Real-time power dispatch in a DC microgrid

Auteur : Stegen, Thomas
Promoteur(s) : Cornélusse, Bertrand
Faculté : Faculté des Sciences appliquées
Diplôme : Master : ingénieur civil électricien, à finalité spécialisée en "electric power and energy systems"
Année académique : 2020-2021
URI/URL : http://hdl.handle.net/2268.2/13301

Real time power dispatch in DC microgrids

Master Thesis

Submitted for the degree of Msc in Electrical Engineering

by Thomas Stegen

Supervised by Prof. Bertrand Cornélusse

FACULTY OF APPLIED SCIENCE

School of engineering and computer science

Academic year 2020–2021

Acknowledgements

First of all, I would like to express my gratitude to Pr. Bertrand Cornélusse, my supervisor. I would like to thank Pr. Cornélusse for his extensive availability during this semester, for all the relevant advice he gave me, for his expertise on the various topics that this work has lead me to discover and for the opportunity to work on this very interesting topic.

I would also like to thank Marine, Maxime and Paul for their help with the proof-reading of this work.

Thomas Stegen

Abstract

This thesis is motivated by challenges in the data center industry. With their critical loads and the significant power they consume, data centers are well suited for the implementation of microgrids, as they could become more reliable and energy efficient.

Moreover, the nature of the loads and energy storage in data centers designate DC microgrids as a perfectly fitting solution. Indeed, for the control of microgrids, a non-centralized solution is ideal as it provides more reliable control than a centralized controller, which is prone to communication errors and single point of failure.

The aim of this thesis is to implement a two level controller: lower-level as decentralized droop control and upper-level as a distributed optimal controller using Optimal Power Flow (`OPF`). Then, to test this controller, a Hardware In the Loop (HIL) device is used to enable a real-time simulation of the network.

For the HIL simulation, `Typhoon HIL Control Center` software is used with a `Typhoon HIL 402` hardware device. This structure provides a very efficient and user-friendly procedure for the modeling and the control of DC microgrids.

The distribution of the `OPF` has not shown satisfactory convergence. Hence, the tertiary controller of the real-time HIL simulation was implemented with a centralized algorithm.

Contents

Chapter 1

Introduction

The interest in microgrids has been growing drastically in the last decades. An important part of the scientific research in power systems is now carried out on this subject.

The genuine interest of scholars in decentralized and locally efficient power systems is partially due to the motivation to decarbonize the energy industry. The current increase in penetration of Distributed Energy Resources (DER), that are for a significant part renewable and intermittent, is creating new challenges in the management of bulk power systems. A decentralized context with microgrids instead of a national network could create opportunities to solve this problem. Indeed, scaling it down to a simpler and more predictive problem may be a solution to reduce complexity.

However, this technology is far from mature enough to be implemented widely and have a tangible impact on a national grid. Currently, most of the applications that are being developed are for providing electricity to isolated areas. Those areas are often in regions where a reliable public network cannot be installed due to the lack of uninterrupted electrical supply and demand, or simply in islands or other unreachable areas.

Some research projects are also being developed for local energy communities. These can be either industrial, domestic, military, or academic with a research purpose. These communities can have multiple objectives:

- **Economical goal:** maximize energy efficiency and provide ancillary services to the grid;
- **Ecological goal:** minimize ecological impact and raise awareness;
- **Independence:** pool resources (*e.g.*, storage, generation) and maximize self-consumption.

This work is inspired by a project in the data center industry. Indeed, the electrical loads (*i.e.*, the servers and cooling system) in those facilities are critical and power intensive. Those two properties make microgrids an interesting addition for data centers. An adapted microgrid can ensure better reliability of the system as well as an improved price efficiency.

Moreover, the nature of the loads (which are consuming DC power) and the energy storage systems (DC batteries usually) make this application very well suited for DC microgrids.

Each microgrid needs a controller, an entity that ensures the point of operation is within the hardware limits of each network element. There are different ways of implementing this controller and it can achieve different goals. This will be presented in further details in SECTION 1.2.

In data centers, the actual necessity for a thoroughly reliable system discourages using a centralized controller (*i.e.*, a single entity handling the control task for the whole system). Indeed, the centralized controller can bring a single point of failure in such a network. This issue brings a new challenge to the application: finding a reliable and efficient decentralized controller.

In the case study, a data center will be modeled by several racks with multiple servers and a cooling system simplified as a single load. These racks connect to the microgrid through an energy management system (responsible for monitoring and managing the energy at the bus). Each bus will consist of a converter with a controller, a load, and an energy storage system. The different nodes will be tied through both communication and power links.

SECTION 1.1 introduces the concept of microgrids, gives a succinct description of their history and motivations. It also provides a fundamental perspective of the properties and types of microgrids. It concludes in SECTION 1.1.3 with an overview on the specific case of DC microgrids, their advantages, and drawbacks.

SECTION 1.2 presents the most common ways of controlling networks: hierarchical, centralized, decentralized, and distributed control. This SECTION also explains the most common bus control strategies: voltage control, power control, droop control, and load shedding.

1.1 Concept and goals of microgrids

In this SECTION, the main concept of microgrids will be explained. Their objectives will be presented, along with the constraints they are subject to and the properties that they can display. Next, DC microgrids, the main subject of interest of this master thesis, will be presented with their specific advantages and drawbacks.

1.1.1 Description

The United States Department of Energy's Microgrid Exchange Group defined a microgrid as follows [1]:

"A microgrid is a group of interconnected loads and distributed energy resources (DER) within clearly defined electrical boundaries that acts as a single controllable entity with respect to the grid. A microgrid can connect and disconnect from the grid to enable it to operate in both grid-connected or island-mode."

Although it is commonly accepted as the definition of a microgrid, this description is very general. For a better understanding of the goals and stakes of modern microgrids, a more precise definition may be helpful.

Another interesting definition has been stated in [2], which suits the modern microgrid vision better. This definition is fully in accordance with the concept of microgrids and their objectives as they will be considered in the following CHAPTERS.

"A microgrid is a modern distributed power system using local sustainable power resources designed through various smart-grid initiatives. It also provides energy security for a local community as it can be operated without the presence of wider utility grid. Microgrid technology generally represents three important goals of a society such as reliability (physical, cyber), sustainability (environmental considerations), and economics (cost optimizing, efficiency)."

A microgrid is a local energy system that aims at being self-sufficient. It can be defined by three key characteristics [3]:

- **A microgrid is local** – it is designed for a limited number of actors. This focus allows meeting the interests of each actor better. It also minimizes the transfer losses because the lines are shortened.
- **A microgrid is independent** – a microgrid can operate in islanded mode, completely disconnected from the grid. This characteristic gives the microgrid an improved reliability in case of a blackout.
- **A microgrid is intelligent** – a microgrid should implement a control strategy that is not naive. It has some degrees of freedom in its operation and should retrieve the best out of its capacity.

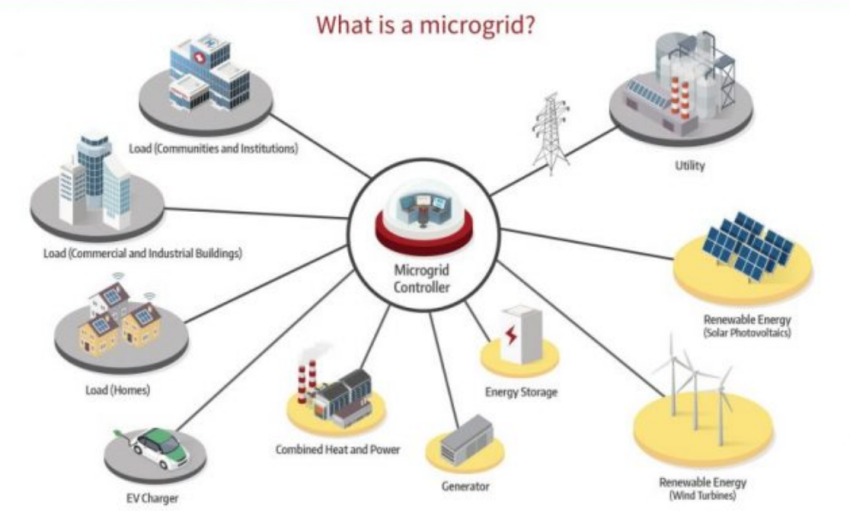


Figure 1.1: Schematic view of a microgrid [3]

According to [4], the essential properties that microgrids are designed to have are the following:

- performing tasks in both islanded or grid-connected modes;
- introducing the microgrid as a solidary and solitary controlled element;
- mixing interconnected burdens and co-found powerage sources;
- arranging fluctuated stages of intensity quality & unwavering quality for end customers.

1.1.1.1 History

The concept of microgrid has existed as long as the electric industry. The first system that can be considered a microgrid was Edison's Pearl Street station. There was no standardized distribution-generation system, so Edison just created its network, locally and isolated from the main grid (as there was none at that time). In practice, it was a small generation and distribution network operated using DC transmission over a few blocks. Surprisingly, this network fits the modern definition of microgrids.[5]

The rise of popularity of modern microgrids seems to have begun in the late 1990s, this is when the term has been used first [6]. Since then, several projects have emerged in places where the research and electricity needs are extensive, such as universities, military bases, industrial parks, or substantial public facilities (school districts, prisons, or hospitals).

University campuses were pioneers in this field. They have been studying and materializing microgrids for decades, showing the benefits they can bring. In the United States of America, interest started rising in 2011–2012. A series of severe storms hit the Northeast, causing multiple electricity outages. However, like Princeton, microgrid operators managed to achieve uninterrupted power supply using their technology [7].

The first goal of a microgrid is to localize energy production and consumption between neighboring actors. This action aims to improve reliability and resilience while maximizing the use of distributed generation resources.

1.1.2 Motivations

The definitions stated in SECTION 1.1.1 directly reflect the 3 D's (Decarbonization, Decentralization, and Digitalization) [8]. This concept is a pathway to develop a smart and sustainable power system industry. It is focusing on three main axes:

- **Decarbonization:** to reach a more sustainable power grid, carbon emissions are one of the main problems to tackle. Most of the conventional energy sources are emitting abundant greenhouse gases. Eliminating carbon-based energy sources often means switching to highly intermittent renewable sources. This can become a problem at a large scale for a power system. However, this intermittency is easier to deal with at a lower scale. For example, Energy storage systems can be more easily sized and installed for a microgrid to operate continuously with intermittent resources than for a whole country. Hence, microgrids can boost this axis;
- **Decentralization:** the very essence of the microgrid is to decentralize. From a national or even international grid management, local and distributed networks or communities are created. Decentralization is critical at a time where an increasing number of big power plants are decommissioned to be replaced by smaller distributed energy resources;
- **Digitalization:** with the increasing complexity of energy markets, effective energy monitoring and management is imperative. On the microgrid topic, efficient control of the assets and loads of each actor is critical, and that is most likely to be achieved through digitalization.

Moreover, the continuous development of renewable energy sources (RES) is also changing power network management. The International Renewable Energy Agency (IRENA) estimated the energy supply provided by RES to be 66% in 2050 [9]. The main problem is the intermittent nature of these sources. A potential solution would be to integrate these sources directly to a DC bus of a DC microgrid [10].

FIGURE 1.2 presents several drivers for different actors around microgrids. The following SUBSECTIONS will present in more detail the most prominent advantages of microgrids.

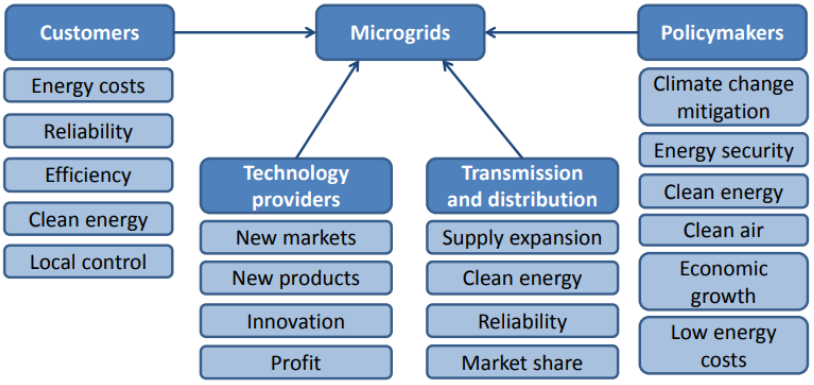


Figure 1.2: Drivers for microgrids across four stakeholder groups [11]

1.1.2.1 Reliability

One of the goals of microgrids is to provide a more reliable power delivery. This can be achieved using an appropriate control mechanism which will be presented in CHAPTER 1.2.

However, the reliability of the microgrid is also dependent on its implementation. In case of incomplete or flawed control, the microgrid's impact on the network's reliability can be deleterious.

1.1.2.1.a Physical reliability

As already stated, the physical reliability of a microgrid can be greater than that of a large-scale grid. Namely, in case of cascading outages due to severe weather conditions, the microgrid keeps power distribution uninterrupted.

Microgrids also segment the network into smaller units, preventing the domino effect that can occur and taking down the entire grid. In case of emergency, it is possible to totally isolate the microgrids so their operation is uninterrupted [12].

1.1.2.1.b Cyber reliability

The public grid is vulnerable to cyberattacks because it relies heavily on information and communication technologies [13]. The decentralized nature of microgrids makes them invulnerable to a single point of failure at network-level and less prone to attacks.

Also, some microgrids are operated in a fully decentralized manner. They do not need any communication devices to exchange data between the different buses. This makes microgrids much more cyber-reliable.

1.1.2.2 Sustainability

The production of certain renewable energy sources is very intermittent. The power in assets like photovoltaic or wind farms is variable and non-controllable. That can cause

serious challenges in a large-scale grid, but microgrids are designed to handle variable generation, using storage technologies to balance generation and loads [14].

1.1.2.3 Economical incentive

Even though the primary goal of microgrids in campuses is usually research, in industrial parks, the main motivation is often economical. Indeed, when a new project is launched, the best incentive to invest in it is the opportunity of increased profit.

1.1.2.3.a Cost optimization

The energy cost can be optimized in a microgrid through efficient energy management. The power produced can be determined by energy markets, with the electricity being sold to the grid when tariffs are advantageous. Also, microgrids can provide ancillary services, which can also be financially compensated.

1.1.2.3.b Energy efficiency

Microgrids offer several advantages for energy efficiency. They reduce losses in the lines or converters and can better allocate energy resources due to the mutualization of the assets.

The assets are used to maximize the value over time. This can be achieved by minimizing the electricity bought through the grid or using an advanced strategy of both short and long-term planning [15].

1.1.3 DC microgrid

This work focuses on the specific field of DC microgrids. This field of research is more recent and less mature than AC microgrids. Even though their implementation is still less common, DC microgrids present many advantages.

They feature fewer conversion stages, which removes unnecessary losses and improves reliability. In DC microgrids, a more straightforward control structure is possible as they do not deal with inrush currents, frequency synchronization, and power quality management [16].

Moreover, by nature, some DERs and loads produce or consume DC power. This also encourages the use of the DC paradigm for the complete microgrid. For example, in the data center case study, the load is exclusively in DC. With the addition of PV panels as a RES is a DC source, which can be used with a battery storage system, also in DC, the DC microgrid becomes more appealing.

Regarding the case study, in data centers, the need for high reliability and the magnitude of the power flows legitimates the use of a microgrid. Low-voltage DC microgrid can be superior to AC. The reason is that source, storage and loads can be directly connected without AC/DC or DC/AC converters, eliminating losses in the conversion step. [17].

The interest in DC microgrids has grown significantly over the last 15 years. As FIGURE 1.3 shows, in the ten-year period between 2007 and 2017, the number of scientific publications on this subject has been in constant increase.

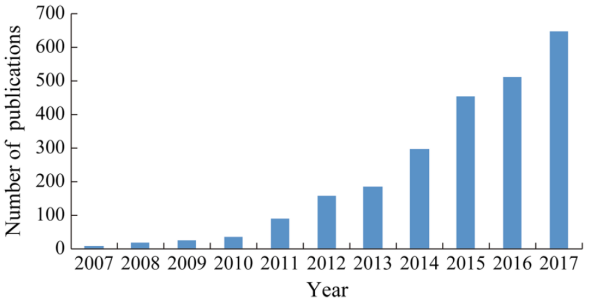


Figure 1.3: Evolution of the number of publications on DC MGs per year [18]

1.1.3.1 Advantages

In addition to the previously enumerated basic advantages of the DC technology, some specific properties of the DC standard also improve its suitability for microgrids implementation.

1.1.3.1.a Cost optimization

In undeveloped or remote areas, the load demand is often lower than usual. Therefore, due to the maintenance and transmission costs for a large-scale grid, this is not the economically optimal choice. DC microgrids can be a cost-efficient alternative to deliver power to those areas [19].

1.1.3.1.b Energy efficiency

Moreover, DC microgrids can still operate while completely separated from the grid. In developing countries, conventional AC grids can be unavailable or unreliable. Local distribution units would greatly benefit from using DC microgrids rather than the AC system, as selecting a DC interconnection bus is simpler, more reliable, and more efficient [20].

1.1.3.1.c Energy storage systems

At the scale of microgrids, ESSs are often chemical batteries, and those are using direct current. Their integration in the DC microgrid is thus profitable in comparison with an AC microgrid.

Moreover, a proper control system can implement a battery management system (BMS). This system can equalize the state-of-charge (SoC) level of all the storage systems in the microgrid [21].

1.1.3.1.d Load management

In microgrids, the stability and reliability of the system are greatly influenced by the loads. The use of DC microgrids can significantly diminish the risk of power failure. Indeed, the lack of frequency variation and active/reactive power fluctuations allow control to be more reliable, as shown in [22].

1.1.3.1.e Electric vehicles

The rise in popularity of electric vehicles is also encouraging research in new applications of DC microgrids. The existence of DC battery systems in those cars and the fact that they are directly connected to the grid allow control strategies involving them to emerge.

Using the vehicles' batteries that are plugged, the controller can use the excess in storage capacity or the remaining power to mitigate power fluctuations and achieve better reliability of the microgrid [23]. More advanced methods involving predictions for the future usage of car batteries are also possible [24].

1.1.3.2 Drawbacks

Nevertheless, DC microgrids are a very immature field. The research in this domain is relatively recent and some challenges are yet to be resolved. Moreover, the nature of the DC paradigm can also bring new hindrances in the developments of these technologies.

1.1.3.2.a Protection of the microgrid

The absence of zero-crossing currents represents a major safety-critical problem. This means the protection devices for DC microgrids require much more attention than their AC counterpart. There is still a substantial research gap in this area of DC microgrid protection schemes. Better operation and safety of the operators, end-users, and other participants involved still needs to be investigated [21].

1.1.3.2.b Islanding

The nature of the DC power and the absence of zero-crossing currents also exacerbate the difficulty of achieving smooth transitions from off-grid to grid-connected operation [25]. However, this problem can be solved by reducing the power transfer between the grid and the microgrid to zero, then disconnecting the network coupling point. A seamless transfer is the key to guarantee smooth transitions between islanding and connected operation modes [26].

1.1.3.2.c Stability issues

The constant nature of the loads in DC microgrids can provoke stability issues when the loads get coupled with a negative impedance characteristic [27]. This phenomenon can also cause resonant conditions due to a virtual negative inductance [28].

1.2 Control strategies for DC microgrids

This SECTION introduces the basic notion of control in power systems. The goal here is to introduce the basic concepts and strategies relative to control in the specific case of a microgrid, focusing on the way the reference is set for the control to operate correctly, not on the way the controller reaches this reference.

In this work specific details of control theory will not be discussed. Problems like closed-loop PID control and error minimization in the controller implementation are not considered.

1.2.1 Introduction to control

In every power system, either a micro- or a macrogrid, there are constraints that need to be met at all time. These constraints are due to the nature of the hardware elements and their limitations, or just emanates from the present demand in energy consumption. In power systems, a proper dynamic controller is thus necessary to ensure that each component always operates in its physical boundaries.

The power in each line cannot be directly controlled individually. It depends on physical laws, power flows in a DC microgrid are dependent on the voltage difference between the two ends of each line. Thus, the control occurs on the buses of the network.

The nodal voltage or injections determine the whole state of the system. In practice, the injection considered in each node is a power injection. It is essentially equivalent to consider both, as the power is simply equal to the voltage multiplied by the current injection. However, in practice, the voltage control can bring better stability in the network as it sets more precisely the power flows between the nodes.

The control aims at always having a stable and feasible state in the system. Most of the time, it needs a realistic model of the network and its parameters, as well as a set of constraints with variables that must be met at each time period. However, some new techniques using reverse engineering allow the network not to be modeled directly [29].

1.2.1.1 Objectives of control

As stated hereabove, the main goal of control is to have a reliable grid, which stays within the operating limits of the different components and provides the necessary power to all the critical loads. Furthermore, some controllers also aim to reach some objectives (*e.g.*, minimizing losses, optimizing economically, or maximizing renewable energy usage).

In the reference book "Microgrid: architectures and control" from N. Hatziargyriou [30], several control targets are listed:

- Smooth switching from islanded to the grid-connected method of operation;
- Regulation of voltage and current sharing;

- Stable operation with nonlinear as well as constant power load;
- Optimizing the Micro Source (MS) production to participate in the energy market;
- Controlling the power flow among MG and the rest of the network using an efficient and proper Energy Management Scheme (EMS);
- Controlling the power flow among MG and the rest of the network using an efficient and proper Energy Management Scheme;
- Efficient load power-sharing and proper communication between DERs;
- Proper control mechanisms to prevent grid failure and potentiality to black start;
- Generation cost optimization and economic dispatching of loads;
- Maximizing the potentiality of DERs and reducing the transmission losses;
- Capability to provide uninterrupted power supply to critical loads like hospitals, industries, and other crucial utilities.

1.2.2 Type of bus controllers

Besides the objective, another essential aspect of microgrid control is the nature of the controller in each bus. Depending on the variable that is regulated, the behaviors of the systems can be entirely different. This SECTION solely focuses on the different bus variables that can be controlled, not on the strategy or implementation of this control.

1.2.2.1 Voltage control

The most basic bus control method is voltage control. The considered bus is maintaining its voltage level to a given voltage reference. The microgrid controller is responsible for finding the adequate voltage for each bus in voltage control mode.

In practice, this kind of control can be implemented by a DC/DC converter. This technology is designed to operate with a voltage reference. The converter controller is responsible for maintaining the power to this reference.

1.2.2.2 Power control

Another way of controlling a bus in a DC microgrid is power control. In this case, the controller aims to reach a given instruction for the value of injected power at this node. The microgrid controller sets this power reference.

Some DERs are by default operated in power control mode, this is notably the case for buses with PV power plants or wind turbines, which are designed to inject the maximum power that can be produced at all times. The power magnitude is dependent on the weather conditions and are determined by MPPT devices.

1.2.2.3 Droop control

Droop control is a technique commonly used for local power or current sharing in the buses of DC microgrids [31]. It is implemented in power converters. It consists of a linear relationship between the voltage and either the current or the power output of the converter.

$$V = \hat{V} - k_p P \tag{1.1}$$

$$V = \hat{V} - k_c I \tag{1.2}$$

EQUATION (1.1) gives the voltage output V of a power-based droop controller and EQUATION (1.2) shows the same for a current-based droop controller. $\hat{V}$ is the DC voltage reference of the converter, the value at which the power or current output is null. k_p or k_c is the droop coefficient which is also called virtual equivalent resistance. This coefficient is mimicking the effect of a generator's internal resistance in the converter. It determines the sensitivity between the power fluctuations and the voltage fluctuations [32].

The base principle of a droop controller is simple. When the voltage increases, the power produced also increases and reciprocally. This regulation allows the voltage magnitudes to stay more stable than with a basic power control strategy.

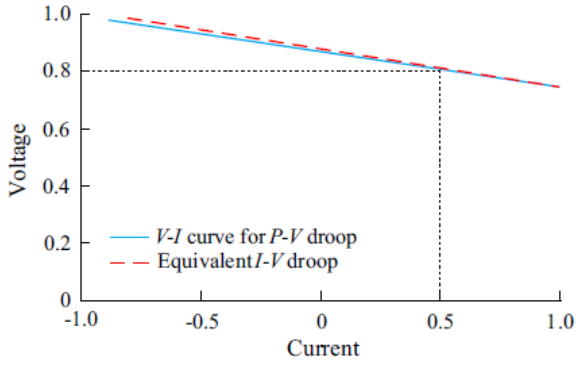


Figure 1.4: Power voltage droop [33]

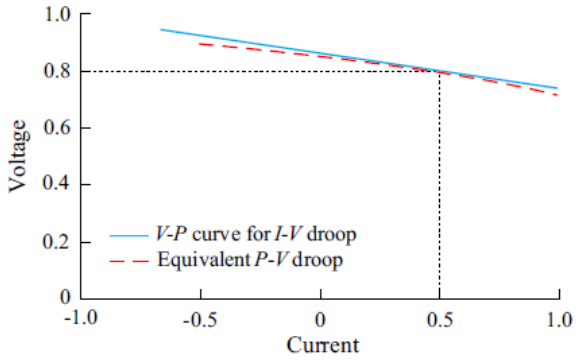


Figure 1.5: Current voltage droop [33]

FIGURE 1.4 and FIGURE 1.5 show the characteristic curves of two equivalent droop controllers. One is controlled with a power reference, the second is controlled using current. Both models are almost equivalent with little differences when small voltage errors are considered [18].

1.2.2.3.a Limitations

Two main limitations of the conventional droop control method for DC microgrids are identified in [34]:

- **Current sharing accuracy degradation:** the controller is implemented using a virtual resistance k. If the line resistances are also taken into account, the DC voltage for the converter differs. Hence, there is current sharing accuracy degradation;

- **DC voltage deviation:** when the output current is null (*i.e.*, the converter operates in open-circuit), the voltage deviation is zero. However, the load current induces a voltage deviation. Thus, the value of k, the virtual internal resistance, should be limited so that the voltage deviation is not too important.

These issues are being investigated, and improved methods are being developed to overcome them [35]. Droop control can also be implemented to achieve different modes of operation depending on several strategies [36] or depending on the voltage level of the bus [37].

1.2.2.4 Load shedding

To improve the controllability of the network, it is also possible to allow load shedding. By defining a cost (*i.e.*, a restraining factor) for renouncing to provide part of particular loads, the system can have a new control over non-generating buses. That would be useful in situations where producing more would be less beneficial than consuming less in some loads.

However, in this specific work, this option will not be considered due to the critical nature of the loads.

1.2.3 Hierarchical control

Another way of distinguishing types of controls is the time scale and the objective of the controller. This is how primary, secondary and tertiary controls are defined.

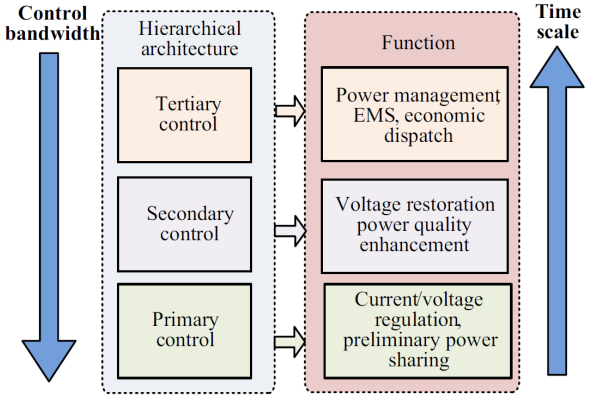


Figure 1.6: Hierarchical control architecture as presented in [18]

Sometimes, only two levels of control are considered: lower-level control and upper-level control. In these cases, the lower-level control mainly corresponds to primary and secondary control, whereas the upper-level control corresponds to the tertiary control. In other words, the lower-level control is in charge of the regulation and stability of the network, and upper-level control is in charge of finding the optimal operating point within the stability limits.

1.2.3.1 Primary control

Primary control is responsible for rapid current/voltage regulation and preliminary power-sharing. It should be very fast to react directly to variations in the microgrid.

This control is implemented locally at each bus and is automatic. It does not try to reach an optimum. It only focuses on keeping the operating point of the DC bus voltage within acceptable boundaries. It is fully automatic and operates mainly to mitigate voltage variations.

It often takes the form of droop control for controllable generators. Other possibilities are DC-bus signaling, fuzzy logic control, and master-slave control [38].

1.2.3.2 Secondary control

The primary controller is not always sufficient to manage the microgrid correctly. It can be the case when line resistances are too large for the droop control to operate flawlessly [34]. Thus, the hierarchical control can use a secondary controller to provide more adapted voltage regulation. It is often implemented in a centralized manner, as it is simpler to achieve an overall voltage regulation in these conditions. Yet, distributed secondary control is also totally feasible [39].

It is interesting to note that this network-wide voltage regulation is not always needed. Henceforth, the secondary controller as it is described here is not invariably implemented.

1.2.3.3 Tertiary control

The tertiary control in DC microgrids acts after and on top of both primary and secondary controls. Its goal is to optimize network usage. To achieve that, it solves an optimization problem with the network constraints. Tertiary control follows a chosen strategy which is determined by the objective function of the optimization problem. The objective function can model loss minimization, equilibrium in the storage systems state of charge, and cost minimization, among others.

The tertiary control can provide new references for the primary and secondary control, such as droop control used as primary control with power references given by an OPF [40].

1.2.4 Types of system controllers

There exist mainly three categories of controllers differing in the way they are implemented at a system scale. Centralized, decentralized, and distributed controls are the three main modes of interactions between the different local controllers (LC) of a DC microgrid. The local controller is responsible for the control of only one bus of the model.

1.2.4.1 Centralized

The centralized controller is a complex system that uses the state of the complete network to control all of its elements. It must use a complex and reliable communication system. There is a central processing unit, which receives data from all the elements and processes them to determine the control variable for each piece of equipment. The references are then sent back to the controlled hardware [41].

Centralized control is prone to communication errors and leads to a potential single point of failure. However, it is still widely used for secondary and tertiary control due to the simplicity of its implementation [42].

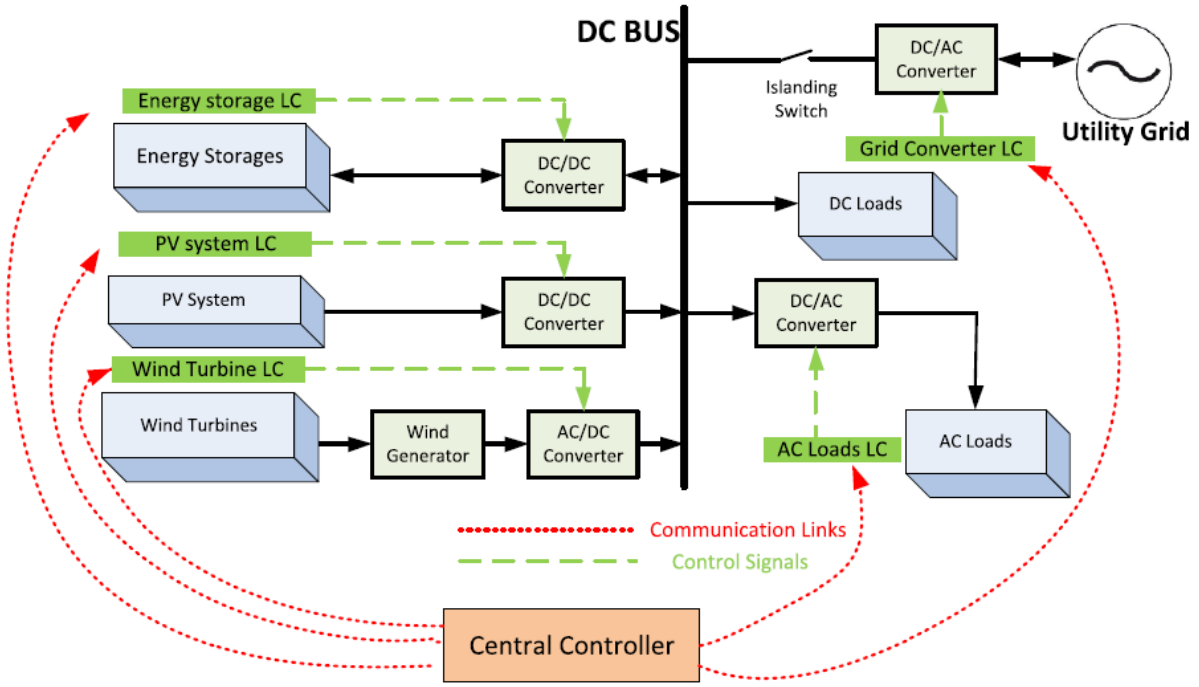


Figure 1.7: Centralized controller for DC microgrid [38]

1.2.4.2 Decentralized

Decentralized control implements a separate control mechanism at each local controller (LC). The strategy is completely localized, and there are no interactions between the components. The most common decentralized controller is the conventional droop which simply stabilizes the voltage at the considered bus.

This strategy has some advantages (*e.g.*, simplicity of control and unaffected by communication) and inherently has some performance limitations. It does not exchange information with other buses, which can cause reduced effectiveness and reliability [42].

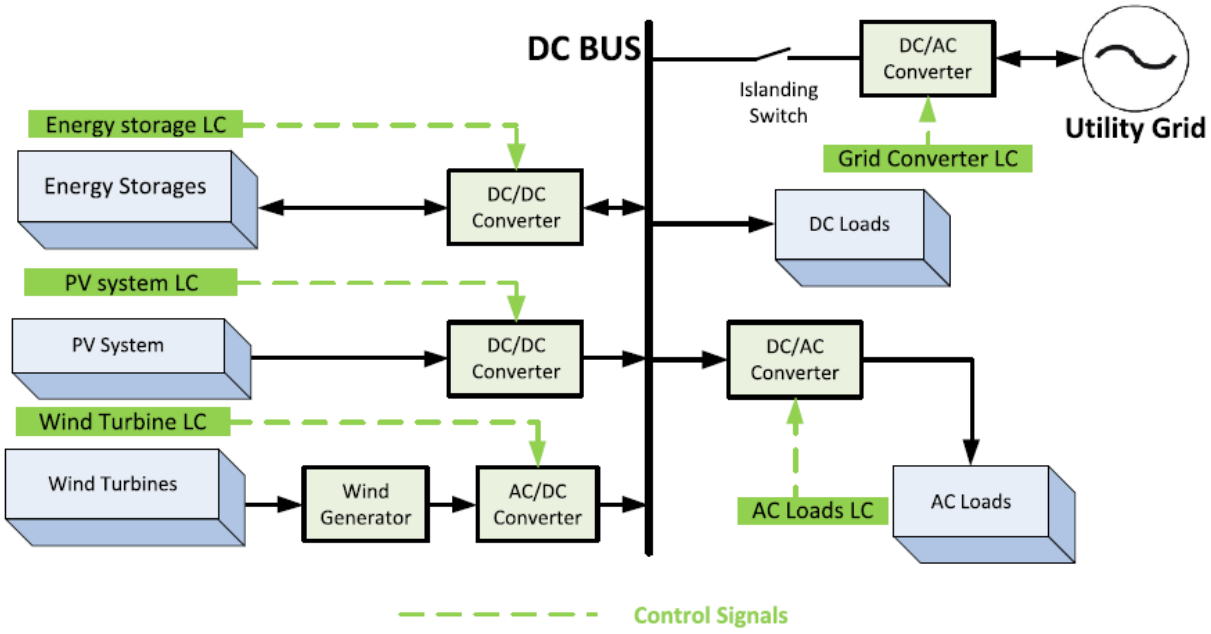


Figure 1.8: Decentralized controller for DC microgrid [38]

1.2.4.3 Distributed

Distributed control tries to combine the advantages of both of the previous strategies. It uses local communication links to communicate between neighboring units. The local controller uses the information from neighboring buses, which can be voltage magnitude, output current, or others, to better control the microgrids at a larger scale.

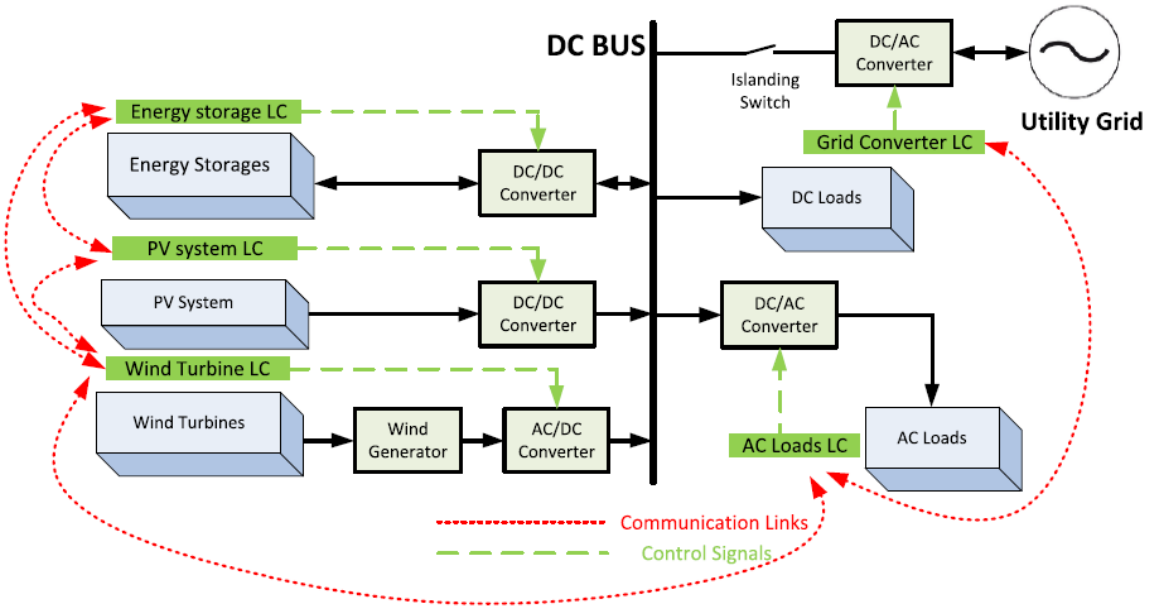


Figure 1.9: Distributed controller for DC microgrid [38]

1.2.5 Summary of control

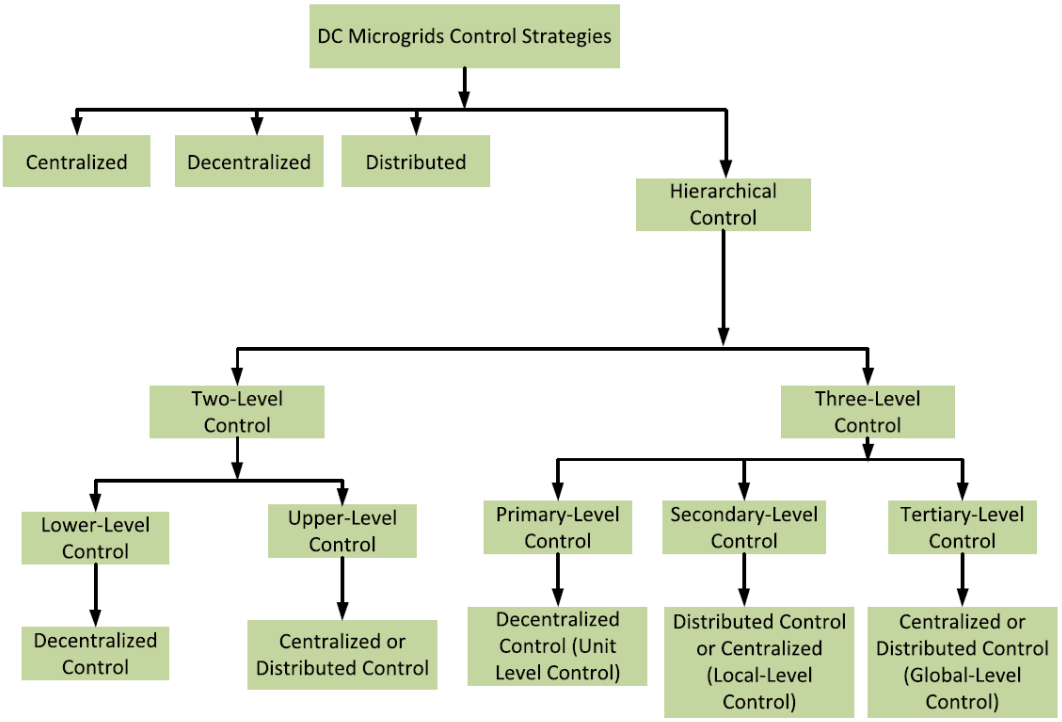


Figure 1.10: Diagram of the main control strategies for DC microgrids [38]

This diagram shows a complete overview of the control strategies usually implemented in a DC microgrid. This work aims to implement a distributed upper-level control, with a decentralized lower-level control implemented using conventional droop control.

Decentralized primary control, as previously explained, can be realized relatively quickly with droop control. A proper tertiary control strategy must be implemented to achieve better power or better cost efficiency in the microgrid. That brings a new challenge, as the objective is often on the scale of the whole microgrid while distributed control must be implemented separately in each bus of the network. However, some strategies to solve this problem have been found, namely in [43], [44], and [45].

Upper-level control involves the resolution of an optimization problem to find the ideal operating point. This point will depend on the instantaneous conditions of the network (*i.e.*, the loads and power production variables) and on the strategy that is followed. An objective function models this strategy. This is the topic of the next CHAPTER.

1.3 Literature review

Eghtedarpour and Farjah [46] propose a control strategy for distributed integration of PV and ESS in DC microgrids. This strategy implements DC bus signaling, a technique using the bus voltage levels to transmit information without a proper communication network.

However, the information distributed is only the operating mode of each bus; the controller itself is centralized.

Khorsandi *et al.* [35] aim to develop a new decentralized control strategy for low-voltage DC microgrids. A new strategy is proposed based on the disadvantages of the conventional droop control strategy introduced in SECTION 1.2.2.3, notably the deterioration of the control due to the effects of line resistances. This method is an improved droop control using estimation of the parameters of the network. It implements multiple operating modes and can manage battery states of charge and distributed PV systems with good results.

Gu *et al.* [37] introduce a mode-adaptive decentralized control scheme. Their goal is to improve the classical droop control. In the presented technique, the droop curves of different types of terminals are divided into three voltage ranges. The DC bus voltage can be used to designate operation modes and facilitate adaptive mode transitions. To do so, three main modes are defined: at high voltage, the bus is in generation mode; at medium voltage, it is in storage mode, and at low voltage, it is in utility mode. With this mode-adaptive operation mechanism, greater control freedom can be obtained than with the conventional DC voltage droop control scheme. More importantly, this approach features fully automatic regulation of distributed converters without an extra control center or communication link.

Moayedi and Davoudi [47] propose a distributed tertiary control method for DC microgrid clusters. They implement power-sharing between different neighboring microgrids, whereas primary and secondary controllers act in each microgrid individually. Power transfer between microgrids enables maximum utilization of renewable sources and suppresses stress and aging of the components. This improves reliability and availability, reduces maintenance costs, and expands the overall lifespan of the network. The proposed solution works with the distributed secondary control system to benefit from its communication structure in propagating voltage set points. It does not require prior knowledge of the number of microgrids in the cluster and, therefore, provides a plug-and-play capability. Experimental results verify the controller performance in load sharing among microgrids and its physical- and cyber-fault resiliency.

Morstyn *et al.* [48] developed a strategy that does not use bus voltage signaling or mode detection mechanism. The control strategy is based on a distributed average consensus protocol for balancing the ESSs, distributed controllers to regulate the average DC microgrid bus voltage, and a new method to control the grid-connected rectifier that maintains the distributed control structure. The model can operate in three modes: grid-connected with rectifier providing load balancing, grid-connected with rectifier charging the ESS, and islanded mode. The main advantage of the proposed control strategy is that the average DC microgrid bus voltage is regulated during all modes and mode transitions. Also, the energy storage systems achieve a balanced energy level and maintain it through accurate load sharing, independently of the operating mode.

Zhang *et al.* [29] consider an AC network modeled as a linear system. They use a distributed algorithm to achieve optimal steady-state control. It is based on recent research in cyber-physical systems. Starting from a standard dynamic system, reverse engineering

is used to obtain the gradient descent of the optimal power flow of the network. Then, distributed control that can react to network modifications is implemented. This framework consists of two steps: seeking an appropriate optimization problem that the system dynamics implicitly solve (reverse engineering) and modifying the resulting optimization problem by incorporating a predefined optimization problem and derive control mechanisms to solve the augmented optimization problem (forward engineering). Necessary and sufficient conditions for the use of this method have also been developed.

Nasirian *et al.* [16] study distributed adaptive droop controllers in the context of DC microgrid control. The aim of this paper is to provide a regulated rated voltage and proportional load sharing, just like in AC grids. This is achieved using a cooperative algorithm that estimates the average voltage in the network. The algorithm is adaptive, which means that it reacts to both regulate the voltage and share the loads proportionally when there is a variation in the network. In this work, each converter communicates with a few neighbors. A voltage observer processes neighbors' information to estimate average voltage across the microgrid. This estimation is also used to improve the voltage reference point and compensate for the voltage drop caused by the droop mechanism. Adaptive-droop control is also introduced, where a current regulator adjusts the virtual impedance of the droop mechanism to ensure proportional load sharing. Comparative studies show that the proposed controller successfully boosts the grid voltages and regulates the average voltage at the rated value.

Salomonsson *et al.* [17] tackle the reliability issue in data center power systems. Through their paper, they propose a solution using an adaptive control system with a DC microgrid in data centers. It also states a motivation for using such a solution. The presented solution allows the microgrid to operate in 8 different modes, depending on the emergency, standby power, and status of the grid. It handles sensitive AC loads, reduces transients in the voltage, and includes brief utility outage detection and fast switches. However, this article is quite old (2007) and only focuses on the safety issues of the DC microgrid. It does not implement an efficient way of reaching an optimal point regarding a particular objective. The only objective here is a reliable and uninterrupted operation of the microgrid.

Korompili *et al* in [43] use the alternating direction method of multipliers (ADMM) to achieve distributed optimal control in DC microgrids that implement active network management. They define different strategies for three types of terminals and provide a methodology to solve the problem for every combination of strategies. Results show high-speed convergence and coherent operating points (*i.e.*, close to centrally computed solution). This paper states that ADMM is a powerful tool to achieve active network management in a distributed manner.

M. Wang *et al.* [44] also use ADMM to achieve optimal power flow control. Their paper proposes a penalty-based ADMM approach using difference-of-convex programming to solve the non-convex optimization problem. If the solution is not feasible due to the non-convexity, a penalty item is added to drive the solution to be feasible. Then, an optimal solution can be obtained using a local nonlinear programming method. The proposed method can solve many problems in DC systems, such as PV dispatch and battery charging/discharging strategy. Moreover, the DP-SOCP method can also be combined with other methods, such as Newton-like algorithms, to accelerate the convergence speed.

Finally, Z. Wang *et al.* [45] propose a distributed optimal power flow model. Their paper proves that using a specific Optimal Power Flow (`OPF`) problem, the computation of the optimum operating point can be realized locally by each bus of the system while only communicating with its electrical neighbors. This method is based on gradient descent. Its advantages are the versatility of the control methods. This means that each bus of the network can implement either voltage control, power control, or droop control. It can also implement a plug-n-play property. The attempt to distribute the controller and to update dynamically the references for the network, as presented in CHAPTER 3, is directly inspired by this paper. CHAPTER 3 first explains the implementation of the methods and then will show the results obtained on different test networks.

	Objective	Strategy	Communication	Specificities
[46]	Stability of DC voltage	Centralized control	DC bus signaling	PV and ESS integration
[35]	Stability of DC voltage Improve droop method	Improved decentralized droop	N/A	PV and ESS integration
[37]	Stability of DC voltage Improve droop method	Improved decentralized droop	N/A	Different modes of operation according to voltage
[47]	Proportional power-sharing	Distributed tertiary control	with neighboring MGs	Clusters of MGs, each with primary and secondary control Tertiary control distributed among the MGs
[48]	Balancing ESS SoC	Distributed tertiary control	with neighboring buses	Three modes of operations A consensus algorithm is used
[29]	Optimal steady-state	Dynamic optimization	with neighboring subsystems	AC networks reverse- and forward-engineered No model needed *a priori*
[16]	Regulated voltage Proper load sharing	Adaptive droop control	Sparse communication to estimate V	Fixes the main issues of conventional droop
[17]	Reliable and uninterrupted operation	Centralized control	Full communication network	Models a data center 8 different operating modes
[43]	Optimal control	Distributed ADMM	with neighboring buses	Different strategies can be implemented Fast and close to optimal convergence
[44]	Optimal control	Distributed ADMM	with neighboring buses	Uses penalty term in non-feasible solutions to ensure exact relaxation of the non-convex problem
[45]	Optimal control	Distributed `OPF`	with neighboring buses	Distribute the gradient descent of the `OPF` saddle point research problem

Table 1.1: Comparison of the different approaches presented in the literature review

1.4 Structure of the manuscript

This thesis is motivated by challenges in the data center industry. Thus, DC microgrids will be investigated. The optimal goal is to implement a two-level controller: lower-level as decentralized droop control and upper-level as a distributed optimal controller using Optimal Power Flow (OPF). Then, to test this controller, a Hardware In the Loop (HIL) device will be used to have a real-time simulation of the network.

In CHAPTER 2, a general description of optimization problems is given, with the dual problem, and an introduction to the gradient descent solving method. Then the notions of power flows and OPF are presented, as well as the specific OPF problem that will be solved in this work.

In CHAPTER 3, the way the problem from CHAPTER 2 is distributed is presented. An attempt to solve this very complex distributed optimal power flow problem with two different test networks is then described. The networks and the results of the distributed algorithm are also presented.

In CHAPTER 4, the network that will need a dynamic controller is described. The controller will implement tertiary control using the algorithm from CHAPTER 2. Diverse test scenarios are defined to illustrate the dynamic response of the controller. Then, the real-time power dispatch is presented, with the result obtained experimentally.

In CHAPTER 5 the work is recapitulated, with its achievements, issues and drawbacks of the methods used. Some suggestions for further research on the subject are also given.

Chapter 2

Optimal Power Flow problem

A very frequent way of achieving optimal control in a microgrid is by using computations called Optimal Power Flow (OPF). This method models the network as a set of nodes and edges representing the buses and lines, respectively.

In this CHAPTER, the implementation of a distributed DC microgrid controller is studied. SECTION 2.1, introduces the concept of optimization, while SECTION 2.2 presents OPF problems, as well as the specific OPFs used and solved for this application.

Each line is commonly represented with an impedance and a maximum power rating, while each bus usually has power generation and voltage limits. Some buses also represent loads and must consume a given amount of power.

The state of the system is physically governed by electrical laws. Power flow algorithms are tools used to find a feasible point of operation, when OPF goes one step further and tries to find the best feasible point of operation by finding the optimum of an objective function in the feasible set of points.

2.1 Optimization problem

An optimization problem is a mathematical problem usually represented by an objective function ($f_0 : \mathbb{R}^n \to \mathbb{R}$), depending on some variables ($x \in \mathbb{R}^n$) and subject to some constraint functions ($f_i := 1, ..., m$). The objective function is given with a direction stating if it should be minimized or maximized. All the general descriptions about optimization problems in this SECTION are directly adapted from [49].

In the standard form, an optimization problem is written with an objective to minimize (f_0), non-strict lower inequality constraints (f_i) and equality constraints (h_i) as follows:

$$\min_x \quad f_0(x) \tag{2.1}$$

$$\text{subject to} \quad f_i(x) \leq 0 \qquad i = 1, ..., m \tag{2.1a}$$

$$h_i(x) = 0 \qquad i = 1, ..., p \tag{2.1b}$$

The optimal solution $x^\star$ reaches the smallest value of f_0, while satisfying all constraints

(*i.e.*, $x^\star \in \mathcal{D}$).

2.1.1 Dual problem

From this problem, a dual formulation can be constructed, using a Lagrangian function. For the standard form (PROBLEM (2.1)), the Lagrangian is:

$$L(x, \lambda, \nu) = f_0(x) + \sum_{i=1}^{m} \lambda_i f_i(x) + \sum_{i=1}^{p} \nu_i h_i(x) \tag{2.2}$$

With $L : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^p \to \mathbb{R}$ and $\mathbf{dom}\, L = \mathcal{D} \times \mathbb{R}^m \times \mathbb{R}^p$.

This function is a weighted sum of the objective and constraint functions. λ_i are the Lagrange multipliers associated with $f_i(x) \leq 0$ and ν_i are the Lagrange multipliers associated with $h_i(x) = 0$.

This Lagrangian can be extended to a Lagrange dual function $g : \mathbb{R}^m \times \mathbb{R}^p \to \mathbb{R}$:

$$g(\lambda, \nu) = \inf_{x \in \mathcal{D}} L(x, \lambda, \nu) \tag{2.3}$$

This Lagrange dual function can provide a lower bound to the base (primal) problem. If $\lambda \succeq 0$: $(\lambda, \nu) \in \mathbf{dom}\, g$, then $g(\lambda, \nu) \leq p^\star$.

A new dual problem can thus be formulated:

$$\begin{aligned} &\max_{(\lambda,\nu)} && g(\lambda, \nu) && (2.4)\\ &\text{subject to} && \lambda \succeq 0 && (2.4a) \end{aligned}$$

The purpose of this problem is to find the highest lower bound on $p^\star$ from the Lagrange dual function: in other words, the point of g in its domain that approaches the most $p^\star$. This is a concave optimization problem and its optimal value is denoted $d^\star$.

This dual problem can prove to be extremely useful when strong duality holds. This means that the lower bound is exact: $p^\star = d^\star$, thus, the optimal value of the dual problem is the same as the primal one. To ensure strong duality, additional constraints must be met, for example, Slater's condition.

When strong duality holds, another interesting property occurs: complementary slackness. Assuming $x^\star$ is primal optimal and $(\lambda^\star, \nu^\star)$ are dual optimal: $f_0(x^\star) = g(\lambda^\star, \nu^\star)$. Then $x^\star$ minimizes $L(x, \lambda^\star, \nu^\star)$.

$$
\begin{aligned}
f_0(x^\star) = g(\lambda^\star, \nu^\star) = \inf_x \left(f_0(x) + \sum_{i=1}^{m} \lambda_i^\star f_i(x) + \sum_{i=1}^{p} \nu_i^\star h_i(x) \right) & \\
\leq f_0(x^\star) + \sum_{i=1}^{m} \lambda_i^\star f_i(x^\star) + \sum_{i=1}^{p} \nu_i^\star h_i(x^\star) & \leq f_0(x^\star)
\end{aligned}
\tag{2.5}
$$

With strong duality, these inequalities hold with equality. Hence, the two sums are nullified. For $\sum_{i=1}^{p} \nu_i^\star h_i(x^\star)$, this is already set by the constraints $h_i(x \in \mathcal{D}) = 0$ whereas, for the first sum this brings a new criterion called complementary slackness:

$$
\begin{aligned}
\lambda_i^\star f_i(x\star) = 0 \text{ for } i = 1, ..., m \qquad \Longrightarrow \qquad & \lambda_i^\star > 0 \Rightarrow f_i(x^\star) = 0 \\
& f_i(x^\star) < 0 \Rightarrow \lambda_i^\star = 0
\end{aligned}
\tag{2.6}
$$

2.1.2 Karush-Kuhn-Tucker conditions

Karush-Kuhn-Tucker (KKT) conditions apply to optimization problems whose functions f_i and h_i are differentiable. They were presented by Kuhn and Tucker in 1951, and retroactively, scholars realized that they had already been demonstrated by Karush in his master's thesis in 1939.

KKT conditions are necessary conditions for the optimality of a problem. In the case of a convex problem, they are also sufficient and can be used in the optimum research.

For a standardized problem of the form 2.1, the KKT conditions are formulated:

- Primal feasibility: $f_i(x) \leq 0, \quad i = 1, ..., m \quad ; \quad h_i(x) = 0, \quad i = 1, ..., p$
- Dual feasibility: $\lambda \succeq 0$
- Complementary slackness: $\lambda_i f_i(x) = 0, \quad i = 1, ..., m$
- The gradient of Lagrangian with respect to x vanishes:

$$
\nabla f_0(x) + \sum_{i=1}^{m} \lambda_i \nabla f_i(x) + \sum_{i=1}^{p} \nu_i \nabla h_i(x) = 0
$$

Those conditions prove to be especially useful in convex and differentiable optimization. In this case, finding a point satisfying them is equivalent to finding the optimum of the problem. In these cases, the problems can sometimes be simplified into an unconstrained optimization algorithm, which can easily be solved iteratively.

2.1.3 Unconstrained optimization problems

With an unconstrained problem of the form:

$$\min_{x} \qquad f(x) \tag{2.7}$$

PROBLEM (2.4) can be rewritten to take the form of a general unconstrained Lagrangian problem:

$$\max_{\nu} \qquad \min_{x} \qquad L(x, \nu) \tag{2.8}$$

In the case of a convex optimization with strong duality, this unconstrained problem is equivalent to the initial one. So the methods described hereunder are relevant for the optimization problem that will be solved in the scope of this thesis.

Unconstrained optimization usually starts from a point in the domain of the function to optimize. Iteratively, this point will move to get closer to the optimum. At each iteration step, a descent direction as well as a step size have to be found. To do that efficiently, one needs to find a coherent direction first and then an appropriate step size using that direction.

$$x^{+} = x + t\Delta x \tag{2.9}$$

2.1.3.1 Search direction methods

The most trivial direction and the first to be used for the update vector (Δx) is the direction of the gradient. This straightforward method is called "gradient descent" in case of the search of a minimum, and "gradient ascent" in the opposite case. For the sake of simplicity and considering the standard form of problems written with minimization objectives, this description will be about the gradient descent. However, the following development can be used in the opposite direction for gradient ascent.

$$\Delta x = \nabla f(x) \tag{2.10}$$

Gradient descent method is very simple but can be slow in some cases.

2.1.3.2 Line search methods

Once the search direction Δx has been found, the most exact value for the step length is $t = argmin_{t=0} f(x + t\Delta x)$. However, finding such a value is not always trivial. Usually, finding a correct step length with another method such as backtracking line search also shows good results.

Backtracking line search is also an iterative method using two parameters $\alpha \in (0, 1/2)$

and $\beta \in (0, 1)$. It starts with $t = 1$ and t is updated with β by $t := \beta t$ until:

$$f(x + t\Delta x) < f(x) + \alpha t \nabla f(x)^T \Delta x \tag{2.11}$$

Usual values for α and β are 0.1 and 0.5 respectively.

2.2 Optimal power flow in DC

Using a regular power flow model combined with an objective function (usually a cost function to minimize), an optimization problem can be created. This is the optimal power flow problem.

This SECTION presents some mathematical models that are efficient to compute the optimal power flow in DC grids. This subject has been extensively studied for AC grids, due to the nature of the transmission and distribution grids globally. As presented in SECTION 1.1.3, the study field of DC microgrids is more recent and less mature.

2.2.1 Power flow model

Before finding the optimal operation point of the network, a basic power flow model will be presented. The goal of this model is to find a feasible operation point, without considering the objective (or cost) function and without optimizing.

The DC network is primarily composed of buses and lines. The whole system is modeled as a connected graph $\mathcal{G} := (\mathcal{N}, \mathcal{E})$, where $\mathcal{N} = 1, ..., n$ is the set of nodes representing the n buses and $\mathcal{E} \subseteq \mathcal{N} \times \mathcal{N}$ is the set of lines. If two buses i and k are connected: $(i, k) \in \mathcal{E}$ and this is denoted $i \sim k$.

This network is subject to physical laws and constraints which should also be considered and modeled.

2.2.1.1 Parameters and variables

For each bus $i \in \mathcal{N}$:

- Bus voltage: V_i;
- Current injection: I_i;
- Power injection: p_i.

For each line $i \sim j$:

- Line admittance: y_{ik};
- Line impedance: $z_{ik} := 1/y_{ik}$;
- Line current: I_{ik}.

2.2.1.2 Constraints

Physical laws:

- Ohm's law: $I_{ik} = y_{ik}(V_i - V_k)$ for $(i,k) \in \mathcal{E}$;
- Current balance: $I_i = \sum_{k:k\in\mathcal{N}_i} I_{ik}$, for $i \in \mathcal{N}$;
- Power balance: $p_i = V_i I_i$, for $i \in \mathcal{N}$.

Device constraints:

- Power injection constraint on each bus: $\underline{p}_i \leq p_i \leq \overline{p}_i$;
- Voltage constraint on each bus: $\underline{V}_i \leq V_i \leq \overline{V}_i$;
- Current constraint on each line: $I_{ik} \leq \overline{I}_{ik}$

All of these constraints are not directly used in practice. Some are modified because the current injection at each bus is not a relevant measure. The value that matters here is the power injection, which is directly linked to energy production and consumption. The power injection is a function of both current and voltage. Therefore, the current injection variables in the hereabove constraints are modified. The new constraints are:

$$p_i = V_i \sum_{k:k\in\mathcal{N}_i} (V_i - V_k) y_{ik} \tag{2.12}$$

2.2.2 Optimal power flow model

From the previous variables and EQUATION (2.12), an optimization problem can be described. But it needs an objective first. This function is represented by a cost to minimize:

$$C(p) = \sum_{i\in\mathcal{N}} f_i(p_i) \tag{2.13}$$

This cost function is usually determined as a function of the power injected at each bus. Two simplistic approaches would be either to model the monetary generation cost at each bus or simply use the losses in the whole system by using the sum of generated powers as a cost function.

To ease further computation, the lines power rating will not be taken into account in this work.

From the system described in SECTION 2.2.1, another constraint is added to have a voltage reference on one of the buses:

$$V_0 = V_{ref} \tag{2.14}$$

The basic `OPF` in DC can thus be written:

$$
\begin{aligned}
\min_{p,V} \quad & \sum_{i\in\mathcal{N}} f_i(p_i) && && (2.15)\\
\text{subject to} \quad & p_i = V_i \textstyle\sum_{k:k\in\mathcal{N}_i}(V_i - V_k)y_{ik} && i\in\mathcal{N} && (2.12)\\
& p_i \in \mathcal{P}_i && i\in\mathcal{N}^+ && (2.15a)\\
& V_0 = V_0^{ref} && && (2.14)\\
& \underline{V}_i \le V_i \le \overline{V}_i && i\in\mathcal{N}^+ && (2.15b)
\end{aligned}
$$

The power injection constrained is modified to take into account generation limits as well as load constraints at consumption buses and becomes EQUATION (2.15a) which represent a nonconvex set. The cost function can also sometimes be nonconvex.

2.2.3 SOCP relaxation

In this SECTION, both EQUATION (2.15) and EQUATION (2.15a) are assumed to be convex.

In PROBLEM (2.15), EQUATION (2.12) is nonconvex. This can be problematic in the solving, as the reach for global optimum is not ensured for nonconvex problems. Fortunately, [50] also proposes a convex second-order cone programming (`SOCP`) relaxation of this problem.

This is achieved by successively replacing EQUATION (2.12) to a positive semidefinite constraint and a rank constraint, then removing the rank constraint. First, two slack variables must be defined:

$$v_i = V_i^2, \quad i\in\mathcal{N}; \tag{2.16a}$$

$$W_{ik} = V_i V_k, \quad i\sim k. \tag{2.16b}$$

The operator $i\sim k$ indicates that i and k are linked by a line in the graph of the network and that $i<k$.

Using these variables EQUATION (2.12) is transformed:

$$p_i = \sum_{k:k\in\mathcal{N}_i} (v_i - W_{ik})y_{ik}, \quad i\in\mathcal{N} \tag{2.17}$$

To represent these relations while keeping the convexity of the problem, matrix R_{ik} is introduced:

$$R_{ik} = \begin{bmatrix} v_i & W_{ik} \\ W_{ki} & v_k \end{bmatrix} = \begin{bmatrix} V_i \\ V_k \end{bmatrix} \begin{bmatrix} V_i & V_k \end{bmatrix} \tag{2.18}$$

This matrix is used to replace the constraints EQUATION (2.16a) and EQUATION (2.16b)

by the rank constraint (2.19d). For the new problem to be equivalent to PROBLEM (2.15), a positive semidefinite constraint and a symmetry constraint must also be added to R_{ik}.

$$\begin{aligned}
\min_{p,v,W} \quad & \sum_{i\in\mathcal{N}} f_i(p_i) && && (2.19)\\
\text{subject to} \quad & p_i = \textstyle\sum_{k:k\in\mathcal{N}_i}(v_i - W_{ik})y_{ik} && i\in\mathcal{N} && (2.17)\\
& p_i \in \mathcal{P}_i && i\in\mathcal{N}^+ && (2.19a)\\
& v_0 = [V_0^{ref}]^2 && && (2.19b)\\
& \underline{V_i}^2 \le v_i \le \overline{V}_i^2 && i\in\mathcal{N}^+ && (2.19c)\\
& W_{ik} = W_{ki} && i\sim k && (2.19d)\\
& R_{ik} \succeq 0 && i\sim k && (2.19e)\\
& rank(R_{ik}) = 1 && i\sim k && (2.19f)
\end{aligned}$$

These modifications are equivalent to the base problem under the assumptions that $v_i > 0$ $\forall i \in \mathcal{N}$. This is not a very hazardous assumption, as voltage magnitudes often have boundary constraints around 1 pu. It also means that $W_{ik} \in \mathbb{R}$ $\forall i \sim k$. In this problem, only the constraint (2.19f) is nonconvex. Exact SOCP relaxation can be obtained by removing this constraint to achieve convexity provided that [50]:

- The upper bound constraints on voltages are non-binding;
- f_0 is strictly increasing.

2.2.4 ESOCP problem

In [45] another base approach is considered using real resistance $r_{ik} = 1/y_{ik}$, power consumption and generation at each bus (p_i^d, p_i^g) separately and droop control on each bus of the network instead of the voltage reference from EQUATION (2.14):

$$v_i - v_i^* = -k_i(p_i^g - \hat{p}_i) \tag{2.20}$$

$$\begin{aligned}
\min_{p^g,P,l,v,\hat{p}} \quad & \sum_{i\in\mathcal{N}} f_i(p_i^g) && && (2.21)\\
\text{subject to} \quad & v_i - v_i^* = k_i(p_i^g - \hat{p}_i) && i\in\mathcal{N} && (2.20)\\
& \underline{p_i^g} \le p_i^g \le \overline{p_i^g} && i\in\mathcal{N} && (2.21a)\\
& \underline{V_i}^2 \le v_i \le \overline{V_i}^2 && i\in\mathcal{N} && (2.21b)\\
& p_i^g - p_i^d = \textstyle\sum_{k:k\in\mathcal{N}_i}(v_i - W_{ik})/r_{ik} && i\in\mathcal{N} && (2.21c)\\
& W_{ik} \le 0 && i\sim k && (2.21d)\\
& W_{ik} = W_{ki} && i\sim k && (2.21e)\\
& R_{ik} \succeq 0 && i\sim k && (2.21f)
\end{aligned}$$

Using a new variable $l_{ik} = |I_{ik}|^2$, an equivalent problem with increased numerical stability

can be stated. Parameters y_i and z_i are also added for this problem to be more stable and time efficient. They only change the objective function by being "penalty terms". When the associated constraints are met, they are null, but while they are not, they add up to the objective function. When the gradient descent is conducted, the direction is thus chosen to also decrease this part of the sum. Hence, this improves the convergence speed.

This new `ESOCP` can be linked to PROBLEM (2.21) through:

$$P_{ik} = (v_i - W_{ik})/r_{ik}, \quad i \sim k; \tag{2.22a}$$

$$l_{ik} = (v_i - W_{ik} - W_{ki} + v_k)/r_{ik}^2, \quad i \sim k. \tag{2.22b}$$

With all aforementioned modifications, the `OPF` problem has become:

$$\begin{aligned}
\min_{p^g,P,l,v,\hat{p}} \quad & \sum_{i\in\mathcal{N}} f_i(p_i^g) + \sum_{i\in\mathcal{N}} \frac{1}{2} y_i^2 + \sum_{i\in\mathcal{N}} \frac{1}{2} z_i^2 && && (2.23)\\
\text{subject to} \quad & v_i - v_i^* = -k_i(p_i^g - \hat{p}_i) && i \in \mathcal{N} && (2.23a)\\
& p_i^g - p_i^d = \textstyle\sum_{k:k\in\mathcal{N}_i} P_{ik} && i \in \mathcal{N} && (2.23b)\\
& \underline{p_i^g} \le p_i^g \le \overline{p_i^g} && i \in \mathcal{N} && (2.23c)\\
& \underline{V_i}^2 \le v_i \le \overline{V_i}^2 && i \in \mathcal{N} && (2.23d)\\
& P_{ik} + P_{ki} = r_{ik} l_{ik} && i \sim k && (2.23e)\\
& v_i - v_k = r_{ik}(P_{ik} - P_{ki}) && i \sim k && (2.23f)\\
& l_{ik} \ge P_{ik}^2/v_i. && i \sim k && (2.23g)\\
& l_{ik} \ge P_{ki}^2/v_k. && i \sim k && (2.23h)\\
\text{with} \quad & y_i = v_i - v_i^* + k_i(p_i^g - \hat{p}_i) && i \in \mathcal{N} && (2.23i)\\
& z_i = p_i^g - p_i^d - \textstyle\sum_{k:k\in\mathcal{N}_i} P_{ik} && i \in \mathcal{N} && (2.23j)
\end{aligned}$$

PROBLEM (2.23) is the one which will be developed and used in practice to compute the optimal control references for the system. This problem is very versatile, as it implements a droop controller for each bus, but also computes their optimal voltage and power values. It means that each bus is provided with an optimal value for each of the three main control parameters introduced in SECTION 1.2.

In practice, the controller will be able to set a reference for each bus, whichever its control parameter is, by simply ignoring the other two.

2.2.4.1 Dual problem

This problem is further developed using the Lagrange dual function with Lagrange multipliers. This is done as a first step to find the dual function and the equivalent unconstrained problem.

Lagrange multiplier:	Constraint:	Associated with:
ϵ	Droop control	EQUATION (2.23a)
μ	Buses power balance	EQUATION (2.23b)
λ	Lines power to current balance	EQUATION (2.23e)
γ	Lines power to voltage balance	EQUATION (2.23f)
$\rho \geq 0$	SOCP relaxation	EQUATION (2.23g)

Table 2.1: Lagrange multipliers and their associated constraints

The Lagrangian formulation of PROBLEM 2.23 can then be expressed:

$$\begin{aligned} L(p^g, v, \hat{p}, P, \epsilon, \mu, , \gamma, \rho) = & \textstyle\sum_{i\in\mathcal{N}} f_i(p_i^g) + \sum_{i\in\mathcal{N}} \frac{1}{2} y_i^2 + \sum_{i\in\mathcal{N}} \frac{1}{2} z_i^2 \\ & + \textstyle\sum_{i\in\mathcal{N}} \epsilon_i (v_i - v_i^* + k_i(p_i^g - \hat{p}_i)) \\ & + \textstyle\sum_{i\in\mathcal{N}} \mu_i (p_i^g - p_i^d - \sum_{k:k\in\mathcal{N}_i} P_{ik}) \\ & + \textstyle\sum_{(i,k)\in\mathcal{E}} \lambda_{ik} (P_{ik} + P_{ki} - r_{ik} l_{ik}) \\ & + \textstyle\sum_{(i,k)\in\mathcal{E}} \gamma_{ik} (v_i - v_k - r_{ik}(P_{ik} - P_{ki})) \\ & + \textstyle\sum_{(i,k)\in\mathcal{E}} \rho_{ik} (P_{ik}^2 / v_i - l_{ik}) \\ & + \textstyle\sum_{(i,k)\in\mathcal{E}} \rho_{ki} (P_{ki}^2 / v_k - l_{ik}) \end{aligned} \tag{2.24}$$

Thus, the gradient of the Lagragian from EQUATION (2.24) which will be used for the unconstrained problem is:

$$\frac{\partial L}{\partial p_i^g} = \frac{\partial f_i(p_i^g)}{\partial p_i^g} + k_i y_i + z_i + \epsilon_i k_i + \mu_i \qquad \forall i \in \mathcal{N} \tag{2.25a}$$

$$\frac{\partial L}{\partial v_i} = y_i + \epsilon_i + \sum_{k:k\in\mathcal{N}_i} \gamma_{ik} - \sum_{k:k\in\mathcal{N}_i} \rho_{ik} \left(\frac{P_{ik}}{v_i}\right)^2 \forall i \in \mathcal{N} \tag{2.25b}$$

$$\frac{\partial L}{\partial P_{ik}} = -z_i - \mu_i + \lambda_{ik} - r_{ik}\gamma_{ik} + 2\rho_{ik}\frac{P_{ik}}{v_i} \qquad \forall (i,k) \in \mathcal{E} \tag{2.25c}$$

$$\frac{\partial L}{\partial l_{ik}} = -r_{ik}\lambda_{ik} - \rho_{ik} - \rho_{ki} \qquad \forall (i,k) \in \mathcal{E} \tag{2.25d}$$

$$\frac{\partial L}{\partial \hat{p}_i} = -k_i(y_i + \epsilon_i) \qquad \forall i \in \mathcal{N} \tag{2.25e}$$

$$\frac{\partial L}{\partial \mu_i} = p_i^g - p_i^d - \sum_{k:k\in\mathcal{N}_i} P_{ik} \qquad \forall i \in \mathcal{N} \tag{2.25f}$$

$$\frac{\partial L}{\partial \epsilon_i} = v_i - v_i^* + k_i(p_i^g - \hat{p}_i) \qquad \forall i \in \mathcal{N} \tag{2.25g}$$

$$\frac{\partial L}{\partial \lambda_{ik}} = P_{ik} + P_{ki} - r_{ik} l_{ik} \qquad \forall (i,k) \in \mathcal{E} \tag{2.25h}$$

$$\frac{\partial L}{\partial \gamma_{ik}} = v_i - v_k - r_{ik}(P_{ik} - P_{ki}) \qquad \forall (i,k) \in \mathcal{E} \tag{2.25i}$$

$$\frac{\partial L}{\partial \rho_{ik}} = \frac{P_{ik}^2}{v_i} - l_{ik} \qquad \forall (i,k) \in \mathcal{E} \tag{2.25j}$$

$$\frac{\partial L}{\partial \rho_{ki}} = \frac{P_{ki}^2}{v_k} - l_{ik} \qquad \forall (i,k) \in \mathcal{E} \tag{2.25k}$$

EQUATIONS (2.25a) to (2.25e) represent the partial derivatives of the Lagrangian according to the primal variables and EQUATIONS (2.25f) to (2.25k) according to the dual variables.

To find the saddle point of the Lagrangian (*i.e.*, the point which maximizes the dual objective), while minimizing the primal one, an iterative method is implemented. The dual variables are updated in the direction which makes EQUATION (2.24) increase, while the primal variables are updated with an opposite strategy. This should be done while keeping the non-strict constraints in feasible range:

$$\Delta p_i^g = \min\left\{\overline{p}_i^g \,;\, \max\left\{\underline{p}_i^g \,;\, p_i^g - (G_i(p_i^g) + \mu_i + z_i + k_i(\epsilon_i + y_i))\right\}\right\} - p_i^g, \tag{2.26a}$$

$$\Delta v_i = \min\left\{\overline{V}_i^2 \,;\, \max\left\{\underline{V}_i^2 \,;\, v_i - \left(y_i + \epsilon_i + \textstyle\sum_{k:k\in\mathcal{N}_i} \gamma ik - \sum_{k:k\in\mathcal{N}_i} \rho_{ik}\left(\frac{P_{ik}}{v_i}\right)^2\right)\right\}\right\} - v_i, \tag{2.26b}$$

$$\Delta P_{ik} = z_i + \mu_i - \lambda_{ik} + r_{ik}\gamma_{ik} - 2\rho_{ik}\frac{P_{ik}}{v_i}, \tag{2.26c}$$

$$\Delta l_{ik} = r_{ik}\lambda_{ik} + \rho_{ik}, \tag{2.26d}$$

$$\Delta \hat{p}_i = k_i\Big(y_i + \epsilon_i\Big), \tag{2.26e}$$

$$\Delta \mu_i = p_i^g - p_i^d - \textstyle\sum_{k:k\in\mathcal{N}_i} P_{ik}, \tag{2.26f}$$

$$\Delta \epsilon_i = v_i - v_i^\star + k_i\Big(p_i^g - \hat{p}_i\Big), \tag{2.26g}$$

$$\Delta \lambda_{ik} = P_{ik} + P_{ki} - r_{ik}l_{ik}, \tag{2.26h}$$

$$\Delta \gamma_{ik} = v_i - v_k - r_{ik}\Big(P_{ik} - P_{ki}\Big), \tag{2.26i}$$

$$\Delta \rho_{ik} = \max\left\{-\rho_{ik} \,;\, \tfrac{P_{ik}^2}{v_i} - l_{ik}\right\}. \tag{2.26j}$$

It should be noted that there is no update function for y_i and z_i because they are not variables of the problems. They are simplified notations for their associated function (EQUATIONS (2.23i) and (2.23j)). The values for y_i and z_i must still be updated at each iteration with the new values of the other variables.

Chapter 3

Distributed Problem

This CHAPTER tries to distribute the previously introduced DC microgrid optimal power flow problem from EQUATION (2.23). The first model used is a 6 bus meshed network, the second one is extremely simplified, with only three buses and two lines.

This CHAPTER shows the result of the distributed algorithm for the two networks. For each of those, it first introduces the distribution of the algorithm and the update formulae for the primal and dual variables in each bus.

3.1 6 bus network

The distribution of the `OPF` problem is a lot more complex. Before tackling the dynamic problem of real-time control, a simpler implementation of the problem has been realized for a single time step. This first phase is especially useful to determine how the algorithm converges and what are its limits from a static point of view.

In this SECTION, the test network and its parameters are introduced first in SECTION 3.1.1, then an optimal solution is shown from a centralized version of the algorithm in SECTION 3.1.2 and finally the distributed algorithm and its results are presented in SECTION 3.1.3.

3.1.1 Network model

The network used for this implementation is a 6 bus network. It is operated at a nominal voltage of 230 V. It is an adapted version of the `case 6ww` power system network of `PandaPower` [51], which is taken from A. J. Wood and B. F. Wollenberg, *Power generation, operation, and control* [52].

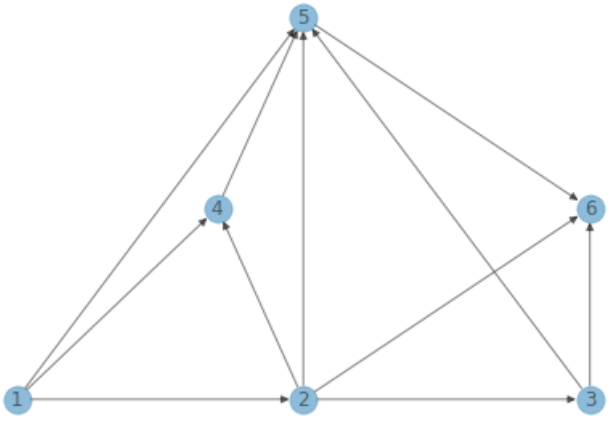


Figure 3.1: Buses and lines of the network.

From `Pandapower`'s network, reactive powers and reactances have been deleted. The values are also adapted to a 230 V network.

Line:	(1,2)	(1,4)	(1,5)	(2,3)	(2,4)	(2,5)	(2,6)	(3,5)	(3,6)	(4,5)	(5,6)
R (pu)	0.1	0.05	0.08	0.05	0.05	0.1	0.07	0.12	0.02	0.2	0.1

Table 3.1: Line resistances in network 3.1

In the algorithm, each bus is considered as a generator. For load buses, generation limits have simply been both set to 0 W. In this test case, buses 1 to 3 act as generators and buses 4 to 6 as loads.

3.1.1.1 Parameters of the generators

After the network and the lines, the buses or generators also need to get parametrized.

	p_d [W]	$\underline{p}$ [W]	$\overline{p}$ [W]	$\underline{V}$ [V]	$\overline{V}$ [V]	k [Ω]	v^* [V^2]	c_1	c_2	c_3
Generator 1	0	50.0	200.0	218.5	241.5	0.1	230.0^2	213.1	11.669	$5.33e{-3}$
Generator 2	0	37.5	170.0	218.5	241.5	0.1	230.0^2	200.0	10.333	$8.89e{-3}$
Generator 3	0	45.0	180.0	218.5	241.5	0.1	230.0^2	240	10.833	$7.41e{-3}$
Generator 4	70.0	0	0	218.5	241.5	0.1	230.0^2	0.0	0.0	0.0
Generator 5	70.0	0	0	218.5	241.5	0.1	230.0^2	0.0	0.0	0.0
Generator 6	70.0	0	0	218.5	241.5	0.1	230.0^2	0.0	0.0	0.0

Table 3.2: Parameters of the generators

3.1.1.2 Costs

The objective function models an energy production cost. This cost is different for each generator and is represented by a second-degree polynomial, the variable is the produced power of the generator.

$$C_i = c_{i,1} + c_{i,2} \cdot P_i^g + c_{i,3} \cdot (P_i^g)^2 \tag{3.1}$$

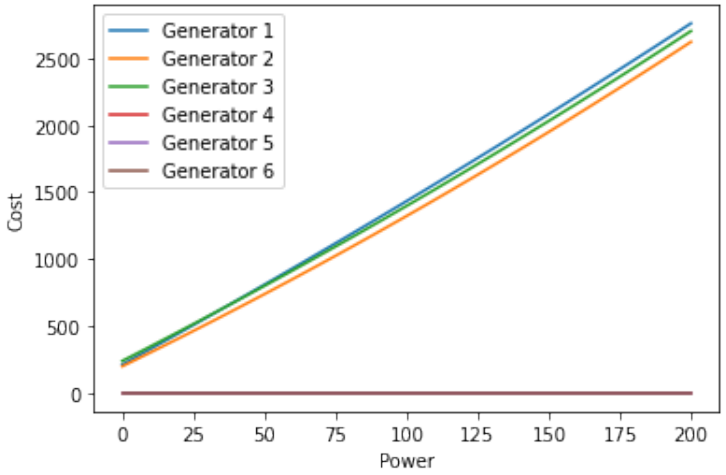


Figure 3.2: Costs functions for generators 1 to 3.

In practice, to obtain faster convergence, the objective function f is a slightly modified version of the initial cost function as explained in SECTION 2:

$$f = \sum_{i \in \mathcal{N}} f_i = \sum_{i \in \mathcal{N}} (C_i + 0.5 y_i^2 + 0.5 z_i^2) \tag{3.2}$$

With $y_i = v_i + k_i p_i^g - v_i^\star - k_i \hat{p}_i$ and $z_i = p_i^g - p_i^d - \sum_{k:k \in \mathcal{N}_i} P_{ik}$. Those two variables are equal to 0 when conditions (2.23a) and (2.23b) are met. Thus, they do not modify the optimal solution.

3.1.2 Non distributed optimal solution

The basic centralized PROBLEM (2.23) is implemented using `pyomo` [53], an open source `python` package used to formulate, model, solve and analyse optimization problems. It is solved with `Ipopt` [54], an open source solver for nonlinear optimization problems.

For this first implementation, the model is simply built with constraints on all the buses and edges. Every node of the network is updated simultaneously and gets updated measures for the whole system.

The optimal solution found using this centralized algorithm has a total cost of 3088.24 and losses of 3.43 W. The following SECTIONS will present the optimal values for both primal and dual variables.

3.1.2.1 Primal variables values

The primal variables of the problem are:

- p_i^g, the power generated [W] for each bus $i \in \mathcal{N}$;
- v_i, the squared voltage [V^2] for each bus $i \in \mathcal{N}$;
- P_{ik}, the power from bus i flowing in line (i,k) [W], for both ends of each line $(i,k) \in \mathcal{E}$, in both directions;
- l_{ik}, the squared current in line (i,k) [I^2] for each line $(i,k) \in \mathcal{E}$.

TABLE 3.3 and FIGURE 3.3 and 3.4 show the optimal values for the primal parameters.

3.1.2.1.a Bus variables: $(p_i^g, V_i, \hat{p}_i)$

	p_g [W]	V [V]	$\hat{p}$ [W]
Generator 1	50	240.58	144.15
Generator 2	88.46	241.5	190.96
Generator 3	74.97	241.21	174.81
Generator 4	0	237.08	62.54
Generator 5	0	236.65	58.74
Generator 6	0	238.56	75.84

Table 3.3: Control variables of each bus

3.1.2.1.b Line powers: (P_{ik})

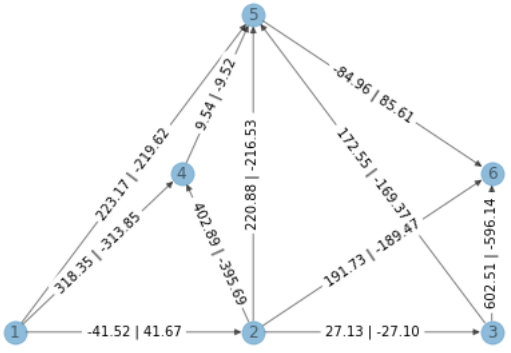


Figure 3.3: Line powers computed with `pyomo` [W]

3.1.2.1.c Line squared currents: (l_{ik})

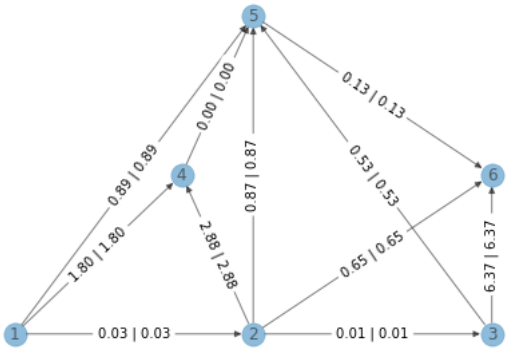


Figure 3.4: Squared line currents computed with `pyomo` $[I^2]$

3.1.3 Distributed algorithm

Using the results obtained in SECTION 3.1.2, the distributed algorithm is developed. It uses a modified version of the updates formulae from EQUATION (2.26), which corresponds to the algorithm used in [45].

$$\Delta p_i^g = \min\left\{\overline{p}_i^g \,;\, \max\left\{\underline{p}_i^g \,;\, p_i^g - (G_i(p_i^g) - \mu_i + z_i + k_i(\epsilon_i + y_i))\right\}\right\} - p_i^g, \tag{3.3a}$$

$$\Delta v_i = \min\left\{\overline{V}_i^2 \,;\, \max\left\{\underline{V}_i^2 \,;\, v_i - \left(y_i + \epsilon_i + \textstyle\sum_{k:k\in\mathcal{N}_i} \gamma ik - \sum_{k:k\in\mathcal{N}_i} \rho_{ik}\left(\frac{P_{ik}}{v_i}\right)^2\right)\right\}\right\} - v_i, \tag{3.3b}$$

$$\Delta P_{ik} = z_i - \mu_i - \lambda_{ik} + r_{ik}\gamma_{ik} - 2\rho_{ik}\frac{P_{ik}}{v_i}, \tag{3.3c}$$

$$\Delta l_{ik} = r_{ik}\lambda_{ik} + \rho_{ik} + \rho_{ki}, \tag{3.3d}$$

$$\Delta \hat{p}_i = k_i\Big(y_i + \epsilon_i\Big), \tag{3.3e}$$

$$\Delta \mu_i = -(p_i^g - p_i^d - \textstyle\sum_{k:k\in\mathcal{N}_i} P_{ik}), \tag{3.3f}$$

$$\Delta \epsilon_i = v_i - v_i^\star + k_i\Big(p_i^g - \hat{p}_i\Big), \tag{3.3g}$$

$$\Delta \lambda_{ik} = P_{ik} + P_{ki} - r_{ik}l_{ik}, \tag{3.3h}$$

$$\Delta \gamma_{ik} = v_i - v_k - r_{ik}\Big(P_{ik} - P_{ki}\Big), \tag{3.3i}$$

$$\Delta \rho_{ik} = \max\left\{-\rho_{ik} \,;\, \tfrac{P_{ik}^2}{v_i} - l_{ik}\right\}. \tag{3.3j}$$

In EQUATION (3.3), the formula for l_{ik} (EQUATION (3.3d)) has been modified to ensure symmetry of this variable in each line. The update function for μ (EQUATION (3.3f)) is also different, as it is the opposite of what was computed in SECTION 2. In the other update functions, the occurrences of μ are thus changed accordingly.

3.1.4 Distributed algorithm results

The distributed algorithm is implemented using a simple iterative method in `python`. The algorithm runs for $40,000$ iterations. The chosen step size is arbitrarily chosen to be 0.01 for the first $20,000$ and 0.001 for the remaining ones. It has been determined empirically that these values bring satisfactory results.

The step is not chosen using a line search method as introduced in SECTION 2.1.3.2. Indeed, the step size must be the same for each bus in order to obtain a convergence, due to the equality constraint on l_{ik} and l_{ki}, each update must be the same for both ends of each line (*i.e.*, each bus of the system by extension). Thus, this type of method would need a centralized controller to determine the ideal step size. This is totally contrasting with the goal of the method.

3.1.4.1 Primal variable values

Most primal values reach a point near the previously computed optimum. Except for the voltage instruction which abnormally tend to the lower bound instead of the upper one, and the droop power reference which vary accordingly. The current value in some of the lines is also oscillating, this is notably the case in lines {12}, {45} and {56}.

3.1.4.1.a Bus primal variables (p_i^g, v_i and $\hat{p}_i$)

	p^g [W]	V [V]	$\hat{p}$ [W]
Generator 1	50.0	221.68	−28.25
Generator 2	88.95	223.06	33.65
Generator 3	73.27	221.80	73.27
Generator 4	0	218.64	−89.35
Generator 5	0	218.5	−98.01
Generator 6	0	220.08	−103.99

Table 3.4: Control variables of each bus after 40,000 iterations

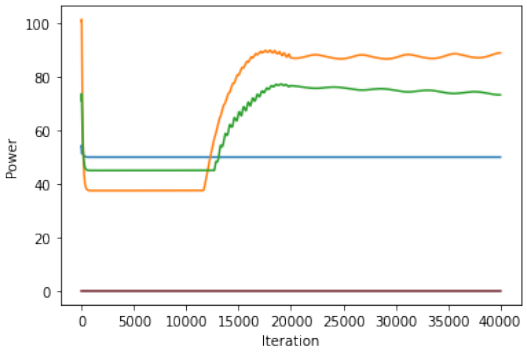


Figure 3.5: Evolution of p^g over 40,000 iterations

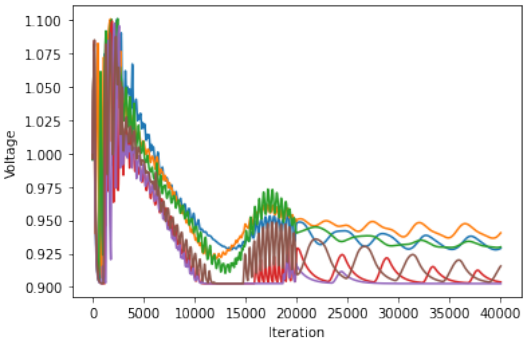


Figure 3.6: Evolution of V over 40,000 iterations

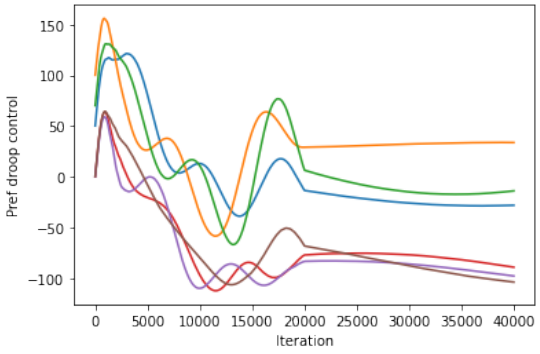


Figure 3.7: Evolution of $\hat{p}$ over 40,000 iterations

3.1.4.1.b Line powers (P_{ik})

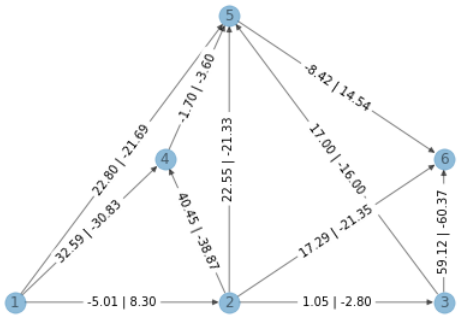
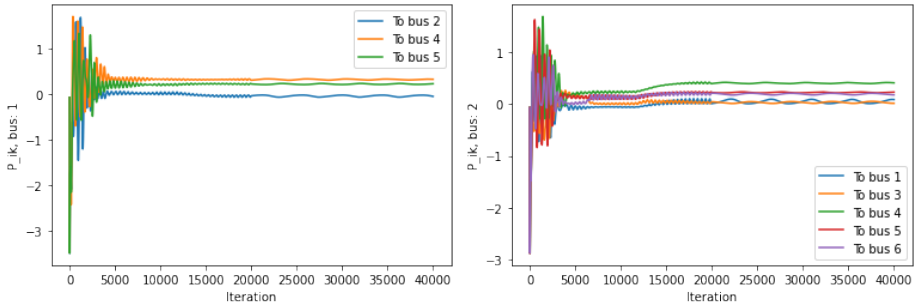
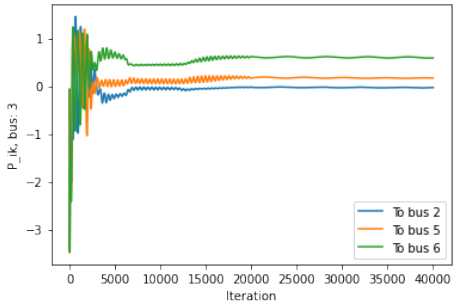


Figure 3.8: Line power variables after 40,000 iterations

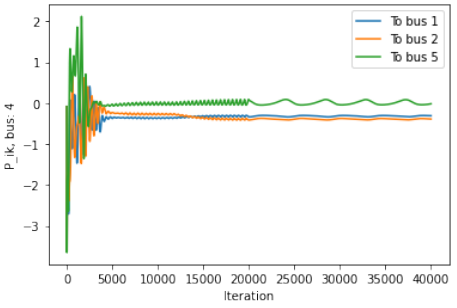


(a) Generator 1

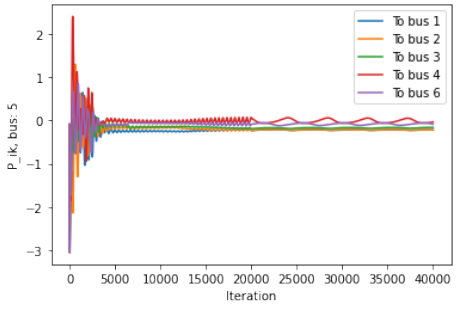
(b) Generator 2



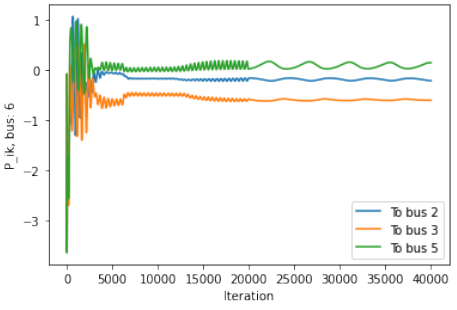
(c) Generator 3



(d) Generator 4



(e) Generator 5



(f) Generator 6

Figure 3.9: Evolution of the values of the line powers over 40,000 iterations

3.1.4.1.c Squared line currents (l_{ik})

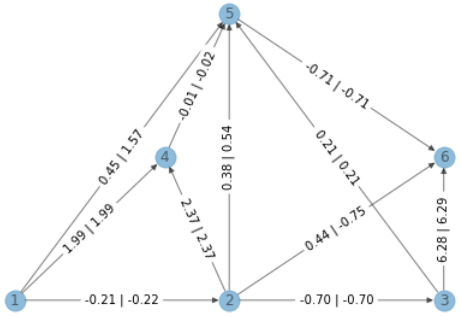
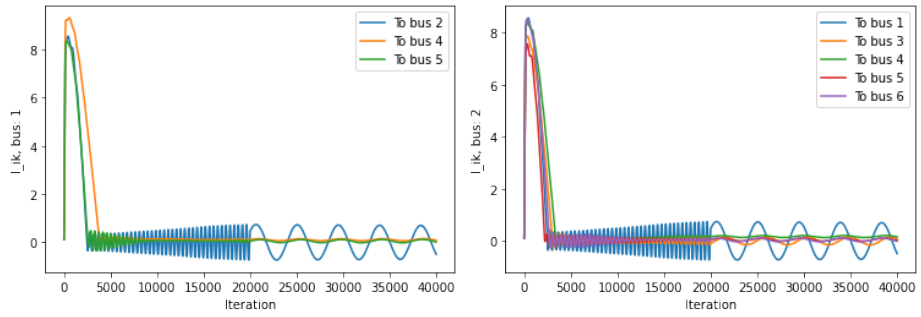
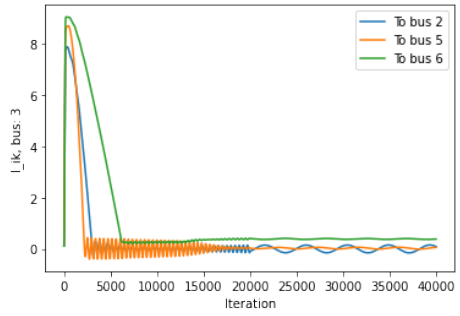


Figure 3.10: Squared line current variables after 100 000 iterations

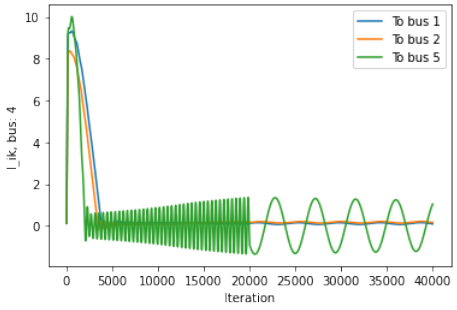


(a) Generator 1

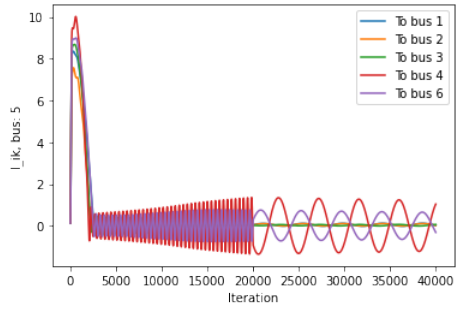
(b) Generator 2



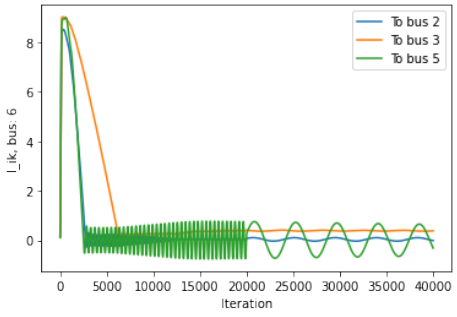
(c) Generator 3



(d) Generator 4



(e) Generator 5



(f) Generator 6

Figure 3.11: Evolution of the values of the squared currents over 40,000 iterations

3.1.4.2 Dual variable values

The convergence of the dual variables is similar to that of the primal variables. Some of the values converge smoothly (μ and λ), others show mitigated results (ϵ and γ), while one oscillates with a wide amplitude (ρ). This difference in convergence emanates directly from the different convergences of the primal variable used in the dual equations.

3.1.4.2.a Bus dual variables (ϵ_i and μ_i)

	ϵ	μ
Generator 1	12.01	0.02
Generator 2	11.91	−0.01
Generator 3	11.93	0.10
Generator 4	12.41	−0.15
Generator 5	12.47	−0.13
Generator 6	12.26	−0.12

Table 3.5: Dual variable values for each bus

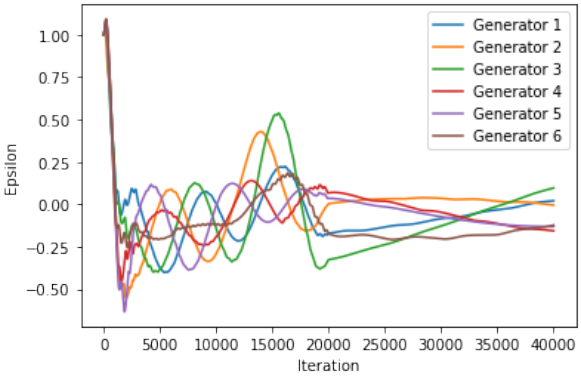


Figure 3.12: Evolution of the value of the dual variable ϵ

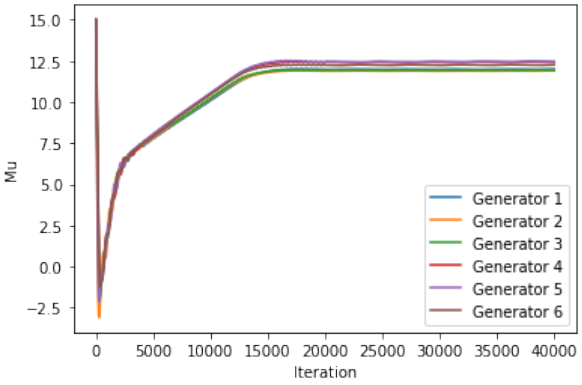


Figure 3.13: Evolution of the value of the dual variable μ

3.1.4.2.b Line power to current balance dual variables (λ_{ik})

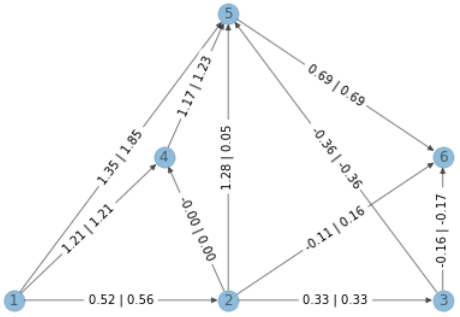
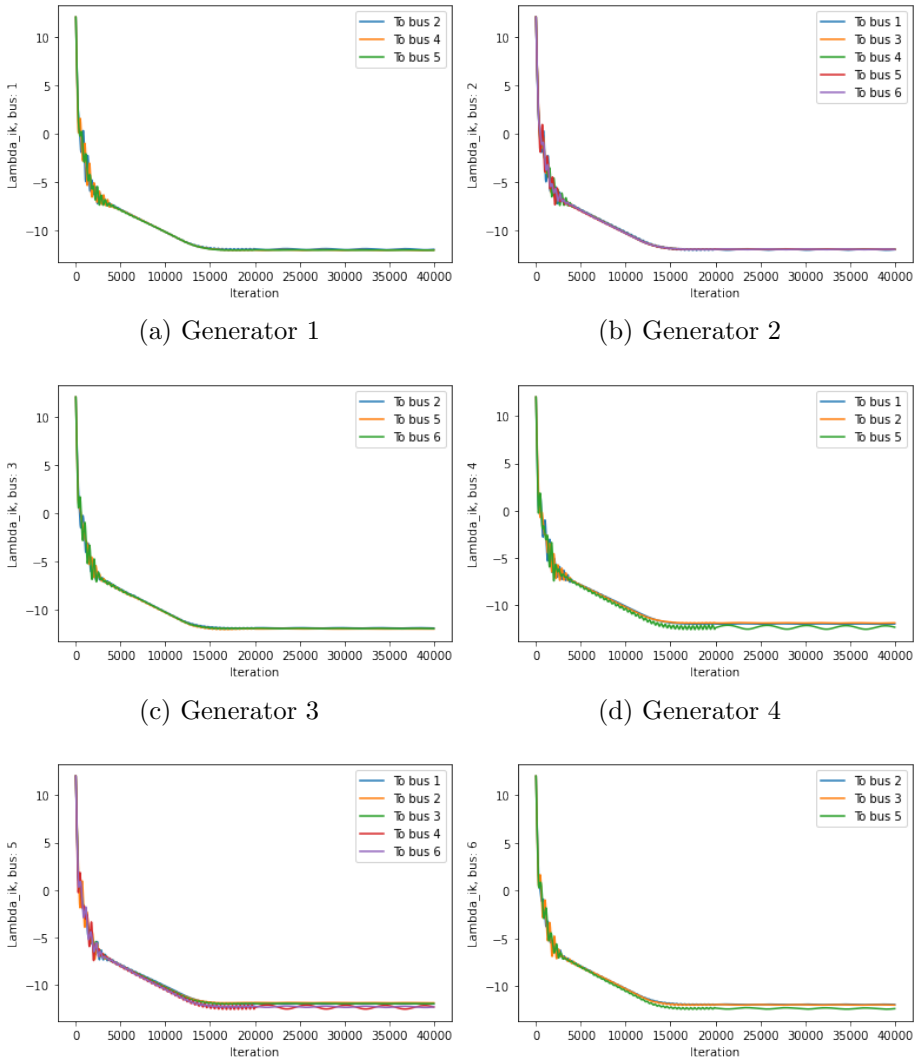


Figure 3.14: Line power to current balance dual variables (λ) after 40,000 iterations



(a) Generator 1

(b) Generator 2

(c) Generator 3

(d) Generator 4

(e) Generator 5

(f) Generator 6

Figure 3.15: Evolution of the values of λ dual variables over 40,000 iterations

3.1.4.2.c Line power to voltage balance dual variables (γ_{ik})

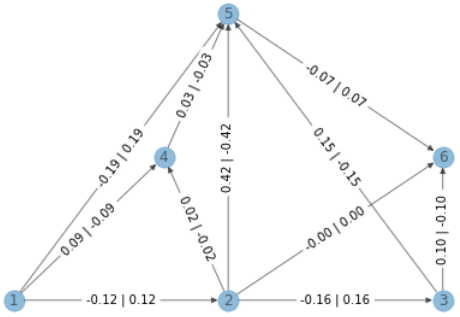
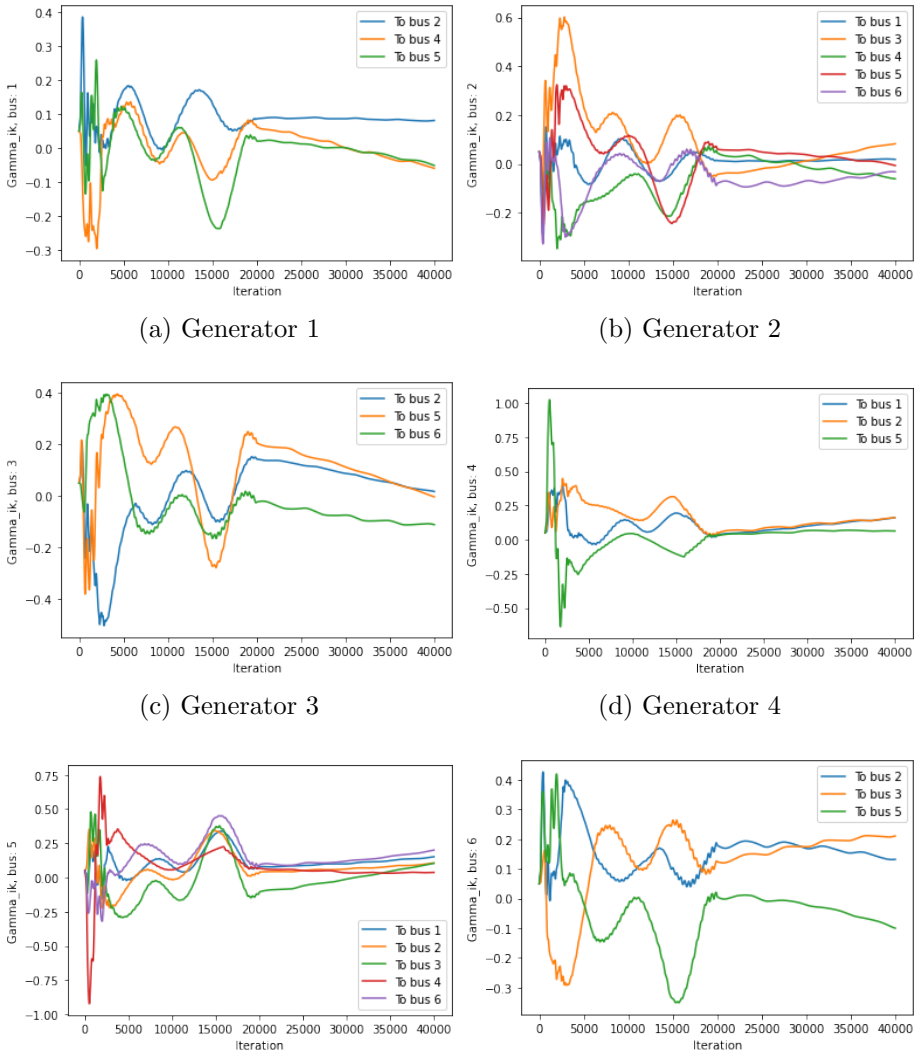


Figure 3.16: Line power to voltage balance dual variables (γ) after 40,000 iterations



(a) Generator 1

(b) Generator 2

(c) Generator 3

(d) Generator 4

(e) Generator 5

(f) Generator 6

Figure 3.17: Evolution of the values of γ dual variables over 40,000 iterations

3.1.4.2.d SOCP relaxation constraint dual variables (ρ_{ik})

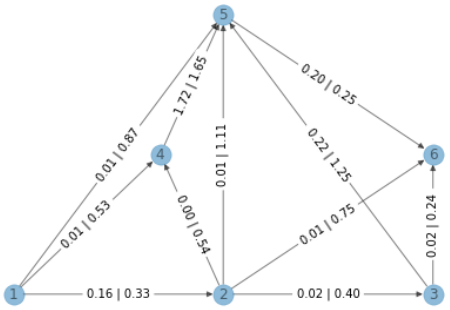
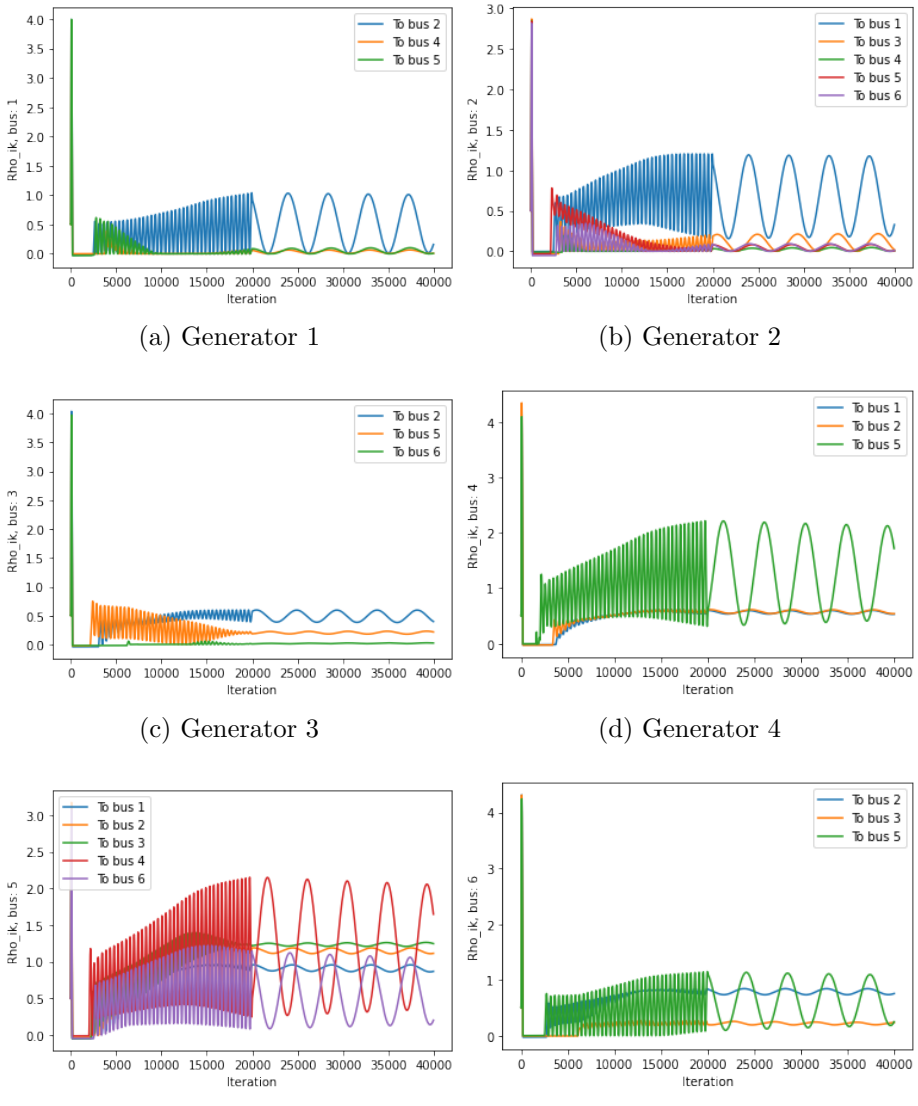


Figure 3.18: SOCP relaxation constraint dual variables (ρ) after 40,000 iterations



(a) Generator 1

(b) Generator 2

(c) Generator 3

(d) Generator 4

(e) Generator 5

(f) Generator 6

Figure 3.19: Evolution of the values of ρ dual variables over 40,000 iterations

3.2 3 bus system

A thorough investigation on the distributed 6 bus system, making sure that the algorithm and the methodology were correct was conducted. Although all computation formulae were accurate, the gradient descent of the optimization problem did not reach an equilibrium point. The scope of the implementation of this distributed controller must then be reduced.

Henceforth, a more simplistic network was to be considered. This network would only consist of three buses and two lines. Indeed, in the first network, the buses have many interconnections and the influence of each update is very strong on many other parameters in the numerous neighbors of each bus.

Such a simple network is purely experimental. In practice, this model would represent a simple line and would not constitute a microgrid. The following developments and results are mainly interesting to check the conditions of convergence and the efficiency of the method.

3.2.1 3 bus system topology

This network is not meshed, it corresponds to a tree with 3 nodes. Two lines are linking bus 1 to 2 and 2 to 3, with resistance values of r_{12} and r_{23}, respectively.

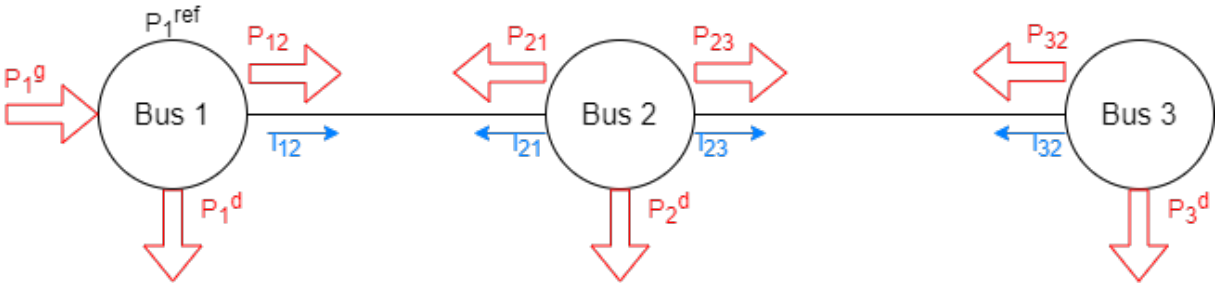


Figure 3.20: Topology of the 3 bus network

Bus 1 is the generator, it produces a power of P_1^g. The power exiting bus i to bus k is denoted P_{ik}, for each line in both directions. The same goes for l_{ik}. Bus 1, 2 and 3 all consume power: P_1^d, P_2^d and P_3^d, respectively.

3.2.2 System parameters and initial values

The convergence of the problem is highly dependent on the system's parameters, as well as the initial values of the variables. Slightly differing approaches would often bring diverging result. For example, the derivative of the cost function must be highly dependent on the variables, with an important slope. The step size which is used to compute the updates must also be small enough to avoid divergences and high enough to allow convergence to be as fast as possible.

3.2.2.1 Parameters

For this circuit the cost function models the generation costs at bus 1, the only bus which injects power in the system. The function is simply a quadratic function of the power generated:

$$C_1(P_1^g) = 0.5P_1^g + 0.02(P_1^g)^2 \tag{3.4}$$

The impedances in lines {12} and {23} are respectively 0.4 Ω and 0.6 Ω, as the network is DC, the lines are purely resistive.

Bus 1 is the only power source, it can provide the system with power values ranging from 0 to 110 W. It is also operated in droop control with a k factor of 1 and a voltage reference of 230 V.

The 3 buses consume power, respectively 20, 40 and 30 W. They have the same voltage boundaries, $V_{min} = 207$ V and $V_{max} = 253$ V, corresponding to $230V \pm 10\%$.

3.2.2.2 Initial values

v_1	v_2	v_3	P_1^g	P_1^{ref}	l_{12}	l_{23}	P_{12}	P_{21}	P_{23}	P_{32}
230	230	230	100	95	1.0	0.5	90	-90	60	-60

Table 3.6: Initial values of the primal variables

μ_1	μ_2	μ_3	ϵ_1	λ_{12}	λ_{23}	γ_{12}	γ_{23}	ρ_{12}	ρ_{21}	ρ_{23}	ρ_{32}
1.0	1.0	1.0	1.0	5.0	4.0	1.0	2.0	-1.0	-0.9	-0.8	-0.7

Table 3.7: Initial values of the dual variables

3.2.3 Optimal power flow problem

$$
\begin{aligned}
\min \quad & C_1(P_1^g) + \frac{z_1^2}{2} + \frac{z_2^2}{2} + \frac{z_3^2}{2} + \frac{y_1^2}{2} && && (3.5)\\
\text{subject to} \quad & v_1 - v_1^{ref} = -k_1(P_1^g - P_1^{ref}) && && (3.5a)\\
& P_i^g - P_i^d = \textstyle\sum_{k:k\in\mathcal{N}_i} P_{ik} && i \in \{1,2,3\} && (3.5b)\\
& \underline{P_1} \leq P_1^g \leq \overline{P_1} && && (3.5c)\\
& \underline{V_i}^2 \leq v_i \leq \overline{V_i}^2 && i \in \{1,2,3\} && (3.5d)\\
& P_{12} + P_{21} = r_{12} l_{23} && && (3.5e)\\
& P_{23} + P_{32} = r_{23} l_{23} && && (3.5f)\\
& v_1 - v_2 = r_{12}(P_{12} - P_{21}) && && (3.5g)\\
& v_2 - v_3 = r_{23}(P_{23} - P_{32}) && && (3.5h)\\
& l_{12} \geq P_{12}^2 / v_1. && && (3.5i)\\
& l_{12} \geq P_{21}^2 / v_2. && && (3.5j)\\
& l_{23} \geq P_{23}^2 / v_2. && && (3.5k)\\
& l_{23} \geq P_{32}^2 / v_3. && && (3.5l)\\
\text{with} \quad & y_1 = v_1 - v_1^{ref} + k_1(P_1^g - P_1^{ref}) && && (3.5m)\\
& z_i = p_i^g - p_i^d - \textstyle\sum_{k:k\in\mathcal{N}_i} P_{ik} && i \in \{1,2,3\} && (3.5n)
\end{aligned}
$$

For this simplified network, the Lagrangian can be fully developed while staying readable (*i.e.*, without the use of sum operators). This allows the distribution to be more specific for each bus.

By denoting the vector of all primal variables $\mathcal{P} = [P_1^g, P_1^{ref}, v_1, v_2, v_3, P_{12}, P_{21}, P_{23}, P_{32}, l_{12}, l_{23}]$ and the vector of all dual variables $\mathcal{D} = [\epsilon_1, \mu_1, \mu_2, \mu_3, \lambda_{12}, \lambda_{23}, \gamma_{12}, \gamma_{23}, \rho_{12}, \rho_{21}, \rho_{23}, \rho_{32}]$

$$
\begin{aligned}
L(\mathcal{P},\mathcal{D}) = {} & C_1(P_1^g) + \frac{z_1^2}{2} + \frac{z_2^2}{2} + \frac{z_3^2}{2} + \frac{y_1^2}{2} + \epsilon_1(v_1 - v_1^{ref} + k_1(P_1^g - P_1^{ref}))\\
& + \mu_1(P_1^g - P_1^d - P_{12}) + \mu_2(-P_2^d - P_{21} - P_{23}) + \mu_3(-P_3^d - P_{32})\\
& + \lambda_{12}(P_{12} + P_{21} - r_{12} l_{12}) + \lambda_{23}(P_{23} + P_{32} - r_{23} l_{23})\\
& + \gamma_{12}(v_1 - v_2 - r_{12}(P_{12} - P_{21})) + \gamma_{23}(v_2 - v_3 - r_{23}(P_{23} - P_{32}))\\
& + \rho_{12}(P_{12}^2/v_1 - l_{12}) + \rho_{21}(P_{21}^2/v_2 - l_{12}) + \rho_{23}(P_{23}^2/v_2 - l_{23}) + \rho_{32}(P_{32}^2/v_3 - l_{23})
\end{aligned} \tag{3.6}
$$

Once again, before tackling the gradient descent, the problem must be distributed. To do so, each bus keeps a local copy of necessary variables on both sides of each line. The said variables are l_{12}, λ_{12} and γ_{12} for line $\{12\}$ as well as l_{23}, λ_{23} and γ_{23} for line $\{23\}$. A copy (x_{ki}) of each variable (x_{ik}) is created and both values must stay equal at all time on both ends of the line.

For the λ and γ variables, the equality is ensured by the equivalence of the update formulae. In this case, a symmetrical initialization of the values is sufficient to comply with this constraint. For the l variables, a specific attention must be paid to stay coherent with the previous formulation. Indeed, the partial derivative according to l_{12} depends on both ρ_{12} and ρ_{21}, so does l_{21}. This means that the buses at both ends of this line must keep an updated copy of both values of ρ.

3.2.3.1 Gradient of the Lagrangian and update formulae

To obtain a distributed system, the update equations from the SECTION are separated according to the bus they refer to. An algorithm using the latter available in APPENDIX B then iterates until it reaches an equilibrium point for the gradient descent of the Lagrangian. This point is the optimum according to the KKT conditions presented in SECTION 2.1.2.

3.2.3.1.a Bus 1

$$
\begin{aligned}
\Delta P_1^g &= -\frac{\partial L}{\partial P_1^g} &&= -\left(\frac{\partial C_1(P_1^g)}{\partial P_1^g} + k_1 y_1 + z_1 + \epsilon_1 k_1 + \mu_1\right) \\
\Delta v_1 &= -\frac{\partial L}{\partial v_1} &&= -\left(y_1 + \epsilon_1 + \gamma_{12} - \rho_{12}\left(\frac{P_{12}}{v_1}\right)^2\right) \\
\Delta P_{12} &= -\frac{\partial L}{\partial P_{12}} &&= -\left(-z_1 - \mu_1 + \lambda_{12} - r_{12}\gamma_{12} + 2\rho_{12}\frac{P_{12}}{v_1}\right) \\
\Delta l_{12} &= -\frac{\partial L}{\partial l_{12}} &&= -\left(-r_{12}\lambda_{12} - \rho_{12} - \rho_{21}\right) \\
\Delta \epsilon_1 &= \frac{\partial L}{\partial \epsilon_1} &&= v_1 - v_1^{ref} + k_1(P_1^g - P_1^{ref}) \\
\Delta \mu_1 &= \frac{\partial L}{\partial \mu_1} &&= P_1^g - P_1^d - P_{12} \\
\Delta \lambda_{12} &= \frac{\partial L}{\partial \lambda_{12}} &&= P_{12} + P_{21} - r_{12} l_{12} \\
\Delta \gamma_{12} &= \frac{\partial L}{\partial \gamma_{12}} &&= v_1 - v_2 - r_{12}(P_{12} - P_{21}) \\
\Delta P_1^{ref} &= \frac{\partial L}{\partial P_1^{ref}} &&= -k_1(y_1 + \epsilon_1) \\
\Delta \rho_{12} &= \frac{\partial L}{\partial \rho_{12}} &&= l_{12} - \frac{P_{12}^2}{v_1}
\end{aligned}
$$

3.2.3.1.b Bus 2

$$
\begin{aligned}
\Delta v_2 &= -\frac{\partial L}{\partial v_2} &&= -\left(-\gamma_{21} + \gamma_{23} - \rho_{21}\left(\frac{P_{21}}{v_2}\right)^2 - \rho_{23}\left(\frac{P_{23}}{v_2}\right)^2\right)\\
\Delta P_{21} &= -\frac{\partial L}{\partial P_{21}} &&= -\left(-z_2 - \mu_2 + \lambda_{21} - r_{12}\gamma_{21} + 2\rho_{21}\frac{P_{21}}{v_2}\right)\\
\Delta P_{23} &= -\frac{\partial L}{\partial P_{23}} &&= -\left(-z_2 - \mu_2 + \lambda_{23} - r_{23}\gamma_{23} + 2\rho_{23}\frac{P_{23}}{v_2}\right)\\
\Delta l_{21} &= -\frac{\partial L}{\partial l_{21}} &&= -\left(-r_{12}\lambda_{21} + \rho_{21} + \rho_{12}\right)\\
\Delta l_{23} &= -\frac{\partial L}{\partial l_{23}} &&= -\left(-r_{23}\lambda_{23} + \rho_{23} + \rho_{32}\right)\\
\Delta \mu_2 &= \frac{\partial L}{\partial \mu_2} &&= -P_2^d - P_{21} - P_{23}\\
\Delta \lambda_{21} &= \frac{\partial L}{\partial \lambda_{21}} &&= P_{12} + P_{21} - r_{12}l_{21}\\
\Delta \lambda_{23} &= \frac{\partial L}{\partial \lambda_{23}} &&= P_{23} + P_{32} - r_{23}l_{32}\\
\Delta \gamma_{21} &= \frac{\partial L}{\partial \gamma_{21}} &&= v_1 - v_2 - r_{12}(P_{12} - P_{21})\\
\Delta \gamma_{23} &= \frac{\partial L}{\partial \gamma_{23}} &&= v_2 - v_3 - r_{23}(P_{23} - P_{32})\\
\Delta \rho_{21} &= \frac{\partial L}{\partial \rho_{21}} &&= l_{21} - \frac{P_{21}^2}{v_2}\\
\Delta \rho_{23} &= \frac{\partial L}{\partial \rho_{23}} &&= l_{23} - \frac{P_{23}^2}{v_2}
\end{aligned}
$$

3.2.3.1.c Bus 3

$$
\begin{aligned}
\Delta v_3 &= -\frac{\partial L}{\partial v_3} &&= -\left(\gamma_{32} - \rho_{32}\left(\frac{P_{32}}{v_3}\right)^2\right)\\
\Delta P_{32} &= -\frac{\partial L}{\partial P_{32}} &&= -\left(-z_3 - \mu_3 + \lambda_{32} + r_{23}\gamma_{32} + 2\rho_{32}\frac{P_{32}}{v_3}\right)\\
\Delta l_{32} &= -\frac{\partial L}{\partial l_{32}} &&= -\left(-r_{23}\lambda_{32} + \rho_{23} + \rho_{32}\right)\\
\Delta \mu_3 &= \frac{\partial L}{\partial \mu_3} &&= -P_3^d - P_{32},\\
\Delta \lambda_{32} &= \frac{\partial L}{\partial \lambda_{32}} &&= P_{23} + P_{32} - r_{23}l_{32}\\
\Delta_{\gamma_{32}} &= \frac{\partial L}{\partial \gamma_{32}} &&= v_2 - v_3 - r_{23}(P_{23} - P_{32})\\
\Delta \rho_{32} &= \frac{\partial L}{\partial \rho_{32}} &&= l_{32} - \frac{P_{32}^2}{v_3}
\end{aligned}
$$

3.2.4 Iterative method

The previous EQUATIONS are implemented using `Python`. It has been found empirically that with a step size of 0.01, the algorithm converges to an optimum. A point very close to the optimum is reached after approximately 25000 iterations. The algorithm continues until 40000 iterations to have a value of practically 0 for the gradient norm of each bus.

3.2.5 Perturbation of the optimal state

After reaching this first optimum and noticing that the number of steps needed for convergence was exceedingly high for a network this simple, a small perturbation of the consumption values is modeled. The goal of this operation is to check with a small perturbation from the initial optimal state if the iterative method will converge fast to the new optimum. With all variables initialized at the old operating point, one can assess that the new optimum will be found faster than with a cold start.

For this second solving, the only parameter changed is P_2^d with an increase of 10 W, from 40 to 50 W. The iteration number is still 40000 and the step size is unchanged.

3.2.6 Evolution of the variables

FIGURES 3.21 and 3.22 show that the algorithm does converge. However, the convergence takes several thousands of iterations.

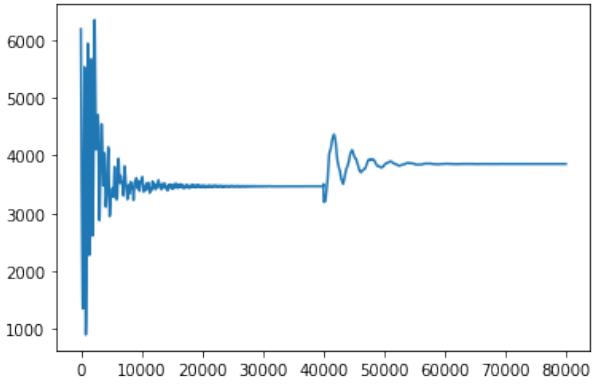


Figure 3.21: Evolution of the objective

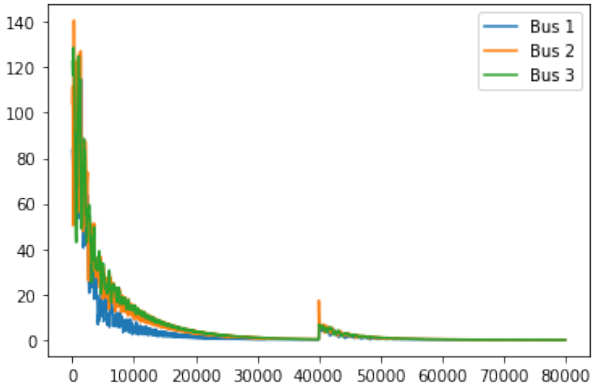
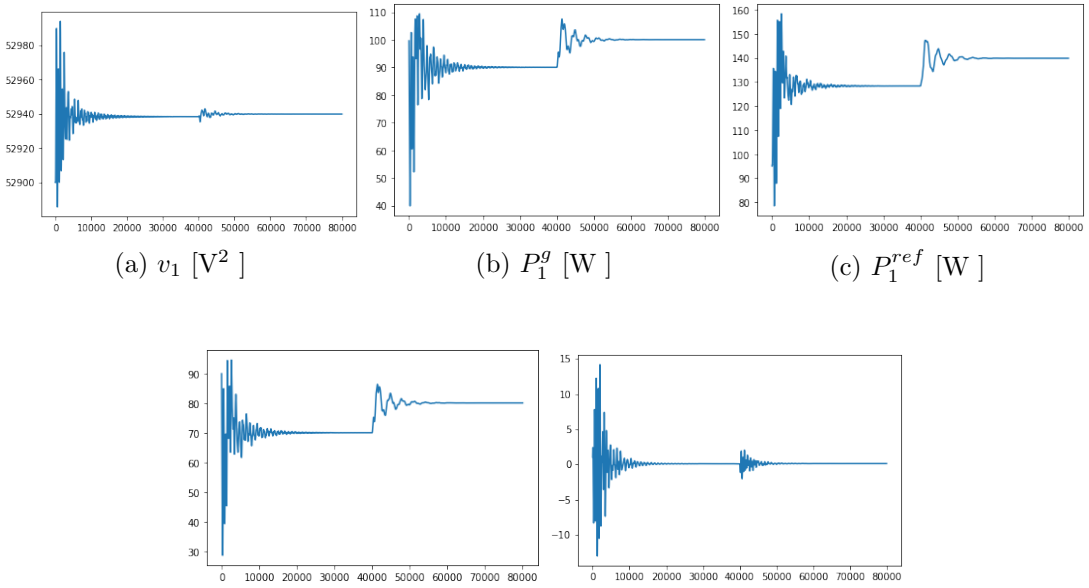


Figure 3.22: Evolution of the gradient norm

The following FIGURES show the evolution of each variable of the model. It can be seen that the number of iterations to achieve convergence is relatively similar in each of the graph. It can also be seen that after the perturbation at the 40000^{th} iteration, the magnitude deviation is much lower, but the number of iterations to retrieve optimality is still several thousand.

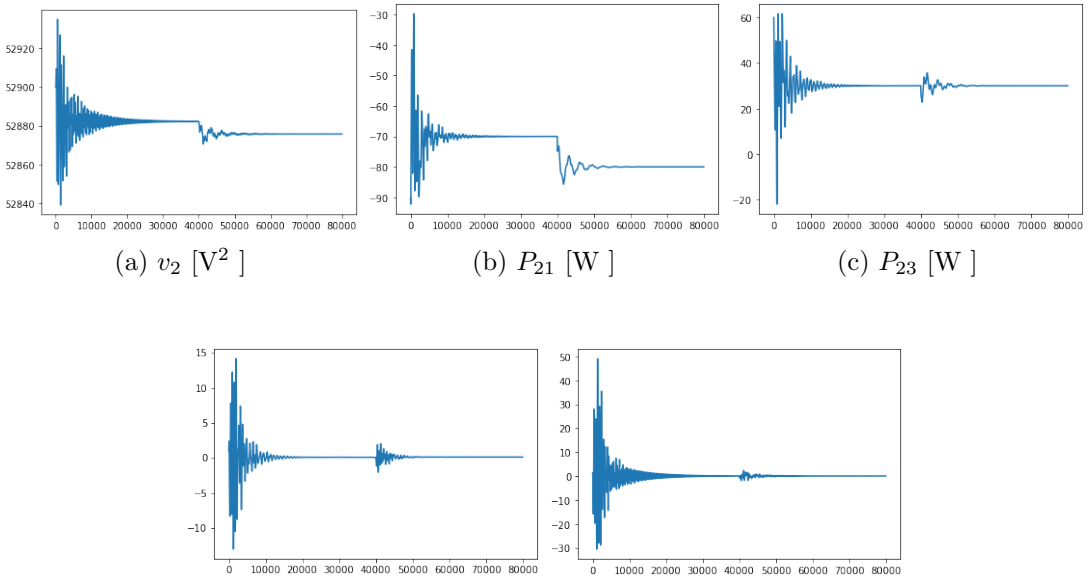
3.2.6.1 Primal variable convergence



(a) v_1 [V^2] (b) P_1^g [W] (c) P_1^{ref} [W]

(d) P_{12} [W] (e) l_{12} [A^2]

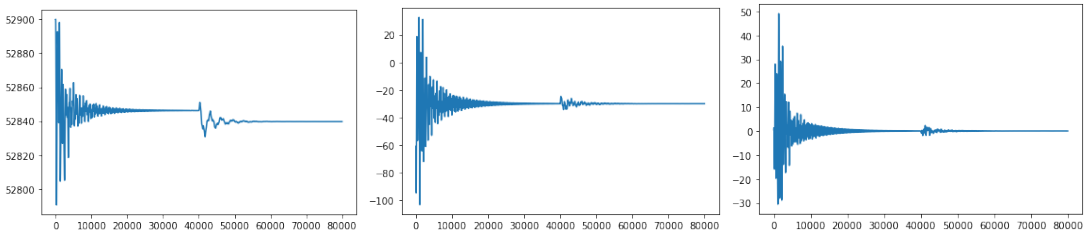
Figure 3.23: Evolution of the values of the primal variables of bus 1



(a) v_2 [V^2] (b) P_{21} [W] (c) P_{23} [W]

(d) l_{21} [A^2] (e) l_{23} [A^2]

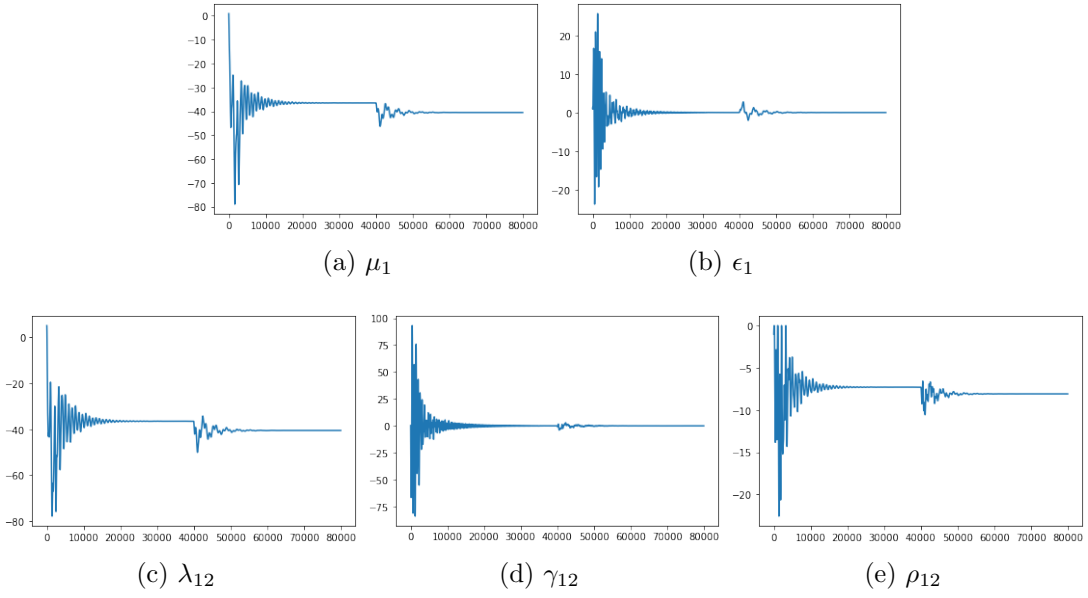
Figure 3.24: Evolution of the values of the primal variables of bus 2



(a) v_3 [V^2] (b) P_{32} [W] (c) l_{32} [A^2]

Figure 3.25: Evolution of the values of the primal variables of bus 3

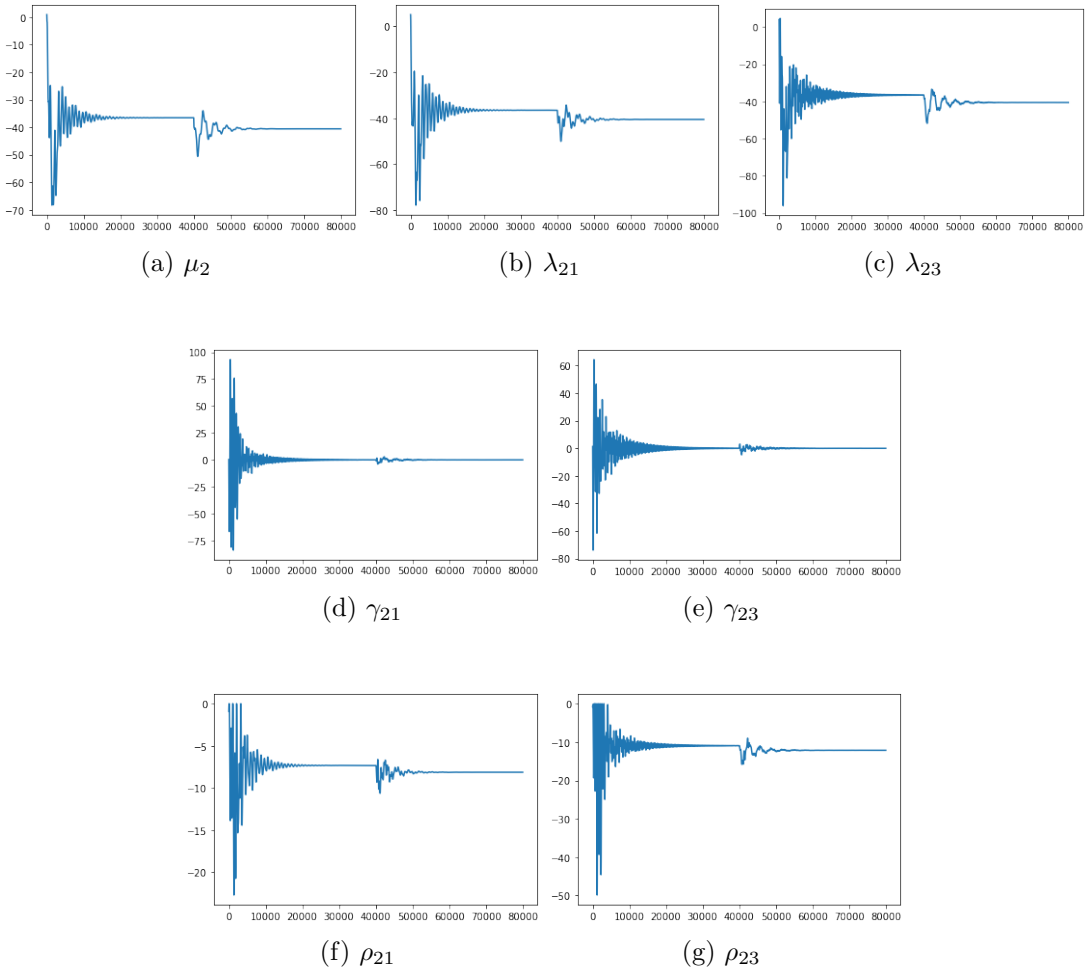
3.2.6.2 Dual variables convergence



(a) μ_1 (b) ϵ_1

(c) λ_{12} (d) γ_{12} (e) ρ_{12}

Figure 3.26: Evolution of the values of the dual variables of bus 1

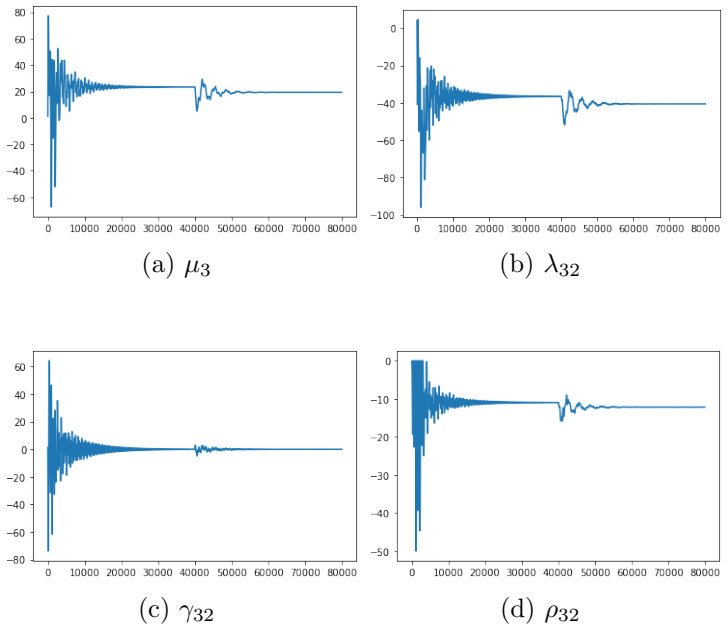


(a) μ_2 (b) λ_{21} (c) λ_{23}

(d) γ_{21} (e) γ_{23}

(f) ρ_{21} (g) ρ_{23}

Figure 3.27: Evolution of the values of the dual variables of bus 2



(a) μ_3 (b) λ_{32}

(c) γ_{32} (d) ρ_{32}

Figure 3.28: Evolution of the values of the dual variables of bus 3

Unfortunately, these results are still unsatisfactory. Some of the variables need up to 20,000 iterations to reach steady state again. Updated values of P_{ik}, v_k and ρ_{ik} from each neighbor k of node i must be exchanged at each iteration. In a real application, this increases convergence time drastically by considering communication delays.

An efficient way of reducing the number of iterations is to use an appropriate line search method. However, to ensure correctness, the step size must be equal at each bus. For an efficient methodology, a centralized algorithm would be necessary. This would be in total contradiction with the objective of this work.

Another way of decreasing the number of iterations would be to implement some sort of naive method to replace the line search. This method would simply decrease the step size according to the increase in iteration number. However, such a method is quite inefficient and the decrease rate must be very well chosen to ensure convergence.

3.3 Outcome of the distributed problem

Wang *et al.* use result from [55] to prove the convergence of this distributed algorithm. Their proof uses dynamic of systems to show that this particular model has a stable convergence point. However, it does not elaborate on how to attain the said point. This is unfortunate, because without the complete methodology, proper convergence was not achieved.

The algorithms implemented in this CHAPTER do not provide a satisfactory result. Hence, the real-time implementation of the tertiary controller will implement the centralized version of the algorithm.

Chapter 4

Dynamic Model

In this SECTION, a dynamic model is studied. The previously developed OPF algorithm is implemented for the control, while a Hardware-in-the-Loop software models the states of the network dynamically.

This is done using the Typhoon HIL technology, which provides both software and hardware solutions for Hardware-in-the-Loop (HIL) controllers. HIL is a technique used to simulate embedded systems in real time. The controller can be tested with both physical hardware and virtual models of some parts of the system.

First a simplistic mode, with ideal generators, is studied. This allows testing of both the model of the system and the link with the OPF solver. After that, a more realistic model is implemented with batteries and droop control. This permits to have a better reactivity of the system. However, the dynamic reaction of the system has yet to be modeled.

Both models are run on a Typhoon HIL 402 machine. This equipment is the basic model of Typhoon HIL simulators. It enables usage of the HIL resources on a laptop-sized device.

The software used to interact with the device is Thyphoon HIL Control Center. It is a proprietary software developed by Typhoon HIL Inc. and divided in several applications. The Schematic Editor and SCADA are the primary ones and will be used in this work.

The Schematic Editor allows to graphically model electrical components and circuits. It consists of a drawing window with a number of libraries and components available for rapid modeling. The model can include both electrical and signal processing elements. They have a different usage, execution rate and are differentiated by color. The signal processing is in blue and the electrical components are black. The Schematic Editor also compiles the model so it can be used with SCADA. The compilation process ensures that the model is complete and correct.

SCADA is the application that runs the dynamic model and displays the variables of interest. It also appears as a whiteboard with multiple widgets available. These widgets can be used to monitor the values of different variables from the schematic model, to change the inputs of a said model or to execute macros.

4.1 HIL model

The network has the same topology as the one presented in SECTION 3.1.1. The values of the parameters have been adapted to a allow more power to be exchanged. Thus, the boundaries for power produced are increased tenfold. Resistances are decreased to allow more power to flow in the lines with the same voltage difference as in SECTION 3.1. The cost coefficients are also modified to keep the same range of values for the objective function.

Line:	(1,2)	(1,4)	(1,5)	(2,3)	(2,4)	(2,5)	(2,6)	(3,5)	(3,6)	(4,5)	(5,6)
R (Ω)	5.0	2.5	4.0	2.5	2.5	5.0	3.5	6.0	1.0	10.0	5.0

Table 4.1: Line resistances for the real-time simulation using `Typhoon HIL`

	$\underline{p}$ [W]	$\overline{p}$ [W]	$\underline{V}$ [V]	$\overline{V}$ [V]	k [Ω]	v^* [V^2]	c_1	c_2	c_3
Generator 1	500.0	2000.0	218.5	241.5	10.0	230.0^2	213.1	0.11669	$5.33e{-5}$
Generator 2	375.0	1700.0	218.5	241.5	10.0	230.0^2	200.0	0.10333	$8.89e{-5}$
Generator 3	450.0	1800.0	218.5	241.5	10.0	230.0^2	240.0	0.10833	$7.41e{-5}$
Generator 4	0	0	218.5	241.5	10.0	230.0^2	0.0	0.0	0.0
Generator 5	0	0	218.5	241.5	10.0	230.0^2	0.0	0.0	0.0
Generator 6	0	0	218.5	241.5	10.0	230.0^2	0.0	0.0	0.0

Table 4.2: Parameters of the generators for the real-time simulation using `Typhoon HIL`

In `Typhoon HIL`, the model must be completed with signal processing units in order to correctly capture the real time states of the network.

4.1.1 Schematic

In large scale, the `Typhoon HIL` schematic replicates the topology of FIGURE 3.1. The six buses are placed at approximately the same position as in FIGURE 4.1. The links between them are two-wire lines, one for the ground and one for the positive terminal. Between the buses, some other elements have been added.

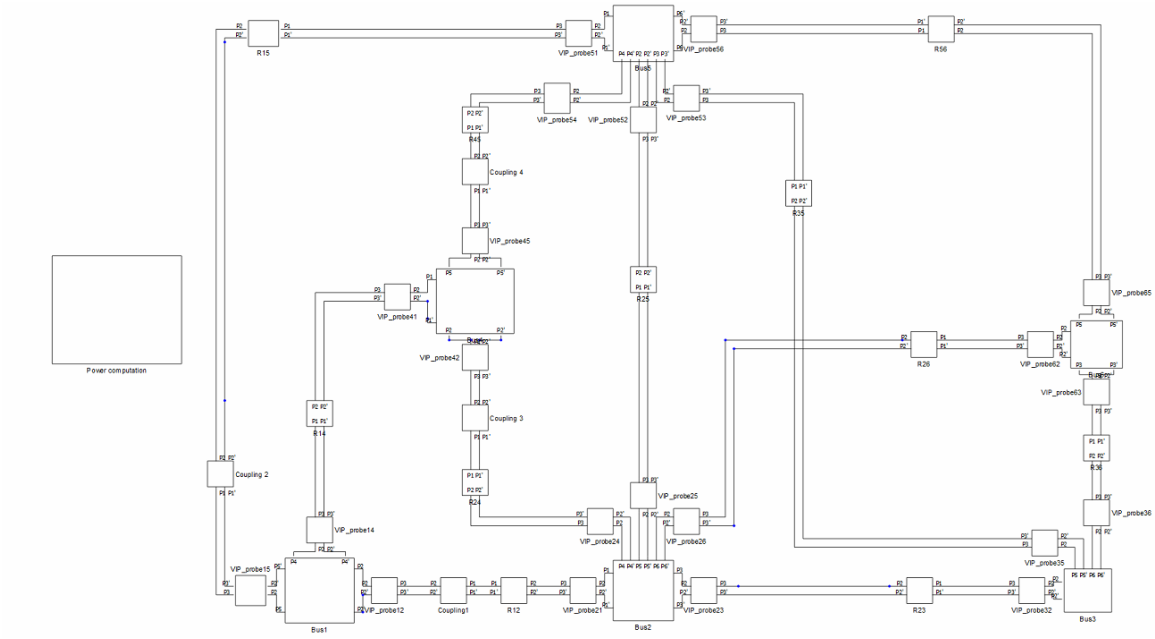


Figure 4.1: Complete schematic of the 6 bus network in `Typhoon HIL`

4.1.1.1 Buses

Different types of bus are modeled using `Typhoon HIL`. They correspond to the different methods of bus control presented in SECTION 1.2.2, except load shedding.

4.1.1.1.a Voltage controlled bus

FIGURE 4.2 shows the schematic of a basic voltage controlled bus. This kind of bus is used to provide an ideal source of power as well as voltage stability for the network.

It consists of an ideal voltage source controlled by `SCADA`, a voltage measurement device, a load and line outputs going to the neighboring buses.

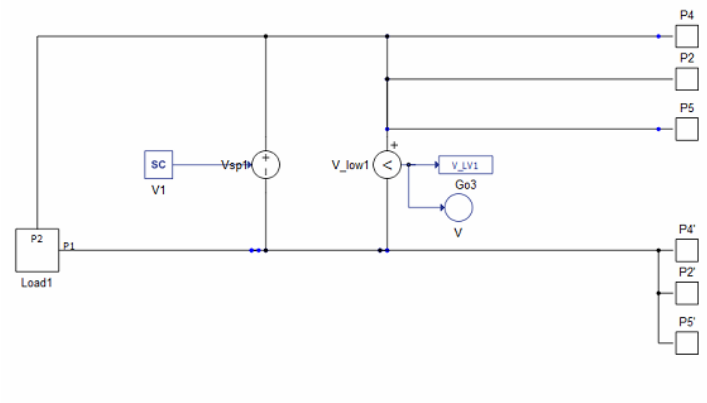


Figure 4.2: Voltage controlled bus schematic in the 6 bus network from `Typhoon HIL`

4.1.1.1.b Power controlled bus

FIGURE 4.3 shows the schematic of a power controlled bus. The controlled power source is modeled by an ideal current source. The injected current value is defined by the power setpoint and the voltage level of the bus.

Similarly to the voltage bus, it contains a load, a voltage measurement device and ports to connect to the neighboring nodes.

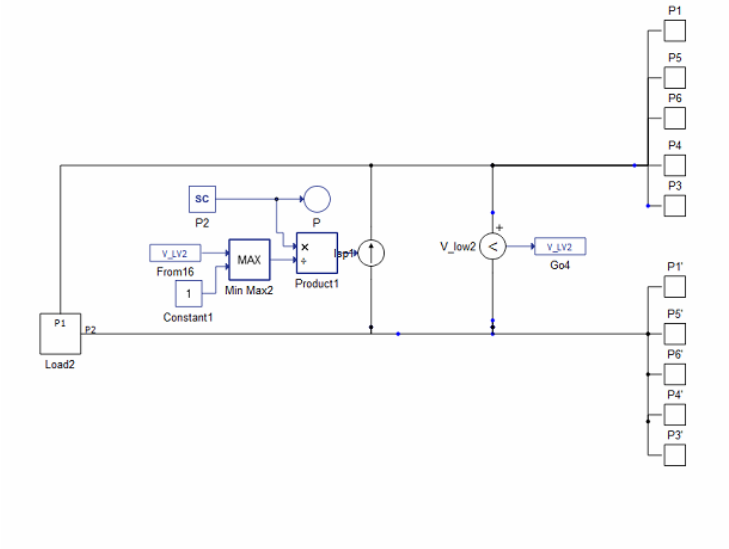


Figure 4.3: Power controlled bus schematic in the 6 bus network from `Typhoon HIL`

4.1.1.1.c Droop controlled bus

The last type of generator bus is droop controlled. As shown in FIGURE 4.4, it consists of a battery behind a DC/DC converter, with a voltage measurement device, a load and the ports to the neighboring nodes on the HV side. Both sides of the converters are also linked to the same processing core of the `HIL 402` device, because the number of cores is limited to two.

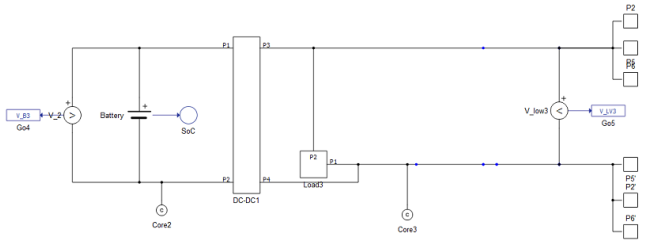


Figure 4.4: Droop controlled bus schematic in the 6 bus network from `Typhoon HIL`

The DC/DC converter is operated by droop control. It is modeled by two ideal controllable current sources. The voltage is measured on both sides of the converter and used to compute the injection current according to the power setpoint computed by the droop controller.

The power setpoint is computed using EQUATION (2.20). It is important to note that in this equation, the voltage values are squared voltage. k_i and v_i^* are parameters set by `SCADA`. $\hat{p}_i$ is a variable, it is also set by `SCADA` and can be modified, either manually, or by the `OPF` tertiary control. The power setpoint is bounded between -2500 W and 2500 W.

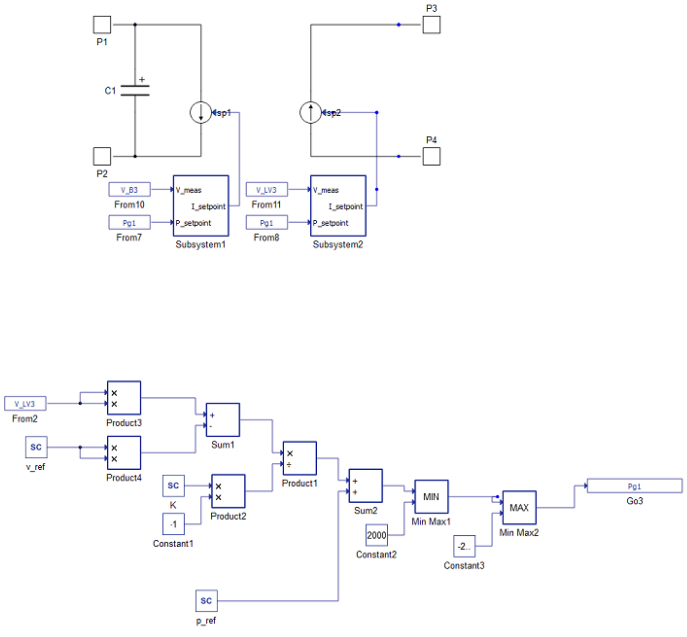


Figure 4.5: Droop controller schematic in the 6 bus network from `Typhoon HIL`

4.1.1.1.d Load bus

The last type of bus does not include any generator. Hence, it only consists of a voltage measurement device, a load subsystem and ports to connect the bus to its neighbors, as shown in FIGURE 4.6.

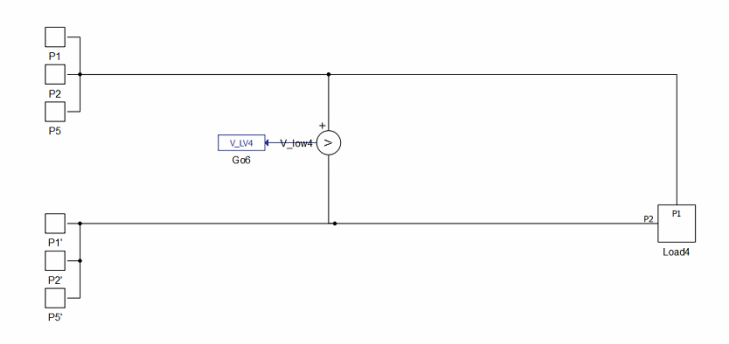


Figure 4.6: Load only bus schematic in the 6 bus network from `Typhoon HIL`

4.1.1.1.e Load subsystem

The load subsystem is present in each bus. The controllable load is modeled by an ideal current source which consumes constant power. Its current value is computed using the voltage of the bus and the power reference set by `SCADA`.

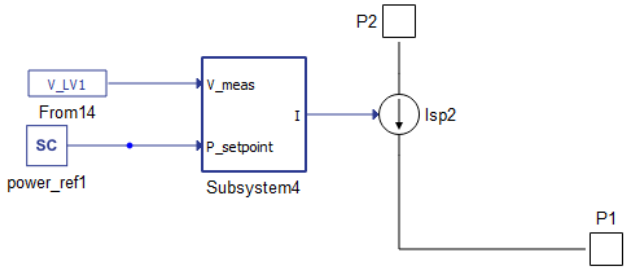


Figure 4.7: Load schematic in the 6 bus network from `Typhoon HIL`

4.1.1.2 Line elements

The buses are connected using two-wire lines. One wire is used for the ground and the other for the positive terminal. These lines are not plain connections, they include other elements: probes, resistances, and coupling.

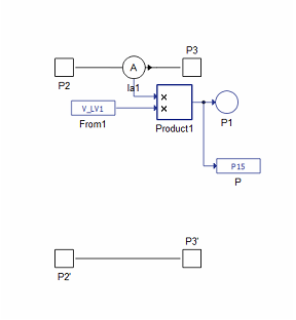


Figure 4.8: Probe schematic in the 6 bus network in `Typhoon HIL`

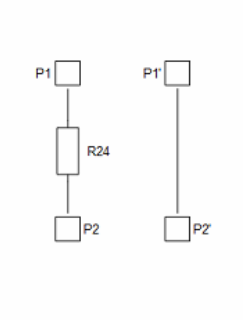


Figure 4.9: Resistance schematic in the 6 bus network in `Typhoon HIL`

To assess the correctness of the method, line currents and power must be monitored. This is done by the probe subsystem shown in FIGURE 4.8. This system consists of an ideal current measurement device and a product operator that takes this signal and the bus voltage to output the power sent through this line. Each line is equipped with two probes, in order to compute the power and current in both directions.

Each line also includes a resistance, whose value is given in TABLE 4.1. The system is grounded, so the resistance is on the high-voltage wire of each line.

Due to the limitations of the `Typhoon HIL` device, the dynamic computation of the state of an extensive model cannot be done in a single core of the machine. Fortunately, in some configurations, the `HIL 402` device can simulate the network in two cores. The cores are linked via a "core coupling" element, an ideal voltage/current converter that handles the communication between the two.

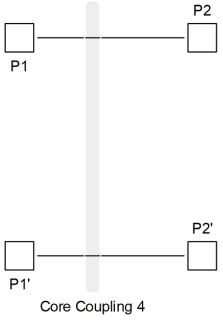


Figure 4.10: Coupling element schematic in the 6 bus network in `Typhoon HIL`

4.1.1.3 Signal computation

To display the different power values as well as other computable data, `SCADA` needs only one signal. It cannot conduct arithmetic between different signals. This is why two signal processing units are needed.

The first one depicted in FIGURE 4.11 performs all the power computations. It outputs the powers generated and sent by each bus, as well as the total power lost in the system.

The second one illustrated in FIGURE 4.12 implements a cost computation using the cost coefficients of each generator from TABLE 4.2 and the cost function introduced in EQUATION (3.1).

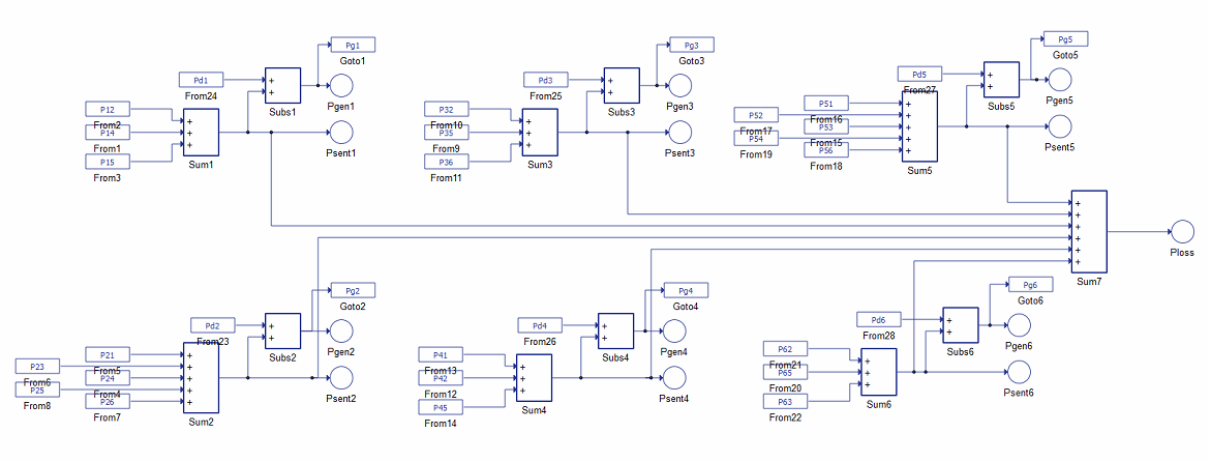


Figure 4.11: Power computation schematic in the 6 bus network in `Typhoon HIL`

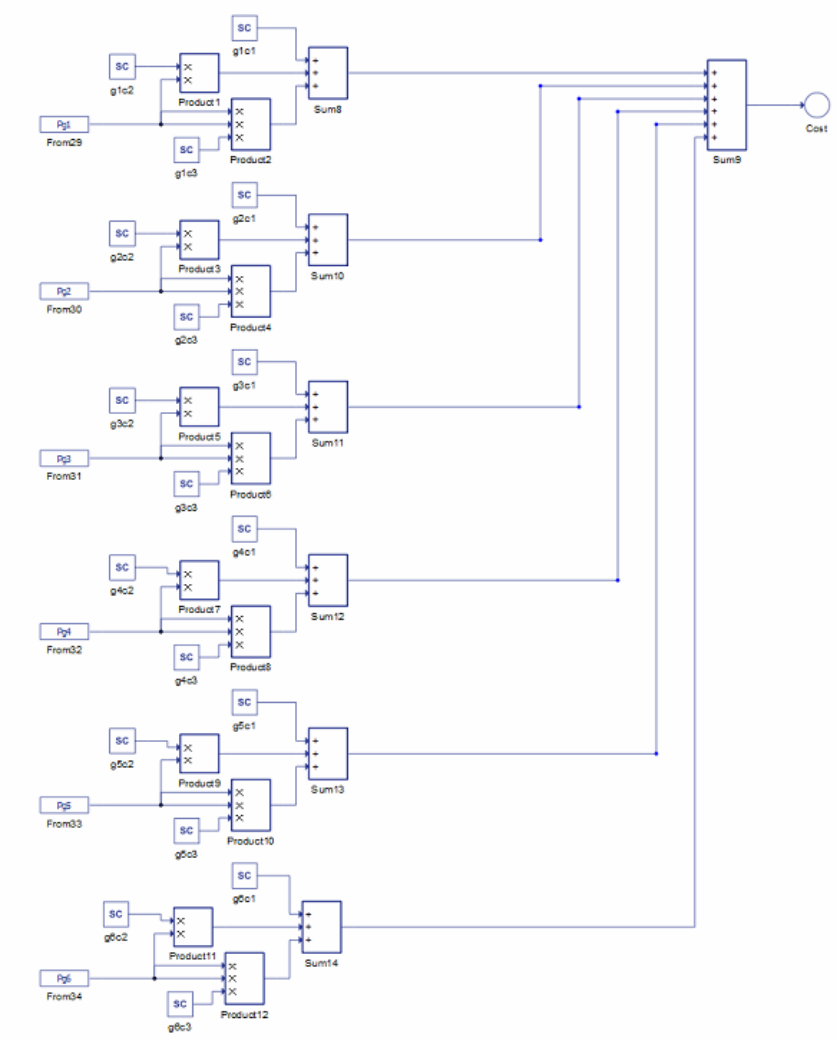


Figure 4.12: Cost computation schematic in the 6 bus network in `Typhoon HIL`

4.1.2 SCADA panel

In `Typhoon HIL`, the `SCADA` panel is the dashboard of the real-time HIL simulation. In this case, the `SCADA` panel is the tool which allows to both monitor and control the current state of the network.

It displays several gauges for the variables tracked in the model. It can also take some actions with the `SCADA` inputs, by modifying their values.

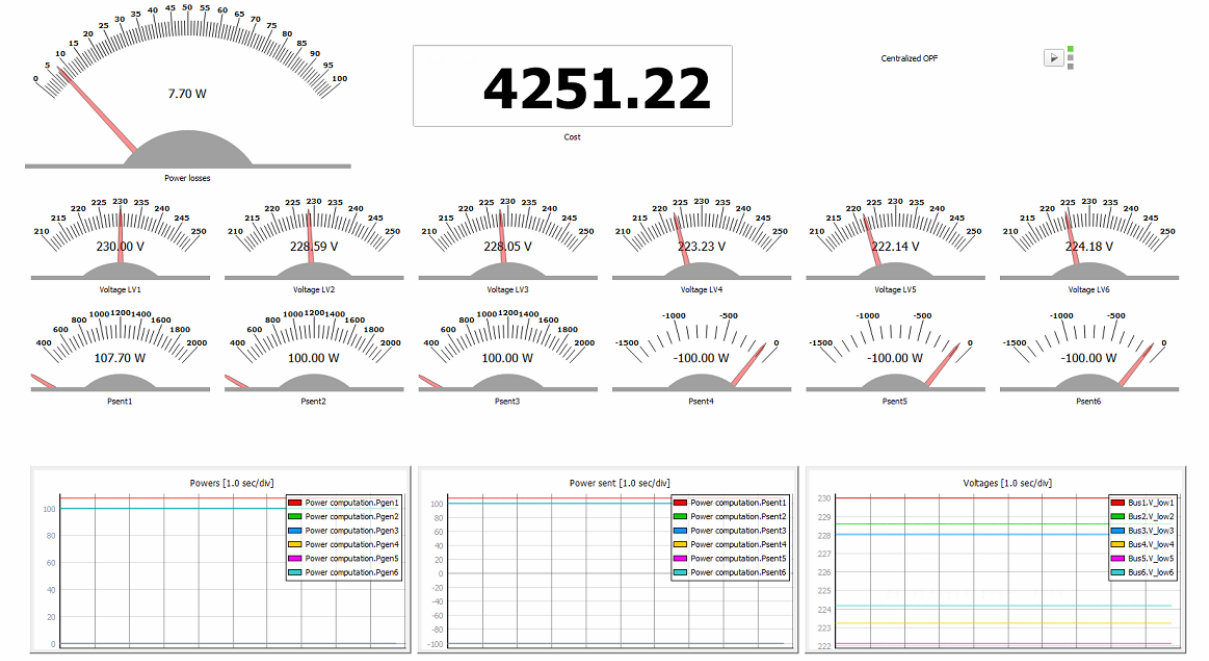


Figure 4.13: Monitoring of the bus variables in SCADA

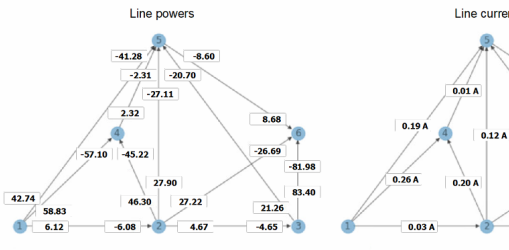


Figure 4.14: Display of the network in SCADA

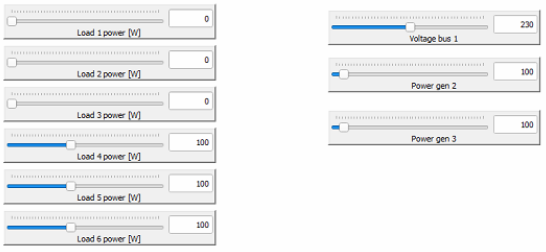


Figure 4.15: Inputs controlled by SCADA

In FIGURE 4.13, all the monitored variables are displayed using gauges or plots. There are also the macro widgets on the upper right of the display. They run the Python code which solves the OPF and send the new reference setpoint to the buses. The macros also perturb the network between the computations for each state of the scenarios.

In FIGURE 4.14, the two models represent the network in real time with the powers and currents flowing through the lines. It can display the line variables value to verify that power flows are correctly computed and that possible maximum values are not reached.

In FIGURE 4.15, the part of the panel where the inputs are managed is shown. Despite the possibility of fully parametrizing the network, the customization is not doable on a very numerous number of parameters. Indeed, the goal here is to check the basic response of the network and the modification of its control variables when the load changes. Thus, only the loads are configurable as a pure parameters. The control instructions are also modifiable, yet this would usually be done by the OPF algorithm.

4.2 Scenarios

To test the prototype of the tertiary controller in the real-time simulation, two scenarios are investigated. They are described individually in SECTIONS 4.2.2 and 4.2.3 and the results are presented in SECTIONS 4.3.1 and 4.3.2.

Two different models are built with the elements presented in SECTION 4.1.1. Both are based on FIGURE 4.1 and only differ on the type of control for the buses:

- **Model A**: Bus 1 is voltage controlled (FIGURE 4.2), buses 2 and 3 are power controlled (FIGURE 4.3) and buses 4, 5 and 6 are load buses (FIGURE 4.6).
- **Model B**: Bus 1 is voltage controlled (FIGURE 4.2), buses 2 and 3 are droop controlled (FIGURE 4.4) and buses 4, 5 and 6 are load buses (FIGURE 4.6).

The two scenarios are tested with the two models. The tertiary controller uses the `OPF` of PROBLEM (2.23). The network corresponds to the one presented in SECTION 3.1.1, with the new parameter values presented in SECTION 4.1. Due to the absence of satisfactory convergence for the distributed algorithm, the centralized version is implemented. This is done using a `Python` program with `pyomo` in the macro widgets of FIGURE 4.13.

Both models give similar and satisfactory results. However, model B implements droop which allows primary control. The system is thus more stable. The deviations right after the perturbations are less pronounced. Only the description of the states for model B are shown in SECTION 4.3. This choice has been made because of the similarities of both model and to keep the presentation concise.

4.2.1 Initial state

At the start of both scenarios, the network operates in optimal steady-state. Loads in buses 4, 5 and 6 each consumes 700 W. FIGURE 4.16 shows the voltage values for the different buses and their incident power. FIGURE 4.17 shows the present values for currents and powers flowing in the lines.

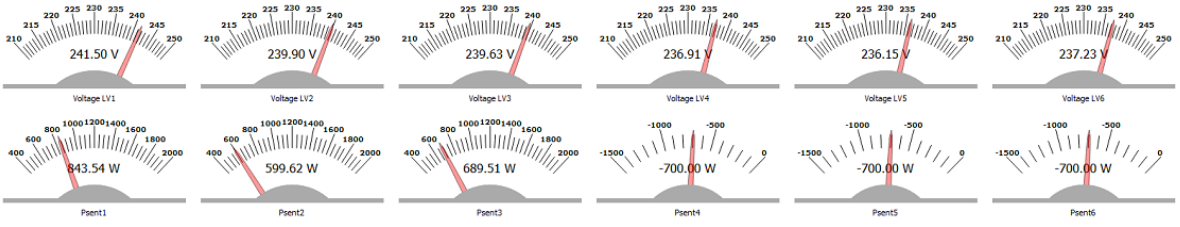


Figure 4.16: Initial state of the buses in `Typhoon HIL`

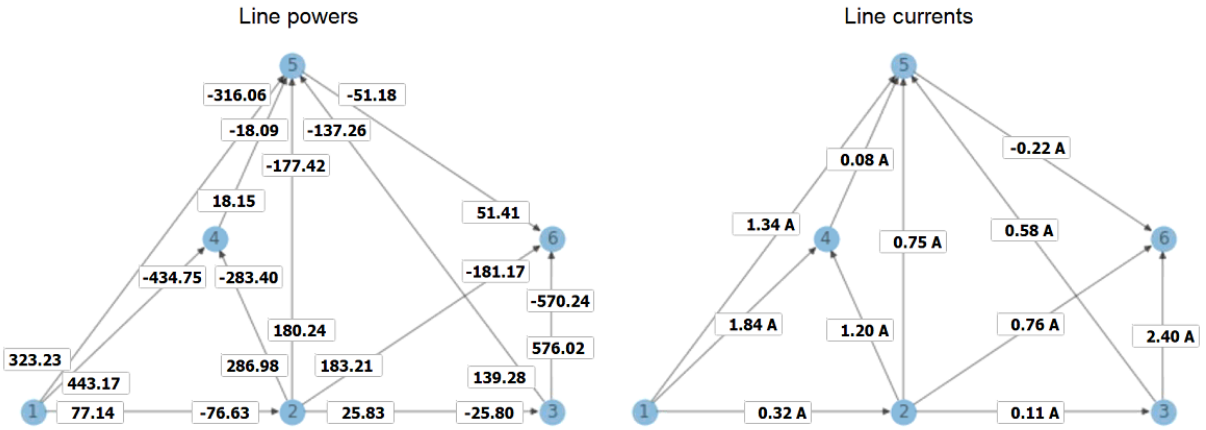


Figure 4.17: Initial state of the lines in `Typhoon HIL`

TABLE 4.3 shows the values of the bus variables, computed by the optimization algorithm using `Pyomo`. They are close to the values of FIGURE 4.16, which is appropriate and demonstrates the correspondence between both models.

	V_i [V]	p_i^g [W]	$\hat{p}_i$ [W]
Bus 1	241.5	844.41	1386.63
Bus 2	239.90	599.73	1064.96
Bus 3	239.63	689.63	1141.94
Bus 4	236.91	0	0
Bus 5	236.14	0	0
Bus 6	237.23	0	0

Table 4.3: Initial values for the bus variables computed by `Pyomo`

4.2.2 Scenario 1

For the first scenario, the goal is to assess the changes in the network due to the changes in the load values. Then, the impact of the loss of one generator is evaluated.

This scenario consists of 5 steps, each of which is separated by a computation of the new optimal state of the system. As soon as the network is once again operating in optimal steady state, the next perturbation occurs.

At the start, buses 4, 5 and 6 each consumes 700 W. The `OPF` has been run and the network is operating in optimal steady-state conditions.

1. Buses 2 and 5 see their consumed power increased to 500 W and 1000 W respectively.
2. The load of bus 6 is deactivated.

3. Bus 4 increases its power consumption to 2000 W.

4. Generator 2 is deactivated, both of the boundaries on the produced power are set to 0 W.

5. Generator 2 is reactivated with the initial boundaries for the produced power.

4.2.3 Scenario 2

For the second scenario, the goal is to assess the impact of a change in the cost function, the loss of several lines and the isolation of one bus. It consists in 3 successive perturbations of the network state.

It is important to note that the line is simply cut in this simulation. As the model does not describe correctly the dynamics of the network, fault analysis and detection are outside the scope of this project. The loss of one line or one bus is simply simulated by opening the lines involved.

At the start, buses 4, 5 and 6 each consumes 700 W. The `OPF` has been run and the network is operating in optimal steady-state conditions.

1. Buses 5 and 6 see their consumed power changed to 2000 W and 500 W, respectively.

2. The price for energy in bus 2 increases.

3. Lines {45} and {56} are lost.

4.3 Results

Both scenarios are tested with the macro widgets from FIGURE 4.13. They implement `Python`code to run the centralized `OPF` algorithm using `Pyomo`.

Each step consists of a perturbation, then the computation of the new optimal steady-state. Once the perturbation has happened, there is a delay of 0.5 second before the `OPF` is solved, then the model stays at steady state for one second before the next perturbation is initiated.

4.3.1 Scenario 1

This SECTION presents the results for the first scenario. It first shows the evolution of the bus and global variables, then it presents the state if the network at each step, before and after the computation of the new optimal state.

4.3.1.1 Evolution

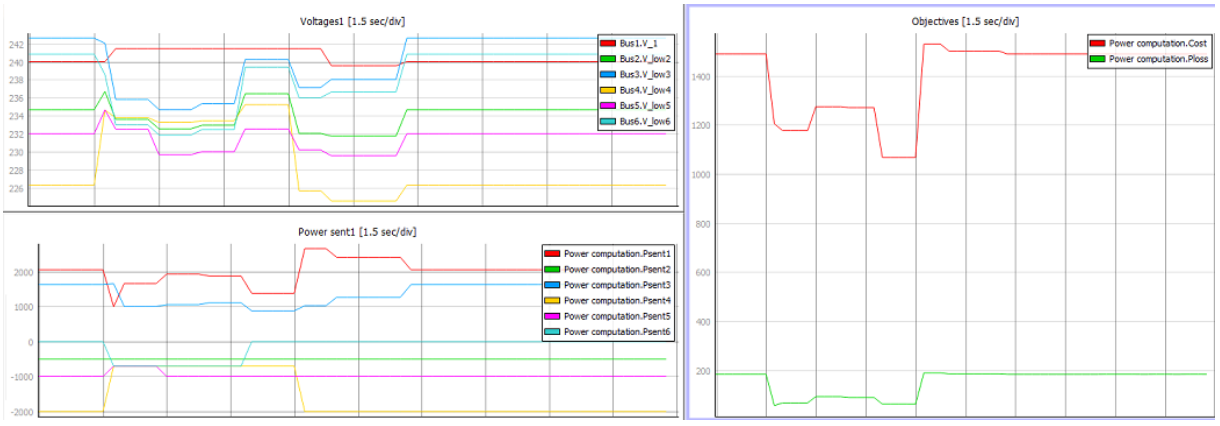


Figure 4.18: Evolution of the bus and global variables for scenario 1

FIGURE 4.18 shows the values for the bus variables on the left. It can be seen that the voltage value for bus 1 stays relatively constant. This bus can be considered as a slack bus, it is voltage controlled, so it can handle the power inaccuracies. When the perturbation occurs, buses 2 and 3 also react, with the power-sharing mechanism of droop control. However, this mechanism is not optimal until the `OPF` is run again. During that time, bus 1 delivers the necessary power to be at equilibrium.

4.3.1.2 Perturbation 1

The powers consumed on buses 2 and 5 are modified to 500 W and 1000 W, respectively.

After the perturbation:

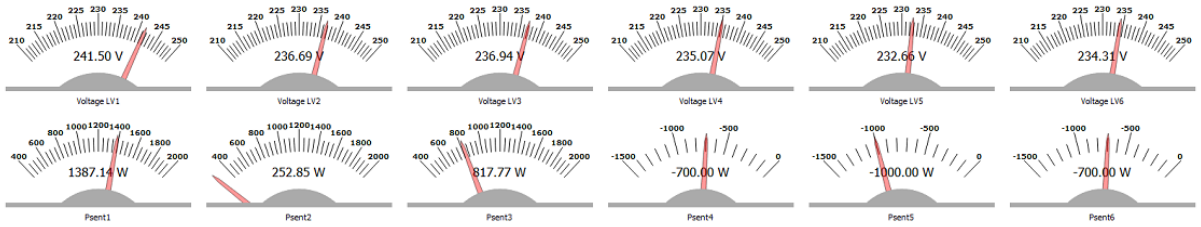


Figure 4.19: State of the buses after the first perturbation of scenario 1

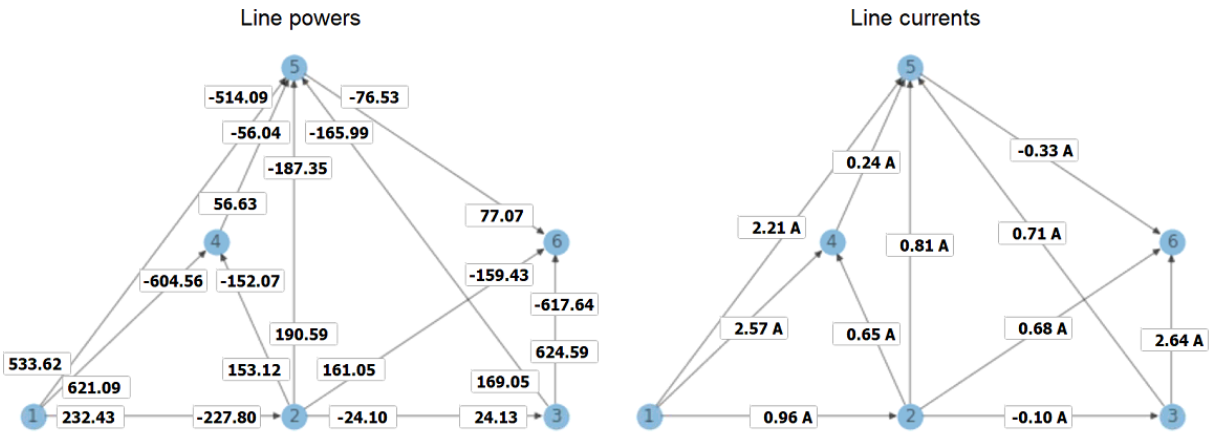


Figure 4.20: State of the lines after the first perturbation of scenario 1

The perturbation causes a modification of the power produced by the slack bus. Buses 2 and 3 are participating in the power sharing, but unevenly. Both of the voltages of the five non-slack buses are decreased to allow the power of bus 1 to be increased, as it is operated at constant voltage. In FIGURE 4.19, the incident power is displayed, the generated power for bus 2 is actually 752.85 W, because 500 W are directly consumed at the bus.

After the OPF:

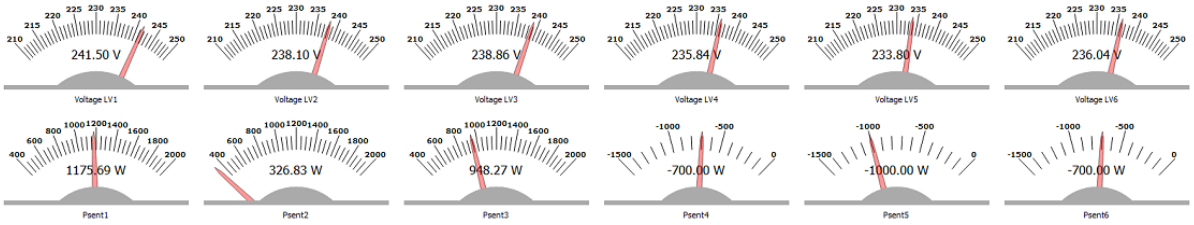


Figure 4.21: State of the buses after the first OPF of scenario 1

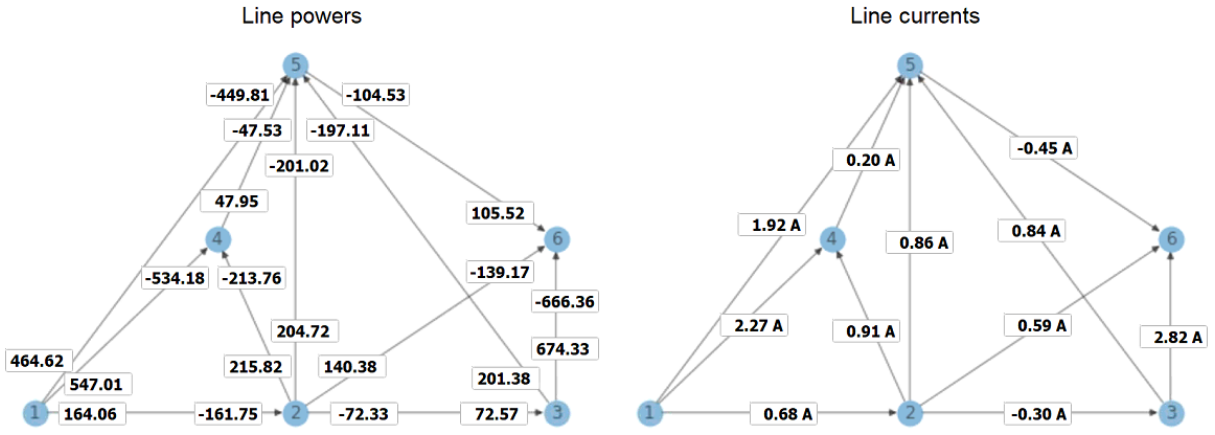


Figure 4.22: State of the lines after the first OPF of scenario 1

The OPF equalizes the power sharing between the three generating buses.

4.3.1.3 Perturbation 2

The load of bus 6 is decreased to 0 W.

After the perturbation:

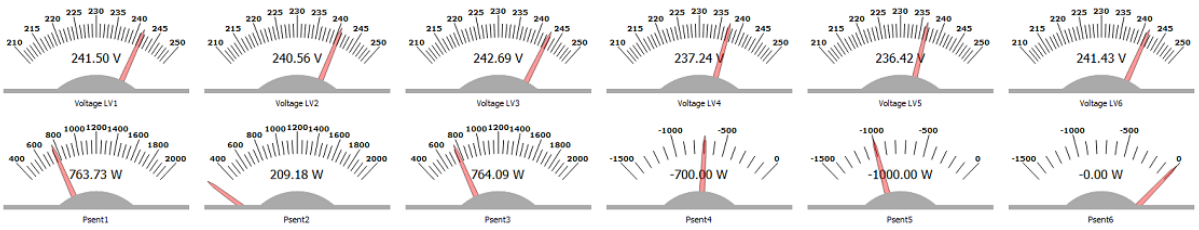


Figure 4.23: State of the buses after the second perturbation of scenario 1

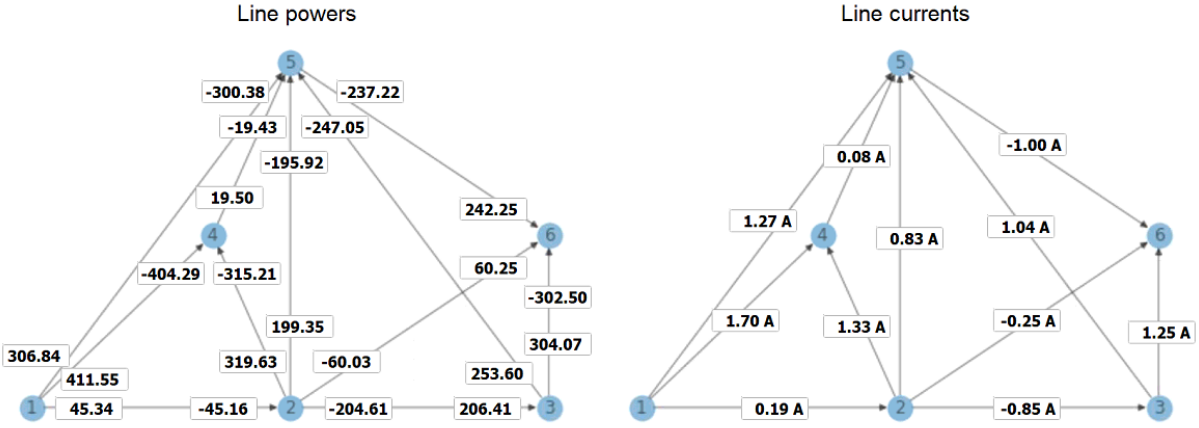


Figure 4.24: State of the lines after the second perturbation of scenario 1

With this decrease of power, once again, most of the power share is handled by bus 1. The effect of the change in the power flows also cause bus 3 to exhibit a small overvoltage. This is due to the fact that the network is constantly operated at maximal voltage in the feasible range, to reduce the losses.

After the OPF:

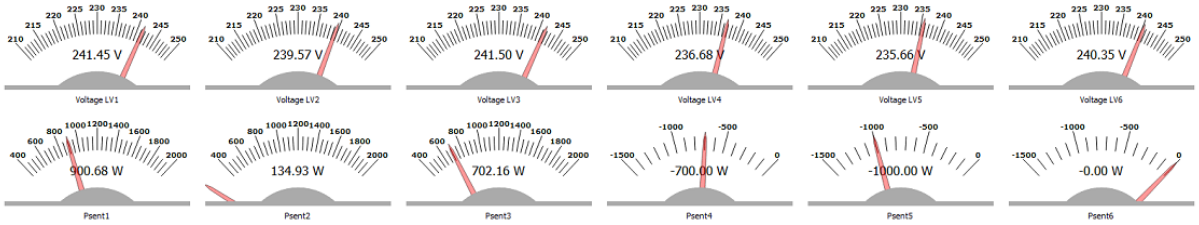


Figure 4.25: State of the buses after the second `OPF` of scenario 1

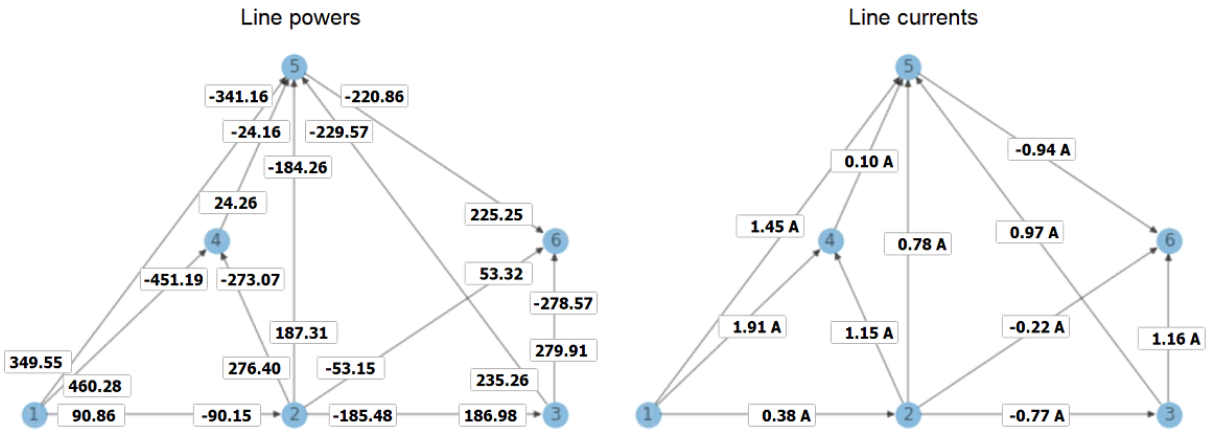


Figure 4.26: State of the lines after the second OPF of scenario 1

The voltage of bus 3 is reduced to return in the feasible range. The OPF also balances the power sharing of the three generators.

4.3.1.4 Perturbation 3

The load of bus 4 is increased to 2000 W.

After the perturbation:

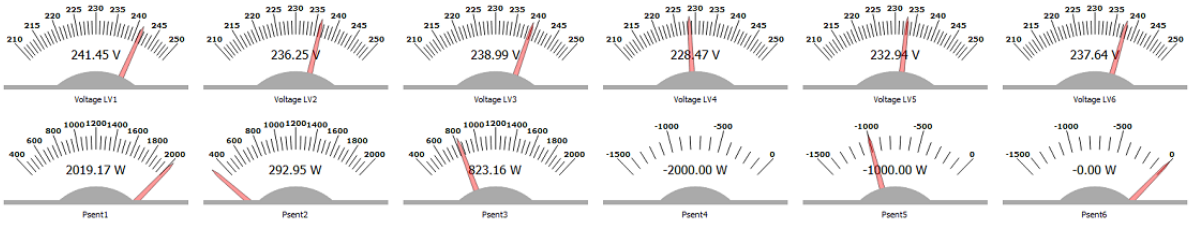


Figure 4.27: State of the buses after the third perturbation of scenario 1

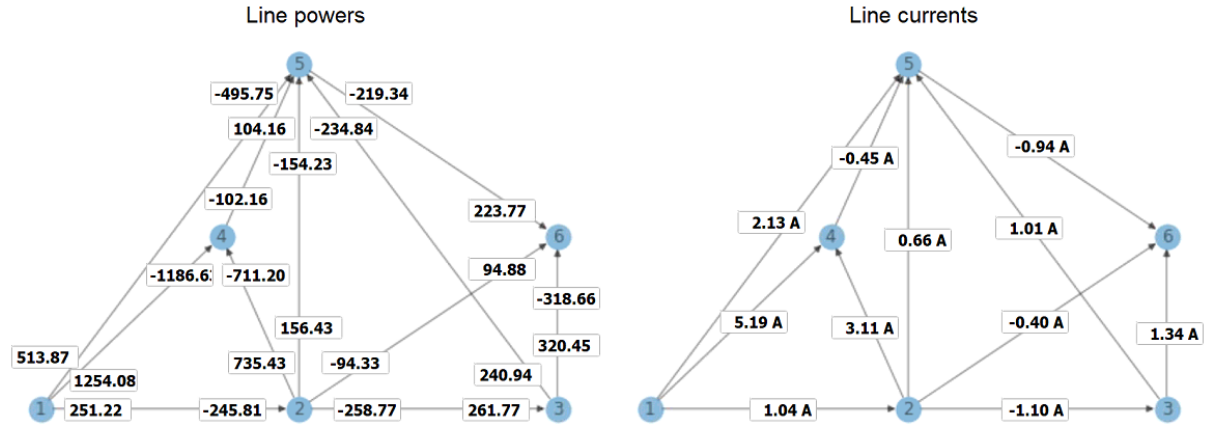


Figure 4.28: State of the lines after the third perturbation of scenario 1

The modifications in the network are similar to those of perturbation 1.

After the OPF:

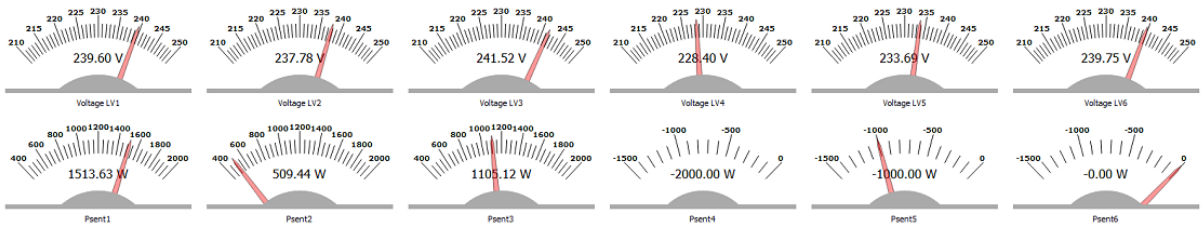


Figure 4.29: State of the buses after the third `OPF` of scenario 1

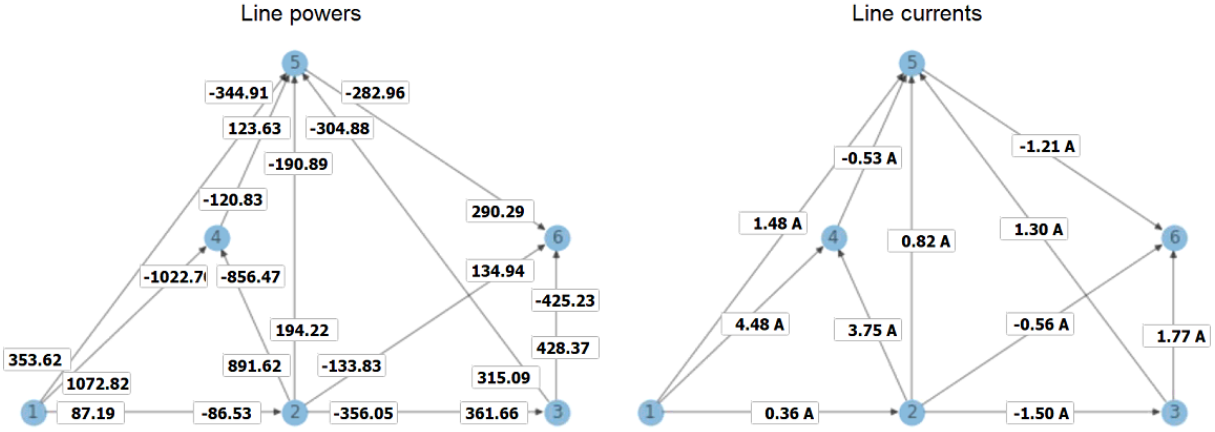


Figure 4.30: State of the lines after the third `OPF` of scenario 1

Power sharing is balanced by the `OPF`. The value of bus 3 is slightly over the maximum, this is due to small inaccuracies in the simulation. The power reference of bus 3 is set to have a voltage value of 241.5 V. There is a small error due to the simulation.

4.3.1.5 Perturbation 4

Generator 2 is deactivated, both of the boundaries on the produced power are set to 0 W.

After the perturbation:

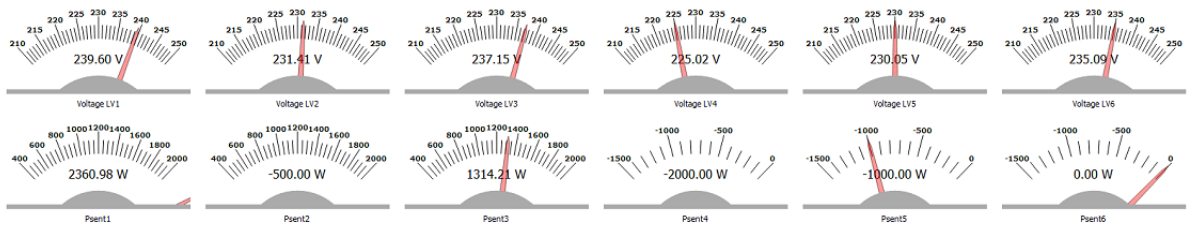


Figure 4.31: State of the buses after the fourth perturbation of scenario 1

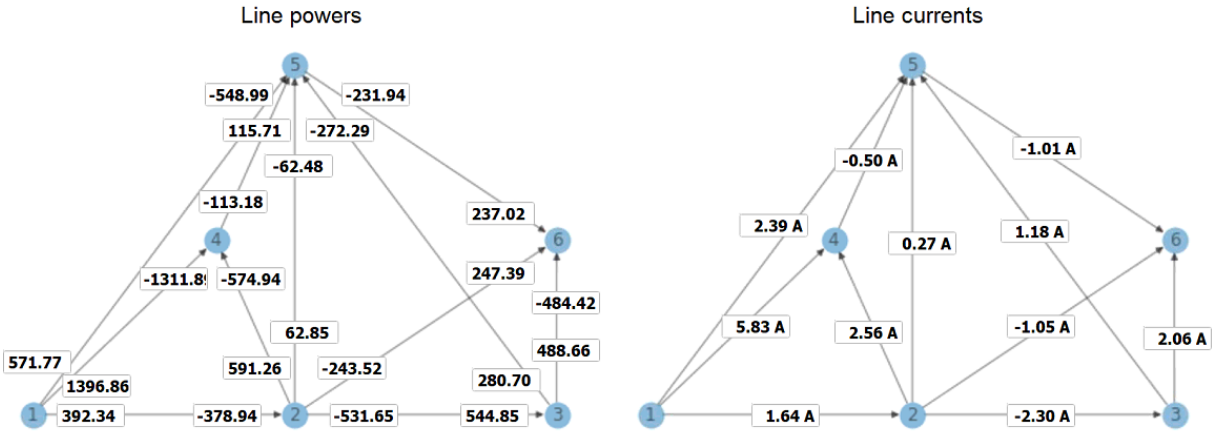


Figure 4.32: State of the lines after the fourth perturbation of scenario 1

The removal of generator 2 causes a voltage drop at buses 2 to 6, this is once again due to the important increase of produced power at bus 1, bus 3 also see its power injection increased, but not as much. In this state, the power constraints for bus 1 are not fulfilled.

After the OPF:

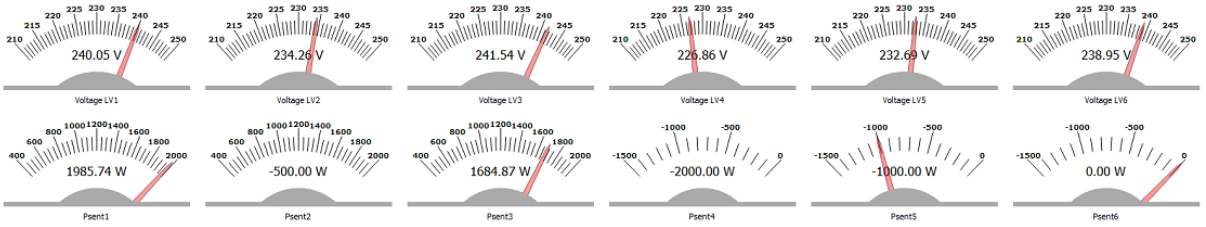


Figure 4.33: State of the buses after the fourth OPF of scenario 1

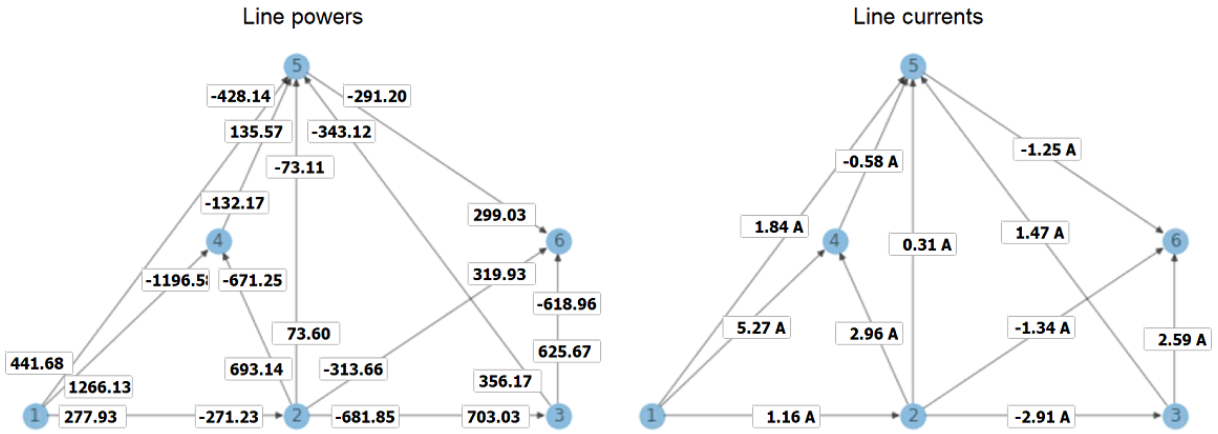


Figure 4.34: State of the lines after the fourth OPF of scenario 1

After the OPF computations, the power injections meet the constraints again. The voltage on bus 3 is back at 241.5 V with a slight error.

4.3.1.6 Perturbation 5

Generator 2 is reactivated, with its initial boundaries for the produced power.

After the perturbation:

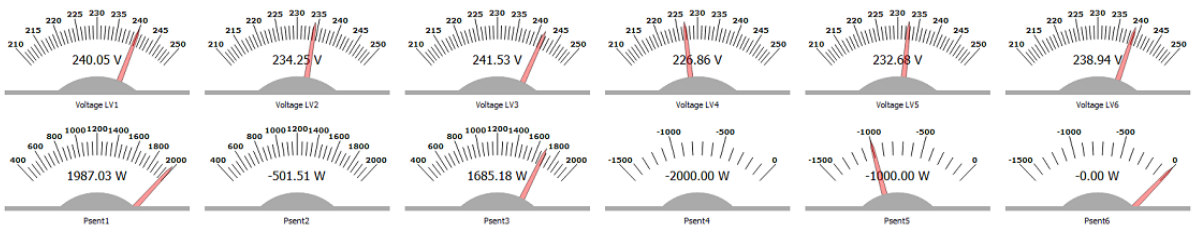


Figure 4.35: State of the buses after the fifth perturbation of scenario 1

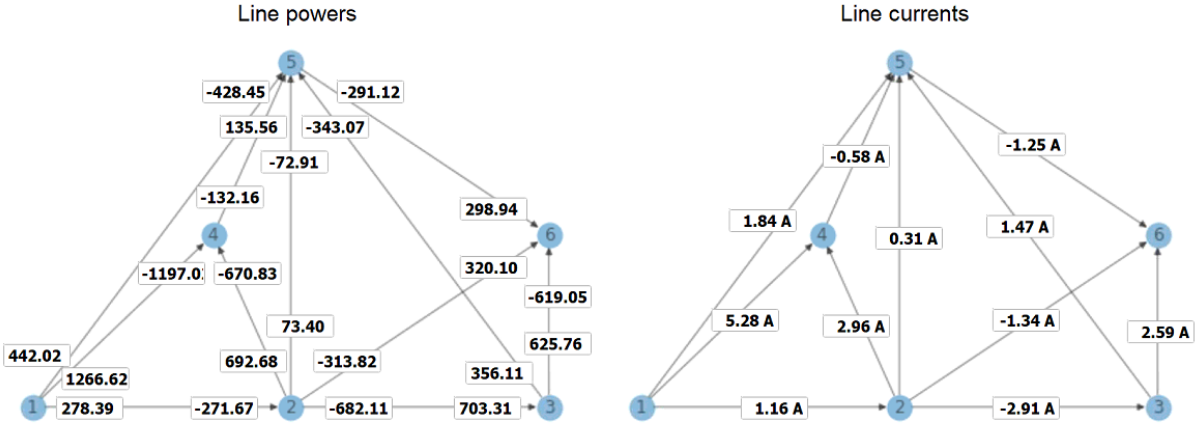


Figure 4.36: State of the lines after the fifth perturbation of scenario 1

The restart of generator 2 does not change the state of the network, except for the 1.51 W that it produces due to the droop controller.

After the OPF:

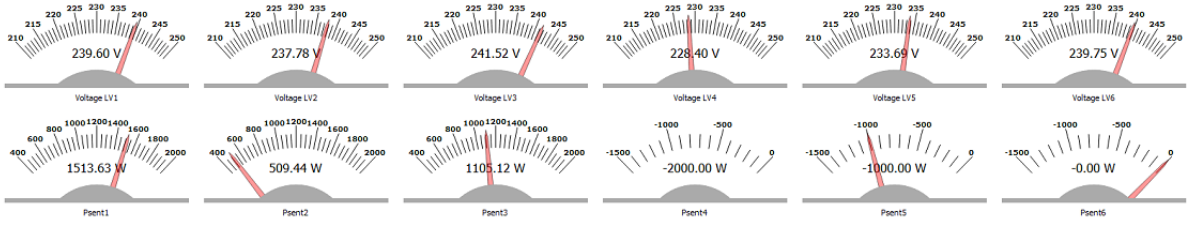


Figure 4.37: State of the buses after the fifth `OPF` of scenario 1

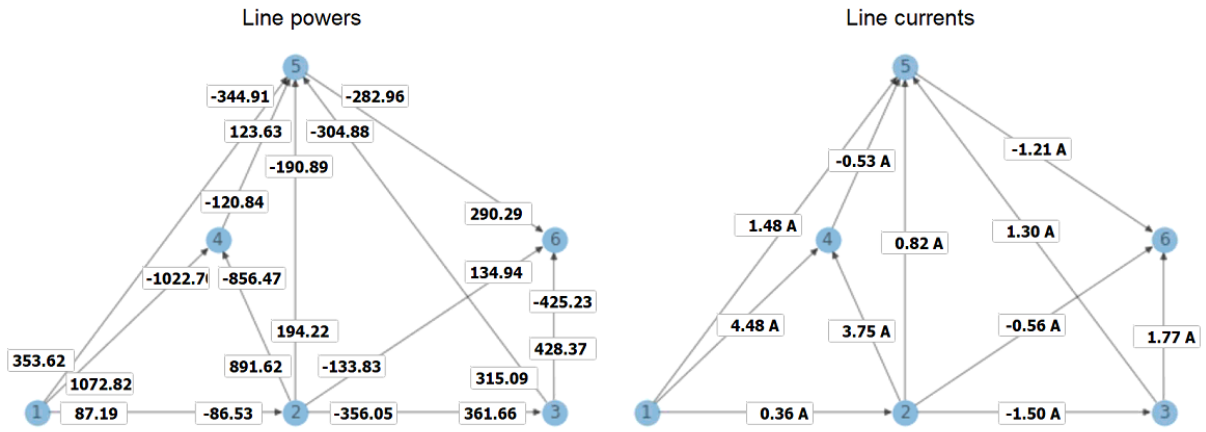


Figure 4.38: State of the lines after the fifth OPF of scenario 1

At optimal operation, the network is at the exactly at the same point of operation as it was before the fourth perturbation.

4.3.2 Scenario 2

This SECTION presents the results for the second scenario. It first shows the evolution of the bus and global variables, then it presents the state if the network at each step, before and after the computation of the new optimal state.

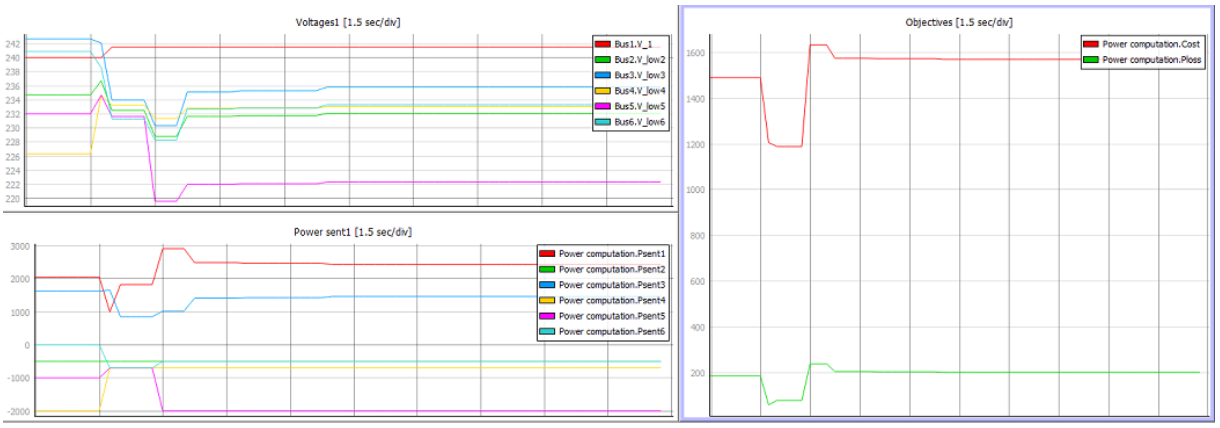


Figure 4.39: Evolution of the bus and global variables for scenario 2

For scenario 2, bus 1 also has a relatively constant voltage level and acts as a slack bus for the microgrid.

4.3.2.1 Perturbation 1

Bus 5 and 6 see their consumed power changed to 2000 W and 500 W respectively.

After the perturbation:

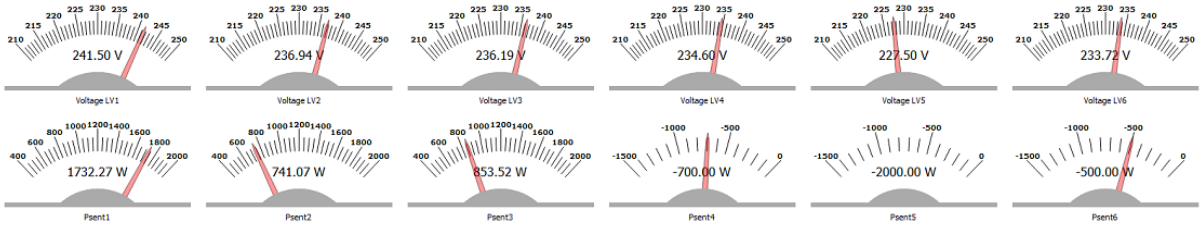


Figure 4.40: State of the buses after the first perturbation of scenario 2

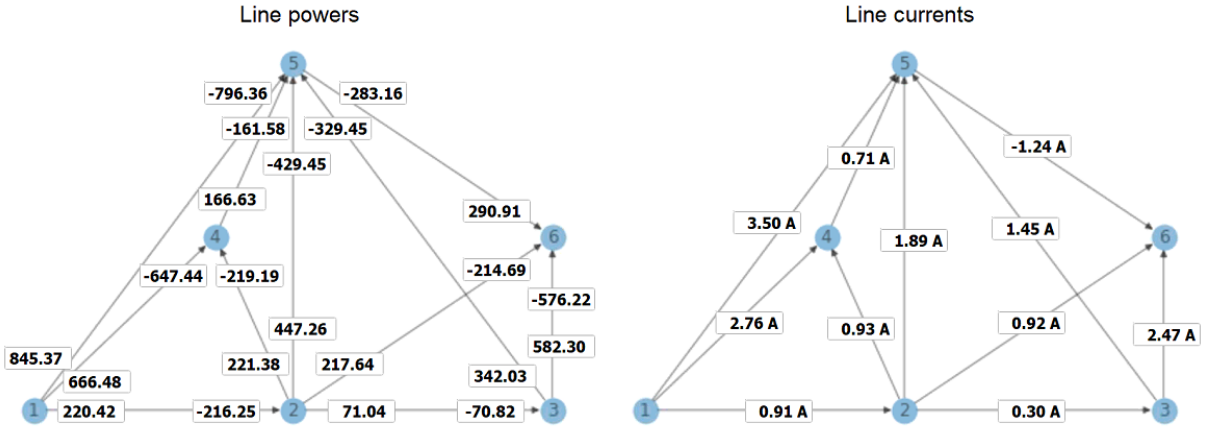


Figure 4.41: State of the lines after the first perturbation of scenario 2

The perturbation induces an important change in the power produced by the slack bus. However, buses 2 and 3 are not extensively participating in the power sharing, this is because the change in power is too important with respect to their value of k. Both of the voltages of the five non-slack buses are decreased to allow the power of bus 1 to be increased, as it is operated at constant voltage.

After the OPF:

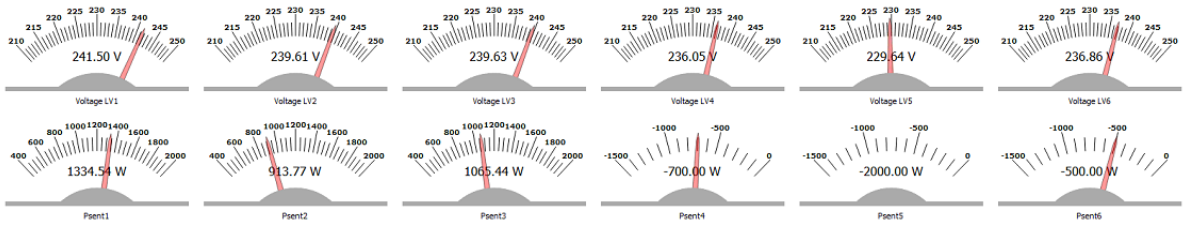


Figure 4.42: State of the buses after the first `OPF` of scenario 2

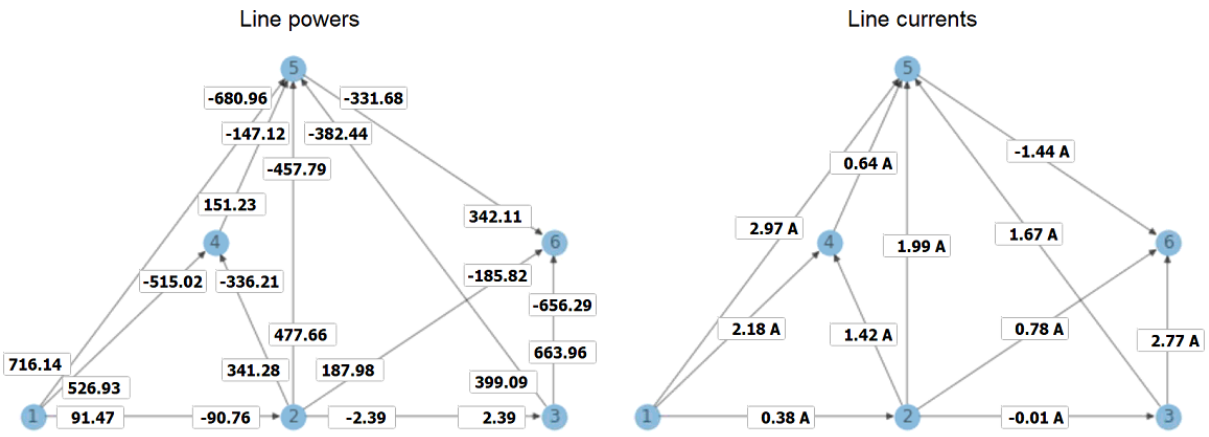


Figure 4.43: State of the lines after the first `OPF` of scenario 2

The power flow equalizes the power sharing between the three generating buses.

4.3.2.2 Perturbation 2

The price for energy in bus 2 is increased. The value of c_2 is now 0.12.

After the perturbation:

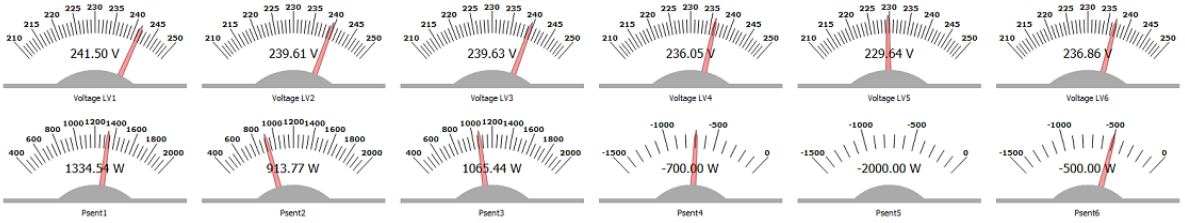


Figure 4.44: State of the buses after the second perturbation of scenario 2

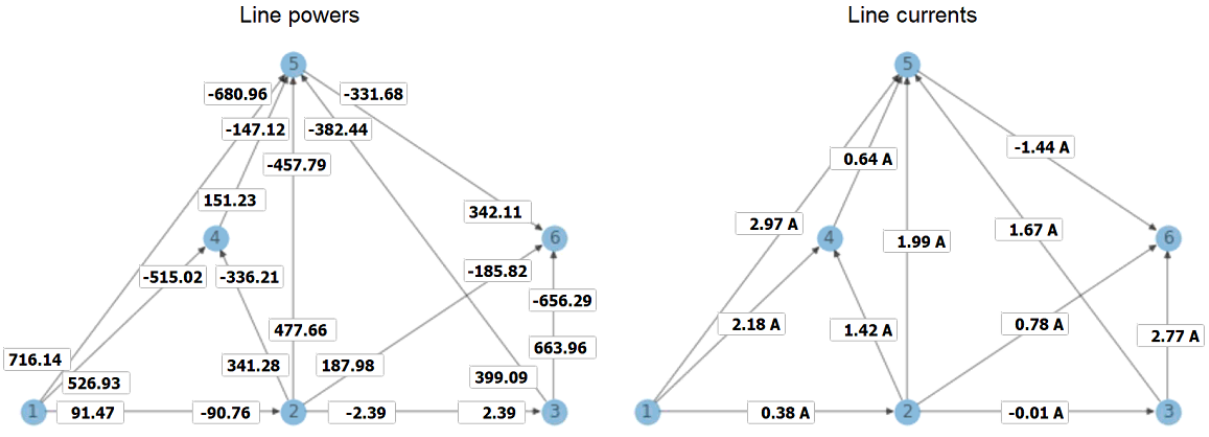


Figure 4.45: State of the lines after the second perturbation of scenario 2

After this change in the model, the power flow stays exactly the same. Indeed, a change in a price coefficient does not modify the physical state of the system. It only modifies the value of the price function and by extension, the optimal state of the network.

After the OPF:

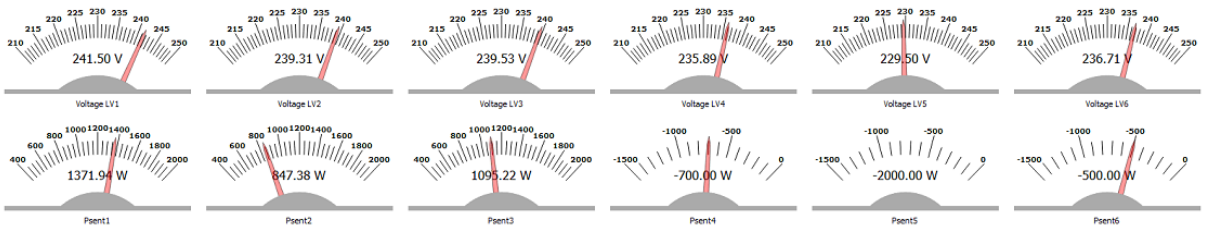


Figure 4.46: State of the buses after the second `OPF` of scenario 2

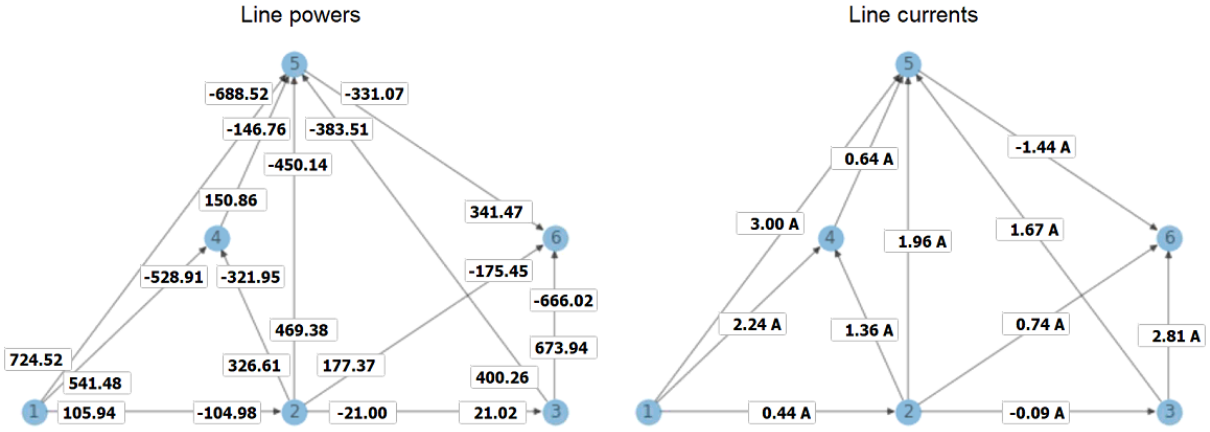


Figure 4.47: State of the lines after the second `OPF` of scenario 2

As expected, the value power produced by generator 2 is decreased and the two other generators provide the difference in power.

4.3.2.3 Perturbation 3

Lines {45} and {56} are lost. A contactor is opened in the `Typhoon HIL` model and they are removed from the network in `Python`.

After the perturbation:

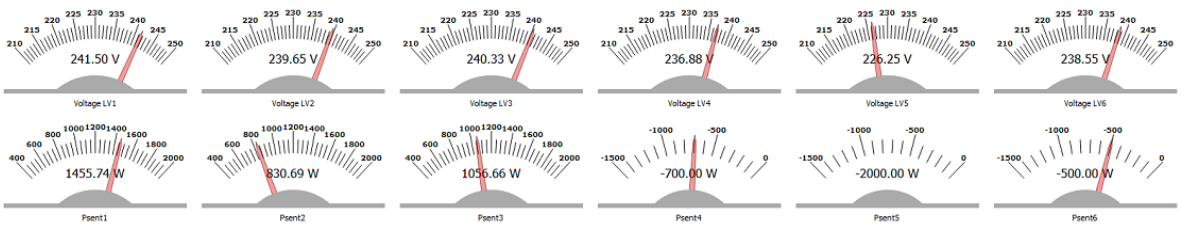


Figure 4.48: State of the buses after the third perturbation of scenario 2

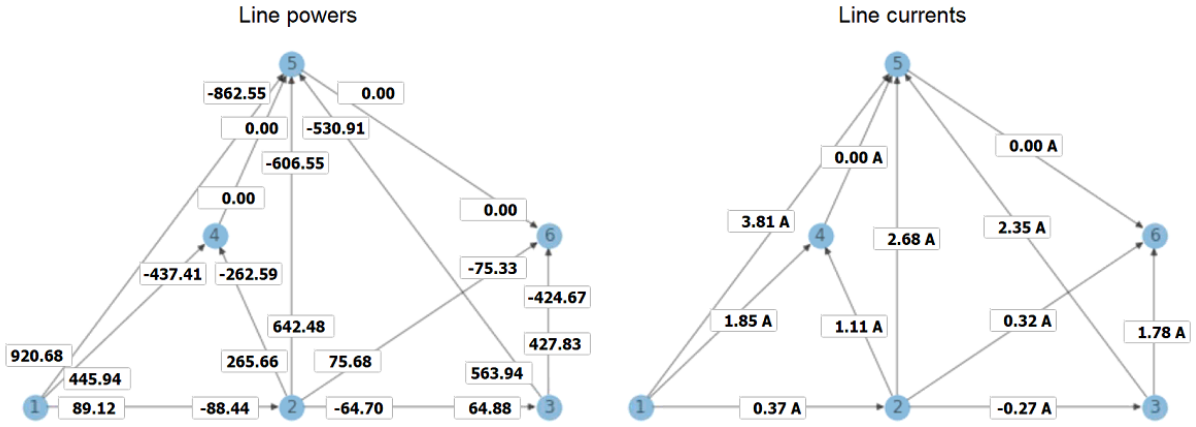


Figure 4.49: State of the lines after the third perturbation of scenario 2

After the two lines are lost, the generating buses adapt themselves using droop for the power flows in the system to be correct again. Bus 1 still has a slack role. The network still operates at a feasible point after this perturbation.

After the OPF:

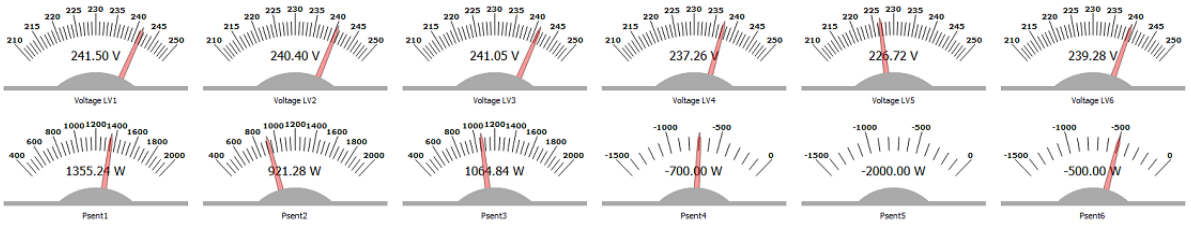


Figure 4.50: State of the buses after the third OPF of scenario 2

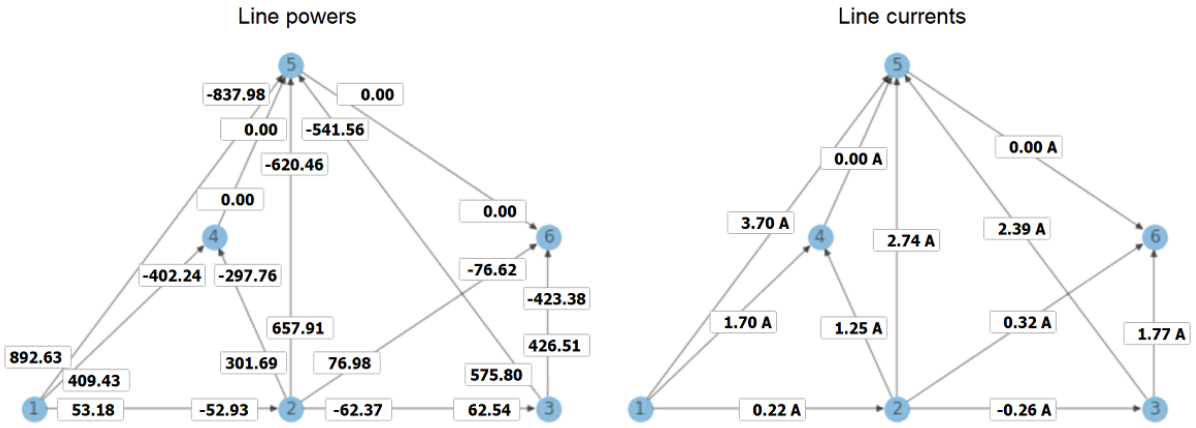


Figure 4.51: State of the lines after the third OPF of scenario 2

The OPF regulates the power flows in the modified network. The complete state of the system does not change drastically because the loss of the lines did not put any element outside of its admissible range of values.

Chapter 5

Conclusion

The goal of this thesis was to investigate a distributed algorithm for optimal control of DC microgrids and to implement it on a real-time controller. This work is introductory in both of these topics. It has been developed from the ground up, based only on research papers. The main goal was to investigate these topics and develop methodologies.

Regarding the distribution of the `OPF` problem for tertiary control, the approach did not give satisfactory results. This is unfortunate, because it was at the core of this subject. The paper from which the method originates [45] states that the technique is robust and indubitably converging. However, the authors only give theoretical proof of convergence. The detailed approach and how they managed to implement the algorithm are let for the reader to discover.

It can be noted that some of the bus primal values converged to a point close to the centrally computed optimum. This is encouraging because if these values are used as setpoint for the control, the state of the model may not be very different from the optimal. However, this is purely speculatory, and a distributed controller should not be implemented without certainty of the exactitude of the solution.

Other factors to be considered are the penalty terms (y_i and z_i). It seems that the constraints where those were used had better convergence rates. A test has been conducted with more penalty terms with no success. However, another technique could help to achieve faster and more stable convergence: Alternating Direction Method of Multipliers (ADMM). This is the method used in [43] and [44].

ADMM has been proven to be efficient against distributed optimization [56]. It is an adaptation of the augmented Lagrangian technique, using partial updates. This method is not exact but gives a very close approximation with an increased convergence speed.

Regarding the real-time control using `Typhoon HIL`, the method shows very promising results. HIL systems are a very efficient method for the simulation of electrical systems. Chapter 4 shows that both primary and tertiary controllers can be implemented relatively easily using this tool. More detailed models could be created using that tool. It could also be used to monitor and control physical networks, by using the inputs and outputs of the machine.

However, to obtain a more accurate simulation of the dynamics of the network, the model needs to be improved. A realistic model was outside the scope of this thesis as it focused primarily on the control problem. This is why the real-time simulation misses proper dynamic results.

A more practical case study, with the exact model of a data center power system, would deliver a more insightful base material for this type of research. Knowing the exact needs, the precise dynamics and the correct implementation of primary control in such a network would allow to have a better understanding of the tertiary controller mechanics and capabilities in a practical situation.

In the present case, with only basic modeling, CHAPTER 4 demonstrates a proof-of-concept for the dynamic control and study of DC microgrids using `Typhoon HIL`. This test is still conclusive given that the scope of this paper did not include advanced modeling of the network.

To conclude, this thesis is a starting point for both the study of distributed `OPF` algorithms and real-time tertiary control in DC microgrids. Given that these two topics are rather recent, many challenges still have to be tackled. Therefore this work is a first step towards a better understanding of the distributed real-time tertiary control of DC microgrids, suggesting several paths for improvements and further research.

Appendix A

Stability of the 6 bus distributed solution

To check the stability of the convergence, the variable values of SECTION 3.1.2 are used as a starting point for the distributed algorithm. Starting from a point nearly exactly at the optimum, the algorithm should not diverge from the initial values.

From this optimal starting point, the algorithm iterates 100,000 times with a very small step size of $5e{-}5$. Theoretically, the variable values should not deviate away from that. The following plots show that this is unfortunately not the case.

A.1 Primal variable values

For the primal variables, the variation is low, except for the current variables, which quickly begin to oscillate.

A.1.1 Bus primal variables

	p_g [W]	V [V]	$\hat{p}$ [W]
Generator 1	500	240.64	1000.65
Generator 2	880.14	241.50	1422.29
Generator 3	744.97	241.22	1273.50
Generator 4	$3.31e{-}11$	237.33	342.59
Generator 5	$3.31e{-}11$	236.93	323.67
Generator 6	$3.31e{-}11$	238.72	408.86

Table A.1: Control variables of each bus after 100,000 iterations

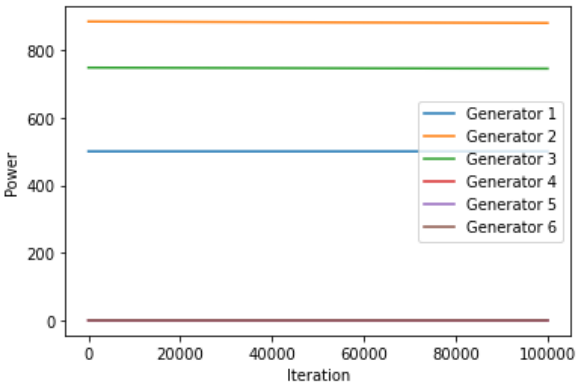


Figure A.1: Evolution of p^g over 100,000 iterations

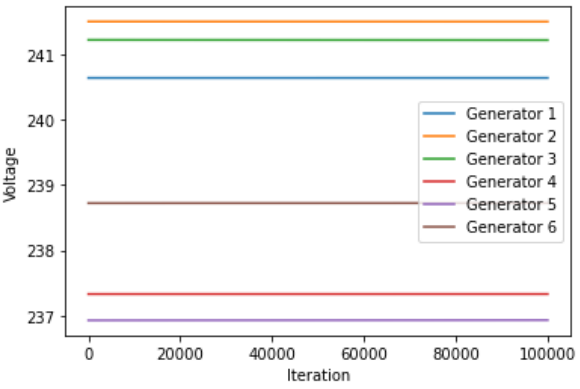


Figure A.2: Evolution of V over 100,000 iterations

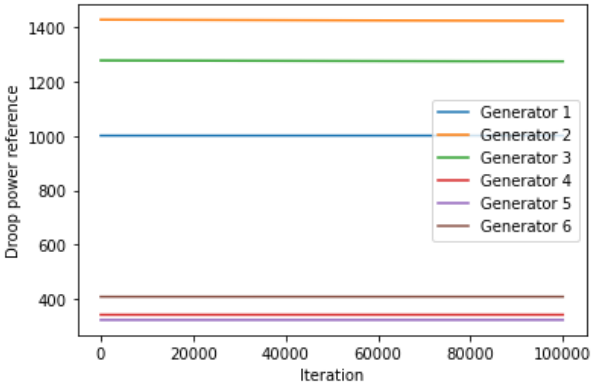


Figure A.3: Evolution of $\hat{p}$ over 100,000 iterations

A.1.2 Line powers

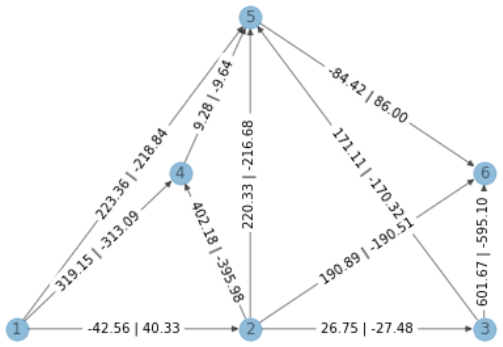
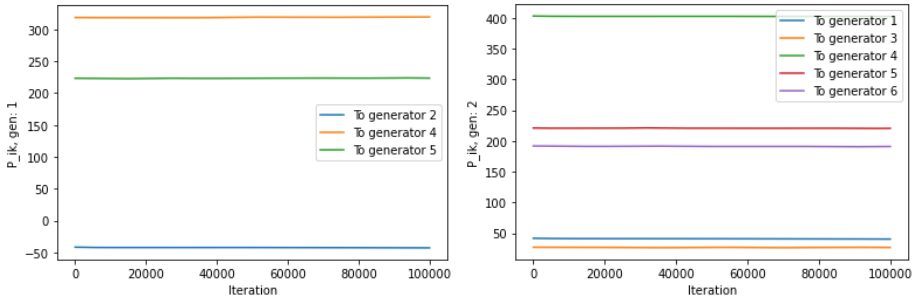
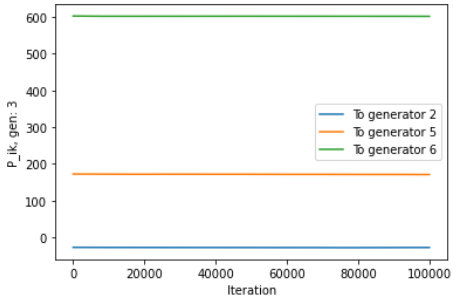


Figure A.4: Line power variables after 100,000 iterations

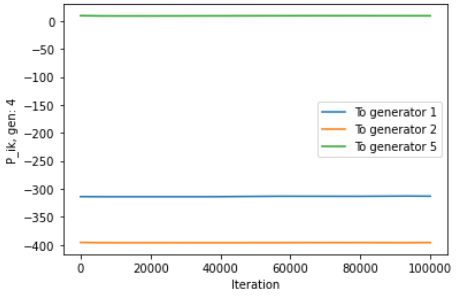


(a) Generator 1

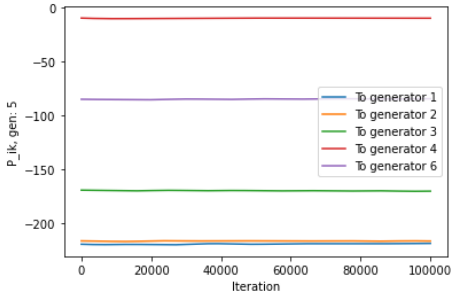
(b) Generator 2



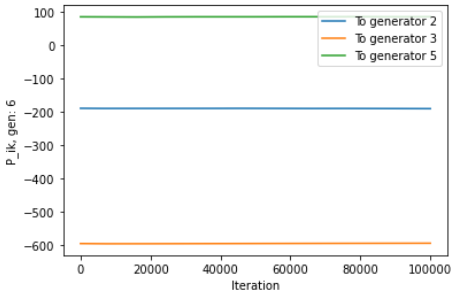
(c) Generator 3



(d) Generator 4



(e) Generator 5



(f) Generator 6

Figure A.5: Evolution of the values of the line powers over 100,000 iterations

A.1.3 Squared line currents

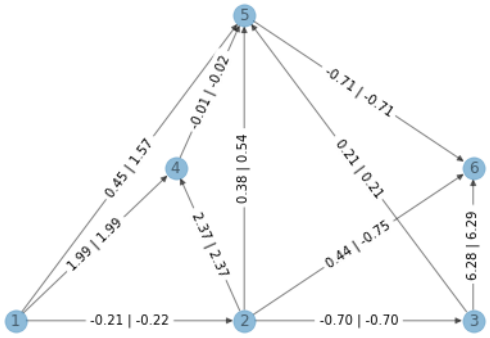
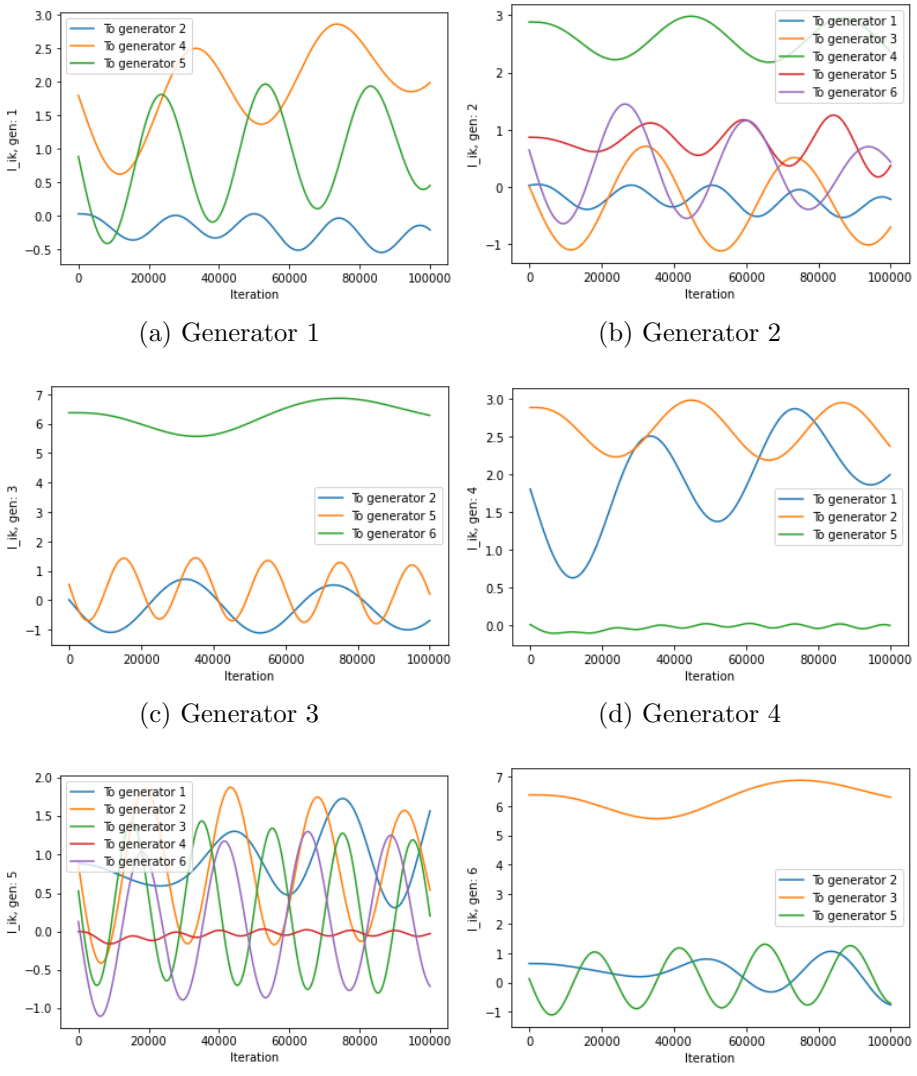


Figure A.6: Squared line current variables after 100 000 iterations



(a) Generator 1

(b) Generator 2

(c) Generator 3

(d) Generator 4

(e) Generator 5

(f) Generator 6

Figure A.7: Evolution of the values of the squared currents over 100,000 iterations

A.2 Dual variable values

Most of the dual variables show big variations or oscillations, especially line power to current balance dual variables and line power to voltage balance dual variables.

A.2.1 Bus dual variables

	ϵ	μ
Generator 1	$9.98e{-4}$	-0.95
Generator 2	$-6.88{-2}$	-0.16
Generator 3	$-4.49e{-2}$	0.33
Generator 4	$2.27e{-12}$	1.24
Generator 5	$-3.18e{-12}$	1.24
Generator 6	$2.83e{-18}$	1.22

Table A.2: Dual variable values for each bus

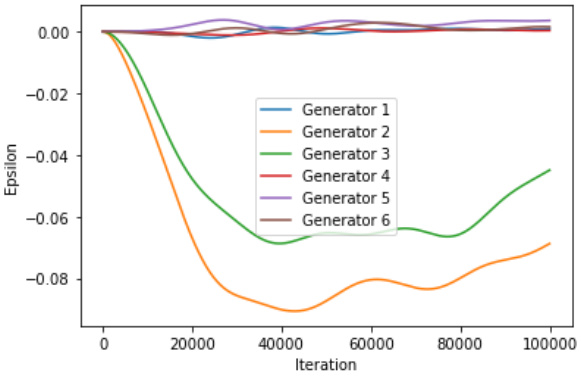


Figure A.8: Evolution of the value of the dual variable ϵ

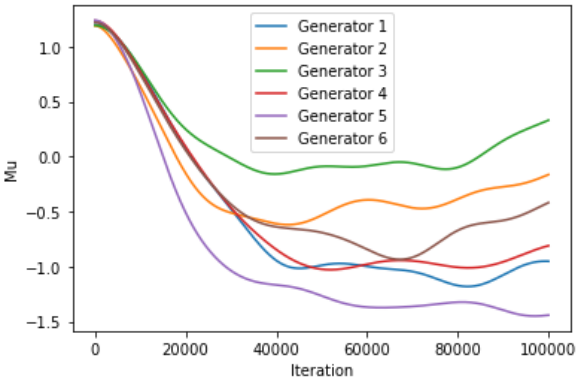


Figure A.9: Evolution of the value of the dual variable μ

A.2.2 Line power to current balance dual variables

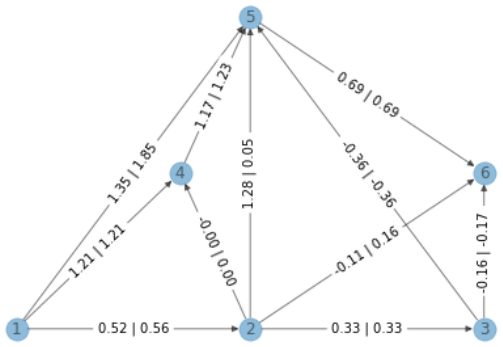
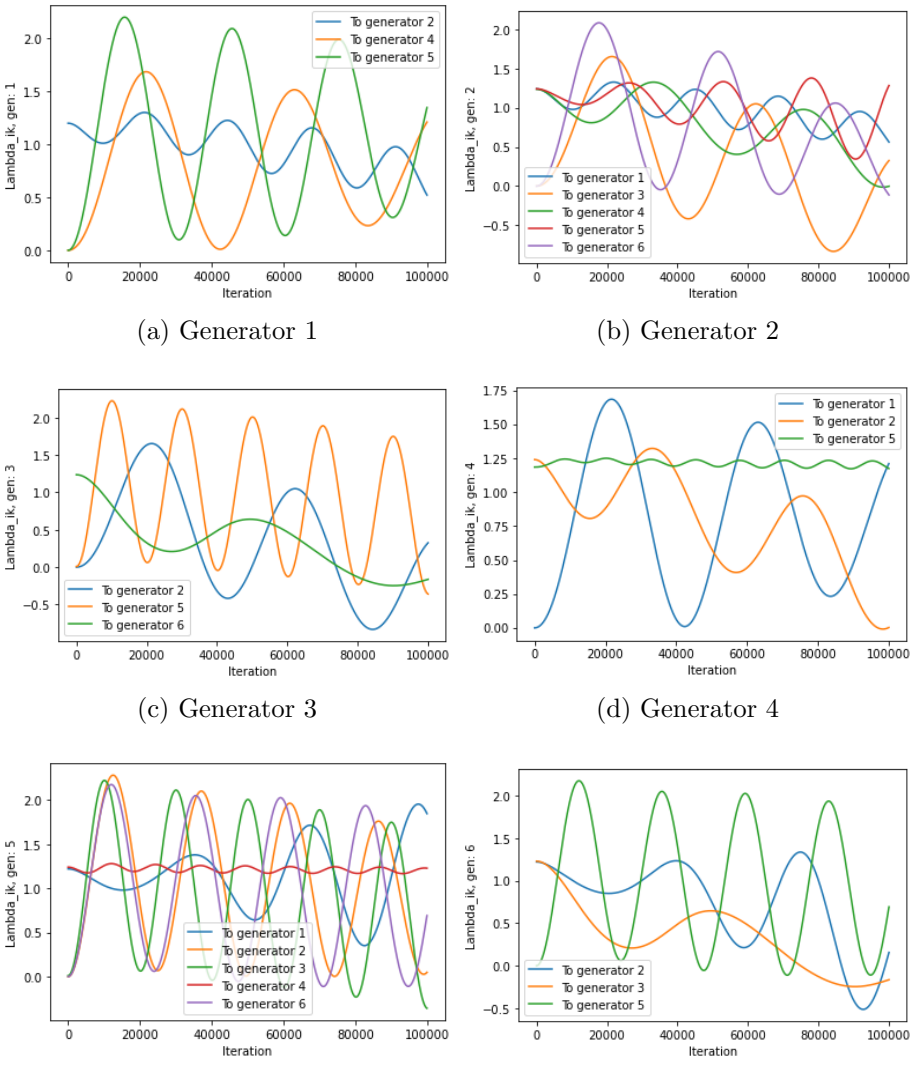


Figure A.10: Line power to current balance dual variables (λ) after 100 000 iterations



(a) Generator 1

(b) Generator 2

(c) Generator 3

(d) Generator 4

(e) Generator 5

(f) Generator 6

Figure A.11: Evolution of the values of λ dual variables over 100,000 iterations

A.2.3 Line power to voltage balance dual variables

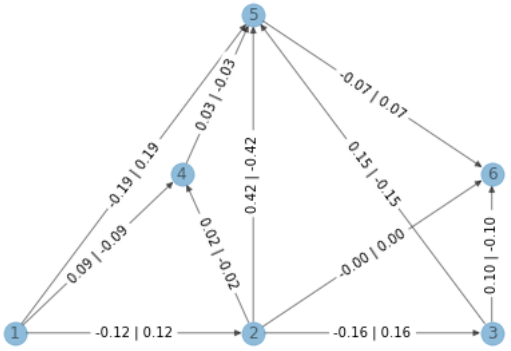
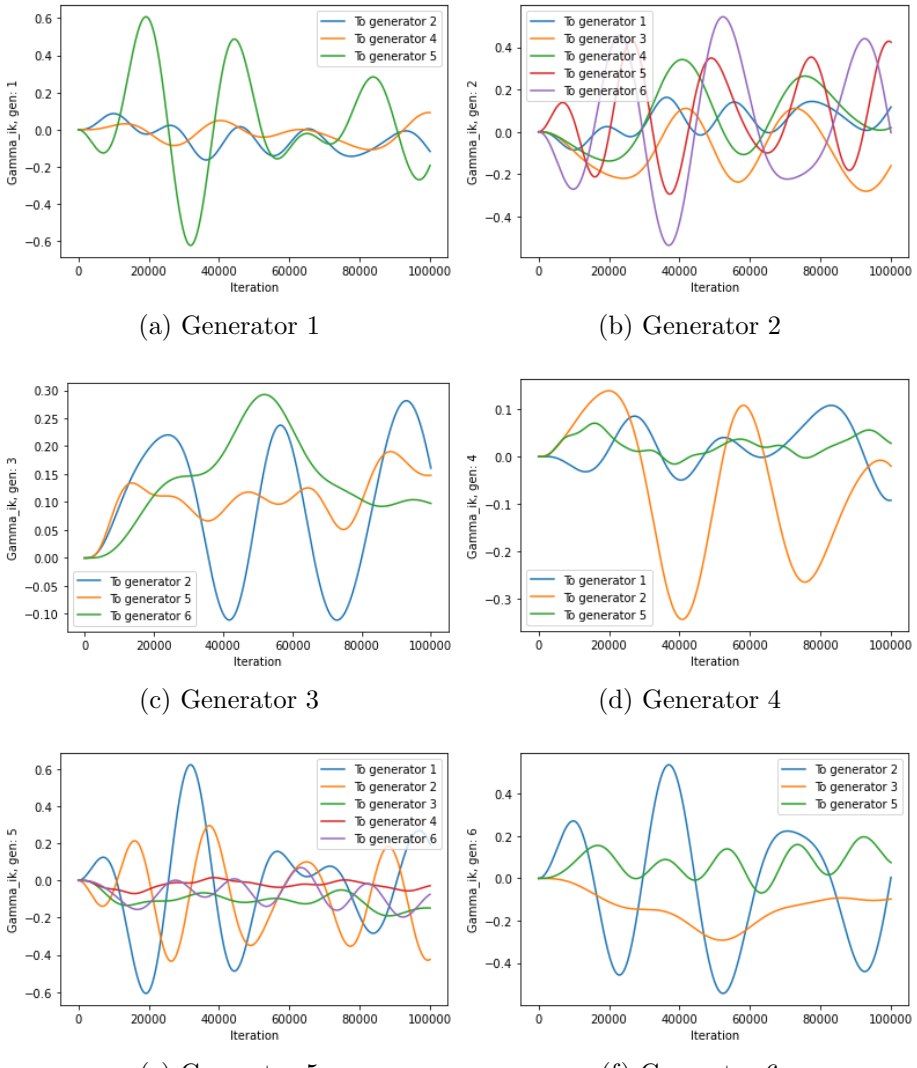


Figure A.12: Line power to voltage balance dual variables (γ) after 100,000 iterations



(a) Generator 1

(b) Generator 2

(c) Generator 3

(d) Generator 4

(e) Generator 5

(f) Generator 6

Figure A.13: Evolution of the values of γ dual variables over 100,000 iterations

A.2.4 SOCP relaxation constraint dual variables

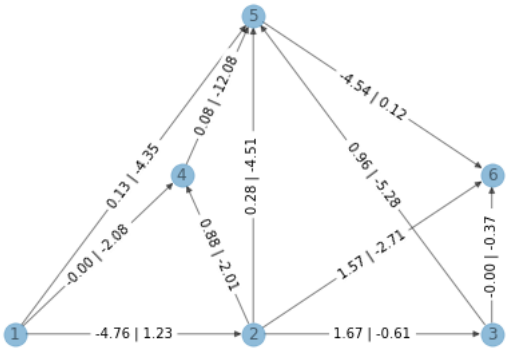
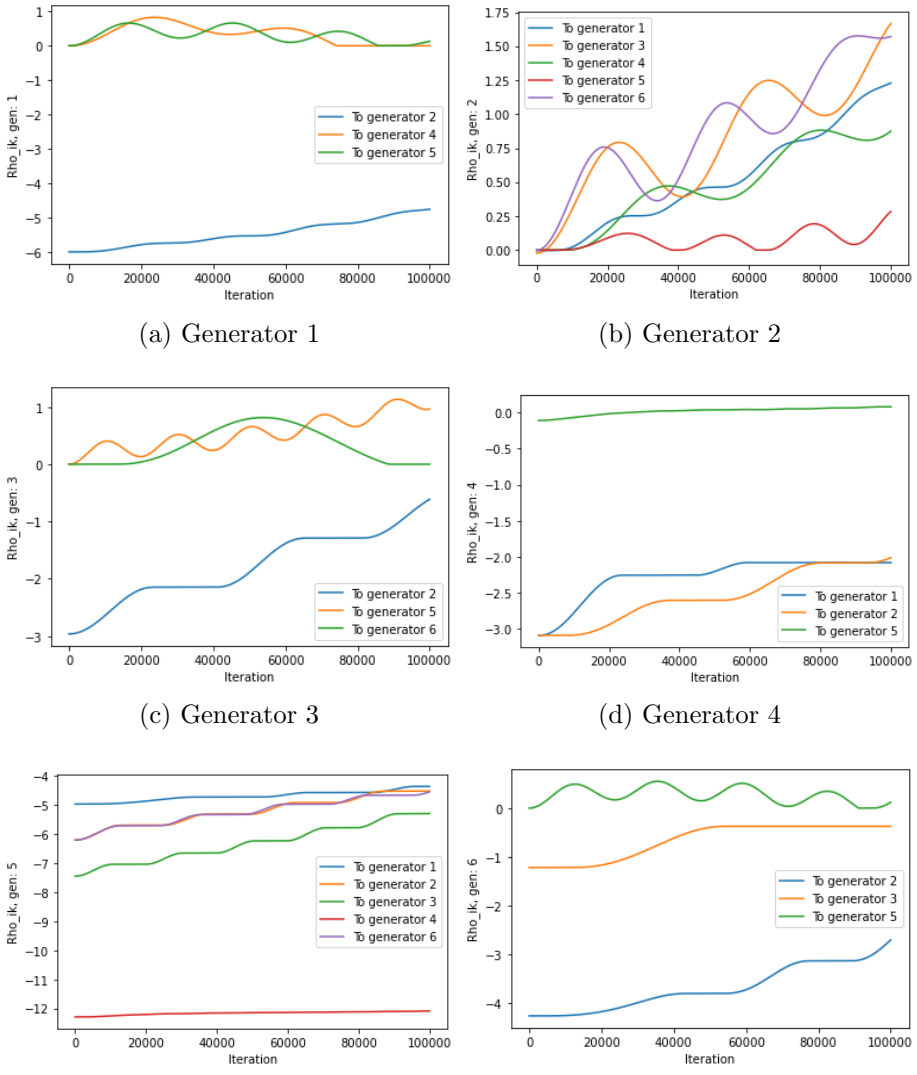


Figure A.14: SOCP relaxation constraint dual variables (ρ) after 100,000 iterations



(a) Generator 1

(b) Generator 2

(c) Generator 3

(d) Generator 4

(e) Generator 5

(f) Generator 6

Figure A.15: Evolution of the values of ρ dual variables over 100,000 iterations

Those results are not promising. Going deeper in the search on this algorithm does not seem profitable. The fact that the starting point is the optimum of the optimisation problem computed centrally should not bring a divergence. The previously computed values are supposed to be extremely close to the point minimizing the cost function. However, some of the variable values diverge using this algorithm.

It can be noted that the bus primal values do not seem to diverge. That can be beneficial when only considering the control point of view. Indeed, those values are the ones that are used as reference in the bus controllers. If implemented dynamically, it is unsure how the network will react. But it is still possible that the algorithm is able to find correct values for these variables that are critical. In practice, the correctness of the power/voltage/droop command is sufficient for the stability of the network. The other parameters like line power and voltages are only consequences of these setpoints.

Still, it may not be a good idea to implement such a method without further research and without solving the convergence issue. From a purely intuitive point of view, it seems that the power, droop control and boundaries constraints are met, while minimizing the objective function. While the other ones are not very stable and oscillating around the optimum value. This may be solved by the usage of more penalty terms associated to the other constraint of the problem. This would maybe improve the convergence speed and increase the stability of the iterative method.

This lack of convergence and unstability are the main reasons that justify the potential need of using ADMM methods.

Appendix B

3 Bus algorithm

B.1 Initialization

```
r12 = 0.4
r23 = 0.6

P1d = 20
P2d = 40
P3d = 30

k1 = 1

def c1(P1g):
  return 0.5 + 0.4*P1g

v1ref = 230**2

Vmax = 253
Vmin = 207
P1max = 110
```

```
v1 = 230.0**2
v2 = 230.0**2
v3 = 230.0**2

P1g = 100.0
P1ref = 95.0

P12 = 90.0
P21 = -90.0
P23 = 60.0
P32 = -60.0

l12 = 1.0
```

```
l21 = 1.0
l23 = 0.5
l32 = 0.5

mu1 = 1.0
mu2 = 1.0
mu3 = 1.0

eps1 = 1.0

lambda12 = 5.0
lambda21 = 5.0
lambda23 = 4.0
lambda32 = 4.0

gamma12 = 1.0
gamma21 = 1.0
gamma23 = 2.0
gamma32 = 2.0

rho12 = -1.0
rho21 = -0.9
rho23 = -0.8
rho32 = -0.7
```

B.2 Functions

```
def get_update1():
  y1 = v1 - v1ref + k1*(P1g-P1ref)
  z1 = P1g - P1d - P12

  dP1g = min(P1max, max(0, P1g - (c1(P1g) + z1 + k1*y1 + mu1 + eps1*k1))) - P1g
  dP1ref = -(-k1*(y1+eps1))
  dv1 = min(Vmax**2, max(Vmin**2, v1-(y1 + gamma12 + eps1 - rho12*(P12**2/v1**2)))) - v1

  dP12 = -(-z1 - mu1 + lambda12 - r12*gamma12 - 2*rho12*P12/v1)
  dl12 = - (-lambda12*r12 + rho12 + rho21)

  dmu1 = P1g - P1d - P12
  deps1 =  v1 - v1ref +k1*(P1g - P1ref)

  dlambda12 = P12 + P21 - r12*l12
  dgamma12 = v1 - v2 - r12*(P12-P21)
  drho12 = l12 - P12**2/v1

  return dP1g, dP1ref, dv1, dP12, dl12, dmu1, deps1, dlambda12, dgamma12, drho12
```

```
20
21
22  def update1(step):
23    P1g_ = P1g + step*dP1g
24    P1ref_ = P1ref + step*dP1ref
25    v1_ = v1 + step*dv1
26
27    P12_ = P12 + step*dP12
28    l12_ = l12 + step*dl12
29
30    mu1_ = mu1 + step*dmu1
31    eps1_ = eps1 + step*deps1
32
33    lambda12_ = lambda12 + step*dlambda12
34    gamma12_ = gamma12 + step*dgamma12
35
36    if rho12 + step*drho12 <= 0:
37      rho12_ = rho12 + step*drho12
38    else:
39      rho12_ = 0
40
41    return P1g_, P1ref_, v1_, P12_, l12_, mu1_, eps1_, lambda12_, gamma12_, rho12_
42
43
44  def get_update2():
45    z2 = - P2d - P21 - P23
46
47    dv2 = min(Vmax**2, max(Vmin**2, v2-(-gamma21 + gamma23 - rho21*(P21/v2)**2 -
48                                        rho23*(P23/v2)**2))) - v2
49
50    dP21 = -(-z2 - mu2 + lambda21 + r12*gamma21 - 2*rho21*P21/v2)
51    dP23 = -(-z2 - mu2 + lambda23 - r23*gamma23 - 2*rho23*P23/v2)
52
53    dl21 = - (-lambda21*r12 + rho12 + rho21)
54    dl23 = - (-lambda23*r23 + rho23 + rho32)
55
56    dmu2 = - P2d - P21 - P23
57
58    dlambda21 = P12 + P21 - r12*l21
59    dlambda23 = P23 + P32 - r23*l23
60
61    dgamma21 = v1 - v2 - r12*(P12-P21)
62    dgamma23 = v2 - v3 - r23*(P23-P32)
63
64    drho21 = l21 - P21**2/v2
65    drho23 = l23 - P23**2/v2
66
67    return dv2, dP21, dP23, dl21, dl23, dmu2, dlambda21, dlambda23, \
```

```
68            dgamma21, dgamma23, drho21, drho23
69
70  def update2(step):
71    v2_ = v2 + step*dv2
72
73    P21_ = P21 + step*dP21
74    l21_ = l21 + step*dl21
75    P23_ = P23 + step*dP23
76    l23_ = l23 + step*dl23
77
78    mu2_ = mu2 + step*dmu2
79
80    lambda21_ = lambda21 + step*dlambda21
81    gamma21_ = gamma21 + step*dgamma21
82    if rho21+step*drho21 <= 0:
83      rho21_ = rho21 + step*drho21
84    else:
85      rho21_ = 0
86
87    lambda23_ = lambda23 + step*dlambda23
88    gamma23_ = gamma23 + step*dgamma23
89    if rho23+step*drho23 <= 0:
90      rho23_ = rho23 + step*drho23
91    else:
92      rho23_ = 0
93
94
95    return v2_, P21_, P23_, l21_, l23_, mu2_, lambda21_, lambda23_,\
96          gamma21_, gamma23_, rho21_, rho23_
97
98
99  def get_update3():
100   z3 = - P3d - P23
101
102   dv3= min(Vmax**2, max(Vmin**2, v3-(-gamma32 - rho32*(P32**2/v3**2)))) - v3
103
104   dP32 = -(-z3 - mu3 + lambda32 + r23*gamma32 - 2*rho32*P32/v3)
105   dl32 = - (-lambda32*r23 + rho23 + rho32)
106
107   dmu3 = - P3d - P32
108
109   dlambda32 = P23 + P32 - r23*l32
110   dgamma32 = v2 - v3 - r23*(P23-P32)
111   drho32 = l32 - P32**2/v3
112
113   return dv3, dP32, dl32, dmu3, dlambda32, dgamma32, drho32
114
115
```

```
116 def update3(step):
117   v3_ = v3 + step*dv3
118
119   P32_ = P32 + step*dP32
120   l32_ = l32 + step*dl32
121
122   mu3_ = mu3 + step*dmu3
123
124   lambda32_ = lambda32 + step*dlambda32
125   gamma32_ = gamma32 + step*dgamma32
126
127   if rho32 + step*drho32 <= 0:
128     rho32_ = rho32 + step*drho32
129   else:
130     rho32_ = 0
131
132   return v3_, P32_, l32_, mu3_, lambda32_, gamma32_, rho32_
```

B.3 Iterative solving

```
1  import numpy as np
2  step = 0.01
3
4  obj = []
5  normGrad1 = []
6  normGrad2 = []
7  normGrad3 = []
8
9  d1 = []
10 d2 = []
11 d3 = []
12
13 p1 = []
14 p2 = []
15 p3 = []
16
17 for i in range(40000):
18   #print(i)
19   dP1g, dP1ref, dv1, dP12, dl12, dmu1, deps1, \
20       dlambda12, dgamma12, drho12 = get_update1()
21   dv2, dP21, dP23, dl21, dl23, dmu2, dlambda21, dlambda23, \
22       dgamma21, dgamma23, drho21, drho23 = get_update2()
23   dv3, dP32, dl32, dmu3, dlambda32, dgamma32, drho32 = get_update3()
24
25   d1.append([dP1g, dP1ref, dv1, dP12, dl12, dmu1, deps1, dlambda12, dgamma12, drho12])
26   d2.append([dv2, dP21, dP23, dl21, dl23, dmu2,
```

```
                dlambda21, dlambda23, dgamma21, dgamma23, drho21, drho23])
  d3.append([dv3, dP32, dl32, dmu3, dlambda32, dgamma32, drho32])

  P1g, P1ref, v1, P12, l12, mu1, eps1, lambda12, gamma12, rho12 = update1(step)
  v2, P21, P23, l21, l23, mu2, lambda21, lambda23, \
        gamma21, gamma23, rho21, rho23 = update2(step)
  v3, P32, l32, mu3, lambda32, gamma32, rho32 = update3(step)

  p1.append([P1g, P1ref, v1, P12, l12, mu1, eps1, lambda12, gamma12, rho12])
  p2.append([v2, P21, P23, l21, l23, mu2, lambda21, lambda23, gamma21, gamma23, rho21, rho23])
  p3.append([v3, P32, l32, mu3, lambda32, gamma32, rho32])

  obj.append(0.5*P1g + 0.2*P1g**2 + 0.5 * ((P1g - P1d - P12)**2+(v1 - v1ref +
            k1*(P1g-P1ref))**2 +(- P2d - P21 - P23)**2+(- P3d - P23)**2))

  normGrad1.append(np.linalg.norm(d1[-1]))
  normGrad2.append(np.linalg.norm(d2[-1]))
  normGrad3.append(np.linalg.norm(d3[-1]))

print("Cost = " + str(obj[-1]))
print("Gradient norms:")
print("Bus 1: " + str(np.linalg.norm(d1[-1]))
      + ", Bus 2: " + str(np.linalg.norm(d2[-1]))
      + ", Bus 3: " + str(np.linalg.norm(d3[-1])))
print()
print("P1g = " + str(P1g) + ", P1ref = " + str(P1ref))
print("V1 = " + str(v1**0.5) + ", V2 = " +str(v2**0.5) + ", V3 = " + str(v3**0.5))
print("P12=" + str(P12) + ", P21=" + str(P21) + ", P23=" + str(P23) + ", P32=" + str(P32))
print("l12=" + str(l12) + ", l21=" + str(l21) + ", l23=" + str(l23) + ", l32=" + str(l32))
```

```
Cost = 3466.225568429156
Gradient norms:
Bus 1: 0.12880764370495665, Bus 2: 0.2933343935928654, Bus 3: 0.28584194915234257

P1g = 90.03629789227988, P1ref = 128.30408239428115
V1 = 230.08318816310313, V2 = 229.96115353993173, V3 = 229.88318024234297
P12=70.03702879633873, P21=-69.99897502421005, P23=29.998340230249237, P32=-30.10590545086948
l12=0.093230651733711, l21=0.093230651733711, l23=0.071568440481263, l32=0.071568440481263
```

B.4 Perturbation

```
P1d = 20
P2d = 50
P3d = 30

```

```
5
6  for i in range(40000):
7    #print(i)
8    dP1g, dP1ref, dv1, dP12, dl12, dmu1, deps1, \
9          dlambda12, dgamma12, drho12 = get_update1()
10   dv2, dP21, dP23, dl21, dl23, dmu2, dlambda21, \
11         dlambda23, dgamma21, dgamma23, drho21, drho23 = get_update2()
12   dv3, dP32, dl32, dmu3, dlambda32, dgamma32, drho32 = get_update3()
13
14   d1.append([dP1g, dP1ref, dv1, dP12, dl12,
15              dmu1, deps1, dlambda12, dgamma12, drho12])
16   d2.append([dv2, dP21, dP23, dl21, dl23, dmu2,
17              dlambda21, dlambda23, dgamma21, dgamma23, drho21, drho23])
18   d3.append([dv3, dP32, dl32, dmu3, dlambda32, dgamma32, drho32])
19
20   P1g, P1ref, v1, P12, l12, mu1, eps1, lambda12, gamma12, rho12 = update1(step)
21   v2, P21, P23, l21, l23, mu2, lambda21, lambda23, \
22         gamma21, gamma23, rho21, rho23 = update2(step)
23   v3, P32, l32, mu3, lambda32, gamma32, rho32 = update3(step)
24
25   p1.append([P1g, P1ref, v1, P12, l12, mu1, eps1, lambda12, gamma12, rho12])
26   p2.append([v2, P21, P23, l21, l23, mu2, lambda21, lambda23, gamma21, gamma23, rho21, rho23])
27   p3.append([v3, P32, l32, mu3, lambda32, gamma32, rho32])
28
29   obj.append(0.5*P1g + 0.2*P1g**2 + 0.5 * ((P1g - P1d - P12)**2+(v1
30               - v1ref + k1*(P1g-P1ref))**2+(- P2d - P21 - P23)**2+(- P3d - P23)**2))
31
32 print("Cost = " + str(obj[-1]))
33 print("Gradient norms:")
34 print("Bus 1: " + str(np.linalg.norm(d1[-1]))
35       + ", Bus 2: " + str(np.linalg.norm(d2[-1]))
36       + ", Bus 3: " + str(np.linalg.norm(d3[-1])))
37 print()
38 print("P1g = " + str(P1g) + ", P1ref = " + str(P1ref))
39 print("V1 = " + str(v1**0.5) + ", V2 = " +str(v2**0.5) + ", V3 = " + str(v3**0.5))
40 print("P12=" + str(P12) + ", P21=" + str(P21) + ", P23=" + str(P23) + ", P32=" + str(P32))
41 print("l12=" + str(l12) + ", l21=" + str(l21) + ", l23=" + str(l23) + ", l32=" + str(l32))
```

```
Cost = 3853.0463642017
Gradient norms:
Bus 1: 0.007954139052246552, Bus 2: 0.018516201500924593, Bus 3: 0.018924818732675916

P1g = 100.05931431899863, P1ref = 139.8011277609948
V1 = 230.08637711202942, V2 = 229.94721216131953, V3 = 229.86888826773523
P12=80.05941835497212, P21=-80.01092689519338, P23=30.01072287900471, P32=-29.993585236070675
l12=0.121103666020251, l21=0.121103666020251, l23=0.0140754926405655, l32=0.0140754926405655
```

Bibliography

[1] U.S. Department of Energy, "DOE Microgrid Workshop Report," *Office of Electricity Delivery and Energy Reliability Smart Grid RD Program, San Diego, California.*, pp. 1–32, 2011.

[2] E. Hossain, E. Kabalci, R. Bayindir, and R. Perez, "A Comprehensive Study on Microgrid Technology," no. January, 2014.

[3] Elisa Wood, "What is a microgrid?." `https://microgridknowledge.com/microgrid-defined/`. Accessed: 09/08/2021.

[4] A. Kumar, D. M. A. Hussain, and M. Z. U. Khan, "Microgrids Technology: A Review Paper," *Gyancity Journal of Electronics and Computer Science*, vol. 3, no. 1, pp. 11–20, 2018.

[5] Gene Wolf, "A short history: The microgrid." `https://www.tdworld.com/digital-innovations/article/20970388/a-short-history-the-microgrid`. Accessed: 14/04/2021.

[6] Martin Anderson, "Microgrid: History, definition uses." `https://brdgstn.com/microgrid/`. Accessed: 14/04/2021.

[7] Lisa Cohn, "History of Microgrids in the US: From Pearl Street to Plug-and-Play." `https://microgridknowledge.com/history-of-microgrids/`. Accessed: 08/08/2021.

[8] GridBeyond.

[9] M. Taylor, *Evolution in the Global Energy Transformation to 2050*. 2020.

[10] S. Ullah, A. M. Haidar, P. Hoole, H. Zen, and T. Ahfock, "The current state of distributed renewable generation, challenges of interconnection and opportunities for energy conversion based dc microgrids," *Journal of Cleaner Production*, vol. 273, p. 122777, 2020.

[11] C. Marnay, N. Zhou, M. Qu, and J. Romankiewicz, "International Microgrid Assessment : Governance , INcentives , and Experience (IMAGINE)," no. June, 2012.

[12] D. Newman, "Right-Sizing the Grid," *Mechanical Engineering*, vol. 137, pp. 34–39, 01 2015.

[13] W. Wang and Z. Lu, "Cyber security in the smart grid: Survey and challenges," *Computer Networks*, vol. 57, no. 5, pp. 1344–1371, 2013.

[14] A. Hirsch, Y. Parag, and J. Guerrero, "Microgrids: A review of technologies, key drivers, and outstanding issues," *Renewable and Sustainable Energy Reviews*, vol. 90, pp. 402–411, 2018.

[15] Elisa Wood, "Microgrid benefits: Eight ways a microgrid will improve your operation...and the world." `https://microgridknowledge.com/microgrid-benefits-eight/`. Accessed: 09/08/2021.

[16] V. Nasirian, A. Davoudi, and F. L. Lewis, "Distributed adaptive droop control for dc microgrids," *Conference Proceedings - IEEE Applied Power Electronics Conference and Exposition - APEC*, pp. 1147–1152, 2014.

[17] D. Salomonsson, L. Söder, and A. Sannino, "An Adaptive Control System for a dc Microgrid for Data Centers," *IEEE Transactions on Industry Applications*, vol. 44, no. 6, pp. 1910–1917, 2008.

[18] F. Gao, R. Kang, J. Cao, and T. Yang, "Primary and secondary control in DC microgrids: a review," *Journal of Modern Power Systems and Clean Energy*, vol. 7, no. 2, pp. 227–242, 2019.

[19] S. K. Kollimalla, M. Mishra, A. Ukil, and G. H. Beng, "Dc grid voltage regulation using new hess control strategy," *2017 IEEE Power & Energy Society General Meeting*, pp. 1–1, 2017.

[20] L. E. Zubieta, "Are Microgrids the Future of Energy ? DC Microgrids from Concept to Demonstration to Deployment," *IEEE Electrification Magazine*, vol. 4, pp. 37–44, 6 2016.

[21] K. R. Bharath, M. Krishnan Mithun, and P. Kanakasabapathy, "A review on DC microgrid control techniques, applications and trends," *International Journal of Renewable Energy Research*, vol. 9, no. 3, pp. 1328–1338, 2019.

[22] T. L. Vandoorn, B. Renders, L. Degroote, B. Meersman, and L. Vandevelde, "Active load control in islanded microgrids based on the grid voltage," *IEEE Transactions on Smart Grid*, vol. 2, no. 1, pp. 139–151, 2011.

[23] M. M. A. Abdelaziz, M. F. Shaaban, H. E. Farag, and E. F. El-Saadany, "A multistage centralized control scheme for islanded microgrids with pevs," *IEEE Transactions on Sustainable Energy*, vol. 5, no. 3, pp. 927–937, 2014.

[24] B. Aluisio, S. Bruno, L. De Bellis, M. Dicorato, G. Forte, and M. Trovato, "DC-Microgrid operation planning for an electric vehicle supply infrastructure," *Applied Sciences (Switzerland)*, vol. 9, no. 13, 2019.

[25] X. Lu, K. Sun, J. M. Guerrero, J. C. Vasquez, L. . Huang, and R. Teodorescu, "SoC-Based Droop Method for Distributed Energy Storage in DC Microgrid Applications," *IEEE. Industrial Electronics*, pp. 28–31, 2012.

[26] W. Huang, M. Lu, and L. Zhang, "Energy procedia survey on microgrid control strategies," 2011.

[27] S. Liu, P. Su, and L. Zhang, "A virtual negative inductor stabilizing strategy for dc microgrid with constant power loads," *IEEE Access*, vol. 6, pp. 59728–59741, 2018.

[28] M. Su, Z. Liu, Y. Sun, H. Han, and X. Hou, "Stability analysis and stabilization methods of dc microgrid with multiple parallel-connected dc–dc converters loaded by cpls," *IEEE Transactions on Smart Grid*, vol. 9, no. 1, pp. 132–142, 2018.

[29] X. Zhang, A. Papachristodoulou, and N. Li, "Distributed optimal steady-state control using reverse- and forward-engineering," *Proceedings of the IEEE Conference on Decision and Control*, vol. 54rd IEEE Conference on Decision and Control,CDC 2015, no. Cdc, pp. 5257–5264, 2015.

[30] N. Hatziargyriou, *Microgrids: Architectures and Control*. Wiley-IEEE Press, 1 ed., 2013.

[31] T. Dragicevic, X. Lu, J. C. Vasquez, and J. M. Guerrero, "DC Microgrids - Part I: A Review of Control Strategies and Stabilization Techniques," *IEEE Transactions on Power Electronics*, vol. 31, no. 7, pp. 4876–4891, 2016.

[32] Y. Han, S. Member, X. Ning, P. Yang, and L. Xu, "Review of power sharing , voltage restoration and stabilization techniques in hierarchical controlled DC microgrids," *IEEE Access*, vol. PP, p. 1, 2019.

[33] J. Beerten and R. Belmans, "Analysis of power sharing and voltage deviations in droop-controlled dc grids," *IEEE Transactions on Power Systems*, vol. 28, no. 4, pp. 4588–4597, 2013.

[34] X. Lu, J. M. Guerrero, K. Sun, and J. C. Vasquez, "An improved droop control method for dc microgrids based on low bandwidth communication with dc bus voltage restoration and enhanced current sharing accuracy," *IEEE Transactions on Power Electronics*, vol. 29, no. 4, pp. 1800–1812, 2014.

[35] A. Khorsandi, M. Ashourloo, and H. Mokhtari, "A Decentralized Control Method for a Low-Voltage DC Microgrid," *IEEE Transactions on energy conversion*, 2014.

[36] S. E. Mhankale and A. R. Thorat, "Droop Control Strategies of DC Microgrid: A Review," *Proceedings of the 2018 International Conference on Current Trends towards Converging Technologies, ICCTCT 2018*, 2018.

[37] Y. Gu, X. Xiang, W. Li, and X. He, "Mode-Adaptive Decentralized Control for Renewable DC Microgrid With Enhanced Reliability and," *IEEE Transactions on Power Electronics*, vol. 29, no. 9, pp. 5072–5080, 2014.

[38] F. S. Al-Ismail, "Dc microgrid planning, operation, and control: A comprehensive review," *IEEE Access*, vol. 9, pp. 36154–36172, 2021.

[39] S. Peyghami, P. Davari, H. Mokhtari, P. C. Loh, and F. Blaabjerg, "Synchronverter-enabled dc power sharing approach for lvdc microgrids," *IEEE Transactions on Power Electronics*, vol. 32, no. 10, pp. 8089–8099, 2017.

[40] T. Morstyn, B. Hredzak, and V. G. Agelidis, "Control Strategies for Microgrids with Distributed Energy Storage Systems: An Overview," *IEEE Transactions on Smart Grid*, vol. 9, no. 4, pp. 3652–3666, 2018.

[41] Pouya Borazjani, Noor Izzri Abdul Wahab, Hashim b. Hizam, Azura Bt Che Soh, "A Review on Microgrid Control Techniques," pp. 749–753, 2014.

[42] L. Z. et al, "A review of control strategies in dc microgrid," *J. Phys*, p. 42035, 2018.

[43] A. Korompili, P. Pandis, and A. Monti, "Distributed OPF Algorithm for System-Level Control of Active Multi-Terminal DC Distribution Grids," *IEEE Access*, vol. 8, no. Cl, pp. 136638–136654, 2020.

[44] M. Wang, Y. Su, L. Chen, Z. Li, and S. Mei, "Distributed optimal power flow of DC microgrids: A penalty based ADMM approach," *CSEE Journal of Power and Energy Systems*, vol. 7, no. 2, pp. 339–347, 2021.

[45] Z. Wang, F. Liu, Y. Chen, S. H. Low, and S. Mei, "Unified Distributed Control of Stand-Alone DC Microgrids," *IEEE Transactions on Smart Grid*, vol. 10, no. 1, pp. 1013–1024, 2019.

[46] N. Eghtedarpour and E. Farjah, "Control strategy for distributed integration of photovoltaic and energy storage systems in DC micro-grids," *Renewable Energy*, vol. 45, pp. 96–110, 2012.

[47] S. Moayedi and A. Davoudi, "Distributed Tertiary Control of DC Microgrid Clusters," *IEEE Transactions on Power Electronics*, vol. 31, no. 2, pp. 1717–1733, 2016.

[48] T. Morstyn, B. Hredzak, G. D. Demetriades, and V. G. Agelidis, "Unified distributed control for DC microgrid operating modes," *IEEE Transactions on Power Systems*, vol. 31, no. 1, pp. 802–812, 2016.

[49] S. Vavasis, *Convex Optimization*. 1998.

[50] L. Gan and S. H. Low, "Optimal power flow in direct current networks," *IEEE Transactions on Power Systems*, vol. 29, no. 6, pp. 2892–2904, 2014.

[51] L. Thurner, A. Scheidler, F. Schäfer, J. Menke, J. Dollichon, F. Meier, S. Meinecke, and M. Braun, "pandapower — an open-source python tool for convenient modeling, analysis, and optimization of electric power systems," *IEEE Transactions on Power Systems*, vol. 33, pp. 6510–6521, Nov 2018.

[52] A. J. Wood and B. F. Wollenberg, *Power Generation Operation and Control*. 1996.

[53] W. E. Hart, C. D. Laird, J.-P. Watson, D. L. Woodruff, G. A. Hackebeil, B. L. Nicholson, and J. D. Siirola, *Pyomo–optimization modeling in python*, vol. 67. Springer Science & Business Media, second ed., 2017.

[54] A. Wächter and L.T. Biegler, "On the implementation of a primal-dual interior point filter line search algorithm for large-scale nonlinear programming.." `http://www.optimization-online.org/DB_HTML/2004/03/836.html`. Accessed: 27/04/2021.

[55] D. Feijer and F. Paganini, "Stability of primal-dual gradient dynamics and applications to network optimization," *Automatica*, vol. 46, pp. 1974–1981, 2010.

[56] N. Parikh, S. Boyd, N. Parikh, and S. Boyd, "Block splitting for distributed optimization," *Math. Prog. Comp*, vol. 6, pp. 77–102, 2014.