle of the Minimum Spanning Tree was used for the communication behavior. This idea will be explained in chapter 3.4.1, in its theory to understand the steps for a successful connection. It becomes clear that any agent can communicate with every other agent. As long as no communication occurs between the agents all stay in a similar status. There is no agent with a higher ranking, like a parent or master agent with special properties.

2.3.1 Agent Communication

An agent is capable of acting in an environment, which means that the agent changes its environment with its actions. For example a powerful solar generator, by altering its production changes the set points of other units, and changes the voltage level in the adjacent buses, and in a more global point of view changes the security level of the system. It affects the stability of the system in case of a short circuit.

Agents communicate with each other and this is part of their capability for acting in the environment. We consider a system that includes a wind generator and a battery system: the battery system receives some power from the wind turbine to charge and to provide it back to the system in time with no wind. In order to achieve this operation, the two agents have to exchange several messages. This is a type of action because the environment is altered in a different way than if the two agents were acting without any kind of coordination.

Agents have a certain level of autonomy, which means that they can take decisions without a central controller or commander and to achieve that, they are driven by a set of tendencies. For a battery system a tendency could be: charge the batteries when the
price for the kWh is low and the state of charge is low, too. The system decides when to start charging based on its own rules and goals and not by an external command. In addition, the autonomy of every agent is related to the resources that it possesses and uses. These resources could be the available fuel for a diesel generator, the bandwidth in the communication channel or the processor time. Another significant characteristic of agents is that they have at best partial information about the environment. For example in a power system the agent of a generator knows only the voltage level at its own bus and maybe it can estimate what is happening in some certain buses, but it does not know what is happening in the whole system. This is the core of the MAS theory, since the goal is to control a very complicated system with minimum data exchange and minimum computational demands. Finally, another significant characteristic of an agent is that it has a certain behavior and tends to satisfy certain objectives using its resources, skills and services. [17].

2.4 Hardware

The hardware available in ERISE laboratory which is used in the project is discussed in this chapter. The agent consists of special software interacting with a hardware structure. For every environment software and hardware look different and are specialized for the particular system. Even in the energy sector there is no final solution or existing hardware framework for an agent. Our focus was on interfacing the agent to simulated power system for the information flow and to test different algorithms for power flow control. For this reason we selected commercial computer components to build up a testbed. In addition, a Real Time Power System Simulator (RTDS) is available in the laboratory. The establishment of the agents and their hardware and software specifications are briefly discussed in the
further sections. This project mainly focuses on the method of interfacing the agents to the
simulated power system [32].

2.4.1 Raspberry Pi

The Raspberry Pi is a small sized single board computer, which was developed by the Rasp-
berry Pi Foundation. The board contains essentially a single-chip system with a Broadcom
BCM 2835 700 MHz processor ARM1176JZF-S. Memory cards (SD or MMC) can be used
as a non-volatile memory or external hard drives and USB flash drives via the USB port.
The recommended Linux distribution is called Raspbian and is based on Debian. A special
Ubuntu distribution is not available, because Ubuntu only supports the ARMv7 architec-
ture [33]. Ubuntu, Raspbian offers a graphical user interface (GUI). Every further explained
configuration step could be realized on both operating systems.

Figure 2.1 Raspberry Pi Model B+ 512MB RAM, 2-USB-Ports und Etherne
(www.vesalia.de)

2.4.1.1 Programmable interface

The Raspberry Pi provides a programmable interface (also known as GPIO: General Purpose
Input / Output), LEDs, sensors, and displays. There are six GPIO ports, but generally only
the connection P1 is used. The GPIO interface P1 consists of 26 pins. Of these pins 17 are
free programmable, which offers a lot of special functions:
• 5 pins can be used as SPI - interface.

• 2 pins have 1.8 kΩ pull-up resistors and can be used as I²C - interface.

• 2 pins can be configured and used as UART - interface.

To control the GPIOs several libraries for many programming languages exist, and even a control terminal through a web interface is possible. In comparison to commercially available computers, easy connections with additional hardware, such as sensors and power electronics would be possible. In addition, this could be a simple way to realize a link to a RTDS. But, at this time no further statement can be made about the speed and the performance of such a configuration.

2.4.2 Real-Time Digital Simulator: RTDS

For studies and research based on power system, it would be recommended to simulate and run the system in real-time. Software such as MATLAB, Power World Simulator, ETAP, PSCAD, etc. can simulate a power system but cannot run in real time speed. For this reason RTDS technologies came up with a solution, and that is the Real Time Digital Simulator (RTDS). Since the speed is in real-time the RTDS is not only used for simulating a power system or a fully functional electrical grid but also is used for system failure scenarios such as control system cyber intrusion or a physical damage event such as a natural disaster [35]. RTDS consists of a hardware unit and a software unit. The RTDS in ERISE laboratory, Michigan State University is given below in Fig. 2.2.

• The Software Unit:

The RSCAD provides the user interface for the system. The system can be simulated using the RSCAD. It is similar to PSCAD. The PSCAD was also developed by RTDS
technologies. The advantages of RSCAD over the PSCAD is that RSCAD can be used for simulation for real time systems, as one time step in the simulation is equal to same out of time in the real world. The RSCAD communicates with the hardware unit over the Ethernet card. For the system simulated the Various features provided by RSCAD will be explained in chapter 2.5.2

• The Hardware Unit:

The Giga-Processor Card (GPC) is the processor card used to solve the equations representing the power system and control system components modelled within the RTDS. The RTDS contain three GPC cards and also contains three PC cards.

WIF- The Work Station Interface card has the following functions:

a- Communication between the RTDS rack and the computer workstation running the RSCAD software. Communication is over an Ethernet based LAN.

b- Communication of data between processors over the rack’s backplane is coordinated by the WIF.

c- WIF performs self-tests and runs diagnostics on other cards installed in its rack.
GTAO- The Gigabit Transceiver Analogue Output Card (GTAO) is used to interface analogue signals from the RTDS to external devices. The GTAO card includes twelve, 16 bit analogue output channels with an output range of ± 10 volts. The 16 bit DACs provide a wide dynamic range. A wide dynamic range is often required when providing measured current signals to protection devices. GTDI- The Gigabit Transceiver Digital Input Card (GTDI) is used to interface digital signals from an external device to the RTDS. The GTDI card includes 64 optically isolated digital input channels. The GTDI card has also been designed to include all the functionality of a DITS card. Therefore the GTDI can be used to read critical firing pulses from an external controller [36]. The GTDI card will send the required timing information to the RTDS software. 17 GTFPI- The Gigabit Transceiver Front Panel Interface (GTFPI) card forms the interface between the digital I/O panel and the GT port on the GPC card. The GTFPI also acts as the interface to the GPC card.

2.5 Interfacing Software

All the software that was required and used for building the testbed is discussed in this chapter. Only some parts of the agent configuration were used during this project and hence only those parts are discussed in the agent, section 2.3.1 of this chapter.

2.5.1 Operating System

The operating system in which the agents were built was Linux. In this case we use the Linux distribution *Ubuntu 12.04.LTS (64 bit)*, as well as a continuously developing hardware support. With a special tool, it was converted to a bootable version running on a USB flash drive. The handling of the installation process is straightforward and completed in a short
time. In the meantime we assigned different system names and passwords for every single computer. For convenience, these are designated by PC1, PC2 and so on, to obtain a simple overview. Furthermore, on each machine a shared directory was created in which the relevant data for the test environment could be stored. These folders are shared in the network, which allows the computer to access the desired data from every supported PC.

In order to create a comfortable to use system for the user, some additional tools were installed. Because only two of the five computers are equipped with input and output devices, a special focus was on setting up a remote connection. Moreover, this should assure access from the outside of the university campus. A brief summary of the setup steps further software components can be found in the following section.

2.5.1.1 Remote Desktop Connection

Due to space constraints not every PC installed in the Lab had its own monitor, keyboard and mouse. Hence, the installation of a remote connection client is the only good alternative. Since most users work primarily with Windows operating systems, a solution had to be found which can establish a connection between the two operating systems.

- Creating a static host name

Usually the IP address of a computer is sufficient to establish a connection with it. But since these are connected to a DHCP network, which dynamically allocates these addresses, this could lead to complications. Thus, Michigan State University offers a service that can be used to create static host names. It can be found at “https://tiny.egr.msu.edu”. The registered host name can be used instead of the IP address to connect to another computer.
• **Setup a connection from outside of the University campus**

To connect from outside of the university campus with the testbed, a secure connection to the servers of the university has to be insured. This can be done with a freely available tool called *Putty*. After a few configuration steps a connection to the servers from any terminal can be established. Once this is done, the computer can dial the testbed with the software, ‘remote desktop connection’.

To remotely control Windows machines, some people prefer to use Remote Desktop Protocol as it performs better than VNC (Virtual Network Computing). VNC has this streak of “JPEG” quality and slow behavior, whereas Remote Desktop Protocol is fast and crystal clear. Since Ubuntu 12.10, xRDP does not seem to work with the Ubuntu desktop anymore unless we install and use an alternative desktop manager. For instance, the desktop manager that has been around is XFCE, which is lightweight and fast. Xfce is a lightweight desktop environment for UNIX-like operating systems. It aims to be fast and low on system resources, while still being visually appealing and user friendly [35].

![Remote connection from a windows operating system to Ubuntu. The registered hostname was used to address the required machine.](image)
2.5.2 RSCAD

The RTDS simulator has an advanced and easy to use graphical user interface - RSCAD. RSCAD is comprised of several modules designed to allow the user to perform all of the necessary steps to prepare and run simulations and to analyze simulation output. The power system that includes the various generation units and transmission line models and control components can be simulated using RSCAD. The RSCAD that is available at ERISE laboratory allows the users to create 18 nodes 3-phase system or 54 nodes 1-phase system. RSCAD provides IEEE standardized models for creating the whole power system. The RSCAD allows the user to simulate the power system using draft sheet and the runtime sheet is used for checking the systems response during the run.

2.5.2.1 Draft Sheet

Fig. 2.2c shows the draft sheet of RSCAD which consists of the following components:

- Power system components: This tab provides all the necessary power system components that are necessary to simulate a system. The power systems components tab contains nodes, Machine models, Transmission line models, breaker models, transformer models, series compensation, instrument transformers, source models, load models and valve group and SVC models.

- Generator control: This tab contains forty five IEEE exciter models, seven generic stabilizer models and twenty five IEEE governor/turbine models. These models are used for the control of the generation and this can be interconnected with any of the machine models for the power system.

- Controls: This tab gives various control option starting from a binary switch to signal generators, logical controllers, and I/O components. By using the I/O components
signals can be outputted from or inputted to the *RTDS* using the analog or digital ports.

- Protection and automation: This feature in RSCAD gives various relay operations such as distance, over current, directional, differential, etc. So by simulating faults, all the operation of the relays could be checked. Usually the relay testing is done as Hardware In Loop (HIL). For this testing it may require continuous loop operation. For this purpose a script feature is explained in detail in a later section.

- Small-$dt$: This tab provides various power electronic components that are necessary for simulating networks containing converters. It becomes useful in the power system when solar photovoltaic panels are used as a generating unit.

### 2.5.3 Runtime Sheet

The runtime sheet in RSCAD allows the user to check the system at runtime. It allows the user to change the values of inputs using sliders, provides switches in case of breakers; hence the breaker can be opened or closed any time while this system is running. This also provides meters and plots to check the systems response. It provides the simulation results and contains the script file. The script file can be programmed to run continuous loops of the simulation in the case of relay testing and power system automation.

#### 2.5.3.1 Script File

The script file is basically a programmable file similar to the MATLAB command window. The system operation can be controlled using the script file. The advantage of using a script file is that in case of testing a relay, the protection schemes are changed every time and relay is tested for various values. This operation requires an operator or a user to perform all these
functions where it will be quite tedious and time consuming. Hence in this case the script file is programmed to record all the operation necessary for the system for one time and is allowed to perform testing for various schemes and various values by running in continuous loops. This will make sure no operator assistance is required. This is not only in case of relay testing, this script file was the main interfacing method in this project. The details of the programming of the script file will be discussed in chapter 4.
Chapter 3

Distributed Power Balance Control and Voltage and Frequency Regulation

Traditional energy supply grids are currently subject to a major transformation. The integration of renewable energy sources, and the liberalization of the energy markets, therefore, require elementary operations in the still predominantly classical structure of the existing power grids in most countries, the conversion and partial automation of the middle and low-voltage distribution grid is a major challenge. It is planned to produce up to 80% of the energy on the basis of renewable energy sources in the future [13].

The future of energy is volatile: it not only involves high performance power sources (e.g. wind, solar), but also model consumers, such as heat pumps, air conditioners, electric vehicles. To maintain the required guidelines for voltage stability and frequency stability, it is essential to align production and consumption together. However; due to the volatile, generation capacity requires a grid, which has a level of automation that allows to observe the technical regulations. Therefore an artificial intelligence is required, which mostly autonomously performs the required processes. To realize this, there is already a wide variety of ideas and approaches. In this paper the possibility of using Multi-Agent systems will be
introduced. Reference [21, 22] share similar shortcomings as chapter 3. To overcome the mentioned drawbacks, this chapter shows a distributed Multi-Agent based power flow algorithm, in which each agent has its local power flow equation and updates state information simultaneously with limited data from their immediate neighbors. Reference [38] shows a distributed power flow algorithm for Multi-Agent platform; However, its algorithm updates state information sequentially from one agent to another, which limits its speed, especially in Multi-Agent framework, where communication speed rather than computation speed is the bottleneck that restricts the speed of algorithm. The distributed power flow algorithm proposed in this chapter fully makes use of communication time, and updates state information synchronously among agents, which offers considerable speed advantage. Based on the proposed power flow algorithm, this chapter also shows that real and reactive unbalanced power can be calculated at slack node. This net power acquisition is implemented fully distributed; therefore, it provides a possible solution for distributed load shedding or restoration.

General equations for three-phase network devices such as transmission lines, transformers and voltage regulators can be found in [19]. These equations can be combined and converted to complex power equations that represent power balance on every phase at every bus in the network. The average power balance and reactive power balance equations for every phase at every bus can be written as follows:

\[
P_m = |V| \sum_{i=1}^{n} |Y_{mn}| |V_n| \cos(\angle|V_m| - \angle|V_n| - \angle|V_{mn}|) \quad (3.1)
\]

\[
Q_m = |V| \sum_{i=1}^{n} |Y_{mn}| |V_n| \sin(\angle|V_m| - \angle|V_n| - \angle|V_{mn}|) \quad (3.2)
\]

where \( n \) is the total number of buses in the distribution system \( m = 1, 2, \ldots, n \), is the index
used to represent every phase at every bus; $P_m, Q_m$ are the average and reactive power, respectively, injected into a phase at a bus; $|V_m|$ is the phasor representation of the sinusoidal voltage on a phase at a bus; $|V_{mn}|$ is the admittance between one phase and another phase at the same or a different bus; $|\cdot|$ denotes the magnitude of a complex quantity; and $\angle$ denotes the phase angle of a complex quantity.

### 3.1 Distribution Network Model and Power Balance

Incremented renewable energy generation and excluding hydropower, holds 23% percents of the growth in electricity generation from 2009 to 2035 [18]. High penetration of renewable energy presents a challenge to the conventional power system by introducing problems as energy management and control, frequency fluctuation and local voltage variation. Microgrids, as systematic organization of distributed generators (DG), intermediate storages, and loads, show more capacity and control flexibility to accommodate these renewable sources as well as other distributed generators, in the range of a few tens of kW, to significantly improve their reliability and better deliver power close to customer loads.

Here, the implements of parent and child control strategy is showed to organize the interfaced distribution generators (DGs) in the microgrids. A DG with largest capacity serves as a voltage source and provides voltage support for microgrids. Other DGs are controlled as current sources. Since the voltage magnitude is determined by parent DG and microgrid configuration, the real and reactive power outputs of parent DGs are adjusted by controlling their current output.

Hierarchical control structure for microgrids is displayed in Fig. 3.1. The upper de-centralized Multi-Agent layer could realize power balance and economic dispatch control.
Local controllers, in response to the power commands required by MAS, regulate local DGs real and reactive power output. This control strategy benefits from simple realization and reliable operation for interfaced microgrids [29].

![Hierarchical control structure of microgrids](image)

**Figure 3.1 Hierarchical control structure of microgrids**

### 3.2 Frequency Droop Control in a Microgrid

For power systems based on rotating generators, frequency and active power are closely interconnected. A load increase implies that the load torque increases without a corresponding increase in the prime mover torque, which means that the rotational speed, and directly the frequency, decreases. The slowing of frequency with increased load is what a droop control is trying to achieve in a controlled and stable manner. The transmission line is modeled in Fig. 3.2 as an \( RL \) circuit with the voltages at the terminals of the line being held constant.
The equivalent transmission line model changes based on the physical line length. For a short line (less than 50 miles), we only consider the series parameters and ignore the shunt parameters. For medium length lines (greater than 50 miles but less than 120 miles), the capacitance of the line becomes significant enough that it impacts the sending end and receiving end voltages and currents; therefore, it is included as shunt components in the equivalent line model. The equivalent \( \pi \) circuit is generally used when modeling medium length transmission lines. In long lines (greater than 120 miles), the distributed effects of the parameters become significant, and the line must be represented by the equivalent \( \pi \) circuit. Alternatively, the line may be represented by smaller cascaded sections, where each section is represented by an equivalent \( \pi \) circuit, similar to the one used for a medium length line. Here, for ease and to show how the power angle depends on the real power and the voltage difference depends on the reactive power, the shunt branches did not consider [19].

\[
\begin{align*}
\frac{I}{\angle \phi} & \rightarrow \bar{Z} \angle \theta \rightarrow + \\
\bar{V}_i \angle 0 & \quad \bar{V}_2 \angle \delta
\end{align*}
\]

Figure 3.2 Power Flowing through a Line

The power flowing into a power line at the terminal is described by the equation

\[
\mathcal{S} = P + jQ = \bar{V} \cdot T^* = V_1 \left( \frac{V_1 - V_2}{Z} \right)^* = V_1 \frac{V_1^2}{Z} e^{j\theta} - \frac{V_1 V_2}{Z} e^{j(\theta + \delta)}
\]

(3.3)

Using Euler’s formula to separate the total power into real and imaginary components gives
the real and reactive power flowing through the line to be

\[ P = \frac{V_1^2}{Z} \cos \theta - \frac{V_1 V_2}{Z} \cos (\theta + \delta) \]  

(3.4)

\[ Q = \frac{V_1^2}{Z} \sin \theta - \frac{V_1 V_2}{Z} \sin (\theta + \delta) \]  

(3.5)

Further defining the line impedance to be \( Z e^{j\theta} = R + jX \), the equations can be written as

\[ P = \frac{V_1}{X^2 + R^2} [R V_1 - V_2 \cos \delta + V_2 X \sin \delta] \]  

(3.6)

\[ Q = \frac{V_1}{X^2 + R^2} [X V_1 - V_2 \cos \delta - RV_2 X \sin \delta] \]  

(3.7)

Typical transmission lines are modeled with the inductance being much greater than the resistance so the resistance is commonly neglected. The equations can then be written as the well known equations

\[ P = \frac{V_1 V_2}{X} \sin \delta \]  

(3.8)

\[ Q = \frac{V_1^2}{X} - \frac{V_1 V_2}{X} \cos \delta \]  

(3.9)

If the power angle \( \delta \) is small, then the small angle formula can be used so that \( \sin \delta = \delta \) and \( \cos \delta = 1 \). Simplifying and rewriting the equations gives

\[ \delta \approx \frac{XP}{V_1 V_2} \]  

(3.10)

\[ V_1 - V_2 \approx \frac{XQ}{V_1} \]  

(3.11)

Equations 3.10 - 3.11 show that the power angle depends heavily on the real power and the voltage difference depends on the reactive power. Stated differently, if the real power can be
controlled, then so can the power angle, and if the reactive power can be regulated, then the voltage $V_1$ will be controllable as well. In the droop method, each unit uses the frequency, instead of the power angle or phase angle, to control the active power flows since the units do not know the initial phase values of the other units in the standalone system. By regulating the real and reactive power flows through a power system, the voltage and frequency can be determined. This observation leads to the common droop control equations

$$f = f_0 - k_p P - P_0 \quad (3.12)$$

$$V_1 = V_0 - k_v Q - Q_0 \quad (3.13)$$

where $f_0$ and $V_0$ are the base frequency and voltage respectively, and $P_0$ and $Q_0$ are the temporary set points for the real and reactive power of the machine. The typical droop control characteristic plots are shown in Fig. 3.3. From the droop equations and highlighted

![Droop Control Characteristic Plots](image)

**Figure 3.3 Droop Control Characteristic Plots**

by Fig. 3.3, as the real power load on the system increases, the droop control scheme will allow the system frequency to decrease. In the droop control, it should be noted that the droop method has the inherent trade-off between the active power sharing and the frequency
accuracy, resulting in the frequency deviating from the nominal frequency [27]. If an active power controller is built to include a frequency restoration loop, the controller is analogous to an engine governor. Engines are equipped with governors to limit the engine to a maximum safe speed when unloaded and to maintain a relatively constant speed despite changes in loading. As the load varies, the speed may droop but over a period of time will return to its nominal speed.

3.3 Voltage Regulated Distribution Transformer (VRDT)

The integration of mechanisms for automating the distribution network presents a particular challenge because they have to be integrated into existing structures. The use of a regulated transformer in cooperation with a Multi-Agent system is, therefore, a useful solution to maintain the voltage stability in a grid. With the local Voltage Regulated Distribution Transformer VRDT a larger voltage bandwidth can be utilized. It adjusts the secondary voltage of a system in several steps. These stages can for example be amounted to 2.5% of the secondary voltage. With 9 steps, up to -10% of the voltage difference can be compensated. For the regulation of the voltage range and to increase the supply of power, connecting a MAS opens completely new possibilities [28]. Several agents can take the voltage measurement across the grid and induce a control intervention of the agents installed on VRDT in excess of the prescribed limits, so the correction algorithm steps control are shown later in this chapter.

In Fig. 3.4 a possible example grid is presented. All agents monitor the voltage levels on their nodes. Agent 6 is connected to a Voltage Regulated Distribution Transformer
(VRDT) and can influence the secondary voltage. The agents 1, 3 and 4 are connected to the power electronic of different renewable generators. They can adjust the real and reactive power of the generators. If the MAS detects a critical power state, then a recommendation for correction of the system state can be calculated. The correction is performed by an algorithm which uses a three-step control method in force: At first the VRDTs and other voltage regulators, like series regulators, are involved. In the second stage, the actuators of renewable generation systems are controlled for targeted control of reactive power. For this, the power factor $\cos \varphi$ of the plant could be adjusted, which has an effect on the local voltage and can influence the result. The third stage should be the shortest period possible. As a last resort the real power could be reduced, however, for economic reasons it is not advisable [20]. During disturbances, the generation and corresponding loads can

![Diagram of a grid and the position of agents](image)

**Figure 3.4 Example for a grid and the position of agents**

...autonomously disconnect from the distribution system to isolate the load of the microgrid...
from the disturbance without damaging the integrity of the transmission grid. This mode is called islanding mode. From the point of view of the customer, it can be seen as a low voltage distribution service with additional features like an increase in local reliability, the improvement of voltage and power quality, the reduction of emissions, a decrease in the cost of energy supply, etc. A microgrid usually connected to the distribution network through a single Point of Common Coupling (PCC) and appears as a single unit to the power transmission network. Power electronics will be a crucial feature for microgrids since most of the power micro-sources must be electronically controlled to gain the characteristics required of the system. The microgrid is therefore not only a more or less autonomous part of the power system, but also has to be a smart system itself. It has to be able to cope with multiple issues, as described below.

As mentioned before, in the last years, the term smart grid has become widespread in the (renewable) energy sector. The introduction of smart grids involves a change from manual operations towards an intelligent, information and communications technology (ICT)-based and controlled network. These changes will especially affect the distribution grid, and in this way, microgrids [23]. Because of the challenges facing the implementations of microgrids, they have to be seen implicitly as smart grids.

1) Microgrid architecture: First, the different types of microgrids will be described. In [24], four classes of microgrids are identified in terms of their architecture.

- Single facility microgrids: These microgrids include installations such as industrial and commercial, residential buildings and hospitals, with loads typically under 2 MW.
- Multiple facility microgrids: This category includes microgrids spanning multiple buildings or structures, with loads typically ranging between 2 and 5 MW. Exam-
Examples include campuses (medical, academic, municipal, etc), military bases, industrial and commercial complexes and building residential developments.

- Feeder microgrids: The feeder microgrid manages the generation and/or load of all entities within a distribution feeder, which can encompass 5-10 MW. These microgrids may incorporate smaller microgrids single or multiple facility within them.

- Substation microgrids: The substation microgrid manages the generation and/or load of all entities connected to a distribution substation, which can encompass 5-10+ MW. Fig. 3.5 shows a Microgrid system with it feeders. The power of each feeder which flows from microgrid into PCC has relation to the outputs of DG and features of loads in microgrid. For example, the output of PV (photovoltaic generation) is affected by sunshine intensity and seasonal variation. Therefore, it is a function of time and in direct proportion with sunshine intensity. Whereas, the output of wind power generation changes over time due to the randomness of the wind.

2) Microgrid control: Concerning the control of microgrids, direct control over local distribution and transmission of electricity from the interconnect point across a facility is a very common practice. Electricity grids are evolving towards intelligent, true complex computer systems, with flexible, controlled power flows supported by advanced information technology [25, 26]. Under the new smart grid paradigm that emerged in the last decade, new capabilities for measurement, control and communication, with a two-way flow of energy and information between customer and supplier, are needed, in order to increase efficiency and achieve cleaner electricity generation.

Some of the new operational measurement and control capabilities that are needed to manage smart grids are:
1) network-based communication and control 2) monitoring of energy generation and consumption 3) optimization of production and consumption 4) generation and load dispatching 5) energy production and consumption control 6) tie line control 7) microgrid active and reactive power control 8) microgrid connection and islanding control 9) security standards 10) user-based billing.

Some of the key technical issues not entirely solved at present are:

1) real-time power flow balancing 2) voltage control and security during disconnection (islanding) from the PCC 3) failure protocol response protection and stability aspects 4) dynamic short- and long-term response 5) active and reactive power control 6) communication technology adaptation 7) tie line control.
3.3.1 Cause of Different Voltage Potentials

Electric potential is a location-dependent quantity that expresses the amount of potential energy at a specified location. This relation is shown in Fig. 3.6, the complete representation of a line.

![Figure 3.6 Complete representation of a line](image)

$I_m$ is the current flowing through the supply end.

$I_n$ is the current flowing through the receiving end of the circuit.

$I_{qm}$ and $I_{qn}$ are the values of currents flowing through the admittances.

By applying KVL to the circuit, we get

$$V_m = V_n + Z_{mn}.I_{qn} + Z_{mn}.(-I_n)$$  \hspace{1cm} (3.14)

$$(-I_n).Z_{mn} = V_m - V_n - Z_{mn}.I_{qn}$$  \hspace{1cm} (3.15)

$$-I_n = \left( \frac{V_m - V_n - Z_{mn}.I_{qn}}{Z_{mn}} \right)$$  \hspace{1cm} (3.16)

$$-I_n = \left( \frac{V_m - V_n}{Z_{mn}} - I_{qn} \right)$$  \hspace{1cm} (3.17)
\[ I_n = \left( \frac{-V_m + V_n}{Z_{mn}} + I_{qn} \right) \]  

(3.18)

Hence for 3-phase the equation 3.18 will be:

\[ I_n = \frac{S^*_n}{3V^*_n} \]  

(3.19)

\[ \Rightarrow S_n = 3V_n I^*_n \]  

(3.20)

\[ \Rightarrow V_m = V_n + Z_{mn}I_{qn} + \frac{Z_{mn}}{3V^*_n}(-P_n) - j\frac{Z_{mn}}{3V^*_n}(-Q_n) \]  

(3.21)

Equation (3.21) is the result of (3.14) through (3.20).

We can see that for real and reactive power changes, it has an influence on the voltage. When feeding reactive or real power, this means that the voltage increases. This effect occurs everywhere in the grid, where we have a power flow. This means, that the consumers and producers in an grid have an effect on the voltages at the several nodes in the grid. Though, critical states can appear. Thus, in Fig. 3.6, it can be seen that the transmission system in addition to the resistive power losses also affects the power factor, which causes influences to the voltage. A distribution grid is made up of several of these equivalent circuits. Between each node in the network you can find an constellation as shown in Fig. 3.6. Therefore, the respective voltage varies from node to node. This can appear especially at nodes to which e.g. large amounts of energy are fed by a photovoltaic system, leading to excessive stresses. These critical points need to be monitored, for which an agent can be used. also, we know a local power flow equation for agent m is (3.22)

\[ S_m = V_m I^*_m = V_m \left[ \sum_{n=1}^{N} Y_{mn}V_n \right]^* \]  

(3.22)
where $S_m$ is the complex power input to node $m$; $V_m$ is the voltage vector at node $m$; $I_m$ is the current input into node $m$. $Y_{mn}$ is built following principles of section 3.3.1. Rewrite this equation:

$$V_m = \frac{1}{Y_{mm}} \left[ \frac{S_m^*}{V_m^*} - \sum_{n=1}^{N} Y_{mn} V_n \right]$$  \hspace{1cm} (3.23)

where, $m = 2, 3, ..., N$

From (3.23), $Y_{mn} = 0$, if agent $n$ is not a neighbor of agent $m$. Then we obtain:

$$V_m = \frac{1}{Y_{mm}} \left[ \frac{S_m^*}{V_m^*} - \sum_{n\in N(m)} Y_{mn} V_n \right]$$  \hspace{1cm} (3.24)

where, $N(m)$ is a set of agents that are neighbors of agent $m$.

To solve power flow, Gauss method is adopted. Differs from traditional Gauss-based power flow, in this Multi-Agent framework, each agent solves its own nodal power flow locally by exchanging the required data with its neighbors. Writing the local power flow equation using the iterative form of Gauss method, we get:

$$V_m^{(k+1)} = \frac{1}{Y_{mm}} \left[ \frac{S_m^{*(k)}}{V_m^{*(k)}} - \sum_{n\in N(m)} Y_{mn} V_n^{(k)} \right]$$  \hspace{1cm} (3.25)

where, $k$ is the iteration number.

The local power flow equation (3.25) shows that each agent only needs to exchange with its neighbors the voltage values from the last iteration in order to update voltage data in its local calculation. It does not need system global information or information beyond its neighbors to calculate its local power flow.

Now, if node $m$ is a PV node, then it first has to estimate $Q_m$: 

36
\[ Q_{m}^{(k+1)} = \text{Im} \left[ V_{m}^{(k)} \sum_{n \in N(m)} \left( Y_{mn} V_{n}^{(k)} \right)^* + V_{m}^{(k)} \left( Y_{mm} V_{m}^{(k)} \right)^* \right] \] (3.26)

where \( Q_{m}^{(k+1)} \) is the reactive power needed to maintain pre-specific voltage magnitude in the \((k + 1)^{th}\) iteration. Then it updates the voltage:

\[
V_{m}^{(k+1)'} = \frac{1}{Y_{mm}} \left[ \frac{P_{m} - jQ_{m}^{(k+1)}}{V_{m}^{(k)}'} - \sum_{n \in N(m)} Y_{mn} V_{n}^{(k)} \right] \] (3.27)

\[ V_{m}^{(k+1)} = V_{m}^{(k+1)'} \left[ \begin{array}{c} V_{m} \\ V_{m}^{(k+1)'} \end{array} \right] \] (3.28)

If \( Q_{m}^{(k+1)} \geq Q_{m}^{\text{max}} \), then \( Q_{m}^{(k+1)} = Q_{m}^{\text{max}} \). If \( Q_{m}^{(k+1)} \leq Q_{m}^{\text{min}} \), then \( Q_{m}^{(k+1)} = Q_{m}^{\text{min}} \).

This means the generator does not have enough reactive power capacity to further maintain voltage magnitude. So the node changes to a PQ node, and voltage magnitude will vary. Voltage update is now updated by (3.29).

\[
V_{m}^{(k+1)} = \frac{1}{Y_{mm}} \left[ \frac{P_{m} - jQ_{m}^{(k+1)}}{V_{m}^{(k)}'} - \sum_{n \in N(m)} Y_{mn} V_{n}^{(k)} \right] \] (3.29)

- Distributed Construction of \( Y_{bus} \)

Conventionally, to perform power flow, we need to first build the bus admittance matrix, \( Y_{bus} \), which contains system structure and admittance information. In Multi-Agent based power flow, we still need system information for power flow calculation. However, this approach differs from traditional centrally computational method, in this distributed based framework, \( Y_{bus} \) is constructed locally. Each agent only obtain part of \( Y_{bus} \) information. The construction principle is shown below:
For agent $i$, it constructs $i^{th}$ row of $Y_{bus}$ matrix:

$$Y_i = [Y_{i1}Y_{i2}...Y_{iN}]$$  \hspace{1cm} (3.30)

$$Y_{ii} = \sum_{n=1}^{N} y_{in}$$  \hspace{1cm} (3.31)

$$Y_{in} = -y_{in} \text{ when } (n \neq i)$$  \hspace{1cm} (3.32)

where, $Y_i$ is the system parameters constructed by agent $i$ for power flow calculation; $Y_{ii}$ is the $i^{th}$ element of $Y_i$; $Y_{in}$ is the $n^{th}$ element of $Y_i$, when $n \neq i$. $y_{ii}$ represents equivalent admittance between bus $i$ and ground. $y_{in}$ when $n \neq i$, is equivalent admittance between bus $i$ and bus $n$.

Recall that in $Y_{bus}$ matrix, if bus $i$ and bus $n$ are not connected, $Y_{in} = -y_{in} = 0$. Since neighborhood is determined by electrical connection, it also means if agent $i$ and agent $n$ are not neighbors, then $Y_{in} = 0$, expressed in (3.33).

$$Y_{in} = 0 \text{ if } (n \notin N(i))$$  \hspace{1cm} (3.33)

where, $N(i)$ is a set of agents that are neighbors of agent $i$.

Construction principles based on equation (3.30)-(3.33) shows that all the information required to build system parameters by agent $i$ are $y_{in}$, where $n = 1, 2, \ldots, N$. Agent $i$ can obtain these parameters locally without global information.
3.4 Steps of Voltage Regulation

As described in 3.3.1, there are three levels of escalation, to regulate the voltage on the network. Each agent knows its own local configuration, thus it can make a contribution to one of the steps explained below. This is called a code word, which is transmitted through the information channel to each agent:

Step 1: VC (voltage control)
Step 2: QC (reactive power control)
Step 3: RC (real power control)

For example, the agent connected to the VRDT can be used for voltage control (VC).

Figure 3.7 Process describing the development of the vector $V_{diff}$

39
Another agent connected to an inverted rectifier of a photovoltaic system could influence the power electronic of the inverter and adjust the real and reactive power (RC, QC). In this case, agent 1, which has detected the critical voltage, first calculates the required voltage adjustment: With a VRDT, which can adjust the secondary voltage -10%, the secondary voltage of the transformer can be regulated as follows:

\[ V_{VRDT,\text{max}} = 1.1 \text{ p.u} \]
\[ V_{VRDT,\text{min}} = 0.9 \text{ p.u} \]

\( V_{corr} \) is the difference between the highest and the lowest value of a node in the grid. The value is transmitted with a command to adjust the voltage. Depending on the voltage at the node of the starting agent, which has detected the critical voltage, it can be positive or negative. If the voltage is too high, \( V_{corr} < 0 \), otherwise \( V_{corr} > 0 \).

If \( V_{agent} > 1 \) then \( V_{corr} = V_{diff,\text{min}} - V_{diff,\text{max}} \)

else \( V_{corr} = V_{diff,\text{max}} - V_{diff,\text{min}} \)

---

**Figure 3.8** Example of the information flow, which transports the highest and lowest voltage of a node to the first agent.
A- Step 1: Voltage Control

Now agent 1 sends the command $VC[V_{corr}]$ to the other child agents. They send it to their child agents as shown in the information flow. This command is now processed by the agents that can make a voltage control over their connected actuator. This is generally only the agent which is coupled to the VRDT. This one processes the request. Depending on $VC[V_{corr}]$ the voltage will be changed. Then the process begins again to detect the voltage level in the network. Agent 1 receives a new view of the situation.

B- Step 2: Reactive Power Control

Agent 1 can now recalculate $V_{corr}$ based on the new information. After this, it sends the $QC[V_{corr}]$ command to all child agents, which in turn pass it on. All child-agents, which are able to influence reactive power, do so now as part of their local conditions. They try to adjust the local voltage by absorption or injection of reactive power. For this, they consider the value $V_{corr}$ and possibilities of their own system. Then the process begins again to detect the voltage level in the network. Agent 1 receives a new view of the situation.

C- Step 3: Real Power Control

Step 3 is analogous to step 2. However, in this case, the call for real power control by agent 1 is started with the command $RC[V_{corr}]$. This is considered as a last resort, so the last possibility which can be used is possible only in exceptional circumstances and for short periods of time. Considering fig. 3.4 with six consumers and three generators for renewable energy, we can make some calculations and do the simulation. These are photovoltaic systems. Each of these systems has the possibility to produce inductive or capacitive reactive power. The power factor of the generators is at most $\cos(\phi) = 0.9$. 
The power values of the different consumers and generators are in Table 3.1.

Parameters for simulation:

\[ R = 0.202 \ \Omega/\text{km} \quad X = 0.085 \ \Omega/\text{km} \]

The used VRDT can adjust the secondary voltage -10%:

\[ V_{VRDT} = 0.9 \ldots ,1.1 \text{p.u} \]

The distances between the several nodes are shown in Fig. 3.9, together with the assumed loads and generator powers. In this case, it could be a small sub-area of a low-voltage network in a rural area. Very often large photovoltaic systems and industrial units can be found on the outskirts of villages. In this case it serves only as an example to present the main idea and to show the influence of each regulation step.

### Regulation Process:

After agent 1 has detected a violation of the permissible voltage, the information flow described in chapter 3.4.1 takes place. In Fig. 3.9 the unregulated voltage is given for every node, before the intervention. To maintain the stability of the grid and to keep the required voltage bandwidth, all of the 3 steps will be performed.
1) Step 1: Voltage Control:

\[ V_{agent1} > 1.2p.u \Rightarrow V_{corr} = V_{diff,\text{min}} - V_{diff,\text{max}} \]

\[ \Rightarrow V_{corr} < -0.2pu \]

In this case the VRDT allows a maximum voltage reduction of 0.1\(pu\).

\[ \Rightarrow V_{VRDT} = 0.9pu \]

The result can be looked up in Fig.3.10 The remaining differences can only be compensated by the steps 2 and 3.

2) Step 2: Reactive Power Control: All of the shown generators can adjust the reactive power. After the injection of maximum reactive power they can handle, the voltage in the grid is reduced again.

\[ G1 = 73 \text{ kVAR}, \ G2 = 97 \text{ kVAR}, \ G3 = 145 \text{ kVAR}, \]
3) Step 3: Real Power Control: In this case we show a worst case scenario. Thus, the real power of the generator at node 7 is scaled down: $P = 250$ kW

Because of this, the reactive power of this generator decreases a little: $Q = 121$ kVAR

### 3.4.1 Communication Structure:

To explain the theory presented here, it is necessary to first provide some definitions. The agent that is processing system information at a given point of time is called the current agent. If agent 1 transmits information to agent 2, then agent 1 is called agent 2’s parent agent, and agent 2 is called agent 1’s child agent [29].

1) Discovery of a Minimal Spanning Tree:

The agents that were established were developed with a working communication structure for an indefinite number of agents. In this section the Discovery of a Minimal Spanning Tree is explained. This approach formed the theory for the further steps in this project. The developed software fulfills every step which is described in this section.
View, either local or global, is an agent’s knowledge of system information. This information consists of maximum real and reactive power generation capacity $P_G, Q_G$, dispatchable real and reactive power generation capacity $P_{DG}, Q_{DG}$, vital real and reactive load demand $P_v, Q_v$, and non-vital real and reactive load demand $P_{nv}, Q_{nv}$. View is mathematically defined as a vector $u$.

$$u = [u_1, u_2, u_3, u_4, u_5, u_6, u_7, u_8] = [P_G, Q_G, P_{SG}, Q_{SG}, P_v, Q_v, P_{nv}, Q_{nv}] \quad (3.34)$$

This process is intended to organize the decentralized Multi-Agent system and steer the flow of information. In the proposed method, there are three basic requirements that must be met in order to efficiently achieve the discovery of power information in the decentralized micro grid. First, a communication protocol should be defined to conduct information flow in a manner that all the nodes in the system are spanned. Second, this protocol should be designed to route the information flow that every node receives and processes information only once. Finally, real time control of a Multi-Agent system requires this protocol to be able to discover system information in parallel, so that it can quickly react to the disturbance in the system. Toward meeting the above requirements, a minimal spanning tree is constructed first. This process is executed as follows:

(a) A token is generated by a starting agent.

(b) Every agent which receives the token, memorizes its parent agent ID, then it transmits the token to all of its other neighbors, and stores its child agents IDs.

(c) Any agent, who receives multiple tokens simultaneously, keeps one and discards others. At the same time, it removes child-parent relationships with those whose tokens are discarded.
To demonstrate this algorithm, consider the example shown in Fig. 3.11. Since all agents are identical, the starting agent can be selected randomly. In this case, let us simply choose one agent as a starting agent. Initially, agent 1 generates a token and transmits it to agent 2. Then agent 2 receives the token and sends it to its neighbors, agent 3 and agent 4, and agent 6 which will discard this token. After that, agent 3 sends the token to its neighbors, agent 7, then agent 7 will send token to agent 8 and agent 6 which now discard one of the token, agent 8 will send the token to its neighbors agent 9 and agent 10. In parallel, agent 4 will send the token to agent 5, the agent 5 will send the token to agent 6 and agent 10 which will discard any extra tokens and so on. Let us assume it discards the token coming from agent 5, thereby removing a redundancy in information flow. Finally, the token will flow from agent 5 to agent 6. At this point, all of the agents in the system have been discovered. During this process, all agents want to store their

<table>
<thead>
<tr>
<th>Agent</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parent ID</td>
<td>-</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>5</td>
<td>3</td>
<td>7</td>
<td>8</td>
<td>9</td>
</tr>
<tr>
<td>Child ID</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>7</td>
<td>5</td>
<td>6</td>
<td>-</td>
<td>8</td>
<td>9</td>
<td>10</td>
</tr>
</tbody>
</table>
parent agent and child agent IDs. Their relationships are shown in Table 3.2. Fig. 3.11 (a) and (b) depict the transmission path of the token and the minimal spanning tree established by stage 1 [29, 30].

2) Information and Feedback Process: When information flow path is established in Structure Construction Process, this step is intended to collect system generation and load data. Its algorithm could be stated as follows:

(a) Agents who have no other neighbors or receive all refuse information from other neighbors, respond their parent agents with local view of the system.

(b) Agents who receive system view information from all their child agents, process these data based on (3.35) and transmit updated view information to their parent agents.

\[ S_t = S_{t \text{local}} + \sum_{i \in \phi} S_{i}^{t}, \quad t \in \{G, DG, v, nv\} \]  \hspace{1cm} (3.35)

where \( S_t \) is current agent’s updated view; \( S_{t \text{local}} \) is current agent’s local view; \( \phi \) is the set of child IDs; \( S_{i}^{t} \) is child agents’ updated view.

Information Feedback Process ensures that each local node view information flows back from child agent to its parent agent. At last, the starting agent could obtain system global view and calculates net power based on (3.36). Fig. 3.11 (c) shows the information flow path of Information Feedback Process.

\[ S_{\text{net}} = S_G + S_{DG} - S_v - S_{nv} \]  \hspace{1cm} (3.36)

where \( S_{\text{net}} \) is system net power; \( S_G, S_{DG}, S_v, S_{nv} \) are elements of starting agent’s
updated view. Once the information flow path is established in stage 1, the next step is designed to collect the different system voltages. If no other agents are discovered, all child agents send the following information to their parent agents. The direction of the information flow is shown in Fig. 3.11 (c)

\[ V_{\text{agent}}: \text{voltage at the node of the agent} \]

\[ V_{\text{diff}} = [V_{\text{max}}, V_{\text{min}}]: \text{vector of the highest and lowest voltage received by unit.} \] Fig. 3.7 describes the development of the vector \( V_{\text{diff}} \). It is shown that the first responding agent sends the vector \( V_{\text{diff}} = [V_{\text{agent}}, V_{\text{agent}}] \), because it did not receive any other information.

Fig. 3.8 shows the 7 agents are listed, which have established a communication structure. Starting from the last two child agents, information about the voltage (in per unit p.u.) is passed to the respective parent agent. Thus, the information about the highest and lowest voltage at the nodes of the agents are delivered to the starting agent 1.

### 3.4.2 Agent Based Power Flow

Power Calculation by Gauss Method:

To solve power flow, Gauss method is adopted. Differs from traditional Gauss-based power flow, in this multiagent framework, each agent solves its own nodal power flow locally by exchanging the required data with its neighbors. Assume in each bus of the microgrid, there is a node agent. It has information about local generation, such as real power capacity, reactive power capacity, voltage magnitude that is going to be maintained; meanwhile, it also has local load information, such as real and reactive power demands by the load. If two nodes are connected electrically, then the corresponding agents are marked as neighbors. Agents with neighborhood can communicate with each other; those without neighborhood relationship cannot communicate.
Assume N nodes in the system, with net power input $P_n, Q_n$. If it is a load node, then $P_n$ and $Q_n$ are negative; if it is a generator node, then $P_n$ and $Q_n$ are positive.

Let us select node 1 as slack node. When calculating the power flow, its voltage magnitude and angle are maintained constant. Other nodes use their original real and reactive power inputs (PQ node) or real power input and voltage magnitude requirement (PV node) to calculate voltage magnitude (PQ node), voltage angle (PQ or PV node) or reactive power demand (PV node).

After agent-based Gauss power flow calculation method, each agent could obtain local voltage magnitude and angle, and they satisfy the following equation:

$$\mathbf{P}_m + j\mathbf{Q}_m = \mathbf{P}_m + j\mathbf{Q}_m = V_m \left[ \sum_{n=1}^{N} Y_{mn} V_n \right]^* \tag{3.37}$$

where $\mathbf{P}_m$ and $\mathbf{Q}_m$ are calculated real and reactive power; $P_m$ and $Q_m$ are original planned real and reactive power input.

Also we could calculate real and reactive power input at node 1:

$$\mathbf{P}_1 + j\mathbf{Q}_1 = V_1 \left[ \sum_{n=1}^{N} Y_{1n} V_n \right]^* \tag{3.38}$$

where $\mathbf{P}_1$ and $\mathbf{Q}_1$ are calculated real and reactive power at node 1.

Losses of lines are categorized by relationship with nodes. If a line connects both node $m$ and node $n$, then we define half of that line loss is contributed to node $m$, the other half of that line loss is contributed to node $n$. If a line connects node $m$ and ground, then we say that all of the line loss is contributed to node $m$. Then local losses related to node $m$ can be calculated by equation (3.39).
\[ P_{\text{loss}}^m + jQ_{\text{loss}}^m = y_{mm}V_mV_m^* + \frac{1}{2} \sum_{n=1}^{N} \sum_{n \neq m} y_{mn} [V_mV_n^* + V_nV_n^* - V_mV_n^* - V_nV_m^*] \] (3.39)

where \( P_{\text{loss}}^m \) and \( Q_{\text{loss}}^m \) stand for real and reactive power losses related to node \( m \) respectively.

So the total losses in the system can be expressed as:

\[ P_{\text{loss}} + jQ_{\text{loss}} = \sum_{m=1}^{N} \left[ \sum_{n=1}^{N} Y_{mn}V_mV_n^* \right] \] (3.40)

where \( P_{\text{loss}} \) and \( Q_{\text{loss}} \) are total system real and reactive power losses.

Therefore, total system generation deficiency or surplus can be calculated:

\[ \Delta P + j\Delta Q = P_1 - P_1 + jQ_1 - jQ_1 \] (3.41)

where \( \Delta P \) and \( \Delta Q \) are generation deficiency or surplus for the system, compared with load demands taking line losses into consideration. Note that if \( \Delta Q \geq 0 \), the reactive power is automatically balanced, because voltage regulator at slack bus will control the reactive power output of the generator to balance reactive power. However, if \( \Delta Q \leq 0 \), load dispatch is required to maintain reactive power balance.

From equation (3.41), it shows that the power deficiency considering line losses in the system can be calculated at slack node. Agent at node 1 knows generator capacity or load demand \( P_1 \) and \( Q_1 \), also it can calculate \( P_1 \) and \( Q_1 \) from voltage and angle data in power flow calculation. Then it can obtain power shortage in the system [29].
3.5 Five-Bus Test System

In order to test the Multi-Agent system, a five bus system from [37] is applied and it is tested as follow:

3.5.1 Example Calculation:

Original system configuration with agents connected to is shown in Fig. 3.12. Also, bus input data and line input data are modified and displayed in Table (3.3) and (3.4). All the data have been transformed into per unit.

![Figure 3.12 Tested five-bus system](image)

Table 3.3 Bus Input Data for Five-Bus System

<table>
<thead>
<tr>
<th>Bus</th>
<th>Type</th>
<th>V</th>
<th>δ</th>
<th>(P_G, Q_G)</th>
<th>(P_L, Q_L)</th>
<th>(Q_G^+)</th>
<th>(Q_G^-)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>slack</td>
<td>1.0</td>
<td>0</td>
<td>(2, 1)</td>
<td>(0, 0)</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>2</td>
<td>PQ</td>
<td>-</td>
<td>-</td>
<td>(0, 0)</td>
<td>(5, 2.8)</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>3</td>
<td>PV</td>
<td>1.05</td>
<td>0</td>
<td>(5.4, -)</td>
<td>(3, 0.4)</td>
<td>4.0</td>
<td>-2.8</td>
</tr>
<tr>
<td>4</td>
<td>PQ</td>
<td>-</td>
<td>-</td>
<td>(0, 0)</td>
<td>(3, 1)</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>5</td>
<td>PQ</td>
<td>-</td>
<td>-</td>
<td>(0, 0)</td>
<td>(1.8, 1.0)</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>
Table 3.4 Line Input Data for Five-Bus System

<table>
<thead>
<tr>
<th>From Bus</th>
<th>to Bus</th>
<th>R'</th>
<th>X'</th>
<th>G'</th>
<th>B'</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>5</td>
<td>0.0015</td>
<td>0.02</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>0.009</td>
<td>0.1</td>
<td>0</td>
<td>1.72</td>
</tr>
<tr>
<td>2</td>
<td>5</td>
<td>0.0045</td>
<td>0.05</td>
<td>0</td>
<td>0.88</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>0.00075</td>
<td>0.01</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>0.00225</td>
<td>0.025</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

3.5.1.1 $Y_{bus}$ Construction:

Based on the construction principles expressed as (3.30)-(3.33), first row of $Y_{bus}$ built by agent 1 is:

$$Y_{12} = Y_{13} = Y_{14} = 0$$

$$Y_{15} = -\frac{1}{0.0015 + j0.02} = -3.73 + j49.72 \implies Y_{11} = 3.73 - j49.72$$

All the data needed to compute local $Y_1$ can be obtained locally by Agent $A_1$. Similarly, other agents establish their $Y_i$ vectors. Results of $Y_{bus}$ matrix is shown in Table (3.5).

Table 3.5 Local $Y_i$ Vectors

<table>
<thead>
<tr>
<th>$Y_i$</th>
<th>$A_1$</th>
<th>$A_2$</th>
<th>$A_3$</th>
<th>$A_4$</th>
<th>$A_5$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$3.73 - j49.72$</td>
<td>$0$</td>
<td>$0$</td>
<td>$0 - 3.73 + j49.72$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$0$</td>
<td>$2.68 - j28.46$</td>
<td>$0$</td>
<td>$-0.89 + j9.92$</td>
<td>$-1.79 + j19.84$</td>
</tr>
<tr>
<td></td>
<td>$0$</td>
<td>$0$</td>
<td>$7.46 - j99.44$</td>
<td>$-7.46 + j99.74$</td>
<td>$0$</td>
</tr>
<tr>
<td></td>
<td>$0$</td>
<td>$-0.89 + j9.92$</td>
<td>$-7.46 + j99.44$</td>
<td>$11.92 - j147.96$</td>
<td>$-3.57 + j39.68$</td>
</tr>
<tr>
<td></td>
<td>$-3.73 + j49.72$</td>
<td>$-1.79 + j19.84$</td>
<td>$0$</td>
<td>$-3.57 + j39.68$</td>
<td>$9.09 - j108.58$</td>
</tr>
</tbody>
</table>

3.5.1.2 Multi-Agent Based Power Flow Calculation

Assume flat start, $V_2^{(0)} = 1.0$, $V_3^{(0)} = 1.05$, $V_4^{(0)} = 1.0$, $V_5^{(0)} = 1.0$. The first iteration of Gauss method for Multi-Agent system is as follows:
agent 2:

\[
V_2^{(1)} = \frac{1}{Y_{22}} \left[ \frac{S_2^*(0)}{V_2^*(0)} - \sum_{n \in N(2)} Y_{2n} V_n^{(0)} \right] = 0.947 \angle -10.30^\circ
\]

agent 3:

\[
Q_3^{(1)} = \text{Im} \left[ V_3^{(0)} \sum_{n \in N(3)} \left( Y_{3n} V_n^{(0)} \right)^* + V_3^{(0)} \left( Y_{33} V_3^{(0)} \right)^* \right] = 4.91
\]

\[
Q_3^{(1)} > Q_3^{max} - Q_{3L} = 4.0 - 0.4 = 3.6 \quad \Rightarrow Q_3^{(1)} = 3.6
\]

\[
V_3^{(1)} = \frac{1}{Y_{33}} \left[ \frac{P_3 - jQ_3^{(1)}}{V_3^*(0)} - \sum_{n \in N(3)} Y_{3n} V_n^{(0)} \right] = 1.039 \angle 1.11^\circ
\]

agent 4:

\[
V_4^{(1)} = \frac{1}{Y_{44}} \left[ \frac{S_4^*(0)}{V_4^*(0)} - \sum_{n \in N(4)} Y_{4n} V_n^{(0)} \right] = 1.033 \angle -1.13^\circ
\]

agent 5:

\[
V_5^{(1)} = \frac{1}{Y_{55}} \left[ \frac{S_5^*(0)}{V_5^*(0)} - \sum_{n \in N(5)} Y_{5n} V_n^{(0)} \right] = 0.996 \angle -0.93^\circ
\]

If we define convergence criterion as a voltage tolerance of 0.001 p.u shown in (3.42), it needs 39 iterations to converge. The first five iteration results are given at Table (3.6).

\[
\left| V_i^{(k)} - V_i^{(k+1)} \right| \leq 0.001 \quad \forall i \quad (3.42)
\]

3.5.1.3 Net Power Calculation

After system node voltages are obtained, total netpower can be computed at slack agent:
Table 3.6 Results for the First Five Iteration

<table>
<thead>
<tr>
<th>No.</th>
<th>agent 2: $V_2$</th>
<th>agent 3: $V_3$</th>
<th>agent 4: $V_4$</th>
<th>agent 5: $V_5$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$0.947 \angle -10.30^\circ$</td>
<td>$1.039 \angle 1.11^\circ$</td>
<td>$1.033 \angle -1.13^\circ$</td>
<td>$0.996 \angle -0.93^\circ$</td>
</tr>
<tr>
<td>2</td>
<td>$0.920 \angle -10.81^\circ$</td>
<td>$1.05 \angle 0.13^\circ$</td>
<td>$1.019 \angle -1.22^\circ$</td>
<td>$0.996 \angle -3.14^\circ$</td>
</tr>
<tr>
<td>3</td>
<td>$0.916 \angle -12.79^\circ$</td>
<td>$1.05 \angle -0.06^\circ$</td>
<td>$1.026 \angle -2.48^\circ$</td>
<td>$0.986 \angle -3.21^\circ$</td>
</tr>
<tr>
<td>4</td>
<td>$0.905 \angle -13.18^\circ$</td>
<td>$1.05 \angle -1.28^\circ$</td>
<td>$1.022 \angle -2.74^\circ$</td>
<td>$0.987 \angle -4.02^\circ$</td>
</tr>
<tr>
<td>5</td>
<td>$0.904 \angle -13.97^\circ$</td>
<td>$1.05 \angle -1.56^\circ$</td>
<td>$1.022 \angle -3.81^\circ$</td>
<td>$0.983 \angle -4.16^\circ$</td>
</tr>
</tbody>
</table>

agent 1:

$$P_1 + jQ_1 = V_1 \left[ \sum_{n=1}^{N} Y_{1n} V_n \right]^* = 7.34 + j1.36$$

$$P_s + jQ_s = P_1 - P_1 + Q_1 - Q_1 = -5.34 - j0.36$$

Therefore, considering generators and transmission network characteristics (PV, PQ, power flow), line losses, the deficiency of generation in the system is $-5.34 - j0.36$. If simply neglecting these factors, netpower is calculated as:

$$P_s + jQ_s = \sum (P_G - P_L) + j(\sum Q_G - Q_L) = -5.4 - j0.2$$

Because of the voltage magnitude requirements at node 3, generators at node 3 will not simply provide maximum reactive power, even though reactive power in the system is deficient. Therefore, neglecting system operation principles and generation characters, the net reactive power estimation are not accurate. In this case, its error is above 44%. Real power difference between this two methods mainly resulted from system losses. In microgrids, with low voltage level and mainly resistant network, this loss sometimes could not be omitted.
Chapter 4

Interfacing Methods

In a microgrid, integrating and interfacing sensing and control devices is challenging because it involves different communication protocols such as Standard number (RS232) serial communication and (RS422/485) modbus communication. To overcome this challenge, we need to convert the information into one standard protocol - Ethernet. We can economically made the conversion by using a communication protocol converter.

4.1 Method 1

Since RTDS and the agents had a different communication speed. An idea of using the same speed device like PC for simulating a power system was dealt. Hence a plan without using the RTDS was considered. Here in this method we decided to program the MPI (Message Passing Interface) cluster, i.e. the five PCs in such a way that they could act as the power system. The MPI was to be programmed using C. The plan was to make each PC to do the functions of a generating machine, a transformer and a load. So when they are interconnected there would be five generating machines, transformers and loads. This would be a primitive power system model. The interfacing could be easy because the socket connection technique can be used and can be programmed as the agents are built in java platform. But this method was soon dropped as the modeling of the machines and transformers using a C coding became tedious. Also the time constrain was an issue with this method. Moreover
the main idea of this project is to create a real time test bed for the control of the micro grid, but this method would give a system which would not work at real time system.

Figure 4.1 a) A fibre optic cable connects the RTDS internal communication system to an FPGA board. The board converts the data and builds an interface with the Ethernet network; b) With the help of an AD/DA converter the outputs of several channels can be converted in digital values and send to a computer or directly in a Ethernet network.

4.2 Method 2

This method was developed based on the interfacing technique adopted by Florida State University. This method used the RTDS and the signals have been taken out from the output ports either (analog or digital) through an optic fiber cable to a Field Programmable Gate Array 27 (FPGA) which acts as the interfacing medium between the RTDS and the agents. This method would suit for quite large system where there are many GTAO or GTDO. As already mentioned due to hardware constrain where only one GTAO port is available...
and so the signals had to be sent sequentially. The agents were neither programmed to take in signals sequentially nor to give signals sequentially. Hence this method could not be implemented. Fig. 4.1 shows the diagrammatic representation of the interface. But this method paved a way to find a better interface with the agents. While working with sending signals sequentially, RSCAD’s script feature proved handy. While working with the script file, a method of interfacing the agents with RTDS using no extra hardware was found. The connection was over the Ethernet itself. This is explained in detail in the next section of this chapter.

4.3 Method 3

As said in previous section this method does not require any special or external hardware and interfacing could be done using the Ethernet over the LAN. The script file in RSCAD was programmed for interfacing in this method. The steps are as follows:

- The power system is simulated using the RSCAD software. Two systems which are shown in Fig. 4.4 and Fig. 4.6. was simulated in which the only for system shown in Fig. 4.4 an interface was done. This system runs in real time on the RTDS. The simulated system results are shown in the Fig. 4.7.

- RTDS imitates or mimics the characteristics of a real power system and hence when a controller or agent is interfaced to it, it provides a real-time working environment.

- The runtime sheet of the RSCAD allows a feature called script. The explanation about the script was given in section 2.5.3.1. The script was programmed to convert the values of the system like voltage, frequency, real power generated, reactive power
generated and vital and non-vital power demand to a comma separated value (.csv) file. The algorithm and programming of the script will be explained in detail in the next chapter.

- The agents programmed to accept the comma separated value file as the input.

- This file generated from the RSCAD is the input to the agents.

- Using file sharing protocols over the LAN, the file is sent to agents.

- Since the RSCAD was on the windows platform and the agents were on the Linux platform, applications like Samba, SMB4K were installed on the Linux to allow the file sharing. The remote desktop connection was used for accessing the PCs and the agents for programming and configuring the sharing protocol through the windows system.

- RSCAD for Linux was available only for 32 bit system and all the updates for the Linux based RSCAD was supported by the RTDS technologies and so the file sharing protocol between the windows and Linux was a necessary.

- In ideal case the agents would perform the required task as per their program, for example if the agents are programmed to do the power balance and economic dispatch. Then from the moment they get the input signals the agents would start the process of finding the power balance and economic dispatch for the system and would output new values of power generated and load distribution in different nodes. This file would be sent back to the system. But the agents that were established in the lab had been programmed to communicate among them which were supposed to be the main work of the agents. Hence communication part was given more importance while the agents were built. Therefore in this case they would output the communication path between
the agents and would output the same values of the file that is being inputted to them. The agents will be further programmed for the economic dispatch and power balance which is the future works.

- This output agents that would be sent to the system would modify the system and the new system values will be again checked for the same. This keeps on repeating until economic dispatch and power balance is obtained.

- The advantage of this method is that the real time speed and the automation.

- The slight disadvantage of this system is that the values of the system cannot be changed at the run time. In order to overcome this disadvantage another method is proposed.

- Fig. 4.2 shows the interfacing using method 4.3.
4.4 Method 4

This method overcomes the disadvantage of the previous method. This is a future method and is yet to accomplished. This method uses external hardware i.e. a Data Acquisition Toolbox (DAQ). The steps for this method are given below:

- The power system is simulated using the RSCAD software and runs in real time on the Real Time Digital Simulator RTDS.

- RTDS imitates or mimics the characteristics of a real power system and hence when a controller or agent is interfaced to it, it provides a real-time working environment.

- The values of the voltage, frequency, power generated, power demand of the power system are taken out from the GTAO (analog output port) of the RTDS. As mentioned in chapter 2.4.2 the RTDS in the lab consists of 1 GTAO card. Therefore 12 ports are available. For one node is 8 signals are to be sent out then for a 6 node system it would be 48 signals at a time. Unfortunately since there is only 12 ports available, we have to send signals sequentially. We use 8 ports for every instance.

- The entire analog signals that are outputted from the GTAO card will be properly scaled on a 5V level (the signals are already tested using the oscilloscope and the scaling was using a multimeter).

- (National Instrument) NI’s LabVIEW is considered as versatile data acquisition software. Even MATLAB can also be used. For the LabVIEW to receive in signals we have to use a hardware i.e. the Data Acquisition Toolbox (DAQ) provided by NI.

- Upon receiving these values the LabVIEW can be programmed to do a looping action
and hence these sequential signals sent from the RTDS can be obtained. These signals can also be programmed to be stored as comma separated value (.csv )file.

- This file will again use the same file sharing protocol to share the file to the agents.
- As mentioned already the agents are programmed to accept the .csv file as an input.
- The agents will process the inputs as per their program and output file in .csv format.
- This file will be acquired by the labVIEW using the file sharing protocol. These values will be converted to digital signals in order to be sent to the RTDS.
- The RTDS available in lab has 1 GTDI card. And hence not more than four signals can be received into the system at the same time. Hence there should be an understanding between the RTDS and the LabVIEW in order to receive the signals. Hence after each four values the port is reset. So once the port is reset the RTDS will understand that the next 4 signals would be another set.
- The control action will be programmed in the RSCAD to check if the port is reset or not. Once the signals are received it would be re-scaled to the original values and sent to the respective nodes.
- Thus the new system values will be inputted to the system.
- The advantage of this system is that it runs at real time and the values of the system can be modified during the run time.
- Another advantage of this is the RTDS can be used to perform HIL where the LabVIEW can send the control signal.
• Fig. 4.3 shows the method of interfacing the RTDS to the agents using the DAQ toolbox.

![Figure 4.3 Interfacing using DAQ (Method 4)](image)

4.5 Simulation and Results

This chapter gives the detailed construction of the testbed. As said in previous chapters the interfacing was given the main importance. In this chapter we discuss the algorithm for the method using the script feature of RSCAD as described as method 3 in 4.3.

4.6 Power System Models

RSCAD gives the user a wide variety of option to construct the power system. Two systems were simulated one consisting of three DG and four load units for a six bus systems shown in Fig.4.4 for which the interface was done. And the second system was an advancement of
the first system with 1 wind turbine, one solar PV unit and one DG as generating units and four load units as shown in Fig.4.6. The number of processors available in our lab limits the number of units to 400 and so only one wind turbine could be used in the system. Also the modifications done to the base system came up to eleven nodes hence node selection for sequentially sending the signals was not possible as the limit was only six (excluding Embedded Bus). Hence the second system was just simulated for which the interface was not performed. The idea of using same simulated system that can used for both method three and method four which are described in section 4.3 and section 4.4 respectively was considered for verifying the results. Unfortunately method four could not be completed. Fig. 4.4 and Fig. 4.6 show the one line diagram of the system that was simulated. Fig. 4.5 shows RSCAD’s one line diagram.

Figure 4.4 System consisting three generating and three load units

RSCAD Draft sheet is used for creating the system. The power system components explained in section 2.5.2.1 were used for creating the nodes. The buses one, five, and nine
Figure 4.5 System consisting of three generating and three load units.
Figure 4.6 System with one wind turbine, one solar PV unit, one DG and four load units in the system shown in Fig. 4.4 and buses one, five for the system show in Fig. 4.6 are embedded buses. RSCAD provides four types of buses.

- Explicit buses: buses which participate in the large time step network solution
- Embed buses: observable but do not participate in the large time step network solution
- Small time step buses: RTDS has a small time step solution that can be embedded inside a large time step solution
- FDNE Buses: In addition to the above three categories of electromagnetic transient (emt) buses, we could embed several frequency dependent network equivalents (FDNE) within a network solution.

Since the buses which are embedded busses i.e. these busses forms the default intercon-
necting node between generator and transformer, these busses doesn’t participate in the simulation. The simulation hence is reduced to six node/bus system for the system with three generating and load unit and as nine node/bus for the second system.

### 4.6.1 Runtime

The Runtime sheet of RSCAD provides the response of the system. The components available in this feature are already explained in section 2.5.3. The simulated three generating unit and three load system and its response in runtime is shown in the Fig. 4.7. The system with wind turbine, solar panel and DG with four load and its response during the runtime unit is shown in the Fig. 4.6

![Runtime Sheet](image)

**Figure 4.7** Shows the plots of simulated system consisting of six buses, four load units, with three generating units
The sliders $PG1, PG2, PG3, QG1, QG2, QG3$ are used for real and reactive power generation control. The sliders $PL1, PL2, PL3, PL4, QL1, QL2, QL3, QL4$, are used for adjusting the real and reactive load. Fig. 4.8 shows the sliders. Meters PMACH1, PMACH2, PMACH3, QMACH1, QMACH2, QMACH3 gives the values of real and reactive power generated. Meters bus 1, bus 2, bus 3, bus 4, bus 5, bus 6 gives the RMS values of the nodes/busses. Fig. 4.9 shows the power generated meters and shows the bus voltages meters.

The extra components in this system are the push button for trip and reclose of the wind turbine to the grid and it is shown in 4.9. The other is additional sliders to control the wind and solar generation.

![Figure 4.8 Sliders to adjust the generation and load](image)

The plots the gives the values of the buses. These plots need to be updated manually.

### 4.7 Script feature of RSCAD

This feature is found in the runtime sheet of RSCAD. Fig. 4.10 shows the scrip feature in runtime sheet. This feature of RSCAD was used for interfacing the agents to the RTDS. This
method has been explained in chapter 4.3. Script file is basically used for loop operation and automation. The main idea of using script is to eliminate the presence to the user. Here in this method we are converting the values of the generation, voltage, frequency, demand, into a comma separated value file which is sent to the agents. The algorithm used for programming the script is shown in Fig. 4.11.

This flow chart gives the overview for the coding done in the script. First of all script_out.csv is the output form the agents and script_in.csv is the input given to the agents. Since the agents were established first the whole procedure is based on the agents. And hence script_out.csv is the input to the system while the script_in.csv is the output from the system. For the first time for running the system a dummy script_out.csv is created with values that are necessary to run the system and once the system start running then it will give the output which will got to agents. The agents now will work on the input and give new script_out.csv as the output. Since all these files are in a shared file between the windows and
Figure 4.11 Flow chart for the programming of the script.
Linux the new script_out.csv will overwrite the old one and so this loop continues. However; we still need a device such as DAQ to make the flow of information between our agents and the system which the RTDS, here. Fig. 4.12 shows the output file given by then agents.

![Table Image]

Figure 4.12 shows the script_in.csv outputted by the system which is the input to the agents

![GUI Image]

Figure 4.13 shows the GUI of the agents

![Table Image]

Figure 4.14 the output script_out.csv file given by the agents then this file is sent to RTDS
Chapter 5

5.1 Conclusion

Using a Multi-Agent system for the micro grid control is advantageous because of the parallel communication and equality among all the agents and that means there is no agent which has higher priority than other agents. Before the start of this project the agents were established which were programmed to communicate based on the minimum spanning tree and token transmission technique. The agents are designed to allow the start of any number of agents on any number of computers simultaneously for increasing the usability.

The MPI cluster can be used for any kind of complex mathematical calculation. It is possible to distribute the computing power to five computers, which increases the performance immensely.

In this project, the main idea of interfacing the power system to the agent was performed is described. Various methods were initiated for the interfacing during this project. The agents which were available from the start of the project was altered a little in order to make the interface. The communication speed and the hardware available were the challenges while accomplishing the required task.

This method of interface using the script feature of the RSCAD can be used for much other purpose. The script can be programmed to take in values from a firing pulse controller or operate a relay that is connect as HIL. The advantage of this using this method is that the values at the time of stable operation can be recorded in a comma separated value file, which can be used as a reference to island a microgrid from the utility grid for a micro grid
controller.

Interfacing using the DAQ tool box explained in method 4 in chapter 4.4 was started shortly before the end of the project. Since the hardware that was available to output was 1 GTAO card signals have to be sent sequentially, hence a dial was designed along with the signal selector to send the signals. Later this was modified to send signals automatically using a counter. But this limitation for this method is the number of nodes that can be used is 6. For receiving the signals sequentially as said in chapter 4.4 a controller was designed using logical components. For the storing the signals in a comma separated value file and vice versa simultaneously LabVIEW is being programmed for performing the loop. Once this part is accomplished the required hardware will be purchased for the interface.

Because of the extension of the testbed with several Raspberry Pis, it is now possible to test the agent behavior with ten different machines. The technology of these machines probably have a realistic behavior. The processors of the Raspberry Pi’s are based on the ARM technology, and thus it is used mainly in mobile phones or microcomputers. It is likely that this processor technology will be used for the development of future agent systems. This is why testing on this basis is very important and interesting.

5.2 Future Work

The solution for future work is, using NI LabVIEW software and NI data acquisition (DAQ) hardware to develop a low-cost microgrid energy management system (MEMS) that includes information and communications technology (ICT), smart meters, advanced optimization applications, and interfacing communication to manage distribution systems that serve as a platform for incorporating renewable energy resources. With the interface being done, this
framework is now giving a lot of options to perform various control actions for a microgrid. Thus, very interesting simulations can be driven. The agents can now be programmed to do economic dispatch and power balance for the system shown in Fig. 4.3. Once the interface through DAQ tool box is done it won’t be a problem to transfer data over a TCP/IP connection. This method could also be used to send data from the cluster to the agents. The range of the developed software for the testbed is very large, but so far not every bug and error could be eliminated. Therefore, it needs additional changes and essential improvements for the future. Another interesting possibility would be to connect the RTDS via the interfaces described in the thesis in section 2.4.2; hence, the implementation of real-time values would be possible. The range of the developed software for the testbed is very large, but so far not every bug and error could be eliminated. Therefore, it needs additional changes and essential improvements for the future.

The testbed created in this project offers a lot of interesting possibilities for further developments and projects in the field of Multi-Agent systems. With the MPI system, large computations for power consuming calculations are now possible. In addition, up to ten agents can be simulated on individual computers. Because of the interoperability, it is easy to increase the number of connected computers and agents. Thus, the testbed is a powerful system that is available for future projects.
BIBLIOGRAPHY
BIBLIOGRAPHY


[34] Jay Sorg, http://www.xrdp.org (08/13/13)

