

Fully Distributed Optimal Power Flow for Low Voltage DC Grids

An optimisation solution using
physical measurements

Pedro Guilherme Lopes Parreira

Technische Universiteit Delft



Fully Distributed Optimal Power Flow for Low Voltage DC Grids

An optimisation solution using
physical measurements

by

Pedro Guilherme Lopes Parreira

in partial fulfillment of the requirements for the degree of

Master of Science
in Electrical Power Engineering

at the Delft University of Technology,
to be defended publicly on Friday November 8, 2019 at 09:00 AM.

Supervisor:	Dr. Ir. Milos Cvetkovic,	TU Delft
Advisors:	Dr. Ir. Laurens Mackay, Ir. Shantanu Chakraborty,	DC Opportunities R&D BV TU Delft
Thesis committee:	Prof. Dr. Peter Palensky, Dr. Ir. Milos Cvetkovic, Dr. Ir. Thiago Batista Soeiro,	TU Delft TU Delft TU Delft

This thesis is confidential and cannot be made public until November 8, 2024.

An electronic version of this thesis is available at <http://repository.tudelft.nl/>.

Acknowledgement

The presented work marks the end of the most challenging and exiting stage of my life so far. In the last five years I went from being an highschool graduate to (hopefully) a graduated engineer. It was been a hard road, one that I am sure I would not be able to walk if it was not for many people that I was lucky enough to have in my life.

I would like to dedicate the work that I here present to:

1. my Mom and Dad, which without any shadow of a doubt, have made me who I am today. Their example of work, dedication and kindness has been one that I try to follow every step of the way.
2. my sister Luísa and my brother Miguel, which I was lucky enough to have as siblings. I strive every day to be the best example for them and to learn from them as well.
3. Mel. Your unconditional support has helped in numerous ways and I can't be more happy to share this moment with you.
4. to Laurens and Shantanu, whose patience, guidance and advisory help me produce a piece of work which I am truly proud of.
5. to Prof. Peter Palensky, Dr. Ir. Milos Cvetkovic & Dr. Ir. Thiago Batista Soeiro thesis supervisors and committee member, for everything you have done.
6. and to all the friends I have made along the way, in Portugal and in the Netherlands, from university, SoSalsa and DC Opportunities. Every gesture of friendship has helped me become who I am.

I am here today because of you and for that I will never have the words to thank you.

I would also like to thank the people at the Municipality of Zoetermeer for kindly allowing me to use their information in this report.

As this stage ends, another one begins. The unknown might be scary, but in the words of a good friend of mine I find comfort for what comes ahead:

Life is tough, but we are tougher.

*Pedro Guilherme Lopes Parreira
Delft, November 2019*

Abstract

Multiple solutions for solving optimal power flow (OPF) have been employed, most of them using a centralised approach. However, there is another approach where every node is responsible for the local problem in order to reach a solution: the decentralised optimal power flow (D-OPF). In this thesis, the aim was to improve the speed and flexibility of a D-OPF algorithm based on the Consensus and Innovation (C+I) method using physical measurements from a direct current (DC) system. While in previous implementations, the system would work towards finding the solution for one point in time, the suggested implementation works by performing online optimisation, meaning that there is less time required in propagating information around the system and faster solutions are reached. A possible interaction between the physical and optimisation layers was suggested, to make online control feasible. Using current droop control for a DC system, it was possible to react to sudden changes the system and, in the long term, optimise the electrical resources. The improvements in speed were then demonstrated by the simulations results, where the time to reach the optimal solution was reduced, when compared to previous implementations. In order to increase the flexibility of the system, adaptive behaviour for the critical optimisation variables was suggested. To reduce the oscillatory behaviour of the system, some gains were made proportional to rate of change of said variables, meaning that the system didn't have to rely on user determine values in order to converge. It was also implemented a solution to calculate the line resistance between two nodes, further reducing the need for external inputs. These implementations were tested and it was concluded that it improved convergence speeds, while increasing the flexibility pf the system. Finally, a test case, based on a real existing lighting grid, was designed in order to test the algorithm under larger networks. The results showed that for a 25% increase in the size of the network there was no significant increase in the time required to reach a solution, indicating that the system can be scaled further, and might be dependent mainly on the network structure, and not its size.

keywords: DC, Optimal Power Flow, Consensus + Innovation, Droop Control, Online Optimisation, Adaptive Behaviour, Parameter Estimation

Contents

1	Introduction	1
1.1	Renewable Energy and Distributed Energy Resources	1
1.2	Meshed Low Voltage Grids and Microgrids	1
1.3	Consensus and Innovation as a Fully Distributed Optimisation Strategy	2
1.4	Research Motivation	2
1.5	Objective and Research Questions	3
1.5.1	Objective	3
1.5.2	Research Questions	3
1.6	Report Structure	3
2	Literature Review	5
2.1	Electrical Energy Market Economics	5
2.2	Grid Physics and Simulation	6
2.2.1	AC Power Flow	6
2.2.2	Exact DC Power Flow	6
2.2.3	Droop Control	7
2.3	Methods for Distributed OPF	8
2.3.1	Augmented Lagrangian Relaxation	8
2.4	KKT based Fully Distributed OPF methods	9
2.4.1	Karush-Kuhn-Tucker Conditions	9
2.4.2	Optimality Condition Decomposition	9
2.4.3	Distributed Interior Point Method	10
2.5	Consensus + Innovations	10
3	Physical Measurements for Increased D-OPF Convergence Rate	15
3.1	Grid Modelling	15
3.1.1	Power Flow equations	15
3.1.2	Droop Control	17
3.2	Nodal Equations	19
3.2.1	Problem Definition	19
3.2.2	Lagrangian Function	19
3.2.3	Karush Kuhn Tucker Conditions	20
3.2.4	Defining the updates	22
3.3	Power and Voltage Setpoints to control the Droop Curve	28
3.3.1	Building the droop curve	28
3.3.2	Droop in the Marginal Generator Region vs Constant Power Region	30
3.4	Simulation vs. Real Life Implementation: Pros and Cons	31
3.5	Synchronous Simulations	33
3.6	Asynchronous Behaviour	48
4	Adaptive Behaviour	53
4.1	Adaptive Optimisation of tuning parameters	53
4.1.1	Gain Adaptation	55
4.1.2	Voltage Error Integration	57
4.2	Power Supply and Demand Changes	63
4.3	Price Changes	67
4.4	Line Parameter Estimation	70
5	Case Study	73
5.1	Description of the Test Case	73
5.2	Test Results	74

6	Conclusion and Future Research	81
6.1	Conclusion	81
6.2	Future Research	83
A	Simulation Results	85
A.1	Results for the simulation of the 6 node grid with congestion	85
A.2	Results for the simulation of the 4 node grid, with voltage congestion, after tuning parameters were made adaptive.	89
A.3	Complete simulation 3 node grid subject to changes in power.	93
B	Schematics of Zoetermeer Lighting Grid	99
	Bibliography	101

Nomenclature

m	Index for the observed node
n	Index for the neighbor of the observed node
l	Iteration number
\mathcal{N}	Total number of nodes
Ω_m	List of nodes which are connected to node m
A_m	Quadratic cost coefficient
B_m	Linear cost coefficient
p_m^S	Power generation or load consumption in node m
$G_{m,n}$	Conductance of line m, n
u_m	Voltage at node m
$p_{m,n}$	Power flow in line m, n
$i_{m,n}$	Current flow in line m, n
$\bar{P}_{m,n}$	Power flow limit in line m, n
$\bar{I}_{m,n}$	Current flow limit in line m, n
\bar{P}_m^S	Maximum power generation at node m
\underline{P}_m^S	Minimum power generation at node m
\bar{U}_m	Maximum voltage at node m
\underline{U}_m	Minimum voltage at node m
\mathcal{L}	Lagrange function
λ_m	Locational marginal price (LMP) at node m
$\mu_{m,n}$	Dual variable for the line limit at line m, n
$\mu_m^{\bar{P}}$	Dual variable for the maximum power generation limit at node m
$\mu_m^{\underline{P}}$	Dual variable for the minimum power generation limit at node m
$\mu_m^{\bar{U}}$	Dual variable for the maximum voltage limit at node m
$\mu_m^{\underline{U}}$	Dual variable for the minimum voltage limit at node m
DC	Direct current
AC	Alternating current
OPF	Optimal power flow

1

Introduction

1.1. Renewable Energy and Distributed Energy Resources

Climate change is one of the big challenges humanity is facing. Proof of this is the amount of attention it has been gathering: from everyday people, striking for change, to world leaders, signing important global treaties, like the Paris Climate Accord, in order to reduce CO₂ emissions.

This shift in mentality, allied to the increase of electricity has a energy source [1], was led to an increase of demand for newer and cleaner energy solutions: Renewable Energy (**RE**). These new energy sources have seen a big push in development, especially wind generation, with big onshore and offshore wind farms being planned and constructed, as well as solar generation, with photovoltaic (**PV**) panels becoming less expensive and more efficient [2], making them, as time goes on, a more economic and environmentally friendly solution the more widely used fossil fuels.

One of the many advantage of using RE sources is the fact that these can be installed in regular homes, meaning homeowners, which could only be consumers, are now able produce their own energy and sell it back to the grid, becoming prosumers. This change in electrical power production from big fossil fuel based power plants to a system where there are smaller distributed energy resources (**DER**) is becoming increasingly likely, but is not without consequences.

This DER, for example, are harder to control because there is no centralised command centre from where power production can be dispatched since power generation might be dependent on weather conditions, in the case of RE, and can also be privately owned and, as such, is subject to privacy laws.

Also, in order to optimise these resources, it is not feasible to have it human controlled. As the system grows and becomes more complex, the amount of control variables becomes exponentially larger and users cannot be expected to manage such complex problems.

As such, to tackle and solve the presented problems, new optimisation strategies are needed.

1.2. Meshed Low Voltage Grids and Microgrids

The increasing number of prosumers will also lead to infrastructure problems, mainly on the low voltage (**LV**) distribution grids.

As it stands, the distribution grid is arranged radially, which is a good solution for a traditional power system, since power flows from medium voltage distribution level and is supplied first from top to bottom. For example, power would first be delivered to a neighbourhood LV grid, then to apartment blocks and finally to individual households. This meant that it would be easier to isolate and react to faults in the system.

However, with the introduction of micro production at the consumer level, the system faces problems which were unseen before. One of which is the lack of information in the distribution grid operator about private production of energy, which leads to sub-optimal line congestion control. This congestion could be an upwards flow of power, meaning power that is being generated in households and flowing upstream. This generated power also leads to less effective protection since the LV distribution system is not built to handle current flow in both directions.

A solution as been presented to tackle such problems: meshing the LV distribution grid, suggested in [3]. This new LV meshed grid would be better prepared to handle faults, since there would have in-built

redundancy, assuring that some power could flow in case of faults, as well as reducing increasing the solution space need to solve congestion problems.

Meshing the grid, however, has some obstacles, especially in alternating current (AC) electrical grids. For example, when connecting two different branches together, the difference in voltage angle, which result of different impedances, can create high currents and trip the protection modules. In [3], is suggested that direct current DC is used as an alternative since it is easier to mesh, current flow depends only on the voltage level, and because most renewable sources are already in DC: PV panels and batteries are inherently DC and wind turbines, in order to have flexible speed, are connected to the grid via two back to back AC-DC converters.

1.3. Consensus and Innovation as a Fully Distributed Optimisation Strategy

In order to reduce costs and better allocate electrical power resources, Optimal Power Flow (OPF) tools have been developed. In its core, OPF is an optimisation problem where the objective is to minimise the electrical power generation costs by despatching different generators, while still meeting the demand and subject to the constraints of the system, e.g, keeping the current under the maximum limit and voltage withing certain boundaries to maintain a stable system. This tool is widely used by transmission system operators [4] in order to keep the grid stable a better allocate the energetic resources.

This OPF calculations are usually performed in a centralised manner, meaning that the information is received from several substations, the optimisation calculation are performed centrally and the generation requirements are sent to power plants. These problems can scale very rapidly with the grid size and for bigger networks, there is a lot of computational power required to reach an optimal solution. There is also the drawback of performing OPF in a centralised way means there is a single point of failure, which can have hefty consequences.

For the scope of the operation of a transmission grids, running centralised OPF makes sense since there are usually only one or few TSO's, which responsible of assuring grid stability. However, in the LV grids, this solution might not be feasible anymore, since there are more players involved in the optimisation process and power supply and demand are more volatile. As such, a new way of preforming OPF becomes more viable: Decentralised OPF (D-OPF).

One of the proposed solutions to run D-OPF is called Consensus and Innovation (C+I). This algorithm is a fully distributed solution for the OPF problem, meaning there is no central controller and every area in the network, a node, is helping the system to reach the optimal solution. The inner workings of this algorithm will be addressed in the following chapters.

1.4. Research Motivation

Although progress has been made in order to make C+I based algorithms a viable solution, there are still some challenge wich remain to be solved. The first problem is that present solutions only solve OPF for one time step, meaning that, the optimal solution is calculated for one point in time and, if there is any shift in the network, the results are not valid anymore.

Another downside of suggested implementations is that there is no feedback from the grid, meaning there is no way to actually check if the setpoints defined by the optimisation process are being applied in the physical system.

A third problem that was found is the fact that tuning parameters of the optimisation process need to be set during the initialisation phase and remain immutable during the whole process, which means that the system might requires heavy human supervision.

And lastly, the whole convergence of the system to an optimal solution needs to be faster in order to make this solution applicable in the real world.

This project was developed with the aim of mitigating, or even eliminating this probelms, in order to path the way to a real world implementation.

1.5. Objective and Research Questions

1.5.1. Objective

The main objective of this thesis is *to improve the speed and flexibility of the C+I fully decentralised online optimisation algorithm for low voltage DC grids using physical measurements.*

1.5.2. Research Questions

In order to successfully meet the objective of this study, 4 different research questions would have to be answered:

1. How can physical measurements be used to improve the convergence rate?
2. What is the impact of communication loss on the convergence rate?
3. How can the optimisation parameters be adapted online to improve convergence rate?
4. What is the impact of changing supply and demand on the convergence rate?

1.6. Report Structure

This report is divided into six chapters. In chapter 1, an introduction to the report was given. In Chapter 2, a state of the art review is done, introducing the concepts that will support this thesis report: grid economics, DC power flow calculations, centralised and decentralised OPF algorithms.

Chapters 3 and 4 represent the main body of the report. In Chapter 3, the grid and droop equations are presented, as well as the formulation for the decentralised optimal power flow optimisation problem. The update strategy is described, the interaction between physical and cyber layers is explained and simulation results are shown. In Chapter 4, the adaptive behaviour of the algorithm is defined and the results are compared to previous implementations.

Chapter 5 presents the test case study of this algorithm on LV electrical power grid in Zoetermeer and, finally, in Chapter 6 the research questions are answered and conclusions are taken about the produced work. Suggestions for future work are given as well.

2

Literature Review

2.1. Electrical Energy Market Economics

In the 1880's, the first steps into building a electrical grid were taken. 140 years past, the transmission of electrical energy from the producer to the consumer became a very complex process with many parties involved. However the basic principle is still the same than other markets: there is a group of entities (*producers*) which produces a certain commodity and the other group (*consumers*) buy said commodity at a certain price [5]. On the electricity market, both entities trade in electrical energy, measured in MWh, for a certain time frame. This transaction can occur in two different ways: Decentralised or Centralised trading [5]. Decentralised trading occurs when producers and consumers have a contract for a certain amount of energy for a certain price, while in centralised trading all consumers and producers submit a bid on how much they are willing to buy/sell and at what price and a third entity, the system operator, gathers all the bids and sets the market clearing price. This last trading option is more relevant for the work developed on this thesis, and therefore it will be the one taken into account in this section.

The price that a producer is willing to sell electricity depends on the costs at which it can be generated. For optimisation purposes, this costs are approximated by a cost function $C(P)$, where C is the cost of generating P , in MWh. The nature of this function depends on the type of generator since different generation methods have different associated expenses. For most gas turbine and diesel fuelled generators, its cost function is usually of quadratic nature[6], while some fuel cell systems have a linear cost function, where its price scale mainly with the amount of fuel need to generate power[7]. Finally, renewable sources like photovoltaic panels have constant cost functions[8], depending on initial investments, for example. In figure 2.1 examples of these functions can be seen.

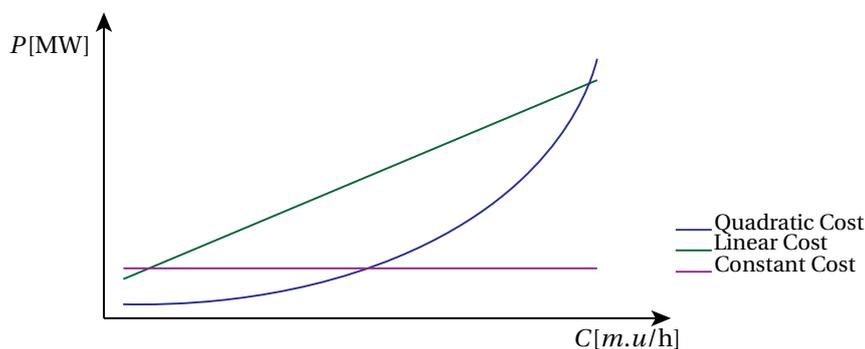


Figure 2.1: Cost function examples

At any given point in time, there is a electricity demand that must be met and so the market works to minimise the overall cost of production in order to meet said demand. This means that, in its core, it is an optimisation problem, which is called Optimised Power Flow (**OPF**)[5]. We can better see it in an example: There are m producers, each one with a cost function C_m , as shown in equation (2.1), and the only constraint

is that the total produced power must be the same as the total demanded power D_{Total} , like in equation (2.2), disregarding any losses in the system. This will be called the unconstrained case.

$$C_m = A_m P_m^2 + B_m P_m + K_m \quad (2.1)$$

$$\sum P_m = D_{Total} \quad (2.2)$$

where A_m, B_m and K_m are, respectively, the quadratic, linear and constant cost coefficients.

The optimisation problem is then solved by minimising all the cost functions, while subject to (2.2).

$$\min \sum C_m(P_m) \quad (2.3)$$

The solution to this problem are the values of P for which all $\frac{dC_m}{dP_m}$ are the same. This derivative is called Marginal Prices (**MP**), since it represents the added cost of increasing production.

The previous example might work for very broad studies, but in the case of this master thesis it does not suffice, since it does not account for any physical limitations of the grid.

For example, no line is capable of transferring an infinite amount of power and there are always some transmission losses on the system. Therefore, this marginal prices at the end of the optimisation might differ from the unconstrained case. Since this MP might vary throughout the network, they are called Locational Marginal Prices (**LMP**) [9].

2.2. Grid Physics and Simulation

2.2.1. AC Power Flow

Most of the installed electrical transmission capacity is in AC [10], therefore most of the tools that exist nowadays to make power flow calculations are also in AC [11]. Hence, power flow is calculated using the equations (2.4), for active power, and (2.5), for reactive power.

$$P_m = \sum_{n=1}^N |U_m| |U_n| (G_{mn} \cos \theta_{mn} + B_{mn} \sin \theta_{mn}) \quad (2.4)$$

$$Q_m = \sum_{n=1}^N |U_m| |U_n| (G_{mn} \sin \theta_{mn} - B_{mn} \cos \theta_{mn}) \quad (2.5)$$

where U_m is the voltage amplitude in node m , G_{mn} and B_{mn} are, respectively, the conductance and susceptance of the branch between node m and node n , and θ_{mn} is the difference between the voltage angles of nodes m and n .

The DC power flow calculations that often appear in literature are an approximation of AC PF, where the voltage magnitude U is taken as being 1 pu, the differences in θ over one line are taken as being very small, meaning $\cos \theta_{m,n} = 1$ and $\sin \theta_{m,n} = \theta_m - \theta_n$, and the line resistances are negligible. This assumptions mean that only active power P flows in the network, $Q_m = 0$, and it is dependent on the voltage angles θ and the line susceptance, as seen in (2.6).

$$P_m = \sum_{n=1}^N B_{mn} (\theta_m - \theta_n) \quad (2.6)$$

Although, for distribution grids this approximations is not be valid, the line resistance is not insignificant when compared to the reactance, and new formulations have been developed to tackle those problems [12].

2.2.2. Exact DC Power Flow

The focus of this thesis project is to preform distributed OPF in a DC network and therefore, this equations are not valid since they do not represent a DC system, where voltage angles and reactances do not exist.

In [13] the equations for Exact DC (**EDC**) Power Flow are show, in terms of current, for a bipolar DC network.

$$i_{m,n} = G_{m,n} \cdot (u_m - u_n) \quad (2.7)$$

$$i_m = \sum_{n|(m,n) \in \mathcal{G}} i_{m,n} - \sum_{n|(m,n) \in \mathcal{G}} i_{m,n} \quad (2.8)$$

$$-i_m = \sum_{n|(m,n) \in \mathcal{S}} \sum_{s|(m,n,s) \in \mathcal{S}} i_{m,n,s}^S - \sum_{n|(m,n) \in \mathcal{S}} \sum_{s|(n,m,s) \in \mathcal{S}} i_{n,m,s}^S \quad (2.9)$$

where $i_{m,n}$ is the current that flows from node m to node n , u_m and i_m are, respectively, the voltage and the total current at node m . Furthermore, $(m, n) \in \mathcal{G}$ is a pair of nodes with a connecting line and $(m, n, s) \in \mathcal{S}$ are the individual sources at nodes (m, n) .

Equation (2.7) is the application of Ohm's law to a branch between 2 nodes, while (2.8) is Kirchhoff's current law applied to a single node m . As for the 3rd equation, (2.9), it states that the source current of a node is the difference between the input current from a source and the drawn current from a load. Since generators and loads also have power limits, it is important to define the power of a source/load:

$$p_{m,n,s}^S = (u_m - u_n) \cdot i_{m,n,s}^S \quad (2.10)$$

where $p_{m,n,s}^S$ is the power of a load connected at (m, n) .

Then, running EDC-PF for any DC grid is a matter of obtaining the solution for a system of equations defined by (2.7)-(2.9).

2.2.3. Droop Control

This report studies a solution for DC OPF in a fully distributed algorithm, where there is no central controller. Therefore, every node of the network, if its a load or a generator, must be able to, until a certain point, be able to control its own voltage and power. A solution can be found on [14], where current droop control is suggested, meaning that the converter in that node could set its own power, current and voltage limits, as well as setting its own current-voltage (IV) characteristics. This would mean that, "(...) even if communication is lost, the system could continue to operate, increasing the resilience of the system." [14]. The suggested droop curve would look similar to the one presented on figure 2.2.

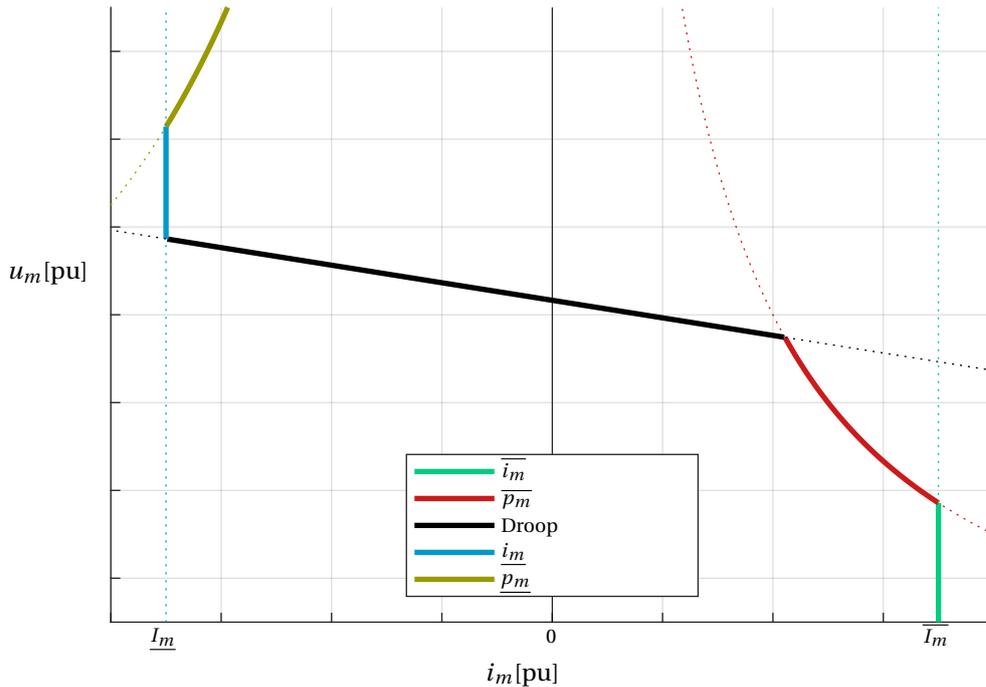


Figure 2.2: Example of a possible droop curve (without deadband)

In this figure it is possible to identify 5 segments which describe the behaviour of the converter depending on its own state. When the converter is supplying or demanding maximum and minimum current, respectively, its behaviour is mandated by the bright green and blue lines, meaning that its own voltage can change but not the current and, therefore, shown as a vertical line. When the converter is at maximum or minimum power, its behaviour is set by the red and olive green respectively, where both current and voltage can change but their product must be constant, i.e., $i \cdot u = \bar{P}$ meaning that $u = \bar{P}/i$, which translates to the hyperbola seen in Figure 2.2. For the case where the converter is not at any operational limit, the behaviour is set by a linear

function of $u = f(i)$, with slope d , called the droop, i.e.:

$$u = d \cdot i + K \quad (2.11)$$

where K the value of voltage for which the current $i = 0$. In [14], droop control is also shown with deadband, which is not present in figure 2.2 since it was not relevant for this project.

2.3. Methods for Distributed OPF

As it was mentioned in section 1, OPF can be done in a centralised, distributed or fully distributed fashion, with the later one being the focus of this report. Many algorithms have been researched for this purpose and studies comparing them have also been realised [15, 16]. These distributed methods can be classified in two groups, depending on the mathematical foundation: Augmented Lagrangian Relaxation (**ALR**) methods or Karush-Kuhn-Tucker (**KKT**) Condition methods, each one with different algorithms [15, 16]. In the following sections, a brief overview of the non used methods will be given.

2.3.1. Augmented Lagrangian Relaxation

Examples of ALR methods are the Analytical Target Cascading (**ATC**) method, the Auxiliary Problem Principle (**APP**) and the Alternating Direction Method of Multipliers (**ADMM**), like the solution presented by [17]. All of this methods require a central coordinator [15], hence, are not fully distributed and thus are not viable of being implemented in this project.

A solution is to use ADMM in combination with Proximal Message Passing (**PMP**) which no longer requires a central coordinator to operate and makes it a fully distributed algorithm [18]. To explain how it works, an example will be given.

Considering the follow optimisation problem:

$$\min_{x,y} f(x) + g(y) \quad (2.12)$$

with the following linear constraint:

$$Ax + By = c \quad (2.13)$$

where x and y are decision variables, A and B are constraint coefficients and c is the specified vector.

For this problem, there is the following augmented Lagrangian function:

$$\mathcal{L}(x, y, \lambda) = f(x) + g(y) + \lambda(Ax + By - c) + \frac{\rho}{2} \|Ax + By - c\|_2^2 \quad (2.14)$$

where $\rho > 0$ is a specified penalty parameter and $\|\cdot\|_2$ is the two norm for absolute value. The ADMM algorithm conducts decomposing the problem in sub-problems of minimizing each decision variables and the dual variable.

$$x(l+1) = \arg \min_x \mathcal{L}(x(l), y(l), \lambda(l)) \quad (2.15)$$

$$y(l+1) = \arg \min_y \mathcal{L}(x(l+1), y(l), \lambda(l)) \quad (2.16)$$

$$\lambda(l+1) = \lambda(l) + \rho(Ax(l+1) + By(l+1) - c) \quad (2.17)$$

where the variables are updated one after the other. In this case, a central coordinator is needed for (2.17) in order to update the dual variable value. Introducing PMP, this is no longer a requirement since every node evaluates a “prox” function:

$$\text{prox}_{f,\rho}(v) = \arg \min_x (f(x) + \rho/2 \|x - v\|_2^2) \quad (2.18)$$

where x is a vector that contains the primal (x, y) and dual (λ) variables and v is a vector of the average values of x in the nodes.

Applying this method the the decentralised OPF problem [18], we get as the decision variables the power plans, $f(x)$ is the local objective function and ρ is scalar for tuning parameter. The *prox* function optimises the local variables and sends it to the neighbouring nodes. The algorithm stops after all nodes agree on a common value of x , in the case of [18], the power P .

2.4. KKT based Fully Distributed OPF methods

The methods which will be described in the following sections are all based on KKT Conditions for optimality, and so it is required an explanation of what they are.

2.4.1. Karush-Kuhn-Tucker Conditions

The KKT Conditions, initially developed in [19, 20], is a method of optimising non-linear constrained problems. Given a certain function, the objective is to minimise said function, in regards to the KKT necessary conditions. Considering the following optimisation problem:

$$\min f(x) \quad (2.19)$$

subject to:

$$h_i(x) = 0 \quad \forall i \in \mathcal{N} \quad (2.20)$$

$$g_j \leq \bar{G}_j \quad \forall j \in \mathcal{M} \quad (2.21)$$

$$g_k \geq \underline{G}_k \quad \forall k \in \mathcal{M} \quad (2.22)$$

where i , j and k are the indexes for equality constraints and the inequality constraints respectively. The problem is to minimise $f(x)$, which is dependent on x and the solution x^* must follow the equality constraints $h_i(x)$ and must not violate maximum and minimum boundaries given by \bar{G} and \underline{G} respectively. In order to solve this problem it is necessary to first define the Lagrangian function \mathcal{L} as:

$$\mathcal{L}(x) = f(x) + \lambda_i h_i(x) + \mu_j^{\bar{G}}(g_j(x) - \bar{G}_j) + \mu_k^{\underline{G}}(-g_k(x) + \underline{G}_k) \quad (2.23)$$

Hence, an optimal solution is only reached when the following 4 sets of conditions are met:

1. Optimality conditions

$$\frac{\partial \mathcal{L}}{\partial x} = 0 \quad (2.24)$$

$$\frac{\partial \mathcal{L}}{\partial \lambda_i} = 0 \quad (2.25)$$

2. Feasibility condition (Equation (2.20)-(2.22))

3. Complementary slackness condition

$$\mu_j^{\bar{G}}(g_j(x) - \bar{G}_j) = 0 \quad (2.26)$$

$$\mu_k^{\underline{G}}(-g_k(x) + \underline{G}_k) = 0 \quad (2.27)$$

4. Positivity condition

$$\mu_j^{\bar{G}}, \mu_k^{\underline{G}} \geq 0 \quad (2.28)$$

Every λ and μ is called a KKT multiplier, or a dual variable of the constraint. The main variables, x , in this case, are called the primal variables. In the case of OPF problems, one of the equality constraints is power mismatch constraint, given by (2.6) for AC systems, and the associated dual variable, λ , is also the LMP, which was discussed in section 2.1.

2.4.2. Optimality Condition Decomposition

The Optimality Condition Decomposition method is used when the optimisation problem has coupling constraints [15]. In this method, the problem is partitioned into many subproblems, which, in every iteration, receive the information from the neighbouring nodes, update their variables by solving OPF locally, and sending that information back to the neighbours until a every node agrees on the solution.

Once decomposed in N subproblems, one for every node, the constraints are divided into coupling and non-coupling. Then, the 1st order derivative of the associated KKT conditions is calculated and the Newton-Raphson method used to reach a local solution. Then, using the Jacobian Matrix of the KKT conditions, the primal and dual variables are updated and that information is sent to the other neighbours, until all the KKT conditions are met, and therefore a solution is reached.

In [21], OCD is used in order to solve a dynamic reactive power optimisation (DRPO) problem, minimising the transmission loss over multiple time periods. DRPO is a problem with continuous decision variables, nodal active power P and reactive power Q as well as voltage magnitude V and angle θ as well as discrete decision variables like controlling variables of the transformers. The later ones have first be relaxed into continuous variables with a help of binary variables in order to preform OCD, making this a mixed integer optimisation problem.

Then, the network is divided into areas by relaxing the coupling constraints, with the neighbouring areas exchanging the necessary information, i.e., V_a, θ_a, λ_a and μ_a and, finally, recurring to non-linear programming, the local problem is solved. The whole method iterates until the change in V and θ is smaller than the defined tolerances $\epsilon_{V,\theta}$.

2.4.3. Distributed Interior Point Method

Distributed Interior Point Method is based on the interior point method and then modified into a distributed algorithm using a unidirectional communication ring, as proposed in [22].

In this approach, an approximation of the centralised interior point method is made by first deriving Lagrangian function \mathcal{L} of the problem, then applying Newton-Raphson to \mathcal{L} and therefore obtaining Δx and $\Delta \lambda$. In the decentralised method, the only variables that are updated are the local primal and dual variables, while the other ones the same and once that update is done, meaning:

$$x_n(l+1) = x_n(l) + \alpha_p(l+1)\Delta x_n \quad (2.29)$$

$$\lambda_n(l+1) = \lambda_n(l) + \alpha_d(l+1)\Delta \lambda_n \quad (2.30)$$

where x_n and λ_n are the local primal and dual variables and l is the current iteration.

The results of one node are then sent to the next node in a circular faction until the result is within the defined tolerance range.

2.5. Consensus + Innovations

Consensus + Innovations is another KKT based method to solve problems in a fully decentralised fashion, which was subject of study in [23–26]. In [23], C+I is suggested as a method to improve on the Consensus algorithm, in order to tackle imperfect communication and randomness in the network. Here, the state estimate of each variable is updated with:

$$x_m(l+1) = x_m(l) - \underbrace{\beta_l \sum_{n \in \Omega_m} x_m(l) - x_n(l)}_{\text{consensus}} + \underbrace{\alpha_l K_m(l) (H_m^T R_m^{-1} (y_m(l) - H_m x_m(l)))}_{\text{local innovation}} \quad (2.31)$$

where x_m is the estimation variable, l the current iteration, α_l and β_l iteration dependant weighting factors, $K_m(l)$ is the iteration dependant local innovation gain, and $(H_m^T R_m^{-1} (y_m(l) - H_m x_m(l)))$ is the local innovation strategy which combines all agents observations y_m with the current one x_m [23].

Although it is a versatile method that can be used in network estimation, it has also been applied to OPF in [24–27]. C + I has the advantage of being a fully decentralised method, which needs no global variable update.

In order to better understand how this method works, we will take the example of the implementation on [25], where the DC approximation for AC power flow is used, as shown previously on (2.6). It is worth noting that $Y_{m,n} = B_{m,n}$ since the line resistance is neglected.

Here the objection function is defined as:

$$\min_{p^S} \sum_{m \in \mathcal{N}} A_m (p_m^S)^2 + B_m p_m^S + K_m \quad (2.32)$$

subject to:

$$p_m^S = \sum_{n \in \Omega_m} Y_{m,n}(\theta_m - \theta_n) \quad \forall (m) \in \mathcal{N} \quad (2.33)$$

$$\underline{P}_{m,n} \leq Y_{m,n}(\theta_m - \theta_n) \leq \bar{P}_{m,n} \quad \forall (m, n) \in \mathcal{N} \quad (2.34)$$

$$\underline{p}_m^S \leq p_m^S \leq \bar{P}_m^S \quad \forall (m) \in \mathcal{N} \quad (2.35)$$

where A_m and B_m are the quadratic and linear cost parameters, respectively, of the observed node m , K_m the constant cost of the same node and the objective function is the sum of all the costs associated with the nodal power P_m . This problem is subject to three sets of constraints: the power flow equation (2.33), the maximum and minimum power transfer limits of a branch (2.34) and the maximum and minimum power limits of a node (2.35). Since all voltage angles are relative to each other, it is necessary do fix one of them, arbitrarily, hence $\theta_1 = 0$.

Given this problem, it is possible then to set the Lagragian funtion \mathcal{L} as:

$$\begin{aligned} \mathcal{L} = & \sum_{m \in \mathcal{N}} A_m (p_m^S)^2 + B_m p_m^S + K_m \\ & + \sum_{m \in \mathcal{N}} \lambda_m \left(\sum_{n \in \Omega_m} Y_{m,n}(\theta_m - \theta_n) - p_m^S \right) \\ & + \sum_{n \in \Omega_m} \mu_{m,n} \left(Y_{m,n}(\theta_m - \theta_n) - \bar{P}_{m,n} \right) + \sum_{n \in \Omega_m} \mu_{n,m} \left(-Y_{m,n}(\theta_m - \theta_n) - \bar{P}_{m,n} \right) \\ & + \sum_{m \in \mathcal{N}} \mu_m^{\bar{P}} \left(p_m^S - \bar{P}_m^S \right) + \sum_{m \in \mathcal{N}} \mu_m^{\underline{P}} \left(-p_m^S + \underline{P}_m^S \right) \end{aligned} \quad (2.36)$$

where λ and μ correspond to the dual variables of respective constraints. The KKT first optimality conditions are:

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial p_m^S} = & \sum_{s \in \mathcal{S}} 2A_m p_m^S + B_m \\ & - \lambda_m + \mu_m^{\bar{P}} - \mu_m^{\underline{P}} = 0 \end{aligned} \quad (2.37)$$

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial \theta_m} = & \lambda_m \sum_{n \in \Omega_m} Y_{m,n} - \sum_{n \in \Omega_m} \lambda_n Y_{m,n} \\ & + \sum_{n \in \Omega_m} G_{m,n}^p (\mu_{m,n} - \mu_{n,m}) = 0 \end{aligned} \quad (2.38)$$

$$\frac{\partial \mathcal{L}}{\partial \lambda_m} = - \sum_{s \in \mathcal{S}} p_m^S + \sum_{n \in \Omega_m} Y_{m,n}(\theta_m - \theta_n) = 0 \quad (2.39)$$

$$\frac{\partial \mathcal{L}}{\partial \mu_{m,n}} = \sum_{n \in \Omega_m} Y_{m,n}(\theta_m - \theta_n) - \bar{P}_{m,n} \leq 0 \quad (2.40)$$

$$\frac{\partial \mathcal{L}}{\partial \mu_{n,m}} = \sum_{n \in \Omega_m} -Y_{m,n}(\theta_m - \theta_n) - \bar{P}_{m,n} \leq 0 \quad (2.41)$$

$$\frac{\partial \mathcal{L}}{\partial \mu_m^{\bar{P}}} = p_m^S - \bar{P}_m^S \leq 0 \quad (2.42)$$

$$\frac{\partial \mathcal{L}}{\partial \mu_m^{\underline{P}}} = -p_m^S + \underline{P}_m^S \leq 0 \quad (2.43)$$

KKT requires then the following conditions to be added in order to reach a true optimal solution, as shown

in equations (2.26), (2.27) and (2.28):

$$\mu_{m,n} \left(\bar{P}_{m,n} - \sum_{n \in \Omega_m} Y_{m,n} (\theta_m - \theta_n) \right) = 0 \quad (2.44)$$

$$\mu_{n,m} \left(-\bar{P}_{m,n} + \sum_{n \in \Omega_m} Y_{m,n} (\theta_m - \theta_n) \right) = 0 \quad (2.45)$$

$$\mu_m^{\bar{P}} (\bar{P}_m^S - p_m^S) = 0 \quad (2.46)$$

$$\mu_m^{\underline{P}} (p_m^S - \underline{P}_m^S) = 0 \quad (2.47)$$

$$\mu_{m,n}, \mu_{n,m}, \mu_m^{\bar{P}}, \mu_m^{\underline{P}} \geq 0 \quad (2.48)$$

An iterative approach is then presented, where each node communicates exclusively with its neighbours during optimisation process. So, for the array of local variables x_m :

$$x_m(l) = [\lambda_m(l), \theta_m(l), \mu_{m,n}(l), p_m^S(l)] \quad (2.49)$$

the general update is given by:

$$x_m(l+1) = \mathbb{P}_m(x_m(l) + \Phi_m g_m(x_n(l))) \quad (2.50)$$

where $g_m(\cdot)$ is the first order optimality constraints of node m , Φ_m a vector of tuning parameters and x_n is the vector of variables from the neighbouring node n . \mathbb{P} is an operator that projects the result into the feasible solution space, e.g., if the $\mu_{m,n} < 0 \Rightarrow \mu_{m,n} = 0$.

Applying this general update rule for the individual local variables of x_m yields the following expressions:

$$\begin{aligned} \lambda_m(l+1) &= \lambda_m(l) - \alpha_\theta^\lambda \left(\frac{\partial \mathcal{L}}{\partial \theta_m} \right) + \alpha_\lambda^\lambda \left(\frac{\partial \mathcal{L}}{\partial \lambda_m} \right) \\ &= \lambda_m(l) - \alpha_\theta^\lambda \left(\lambda_m \sum_{n \in \Omega_m} Y_{m,n} - \sum_{n \in \Omega_m} \lambda_n Y_{m,n} + \sum_{n \in \Omega_m} G_{m,n}^p (\mu_{m,n} - \mu_{n,m}) \right) \\ &\quad + \alpha_\lambda^\lambda \left(- \sum_{s \in \mathcal{S}} p_m^S + \sum_{n \in \Omega_m} Y_{m,n} (\theta_m - \theta_n) \right) \end{aligned} \quad (2.51)$$

$$\begin{aligned} p_m^S(l+1) &= \mathbb{P} \left(p_m^S(l) - \frac{1}{2A_m} \cdot \frac{\partial \mathcal{L}}{\partial p_m^S} \right) \\ &= \mathbb{P} \left(\frac{\lambda_m(l) - B_m}{2A_m} \right) \end{aligned} \quad (2.52)$$

$$\begin{aligned} \theta_m(l+1) &= \theta_m(l) - \alpha_\lambda^\theta \left(\frac{\partial \mathcal{L}}{\partial \lambda_m} \right) \\ &= \theta_m(l) - \alpha_\lambda^\theta \left(- \sum_{s \in \mathcal{S}} p_m^S + \sum_{n \in \Omega_m} Y_{m,n} (\theta_m - \theta_n) \right) \end{aligned} \quad (2.53)$$

$$\begin{aligned} \mu_{m,n}(l+1) &= \mathbb{P} \left(\mu_{m,n}(l) + \beta_\mu^{\bar{P},n} \left(\frac{\partial \mathcal{L}}{\partial \mu_{m,n}} \right) \right) \\ &= \mathbb{P} \left(\mu_{m,n}(l) + \beta_\mu^{\bar{P},n} \left(p_m^S - \bar{P}_m^S \right) \right) \end{aligned} \quad (2.54)$$

$$\begin{aligned} \mu_{n,m}(l+1) &= \mathbb{P} \left(\mu_{n,m}(l) + \beta_\mu^{\bar{P},n} \left(\frac{\partial \mathcal{L}}{\partial \mu_{m,n}} \right) \right) \\ &= \mathbb{P} \left(\mu_{n,m}(l) + \beta_\mu^{\bar{P},n} \left(-p_m^S + \underline{P}_m^S \right) \right) \end{aligned} \quad (2.55)$$

where all the α 's and β 's are tuning parameters.

For the update of the locational marginal price, equation (2.51), the two terms besides the $\lambda_m(l)$ can be discerned, similarly to (2.31). The first term is the consensus part, where the locational marginal prices converge to an agreement and the second term is the innovation part, which is proportional to the power mismatch on the node. For example, if there is negative power balance in m , the λ_m will have a tendency to increase which will lead to an increase of p_m^S , seen in (2.52).

In (2.53) is defined the voltage angle update, which is dependant on the power imbalance on the node, similarly to the LMP. Here, if too much power is flowing into other lines, then the voltage angle is reduced in order to reduce the outgoing power, reducing the nodal power imbalance.

Lastly, equations (2.54) and (2.55) update the dual variable of the line constraints. In case the power flowing in one line is not at the limit, (2.42) and (2.43) wield negative values and are set to 0 by the operator p_m^S . Only if the line is congested, these dual variables will increase.

The tuning parameters can be set by trial and error, [25], but attempts have been made to give them a physical interpretation [27] in order to increase convergence rates.

3

Physical Measurements for Increased Distributed Optimal Power Flow Convergence Rate

This chapter will focus on how physical measurements of a DC network can be incorporated into a Consensus + Innovations method in order to increase its convergence rate, as well as increasing its robustness.

Firstly, in section 3.1, it is defined how DC grid can be modelled, which will provide a foundation on which the optimisation algorithm can be built upon.

Then, in section 3.2, the optimisation problem is specified, and the equations that describe its behaviour are derived. The interaction between the physical and the cyber layer is explained in 3.3 and the differences between the implemented solution and the possible real world implementation are highlighted in 3.4.

In sections 3.5 and 3.6, the difference the synchronous and asynchronous implementations is explained and results for both implementations are shown.

3.1. Grid Modelling

Instead of using an bipolar grid like the one shown in [13], a unipolar DC grid was used. This means there are only 2 wires and voltage is measured between the plus (u^+) and neutral (u^N), which will be named u . It is also important to define that a current that is generated has a positive sign and a current that flows to a load is negative, i.e, if a node is generating power then $i_m^S > 0$.

3.1.1. Power Flow equations

In order to to define power flow in a DC unipolar network, the line current $i_{m,n}$ equations and the source current i_m equations were derived as follow:

$$i_{m,n} = G_{m,n}(u_m - u_n) \quad (3.1)$$

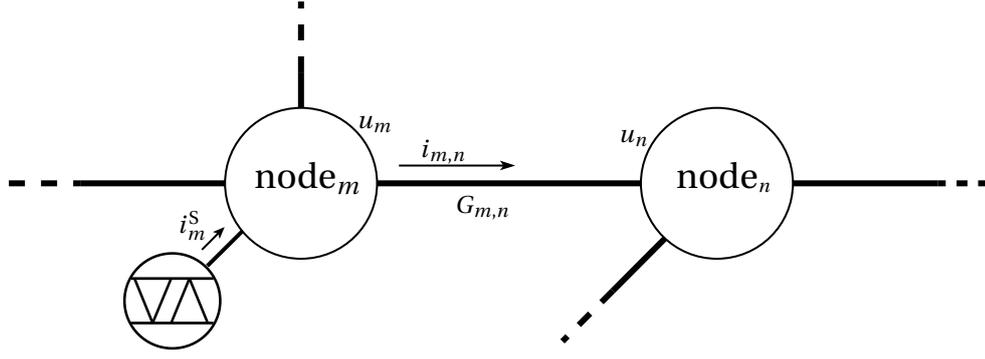
where $G_{m,n}$ is the conductance between node m and n . Throughout this report it is assumed that there are no self connecting nodes, i.e., that there is no line that connects node m to itself, creating a loop, and therefore, $G_{m,m} = 0$. Also, if node m does not have a physical connection to node n , $G_{m,n} = 0$.

From Kirshoff's current law, the expression for source current i_m^S can also be written as:

$$i_m^S = \sum_{n \in \Omega_m} i_{m,n} \quad (3.2)$$

In Figure 3.1 it is possible to better visualise what equations (3.1) and (3.2) mean. The current of the source i_m can be written as the sum of every current that is flowing in or out of that node. Equation (3.2) also holds true in case a node is not generating or consuming any power, meaning that the current that flows in to the node is the same as the one that flows out of said node, and, therefore, $\sum_{n \in \Omega_m} i_{m,n} = 0$.

Combining (3.1) and (3.2) yields:

Figure 3.1: Connection between 2 nodes m and n

$$i_m^S = \sum_{n \in \Omega_m} G_{m,n}(u_m - u_n) \quad (3.3)$$

which can then be written in a general matrix form as:

$$I_m^S = -G \cdot \Delta U \quad (3.4)$$

$$I_m^S = [i_0 \quad \dots \quad i_m \quad \dots \quad i_{\mathcal{N}}]^T \quad (3.5)$$

$$G = \begin{bmatrix} G_{1,1} & \dots & G_{1,\mathcal{N}} \\ \vdots & \ddots & \vdots \\ G_{m,1} & \dots & G_{m,\mathcal{N}} \\ \vdots & \ddots & \vdots \\ G_{\mathcal{N},1} & \dots & G_{\mathcal{N},\mathcal{N}} \end{bmatrix}, \quad \Delta G_{m,m} = 0 \quad \forall m \in \mathcal{N} \quad (3.6)$$

$$\Delta U = \begin{bmatrix} \Delta u_{1,1} & \dots & \Delta u_{1,\mathcal{N}} \\ \vdots & \ddots & \vdots \\ \Delta u_{m,1} & \dots & \Delta u_{m,\mathcal{N}} \\ \vdots & \ddots & \vdots \\ \Delta u_{\mathcal{N},1} & \dots & \Delta u_{\mathcal{N},\mathcal{N}} \end{bmatrix}, \quad \Delta u_{m,n} = u_m - u_n \quad \forall m, n \in \mathcal{N} \quad (3.7)$$

In (3.4) one line of the matrix product gives $i_m^S = \sum_{n \in \Omega_m} G_{m,n}(u_n - u_m)$ and therefore the minus signal is necessary in order to be consistent with (3.3).

For a network of \mathcal{N} node and \mathcal{M} current sources, (3.4) is an equation system with $\mathcal{N} + \mathcal{M}$ independent variables (\mathcal{N} node voltages and \mathcal{M} source currents) and only \mathcal{N} equations. This means that is not mathematically possible to solve this equation system because the it is mathematically undetermined and, therefore, this equation system is not sufficient to simulate powerflow in the grid.

This problem can also be looked at from a physical perspective. In order to know how much current each source is producing it is necessary to know who much current is flowing in each line, as given by (3.2), but in order to know much current is flowing in each line it is required that is the voltage in both ends of that line, shown in equation (3.1). So, it is necessary to know the nodal voltages u , which is unknown, in order to define what will be the source currents i^S .

One solution to a similar problem found in the DC approximation for AC powerflow [28] is to set a reference node. This node will have variable power P but fixed voltage angle θ_{ref} , while every other node will have fixed P_m and variable θ_m .

Applied to EDC power flow, which would mean one node would have its voltage fixed in order to function as a reference and would then accommodate for any current imbalance on the grid. On the other hand, every other node would have fixed current and the local voltage would then vary accordingly to the line currents.

3.1.2. Droop Control

Although is not explicitly said, having a reference node implies that is a central coordinator, which is the reference, which other nodes follow. In this thesis it is explored a solution for optimising grid resources without the necessity for a central coordinator, which means that another solution for running the simulated grid must be developed.

If we assume that every node that every load or generator is connected to the grid with a power electronics (PE) interface, current droop control can then be implemented. This means that for every source \mathcal{M} , the local i_m and u_m have a direct relation, increasing the number of equations that define the system to the same number of variables, $\mathcal{N} + \mathcal{M}$.

In [14], the droop curve is defines u_m as a function of i_m , but as it can be seen in 2.2, that does not translate to a well defined function. As an example, when $i_m = \underline{I}_m^S$, u_m can take a multitude of values. And so, droop as redefined as $i_m = F(u_m)$. As such, limits for current, \overline{I}_m^S and \underline{I}_m^S , and power, \overline{P}_m^S and \underline{P}_m^S , can be defined without incurring in mathematical errors. An example of the redefined control curve can be seen in Figure 3.2.

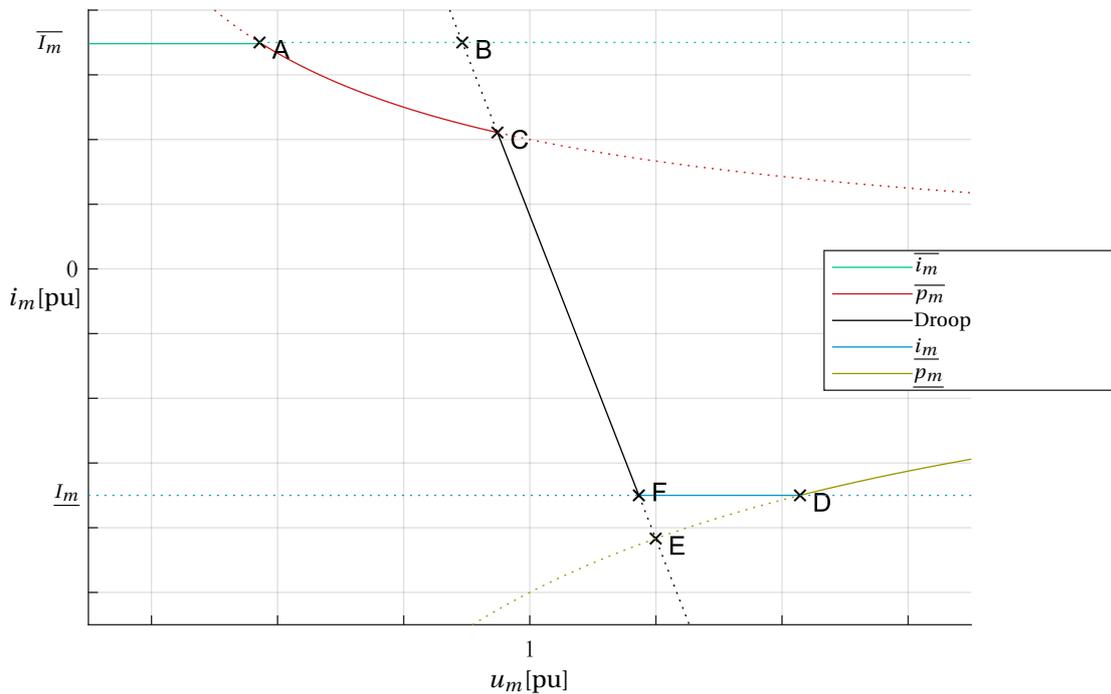


Figure 3.2: Example of the droop curve where current is a function of voltage. The letters A to E label the intersection points between 2 lines. For example, B is the intersection between the droop curve and maximum current line.

Depending on the operating state of the controller, its function is defined by a different equation, which will be explained in the upcoming paragraphs.

PE interface operating within the current and power limits

$$u_m = -d \cdot i_m^S + u_m^0 \quad (3.8)$$

$$\iff$$

$$i_m^S = -d' \cdot u_m + i_m^0, \quad d' = d^{-1} \quad (3.9)$$

where d' is the droop slope and i_m^0 is the desired local current level in case $u_m = 1$ [pu]. While (3.8) is a direct application of (2.11), from the previous chapter, (3.9) is the one that is going to be use throughout the report. When the PE interface is operating within the limits for power and current, i.e., normal operation, the output current is set by this curve. This means that, if there is a disturbance on the grid, the converter will automatically act in order to preserve voltage stability.

The operation of the droop control is demonstrated in Figure 3.3.

If the local voltage increases it means that, somewhere on the network, more power is being generated than consumed. Therefore the controller will decrease its current output, the green arrow, which means less power is supplied to the network. This decreased power output will then work towards lowering the local voltage. Then, it is the task of the optimisation method to find a new optimal voltage, shown as the grey arrow.

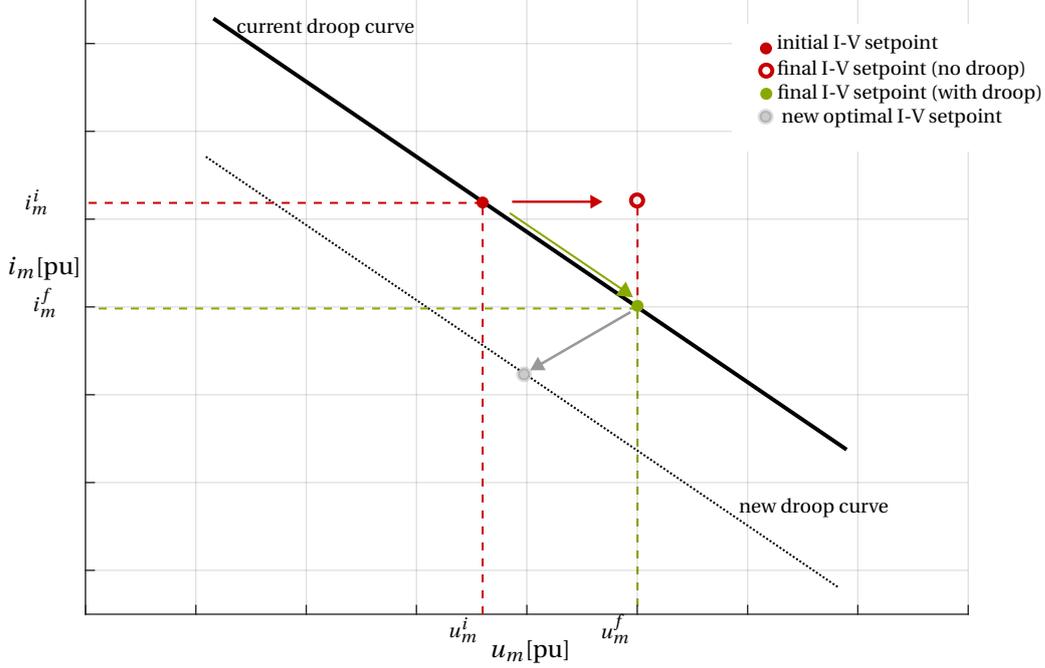


Figure 3.3: Droop Control action example. As a solid red circle, the initial I-V setpoint. As a hollow red circle, the new I-V setpoint in case there was no droop control. As a green circle, the new I-V setpoint after the control action. As a grey point, the possible new optimal set point for I-V.

If no droop control was present, the voltage of that node would increase, but the output current would be the same, red arrow on Figure 3.3, meaning that the power output of that node would also go up and possibility worsening the network instability.

The slope d' is a tuning parameter that can be set as well. The smaller the slope, the less reactive a converter is to a voltage oscillation, meaning that will provide less current to compensate for the fault, but it is more stable to voltage oscillations on the network. On the other hand, a higher droop d' means that a node will help more to compensate for power imbalances, but will be more susceptible to small voltage oscillations which can also create oscillation problems.

PE interface operating within the current and power limits

$$i_m^S = \bar{P}_m^S \cdot u_m^{-1} \quad (3.10)$$

where \bar{P}_m^S is either the maximum or minimum power rating of the converter.

When the PE interface reaches its power limits, it reduces the absolute value of the current output, in order to keep the power output from going the maximum and damaging both the converter and the device it is connected to. Once the voltage reaches u_m^C , the voltage level that corresponds to point C in Figure 3.2, or u_m^E , the converter can then restart normal operation again, given that the current limits are also respected.

PE interface operating at maximum or minimum current

$$i_m = \bar{I}_m^S \quad (3.11)$$

$$u_m = u_n - \frac{i_{m,n}}{G_{m,n}} \quad (3.12)$$

where \bar{I}_m^S is either the maximum or minimum current rating of the converter.

When the PE interface reaches the current limits, the output current is set to \underline{I}_m^S or \bar{I}_m^S , similarly to the operation at the power limit. In this state the power of the node is not controlled since the current is fixed and the voltage may vary, and will remain in this state until it reaches the power limit or when the voltage is between u_m^B and u_m^F .

3.2. Nodal Equations

Having described how the DC grid is simulated, and what we can call the physical layer of the system, we have now to describe how the optimisation process works, on what can be called the cyber layer.

In previous work [16, 27] both Lossless OPF and Exact OPF was discussed, since the algorithm worked detached from any kind of physical feedback. This is not the case anymore, since now exact physical measurements are taken from a lossy network, and so implementing Lossless OPF in a lossy grid would create problems. Therefore, for the rest of this report, only Exact OPF will be taken into account and therefore, it might be referred only as OPF.

3.2.1. Problem Definition

With that said, will define the optimisation problem as:

$$\min \sum_{m \in \mathcal{N}} A_m (p_m^S)^2 + B_m p_m^S \quad (3.13)$$

subject to:

$$p_m^S = u_m \sum_{n \in \Omega_m} G_{m,n} (u_m - u_n) \quad (3.14)$$

$$G_{m,n} (u_m - u_n) \leq \bar{I}_{m,n} \quad (3.15)$$

$$\underline{P}_m \leq p_m^S \leq \bar{P}_m \quad (3.16)$$

$$\underline{U}_m \leq u_m \leq \bar{U}_m \quad (3.17)$$

The constant term K in 2.32 was discarded since it has no relevance on the optimisation problem, since it is a fixed value and, as such, does not have any impact on the solution.

Equation (3.14) represents the power mismatch at node m . The mismatch between power flowing to or from the node, given by the left side of the equality, is the same as the total power produced in said node.

The inequations in (3.15), (3.16) and (3.17) set the maximum current limit in a line, the maximum and minimum power limit and the maximum and minimum voltage limits of the node, respectively.

3.2.2. Lagrangian Function

Having defined the problem, the Lagrangian function can be written as:

$$\begin{aligned} \mathcal{L} = & \sum_{m \in \mathcal{N}} \left(A_m (p_m^S)^2 + B_m p_m^S \right) \\ & + \sum_{m \in \mathcal{N}} \lambda_m \left(u_m \sum_{n \in \Omega_m} G_{m,n} (u_m - u_n) - p_m^S \right) \\ & + \sum_{m \in \mathcal{N}} \sum_{n \in \Omega_m} \mu_{m,n} \left(G_{m,n} (u_m - u_n) - \bar{I}_{m,n} \right) \\ & + \sum_{m \in \mathcal{N}} \bar{\mu}_m^P (p_m^S - \bar{P}_m) \\ & + \sum_{m \in \mathcal{N}} \bar{\mu}_m^P (-p_m^S + \underline{P}_m) \\ & + \sum_{m \in \mathcal{N}} \bar{\mu}_m^U (u_m - \bar{U}_m) \\ & + \sum_{m \in \mathcal{N}} \bar{\mu}_m^U (-u_m + \underline{U}_m) \end{aligned} \quad (3.18)$$

where λ_m and μ 's are, respectively the dual variables of the nodal equality constraint and the nodal and line's inequality constraints.

3.2.3. Karush Kuhn Tucker Conditions

From \mathcal{L} , the First Order Optimality conditions can be derived:

$$\frac{\partial \mathcal{L}}{\partial p_m^S} = 2A_m p_m^S + B_m - \lambda_m + \mu_m^{\bar{P}} - \mu_m^{\underline{P}} \quad (3.19)$$

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial u_m} &= \lambda_m \sum_{n \in \Omega_m} G_{m,n}(u_m - u_n) \\ &\quad + \lambda_m u_m \sum_{n \in \Omega_m} G_{m,n} - \sum_{n \in \Omega_m} \lambda_n u_n G_{m,n} \\ &\quad + \sum_{n \in \Omega_m} G_{m,n}(\mu_{m,n} - \mu_{n,m}) + \mu_m^{\bar{U}} - \mu_m^{\underline{U}} = 0 \end{aligned} \quad (3.20)$$

$$\frac{\partial \mathcal{L}}{\partial \lambda_m} = -p_m^S + u_m \sum_{n \in \Omega_m} G_{m,n}(u_m - u_n) = 0 \quad (3.21)$$

$$\frac{\partial \mathcal{L}}{\partial \mu_{m,n}} = \sum_{n \in \Omega_m} G_{m,n}(u_m - u_n) - \bar{I}_{m,n} \leq 0 \quad (3.22)$$

$$\frac{\partial \mathcal{L}}{\partial \mu_m^{\bar{P}}} = p_m^S - \bar{P}_m^S \leq 0 \quad (3.23)$$

$$\frac{\partial \mathcal{L}}{\partial \mu_m^{\underline{P}}} = -p_m^S + \underline{P}_m^S \leq 0 \quad (3.24)$$

$$\frac{\partial \mathcal{L}}{\partial \mu_m^{\bar{U}}} = u_m - \bar{U}_m \leq 0 \quad (3.25)$$

$$\frac{\partial \mathcal{L}}{\partial \mu_m^{\underline{U}}} = -u_m + \underline{U}_m \leq 0 \quad (3.26)$$

$$(3.27)$$

As well as the slack conditions:

$$\mu_{m,n}(\bar{I}_{m,n} - G_{m,n}(u_m - u_n)) = 0 \quad (3.28)$$

$$\mu_{n,m}(-\bar{I}_{m,n} + G_{m,n}(u_m - u_n)) = 0 \quad (3.29)$$

$$\mu_m^{\bar{P}}(\bar{P}_m^S - p_m^S) = 0 \quad (3.30)$$

$$\mu_m^{\underline{P}}(p_m^S - \underline{P}_m^S) = 0 \quad (3.31)$$

$$\mu_m^{\bar{U}}(\bar{U}_m - u_m) = 0 \quad (3.32)$$

$$\mu_m^{\underline{U}}(u_m - \underline{U}_m) = 0 \quad (3.33)$$

$$(3.34)$$

And the positivity conditions:

$$\mu_{m,n}, \mu_{n,m}, \mu_m^{\bar{P}}, \mu_m^{\underline{P}}, \mu_m^{\bar{U}}, \mu_m^{\underline{U}} \geq 0 \quad (3.35)$$

In order to know λ_n , $\mu_{m,n}$ and u_n , that information had to be sent from the neighbouring nodes. However, since it is assumed that $G_{m,n}$ is known and that the line current $i_{m,n}$ can be measured, $u_{m,n}$ is obtained using (2.7):

$$i_{m,n} = G_{m,n}(u_m - u_n) \iff \quad (3.36)$$

$$\iff u_n = u_m - \frac{i_{m,n}}{G_{m,n}} \quad (3.37)$$

This result can then be applied to the first order optimality conditions and slackness conditions in order to remove u_n from the equations.

Substitution of u_n in $\partial\mathcal{L}/\partial u_m$

In order to make this next part easier to understand, (3.21) was divided in two parts: one that is dependent on either u_m or u_n and another that is not, which will be called \mathcal{K} :

$$\frac{\partial\mathcal{L}}{\partial u_m} = \lambda_m \sum_{n \in \Omega_m} G_{m,n}(u_m - u_n) + \lambda_m u_m \sum_{n \in \Omega_m} G_{m,n} - \sum_{n \in \Omega_m} \lambda_n u_n G_{m,n} + \underbrace{\sum_{n \in \Omega_m} G_{m,n}(\mu_{m,n} - \mu_{n,m}) + \mu_m^{\bar{U}} - \mu_m^{\underline{U}}}_{\mathcal{K}} \quad (3.38)$$

$$\Rightarrow \frac{\partial\mathcal{L}}{\partial u_m} = \lambda_m \sum_{n \in \Omega_m} G_{m,n}(u_m - u_n) + \lambda_m u_m \sum_{n \in \Omega_m} G_{m,n} - \sum_{n \in \Omega_m} \lambda_n u_n G_{m,n} + \mathcal{K} \quad (3.39)$$

Applying (3.37) to (3.39), yields:

$$\frac{\partial\mathcal{L}}{\partial u_m} = \lambda_m \sum_{n \in \Omega_m} i_{m,n} + \lambda_m u_m \sum_{n \in \Omega_m} G_{m,n} - \sum_{n \in \Omega_m} \lambda_n G_{m,n}(u_m) + \sum_{n \in \Omega_m} \lambda_n i_{m,n} + \mathcal{K} \quad (3.40)$$

$$\frac{\partial\mathcal{L}}{\partial u_m} = \sum_{n \in \Omega_m} i_{m,n}(\lambda_m + \lambda_n) + u_m \left(\sum_{n \in \Omega_m} G_{m,n}(\lambda_m - \lambda_n) \right) + \mathcal{K} \quad (3.41)$$

and expanding \mathcal{K} , it gives the full equation for the Lagrangian derivative of the voltage:

$$\begin{aligned} \frac{\partial\mathcal{L}}{\partial u_m} &= \sum_{n \in \Omega_m} i_{m,n}(\lambda_m + \lambda_n) + u_m \left(\sum_{n \in \Omega_m} G_{m,n}(\lambda_m - \lambda_n) \right) \\ &\quad + \sum_{n \in \Omega_m} G_{m,n}(\mu_{m,n} - \mu_{n,m}) + \mu_m^{\bar{U}} - \mu_m^{\underline{U}} \end{aligned} \quad (3.42)$$

This optimality condition is very important because it is going to be used to update the local variables when running the C + I optimisation method, namely the p_m , u_m and λ_m . This is critical because it could be argued that the substitution (3.37) could actually be performed on (3.14) and (3.15) and, once the Lagrangian function was written and its first order optimality conditions were derived, it would be the same as was done in (3.42). That is not the case.

If (3.37) is used in (3.14) and (3.15), then the following expressions are obtained:

$$p_m^S = u_m \sum_{n \in \Omega_m} i_{m,n} \quad (3.43)$$

$$i_{m,n} \leq \bar{I}_{m,n} \quad (3.44)$$

which means:

$$\begin{aligned} \mathcal{L} &= (\dots) + \lambda_m (u_m \sum_{n \in \Omega_m} i_{m,n} - p_m^S) + \\ &\quad + \sum_{m \in \mathcal{N}} \sum_{n \in \Omega_m} \mu_{m,n} (i_{m,n} - \bar{I}_{m,n}) + (\dots) \end{aligned} \quad (3.45)$$

which, when deriving $\partial\mathcal{L}/\partial u_m$, will result in:

$$\frac{\partial\mathcal{L}}{\partial u_m} = \lambda_m \sum_{n \in \Omega_m} i_{m,n} + \mu_m^{\bar{U}} - \mu_m^{\underline{U}} \quad (3.46)$$

It is possible to observe the differences between (3.42) and (3.46). The most fundamental distinction between both equations is the fact that information about λ_n is lost, especially in the $u_m (\sum_{n \in \Omega_m} G_{m,n}(\lambda_m - \lambda_n))$ term. This is a vital loss since that expression represents the consensus part of the C + I algorithm and, without it, the algorithm wouldn't work.

Using (3.46) would then mean that the nodes wouldn't be able to converge to one solution, which defeats the objective of this proposed solution.

Substitution of u_n in $\partial\mathcal{L}/\partial\lambda_m$

The $\partial\mathcal{L}/\partial\lambda_m$ is the power flow condition, meaning that will always have to be 0 at the optimal operating point. Rewriting (3.22) with (3.37) gives:

$$\frac{\mathcal{L}}{\partial\lambda_m} = u_m \sum_{n \in \Omega_m} i_{m,n} - p_m^S = u_m \cdot i_m^S - p_m^S = 0 \quad (3.47)$$

Looking at (3.47), u_m and i_m^S are measured from the physical grid, while p_m^S is an internal variable from the optimisation process. If we define, the product $u_m \cdot i_m^S$ as \hat{p}_m^S , then:

$$\frac{\mathcal{L}}{\partial\lambda_m} = \hat{p}_m^S - p_m^S = 0 \quad (3.48)$$

which means that $\partial\mathcal{L}/\partial\lambda_m$ becomes the difference between the power setpoint defined by the optimisation layer and the actual power that the node is supplying.

Substitution of u_n in $\partial\mathcal{L}/\partial\mu_{m,n}$

Updating the $\partial\mathcal{L}$ expression with is quite strait forward:

$$\frac{\partial\mathcal{L}}{\partial\mu_{m,n}} = \sum_{n \in \Omega_m} i_{m,n} - \bar{I}_{m,n} \leq 0 \quad (3.49)$$

The only difference with previous implementations is that now, since $i_{m,n}$ is measured, $\mu_{m,n}$ will increase or decrease based on physical values and not optimisation variables.

Substitution of u_n on the slackness conditions

The slackness conditions could also be subject to this substitution, but since they are only used during the optimisation process in order to check if the system has converged, it is not very relevant that they are derived.

3.2.4. Defining the updates

Once redefined the first order optimality conditions, the update strategy for the variables has to be described.

Relation between λ_m and p_m^S

The power and λ updates are vital to the optimisation process since the objective is to minimise the costs, which are directly linked to the amount of power being produced in each individual node and how much does it cost to produce it. It would make sense that, in a network with multiple generators and loads that the more expensive generators would only be supplying energy in case the cheaper options where unable to meet the demand on the grid. This makes sense from a market point of view. Looking at the LMP as the price that the network is willing to pay for generation on one node, then this node will only start generating power if the price that it could sell that power was, at least, the same as the cost that it would incur when generating said power. This price will fluctuate until demand is met, meaning that if supply is bigger than demand then the prices will go down and some generators will lower their production, but if demand is higher, then the prices will increase and it is economically viable for generators to produce more.

The locational marginal price is a measurement of the incremental cost of power generation and is measured in m.u./W, where **m.u.** is monetary units and W is power measured in Watts. As such, it is directly related to how cheap or expensive a source is and, therefore, directly influences how much a generator should be supplying. This relation between LMP and power output can be seen in figures 3.4 and 3.5. During the iterative process, either λ_m or p_m should be forced to follow the trajectory shown in one of the figures, with the difference being that in Figure 3.4, the term $A_m = 0$, which means the generator as a linear cost function, which can be seen from (3.13).

And so, following the implementation suggested on [27], two regions of operation can be differentiated: Constant Power Region and Marginal Generator Region. The first is define for LMP values that are either bellow the minimum marginal cost or above the maximum marginal cost while the second is for when λ_m is within those values. This means that, in one hand, if $\lambda_m < B_m + 2A_m \cdot \underline{P}_m^S$ or $\lambda_m > B_m + 2A_m \cdot \bar{P}_m^S$ the power is fixed at the limit value and the LMP is being updated accordingly to an update strategy which will be later described. On the other hand, when $B_m + 2A_m \cdot \underline{P}_m^S < \lambda_m$ and $\lambda_m < B_m + 2A_m \cdot \bar{P}_m^S$, the generator will change p_m^S accordingly to the defined strategy while the LMP will follow accordingly to the curves shown in Figures 3.4 or 3.5. It is important to note that, if the generator has a linear cost function ($A_m = 0$), the LMP will stay

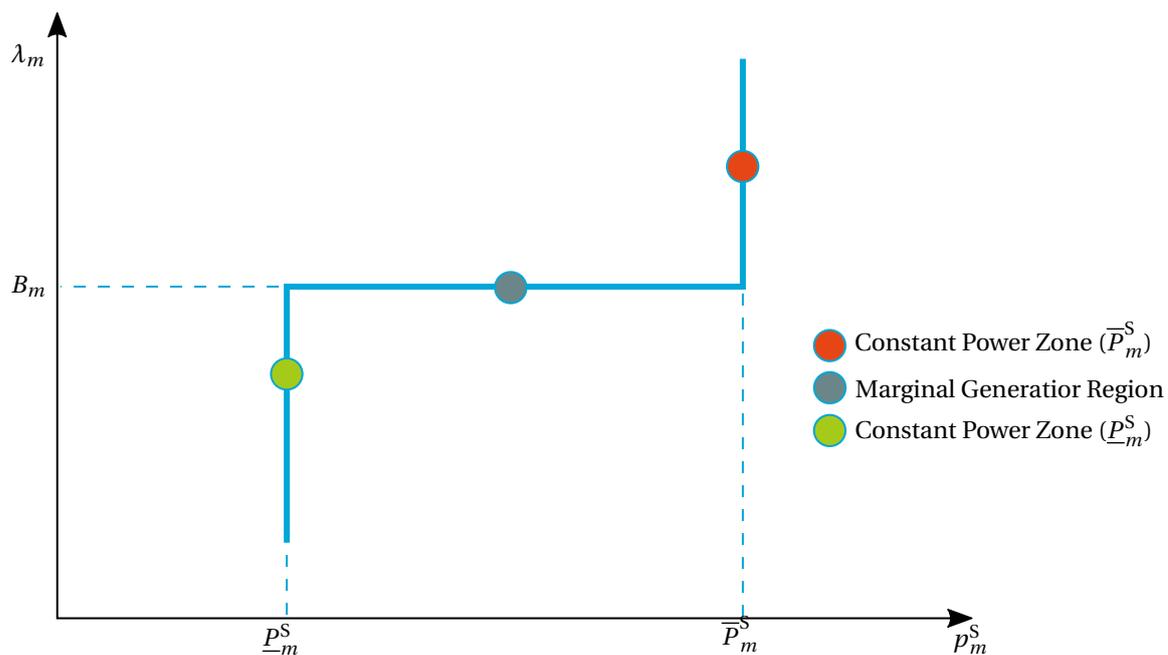


Figure 3.4: Power-LMP characteristic curve of a generator in node m with a linear cost function.

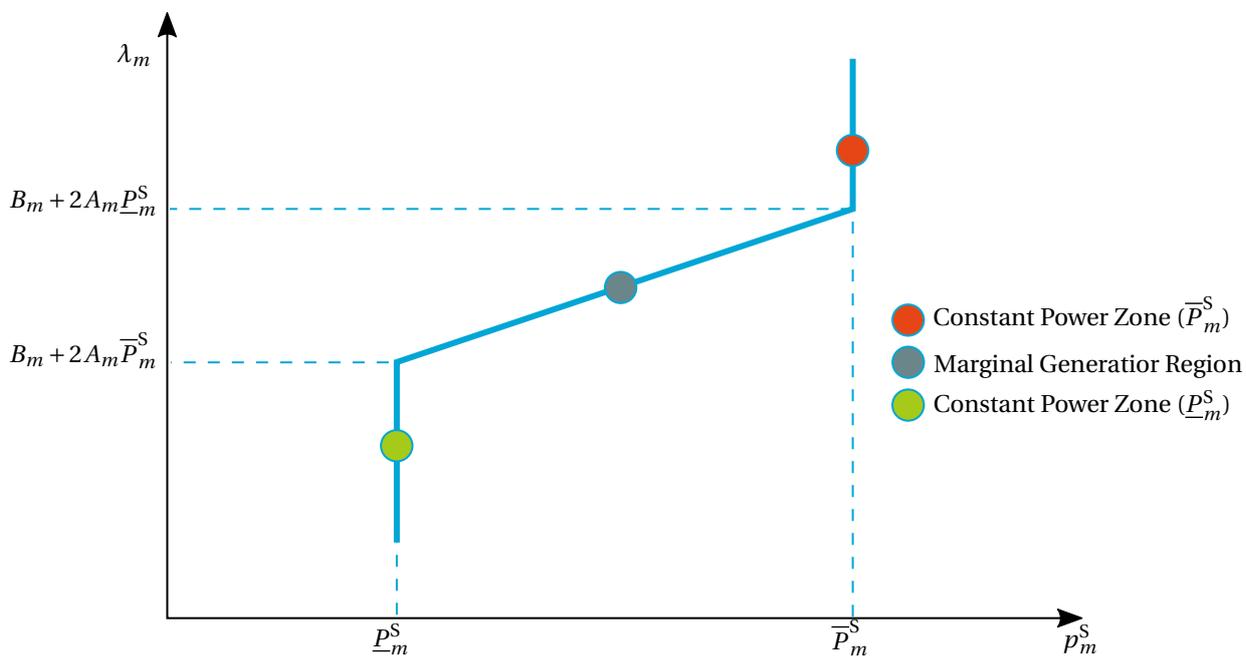


Figure 3.5: Power-LMP characteristic curve of a generator in node m with a quadratic cost function.

the same until the generator reaches one of the power limits, which is not the case for when the cost function is quadratic.

Constant Power Region updating strategy

Starting with the constant power region, the update strategy for the λ_m and p_m^S is given by:

$$\lambda_m(l+1) = \lambda_m(l) - \alpha_u^\lambda \frac{\partial \mathcal{L}}{\partial u_m} + \alpha_\lambda^\lambda \frac{\partial \mathcal{L}}{\partial \lambda_m} \quad (3.50)$$

$$p_m^S(l+1) = \bar{P}_m^S \quad (3.51)$$

where α_u^λ and α_λ^λ are tuning parameters and p_m^S is either at minimum or maximum power output, depending if the LMP is higher or lower than the marginal costs. It is also important to note were the presence of $\partial \mathcal{L} / \partial u_m$, which, as previously mentioned, is the term which contains the part of the update and therefore writing it as (3.46) would remove that term, meaning the whole optimisation would not work as intended.

Both α_u^λ and α_λ^λ can have a physical psychological interpretation, which will help when tuning them.

The α_u^λ formulation

Starting with α_u^λ , this tuning parameter is formulated in a way that λ_m resembles the neighbouring nodes' LMP, λ_n , in order to reach consensus. As such, $\partial \mathcal{L} / \partial \lambda_m$ is discarded and $\partial \mathcal{L} / \partial u_m$ is expanded for (3.50):

$$\begin{aligned} \lambda_m(l+1) = \lambda_m(l) - \alpha_u^\lambda \left(\sum_{n \in \Omega_m} i_{m,n}(\lambda_m + \lambda_n) + u_m \left(\sum_{n \in \Omega_m} G_{m,n}(\lambda_m - \lambda_n) \right) \right. \\ \left. + \sum_{n \in \Omega_m} G_{m,n}(\mu_{m,n} - \mu_{n,m}) + \mu_m^{\bar{U}} - \mu_m^{\underline{U}} \right) \end{aligned} \quad (3.52)$$

For simplification, there is no congestion on the grid and the voltage limit has not been reached:

$$\lambda_m(l+1) = \lambda_m(l) - \alpha_u^\lambda \left(\sum_{n \in \Omega_m} i_{m,n}(\lambda_m + \lambda_n) + u_m \left(\sum_{n \in \Omega_m} G_{m,n}(\lambda_m - \lambda_n) \right) \right) \quad (3.53)$$

For a DC grid with losses, $\lambda_m = \lambda_n$ only if there is no current flowing between node m and node n . If $i_{m,n} \neq 0$, then the power loss in that line is given by $i_{m,n}^2 / G_{m,n} \neq 0$. Those line losses also have a cost associated since it's increasing the power demand and this would mean a disparity in the observed λ_m and λ_n which goes against the initial assumption that $\lambda_m = \lambda_n$. Hence, it is also assumed $i_{m,n} = 0$ and so:

$$\lambda_m(l+1) = \lambda_m(l) - \alpha_u^\lambda \left(u_m \left(\sum_{n \in \Omega_m} G_{m,n}(\lambda_m - \lambda_n) \right) \right) \quad (3.54)$$

$$\alpha_u^\lambda = \frac{k}{u_m \sum_{n \in \Omega_m} G_{m,n}}, \quad k = 1 \quad (3.55)$$

$$\lambda_m(l+1) = \lambda_m(l) - \lambda_m + \lambda_n = \lambda_n \quad (3.56)$$

In (3.56), if $k = 1$ is chosen, a problem might surge where $\lambda_m(l+1) = \lambda_n(l)$ and $\lambda_n(l+1) = \lambda_m(l)$ and, as such, the nodes might oscillate against each other. A solution was found in [27], which was selecting $k = 0.5$ and so both m and n would perform half the work each, hence reducing the oscillations. This then yields:

$$\alpha_u^\lambda = \frac{0.5}{u_m \sum_{n \in \Omega_m} G_{m,n}} \left[\frac{\text{V}}{\text{W}} \right] \quad (3.57)$$

The α_λ^λ formulation

The second tuning parameter in (3.50) is α_λ^λ and it tunes the effect that $\partial \mathcal{L} / \partial \lambda_m$ has on the λ_m update. Since $\partial \mathcal{L} / \partial \lambda_m$ has now a different meaning, also α_λ^λ can be interpreted differently. While in [27] α_λ^λ is interpreted as the rate at which the LMP would vary accordingly to the power mismatch on the node, now it can be thought of as the how fast is every node willing to increase or decrease the local price in order to match the intended power output, p_m with the actual value of power measured in that node, \hat{p}_m . The units are still the same, i.e., since α_λ^λ determines the variation rate of λ_m , measured in m.u./W, accordingly to the measured power and its setpoint, in W, then α_λ^λ is measured in m.u./W².

Setting this parameter too high and the λ_m might become too reactive and cause oscillations and other convergence problems, but setting this value too low and the LMP might not be influenced enough by the difference in power and to converge to a optimal point fast enough. Thus:

$$\alpha_\lambda^\lambda = 0.01 \frac{\text{m.u.}}{\text{W}} \cdot 1 \times 10^{-3} \text{W}^{-1} = 1 \times 10^{-5} \left[\frac{\text{m.u.}}{\text{W}} \right] \quad (3.58)$$

where the defined rate was chosen as a variation of 1% for every 1 kW mismatch.

Marginal Generator Region updating strategy

Once the λ_m is within the the maximum and minimum marginal prices it, then is LMP that is written as a function of the local power setpoint p_m^S . Hence, the update of this variables is given by:

$$p_m^S(l+1) = p_m^S(l) + \alpha_\lambda^p \frac{\partial \mathcal{L}}{\partial \lambda_m} - \alpha_u^p \frac{\partial \mathcal{L}}{\partial u_m} \quad (3.59)$$

$$\lambda_m(l+1) = 2A_m \cdot p_m^S(l+1) + B_m \quad (3.60)$$

where α_λ^p and α_u^p are tuning parameters. Once more, it is possible to observe that the consensus term is also present in this update, which serves to highlight, again, the importance of doing the mathematical substitution only on the differential terms of the Lagrangian function. Similarly to both α_u^λ and α_λ^λ , α_λ^p and α_u^p also have a physical psychical interpretation.

In [27] was found that this update strategy didn't work when, in a grid with multiple generators, the node with the highest voltage (m) was not the marginal generator. Oscillations induced by the $\mu_m^{\bar{U}}$ on the λ_m , via the $\partial \mathcal{L} / \partial u_n$ would translate to an attempt to change λ_n , the LMP of the marginal node n , but since λ_n was fixed at B_m , it would create power oscillations, meaning the system would not converge.

In order to solve this problem, when on the marginal generator region, a node would update its λ_m using (3.50) and the local power would vary accordingly to the following equation:

$$p_m^S(l+1) = \frac{\lambda_m(l+1) - B_m}{2A_m} \quad (3.61)$$

This meant that, in a system, all the generators would have to have quadratic cost functions, something which is not true in real systems.

However, using grid measurements and some parameter tuning, this problem was solved and so this limitation was eliminated.

The α_λ^p formulation

The term $\partial \mathcal{L} / \partial \lambda_m$ from (3.60) determines the difference between last iteration's power setpoint and the measured power output. In [27] it was concluded that, in order to get smooth results, this would mean that $p_m^S(l+1)$ should only go half way towards bridging that gap, and since the update was done simultaneously with the voltage update which will be later discussed, it should only take a quarter step, meaning this tuning parameter is given by:

$$\alpha_\lambda^p = 0.25 \quad (3.62)$$

The α_u^p formulation

The $\partial \mathcal{L} / \partial u_m$ means that power will increase or decrease in regard to the difference in LMP between neighbouring nodes. Using the same reasoning that was used for α_u^λ , and applying it to (3.60), it yields:

$$\alpha_u^p = \frac{k \frac{\text{W}^2}{\text{m.u.}}}{u_m(l) \sum_{n \in \Omega_m} G_{m,n}} \left[\frac{\text{V}}{\text{W}} \right] \quad (3.63)$$

where k is the ratio between the the intended shift in power in regards to the difference in LMP. Through trial and error, it was define that $k = 100$ was a good ratio, hence:

$$\alpha_u^p = \frac{100 \frac{\text{W}^2}{\text{m.u.}}}{u_m(l) \sum_{n \in \Omega_m} G_{m,n}} \left[\frac{\text{V}}{\text{W}} \right] \quad (3.64)$$

The voltage update

In Exact OPF, one of the side effects of trying to minimise costs is also to minimise power losses in the system. One way to minimise those losses is to increase the voltage levels, which decrease the necessary current to transfer the same amount of power, and, therefore, reduces the transmission losses.

As previously stated, u_m is measured from the grid. But at the same time it is also an optimisation variable. As such, in the update, strategy $u_m(l)$ is the measured voltage at the beginning of the iteration and $u_m(l+1)$ is the new voltage setpoint. Both this $u_m(l+1)$ as the $p_m^S(l+1)$ will be later used to set the droop curve, which will be explained in the following section.

As such, it is necessary to make sure voltage changes in order to both reduce the power mismatch in each node and increases to the maximum possible value, in order to reduce losses on the network. Hence, following [27], the voltage update can be given as:

$$u_m(l+1) = u_m(l) - \alpha_\lambda^u \frac{\partial \mathcal{L}}{\partial \lambda_m} - \alpha_u^u \frac{\partial \mathcal{L}}{\partial u_m} \quad (3.65)$$

where α_λ^u and α_u^u are tuning parameters. As before, the formulation and physical interpretation of both tuning parameters will now be explained.

The α_λ^u formulation

The α_λ^u is, in its core, the reverse approach from the formulation of α_λ^P . This means that, for example, if too much power is flowing from the node, it will decrease its voltage, approximating the power setpoint and the real power output. As previously mentioned, α_λ^P does a quarter of the work and so α_λ^u does another quarter step, and so is given by:

$$\alpha_\lambda^u = \frac{0.25}{u_m \sum_{n \in \Omega_m} G_{m,n}} \left[\frac{\text{V}}{\text{W}} \right] \quad (3.66)$$

The α_u^u formulation

This tuning parameter serves to rise, or lower, the voltage in regards to the λ difference between nodes. If power is flowing from m to n and there is no line congestion, $\mu_{m,n} = 0$, and no voltage limit has been reached, $\mu_m^{\bar{U}} = \mu_m^{\underline{U}} = 0$, then, in order to reduce losses, the voltage of node m should rise. Since the current direction is from m to n , this means $\lambda_m < \lambda_n$, which makes it so $\partial \mathcal{L} / \partial u_m < 0$. Thus, in order to make u_m vary positively with this LMP mismatch, an “-” sign is added before α_u^u , in Equation (3.65).

Now, in [27], it is suggested that this variation of u_m accordingly to the LMP mismatch is made so it is linear, therefore, it could be written as:

$$\alpha_u^u \frac{\partial \mathcal{L}}{\partial u_m} \approx k(\lambda_m - \lambda_n) \quad (3.67)$$

Expanding the derivative term, and assuming $\mu_{m,n} = \mu_m^{\bar{U}} = \mu_m^{\underline{U}} = 0$, yields:

$$\alpha_u^u \left(\sum_{n \in \Omega_m} i_{m,n}(\lambda_m + \lambda_n) + u_m \left(\sum_{n \in \Omega_m} G_{m,n}(\lambda_m - \lambda_n) \right) \right) \approx k(\lambda_m - \lambda_n) \quad (3.68)$$

where $i_{m,n}$ is assumed to be relatively small compared to $u_m \sum_{n \in \Omega_m} G_{m,n}$, and so α_u^u is defined to be:

$$\alpha_u^u = \frac{k}{u_m \sum_{n \in \Omega_m} G_{m,n}} \left[\frac{\text{V}^2}{\text{m.u.}} \right] \quad (3.69)$$

Taking the results from [27] as a starting point and through trial and error, it was found that $k = 5 \frac{\text{W}\cdot\text{V}}{\text{m.u}}$ provided the best results and, as such:

$$\alpha_u^u = \frac{5 \frac{\text{W}\cdot\text{V}}{\text{m.u}}}{u_m \sum_{n \in \Omega_m} G_{m,n}} \left[\frac{\text{V}^2}{\text{m.u.}} \right] \quad (3.70)$$

Congestion Management

As mention previously, a line cannot transfer an unlimited amount of current, or power, due to physical limitation. In the simulation context, this limitation is done by setting a limit on the maximum current capabilities of a line: $\bar{I}_{m,n}$. As such, the optimisation algorithm must account for this constraints and act accordingly.

The solution presented in [27] was to bound the nodal voltage to such a value that the resulting current would be the maximum current limit of the conngestion line, as such:

$$u_m(l+1) = \frac{\bar{I}_{m,n}}{\sum_{n \in \Omega_m} G_{m,n}} + u_n(l) \quad (3.71)$$

Since u_n is no longer a variable in the optimisation problem, (3.37) is applied to (3.71), wielding:

$$u_m(l+1) = u_m(l) + \frac{\bar{I}_{m,n} - i_{m,n}}{\sum_{n \in \Omega_m} G_{m,n}} \quad (3.72)$$

The dual variable of the line constraints also needs to be updated and it will depend only on the state of the line, i.e., if it is congested or not. With this in mind, the following update strategy was developed:

$$\mu_{m,n}(l+1) = \mathbb{P} \left[\mu_{m,n}(l) + \beta_{m,n} \frac{\partial \mathcal{L}}{\partial \mu_{m,n}} \right] \quad (3.73)$$

$$\mathbb{P} \Rightarrow \mu_{m,n}(l+1) = 0, \quad \text{if } (\mu_{m,n} \cap \frac{\partial \mathcal{L}}{\partial \mu_{m,n}}) \leq 0 \quad (3.74)$$

where $\beta_{m,n}$ is a tuning parameter, and it translates to the rate of change of price in regards to the excess of current. In (3.73) the \mathbb{P} means, as stated in (3.74), that $\mu_{m,n}$ can update normally unless both the variable and the Lagrangian derivative are negative. As such, the update strategy bounds the minimum value of $\mu_{m,n}$ to 0, since the only way to become negative would be if $\partial \mathcal{L} / \partial \mu_{m,n} < 0$. But if the $\mu_{m,n}$ has a positive value, it is free to increase or decrease in order to reach the optimal value.

In order to better understand this an example can be given. If node m is outputting a high amount of power, the maximum current limit is reached in line (m,n) . This congestion will make it so that $\mu_{m,n}$ increases and, consequently, through (3.50), increases the price in node m and n . Increasing the LMP in node n , makes it so that it will start genmerating its own power and, as such reduce the current flowing through (m,n) , and solving the congestion problem.

The α_u^u formulation

Through trial and error, it was estimated that 5 m.u./A² was a good value and, as such:

$$\beta_{m,n} = 5 \left[\frac{\text{m.u.}}{\text{A}^2} \right] \quad (3.75)$$

Voltage Limits

The voltage limit dual variable update was done in a similar way to the congestion limit update, since the reasoning behind it is analogous. As such, the chosen update strategy, based on [27], was:

$$\mu_m^{\bar{U}}(l+1) = \mathbb{P} \left[\mu_m^{\bar{U}}(l) + \beta^{\bar{U}} \frac{\partial \mathcal{L}}{\partial \mu_m^{\bar{U}}} \right] \quad (3.76)$$

$$\mu_m^{\underline{U}}(l+1) = \mathbb{P} \left[\mu_m^{\underline{U}}(l) + \beta^{\underline{U}} \frac{\partial \mathcal{L}}{\partial \mu_m^{\underline{U}}} \right] \quad (3.77)$$

$$\beta^{\bar{U}} = \beta^{\underline{U}} = 10 \left[\frac{\text{m.u.}}{\text{V}^2} \right] \quad (3.78)$$

where $\beta^{\bar{U}}$ and $\beta^{\underline{U}}$ tuning parameters which, through trial and error, were chosen to be set as 10 m.u./V², compared to the 15 m.u./V² chosen in [27]. The higher gain value meant that the voltage and LMP update were very reactive to the voltage limits and caused an unwanted oscillatory behaviour in the system.

Power Limits

The power limit dual variable update was done, once more, resembling the line and voltage constrain dual variable updates. $\mu_{\bar{P}}$ and $\mu_{\underline{P}}$ have the particularity that they are not used in any other update, and so, they are not vital for the convergence of the algorithm. However, in order to know if the algorithm is working correctly, this values should also be taken into account. That said, the updates are given by:

$$\mu_{\bar{m}}^{\bar{P}}(l+1) = \mathbb{P} \left[\mu_{\bar{m}}^{\bar{P}}(l) + \beta^{\bar{P}} \frac{\partial \mathcal{L}}{\partial \mu_{\bar{m}}^{\bar{P}}} \right] \quad (3.79)$$

$$\mu_{\bar{m}}^{\underline{P}}(l+1) = \mathbb{P} \left[\mu_{\bar{m}}^{\underline{P}}(l) + \beta^{\underline{P}} \frac{\partial \mathcal{L}}{\partial \mu_{\bar{m}}^{\underline{P}}} \right] \quad (3.80)$$

$$\beta^{\bar{P}} = \beta^{\underline{P}} = 0.5 \left[\frac{\text{m.u}}{\text{V}^2} \right] \quad (3.81)$$

3.3. Power and Voltage Setpoints to control the Droop Curve

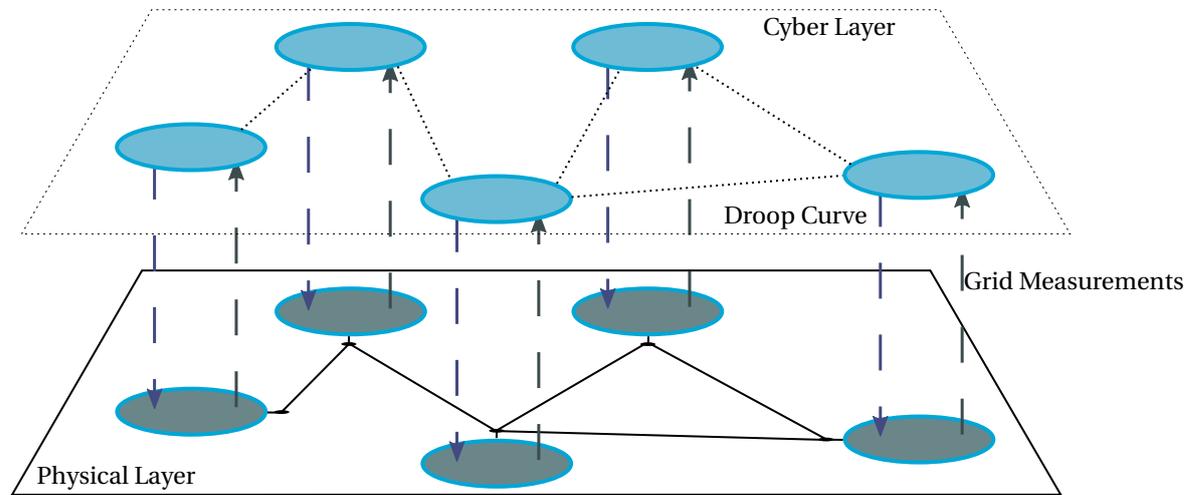


Figure 3.6: Scheme of the interaction of the physical and cyber layers

As it was mention in sections 3.1 and 3.2 there is an interaction between the cyber layer, which takes measurements from the grid in order to preform the optimisation process, and the physical layer, which has droop control implemented in every node with a PE interface that needs to be defined by the optimisation layer. This interdependent interaction it the core of the control and optimisation strategy which will provide the grounds to develop online optimisation.

For simplification purposes, the current limits on the PE converter were set high enough that they can be disregarded, for now.

3.3.1. Building the droop curve

Referring back to Figure 3.2 and the section where Droop Control was previously discussed, five zones of operation can be identified: two for power limits, two for voltage limits and fifth one which would be set by the main droop curve. If it is assumed that neither the power limits nor current limits of a PE interface would change, meaning there would not be a change in the generator/load side of the converter, then only the droop linear curve will be subject to change during the optimisation process. As such, it is important to explain how this line is set in I-V coordinates.

From (3.9) it is possible to see that there are two components that need to be defined in order to obtain the function: d' and i_m^0 .

Defining d'

The term d' is the slope of the droop function and as such it establishes the rate at which the current should change in case there is a voltage variation. As such it can be written as:

$$d' = -\frac{\Delta i^S}{\Delta u} \quad (3.82)$$

This poses a problem since d' , which describes current in terms of voltage, is being calculated in the optimisation layer, which is being described in terms of voltage and power, i.e., the generation setpoint is p_m^S and not i_m^S . And so, the Δi^S must be obtained from a Δp^S . However, this was solved by defining \bar{i}^S and \underline{i}^S as:

$$\bar{i}^S = \bar{p}^S / \underline{u}, \quad \underline{i}^S = \underline{p}^S / \bar{u} \quad (3.83)$$

and by using this result in (3.82), d' is formulated as is then:

$$d' = \frac{p^S / \bar{u} - \bar{p}^S / \underline{u}}{\Delta u} \quad (3.84)$$

In accordance with DC standardisation discussions, $\Delta u = 5V$ while $\Delta p^S = \bar{P}_m^S - \underline{P}_m^S$, which means that in any operating point, it would take a deviation of 5 V to decrease power output from the maximum to the minimum limits. If the PE interface is connected to a variable load, the same 5 V difference would mean that power consumption would go from minimum consumption to maximum. And so, d' is chosen as:

$$d' = \frac{p^S / \bar{u} - \bar{p}^S / \underline{u}}{\Delta u} \left[\frac{A}{V} \right] \quad (3.85)$$

Defining i_m^0

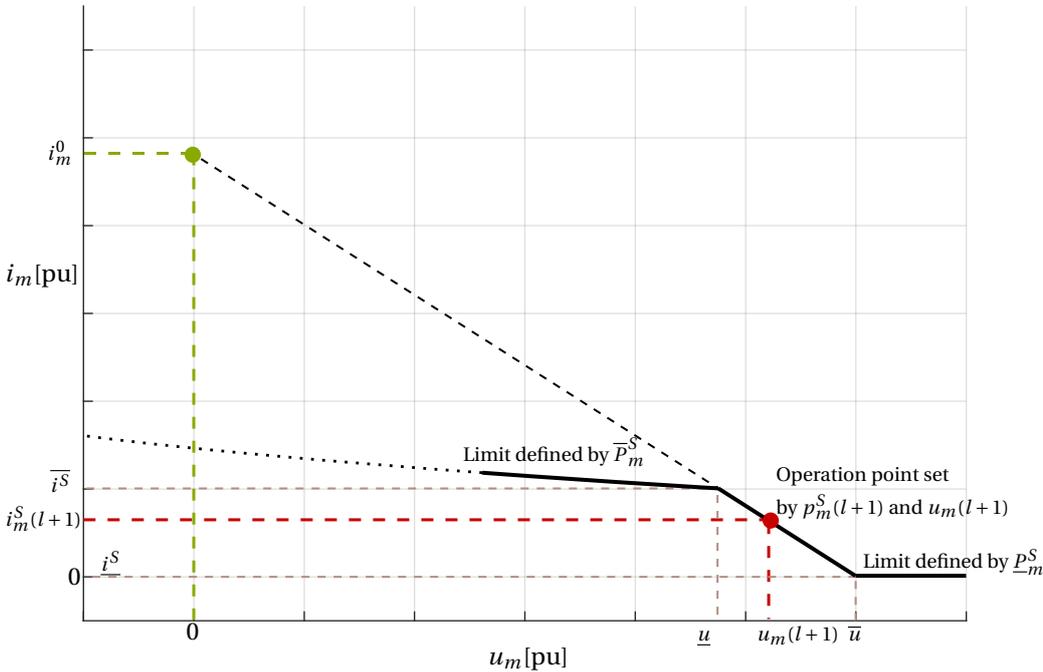


Figure 3.7: Finding the i_m^0 of a generator, for iteration l

Once the slope has been defined, the i_m^0 can be calculated. This point will be derived from the power and voltage setpoints, $u_m(l+1)$ and $p_m^S(l+1)$ for the next iteration. From Figure 3.7 it is possible to see that i_m^0 is the point of the droop function for which $u_m = 0$ and so, from (3.9), its value can be obtained as:

$$i_m^0 = i_m^S(l+1) + d' \cdot u_m(l+1) = \frac{p_m^S(l+1)}{u_m(l+1)} + d' \cdot u_m(l+1) \quad (3.86)$$

3.3.2. Droop in the Marginal Generator Region vs Constant Power Region

Now that the mathematical base behind the generation of the droop curves as been established, the strategy to generate said setpoints is going to be explained for both the Marginal Generator Region and the Constant Power Region, since they differ slightly.

On one hand, when generating d' , the main difference comes from the values that are chosen as \bar{u} and \underline{u} , since it is assumed that the power limits are fixed for the entire duration of the optimisation process. And on the other hand, the setpoints which generate i_m^0 also differ from one zone to the other.

This difference was necessary for the droop curves to be in accordance with the DC standardisation discussions and because it made sure a solution was found when running the power flow calculation, something that will be discussed later in more detail.

Generating Droop in the Constant Power Region

By definition, the Constant Power Region is a region where the node is either at minimum or maximum power. When looking at the whole droop control, this means that the point of operation will not be in the linear droop section defined by (3.9) and, as such, it cannot be used to calculate i_m^0 .

Whenever the converter is in this region, the droop curves need to be set such a way that allows for the correct functionality of the optimisation and at the same time, safeguards the integrity of the devices.

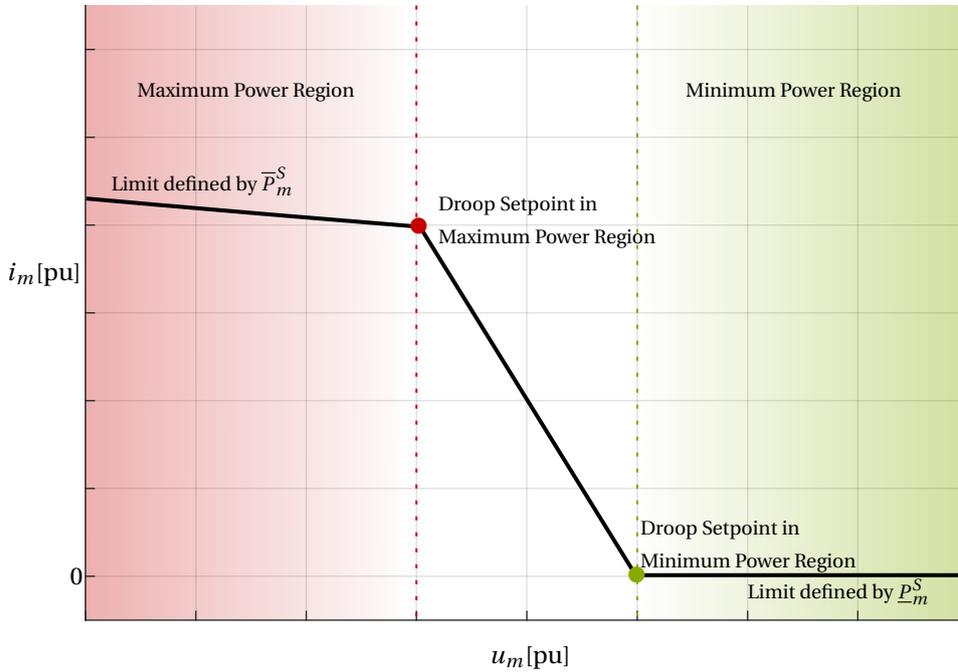


Figure 3.8: Current and Voltage setpoints for the maximum and minimum power region

Generating Droop in the Minimum Power Region

When the converter is in the minimum power region, the chosen values for \bar{u} and \underline{u} were \underline{U}_m and $\underline{U}_m - 5$ V. This allows for a 5 V margin beyond the defined minimum for the node which help act in cases of extreme necessity. As for the values of power and voltage used to calculate i_m^0 , \bar{P}_m^S and $\underline{U}_m - 5$ V are used.

As such, the PE interface will always be providing minimal power unless the local voltage is under the defined nodal limit. As an example, this would mean that, in case of small voltage oscillations when the system is not in stress, $u_m > \underline{U}_m$, would not result in expensive generators turning on and off, which would increase costs, but if the system is subject to stronger disturbances and is near its limits, even the more expensive generators would be used to secure voltage stability, since it would be more important to safeguard the integrity of the devices than to minimise operation costs.

Looking at Figure 3.8, the green dashed line represents the limit between normal operation ($u_m \geq \underline{U}_m$), in a green tint, and emergency operation ($u_m \leq \underline{U}_m$), and the red dashed line represents the voltage limit at which the generator would be generating full power, $\underline{U}_m - 5$ V.

Generating Droop in the Maximum Power Region

The reasoning behind the droop curve generation in this zone is the exact opposite. When in this operation state, it is not optimal for a PE interface to be reducing power in case there are small voltage oscillations, unless it is near the operation limits, since the converter that are at the highest power output are the ones connected to the cheaper sources, and, therefore, to minimise costs, should be the ones providing power while any marginal generator would then manage those voltage oscillations.

Hence, the chosen values for \bar{u} and \underline{u} were $\bar{U}_m + 5\text{ V}$ and \bar{U}_m , in order to account for the same 5 V margin, and, to generate i_m^0 , \bar{P}_m^S and \bar{U}_m were used.

Figure 3.8, the red dashed line represents the limit between normal operation ($u_m \leq \bar{U}_m$), in a red tint, and emergency operation ($u_m \geq \bar{U}_m$), and the green dashed line represents the voltage limit at which the generator would be turned off, $\bar{U}_m + 5\text{ V}$.

Generating Droop in the Marginal Generator Region

The approach to generate d' in this region is pretty straight forward. \underline{u} is chosen to be \bar{U}_m and, as such, $\bar{u} = \bar{U}_m + 5\text{ V}$, in order to be consistent with the Maximum Power Region which was previously discussed.

After the local variable updates are preformed, both $p_m^S(l+1)$ and $u_m(l+1)$ are used to generate i_m^0 . It is important to note that if either $u_m(l+1) > \bar{U}_m$ or $u_m(l+1) < \underline{U}_m$, then the $u_m(l+1)$ is set as the limit value.

Generating Droop in a Load with no Cost Function

There can also be the case that there are loads in the system which don't have any cost function, meaning that they are to be on whatever the local cost is. In that case, the same approach is taken as for the Minimum power region, since it was conventioned that loads would have currents, $i_m^S < 0$, and, therefore, negative power. As such, the \underline{P}_m^S is the maximum power that it can draw and \bar{P}_m^S the minimum, which is usually taken as 0 W.

The same reasoning applies, meaning that if the voltage drops to low, then the PE interface connecting the load to the grid would reduce the power consumption of said load in order to preserve system stability and, otherwise, power consumption would stay at maximum level.

3.4. Simulation vs. Real Life Implementation: Pros and Cons

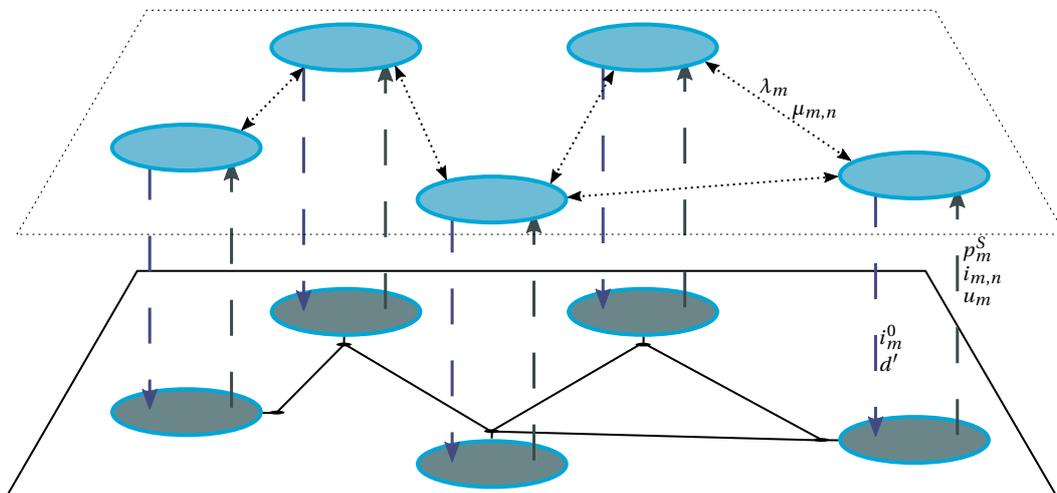


Figure 3.9: Same scheme from 3.6, with the information traded between nodes and between the cyber and physical layer specified.

Up until now, the optimisation process was described in the way that it is intended to be applied in real life scenarios. However, due to the limitations associated with trying to implement and test it in DC grid, such as safety and implementation costs, the whole algorithm was developed and tested with the help of computer simulations.

However, this meant that some approximations had to be made in order for the tests to be feasible and that some limitations were found, meaning that the full potential of this algorithm could not be tested.

In order to run simulations of the whole system, the physical and cyber layers were separated into two different parts. The physical layer was simulated in a central computer, which also served as the simulation

controller and where the simulation data was stored for future analysis. This computer would then communicate with the cyber layer, which represented the different nodes. The central computational unit would then send the grid measurements necessary to the optimisation process, i.e., local power, voltage and line current measurements, and receive the droop curve setpoints, as represented in Figure 3.9.

Physical Network Simulation

In section 3.1, the equations for EDC power flow calculations were presented and lay the foundation for the DC grid simulations. This grid was then implemented in Python [29], with the help of a modified version of the pandapower library [30] which was used to save the network description and run centralised OPF to verify the results obtained with decentralised OPF, as well as the SciPy [31] library, which included the solvers necessary to run EDC power flow, and the pandas [32] library, which was used to store all important data from the simulations. The simulation results were plotted with matplotlib [33].

Node Simulation

The nodal behaviour was also programmed in Python [29] with some of the above mentioned libraries. These nodes were simulated both as different processes running in the same machine as the main grid, if communication induced errors and delays were to be ignored, or in separate microcomputers which were subjected to those communication constraints. The chosen microcomputer was the Raspberry-Pi, due to its flexibility and processing power, and they are connected to each other in a private WiFi network, similarly to the implementation on [27].

In a real world implementation, these nodes would also be subject to communication delays and loss of information, and as such, it was important the algorithm was also tested in those conditions. However, it also introduced a random element to the simulations which made it harder to correct problems that might have been related to the implementation of the algorithm. As such it was important to have both implementations, in order to benchmark and develop solutions for problems that might arise in communication dependent implementations.

Differences between the implemented solution and real life solution

Having described the implementation in the simulated environment, it becomes important to highlight its differences to an implementation on a physical DC grid:

1. Communication between the physical and cyber layer

Since in the simulation environment the physical grid is implemented in one place and the cyber layer in another, there needs to be communication between the nodes and the grid so the droop curves can be generated in each iteration and, after the power flow calculation is done, the grid measurements need to be sent back to the nodes. When running the nodes and the grid simulation on the same machine, this is not a problem because the communication delay is negligible, but, once every node is implemented in a single Raspberry-Pi, the communication between nodes and grid becomes subject to the same loss rate and delays as inter-node communication.

In a real world implementation that would not be the case. The PE interface would have multiple modules, one responsible for actively controlling the power with droop which would be directly connected to a microprocessor responsible for handling the optimisation and the communication with other nodes, meaning the communication delay would be substantially smaller. The grid measurements would also be obtained by the PE converter, and so, obtaining them would be a matter of increasing or decreasing the sample frequency.

2. Grid Simulation is not instantaneous

Once every iteration, the physical layer simulation must run the powerflow calculation. In order to do so, the computer has to solve a system of $\mathcal{N} + \mathcal{M}$ non-linear equations, as described in Section 3.1. This simulation will take, at best, at least 10 ms to complete and, in some cases, it can take up to 1 s, for relatively small grids of 4 or 5 nodes. As the scale of the network increases, so does the calculation time. Many solvers were tried in order to mitigate the problem and the best solution was found in a combination of both the “*hybr*” and “*broyden1*” methods from the SciPy optimise package, but, non the less, the simulation still took a non negligible amount of time.

In the real world, any changes in power or voltage would manifest themselves immediately on the grid.

3. System dynamics are not taken into account

In an ideal DC system, the lines that connect the different nodes would only have a restive element, meaning current and voltage could change instantaneously. However this is not the case. Every line has an associated inductance and capacitance and, as such, no change in current or voltage is instantaneous, meaning there are dynamic phenomena every time the current or voltage changes.

This, however, is not taken into account in the grid simulation since what is done every iteration is a steady state powerflow calculation, i.e, it is calculated what the would be the state of the system after the transient phenomena has passed. The main reason why this was not taken into account is the fact that adding this elements to the simulation would then increase the computational complexity immensely and, as such, trying to simulate and preform online optimisation would not be feasible.

3.5. Synchronous Simulations

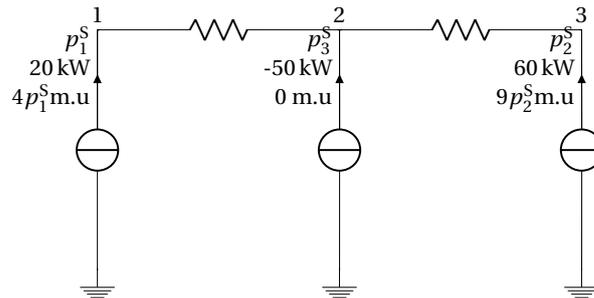


Figure 3.10: A three node radial grid with two differently priced generators and a load.

Having described the algorithm in the previous sections and keeping in mind the limitations that it is subject to while running it in a simulated environment, it is now possible to see how it performs. It is important to note that, in all voltage the figures in this report, \hat{x}_m stands for the measured parameter at node m and \hat{x}_m is the setpoint for said parameter, in that same iteration.

The first tested case was a simple 3 node network with two generators, one which has less capacity, but is cheaper and another one more expensive but with higher capacity.

Looking at Figures 3.11-3.14 it is possible to observe that the system reaches the optimal solution around the 500th iteration, when the dual variable for the maximum voltage limit on node 3, the node with the highest voltage, stabilises at $\mu_3^{\bar{U}} \approx 49$ [m.u./V]. The algorithm converged, for this case, around the same number of iterations as the implementation on [27] and, as such, no conclusion could be drawn from the increase speed of the new strategy, compared to the previous ones.

When the solution is reached, node 1 is at maximum power output, 20 kW, node 2 is at maximum power consumption, 50 kW, and node 3 is generating the 30 kW necessary to meet the demand on node 2, with an additional ≈ 910 W to compensate for the grid losses. This losses are being payed by the load in node 2, seen in Figure 3.11 by the higher value of $\lambda_2 \approx 9.41$ m.u./W.

Analysing the simulation more carefully, it is possible to observe that, in the beginning of the process, the LMP in every node reached 4 m.u./W, where generator in node 1 entered the marginal zone and started to allow the local power set point to increase until it reached the maximum power output, seen as a blue dash-dotted line in Figure 3.13. Then, the generator entered the maximum power region and its LMP started increasing again. This increase in power can also be seen in the local voltage, since more power was put in the grid, the local voltage was allowed to rise a to around 328V, 3 V above the minimum level. But since the load in node 2 also increased its consumption at the same rate, seen as the orange line in Figure 3.13, the overall system voltage could not increase to the maximum voltage limits. Only when the generator in node 3 reached its minimum marginal price, 9 m.u./W, the power was then allowed to increase again and, as such, the voltage also increased.

The small voltage dip around 100th iteration can be explained by the big spike in $\mu_3^{\bar{U}}$, seen in Figure 3.14. Since $\mu_3^{\bar{U}}$ is taken into account on the voltage update, as given by 3.65, the high values taht it takes actually makes it so that $-\alpha_\lambda^u \partial \mathcal{L} / \partial \lambda_m - \alpha_u^u \partial \mathcal{L} / \partial u_m < 0$, and so the voltage starts to decrease. The cause of this large spike is the fact the high voltage increase rate after the generator in node 3 reaches its marginal region, which is given mostly by $\partial \mathcal{L} / \partial u_m$, which, once the limit is reached, is going to translate to high increase rate of $\mu_3^{\bar{U}}$.

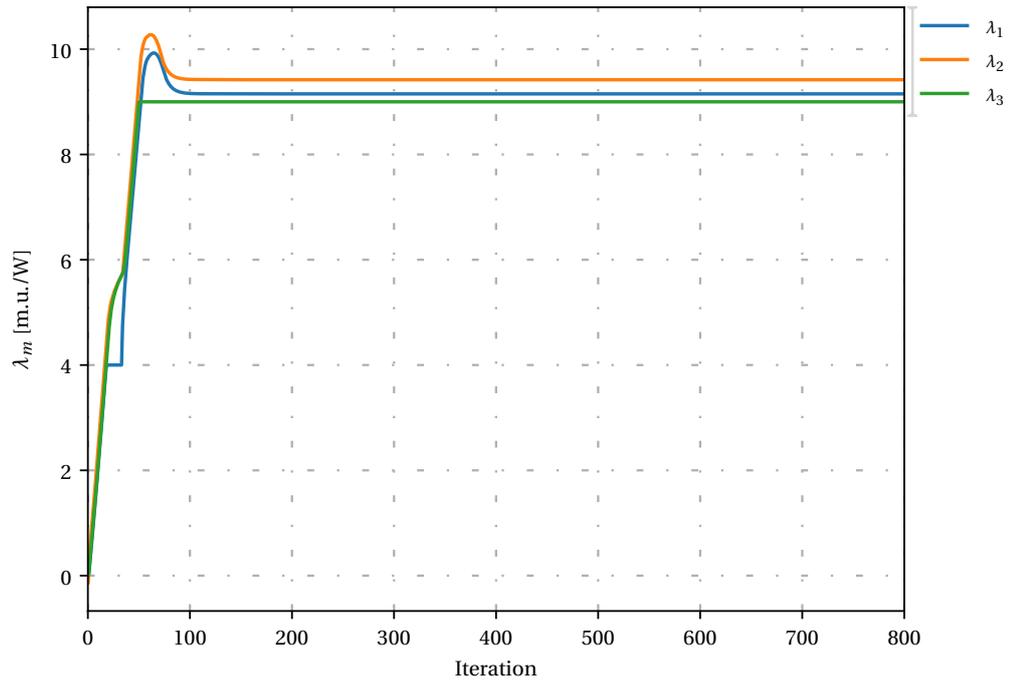


Figure 3.11: LMP at all the nodes in the case presented in 3.10.

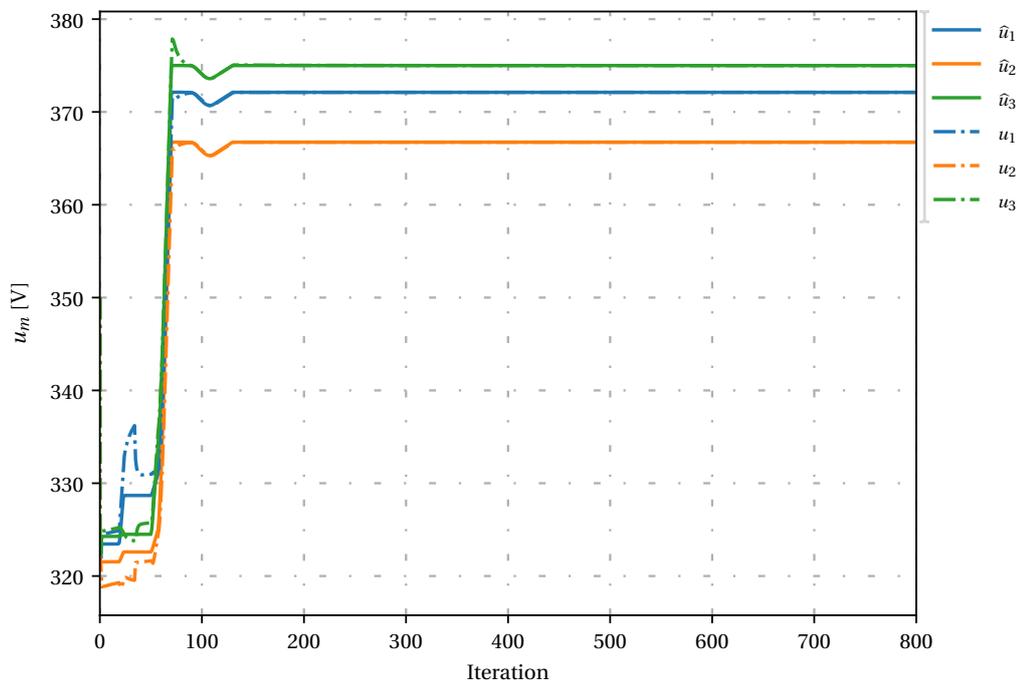


Figure 3.12: Local voltage levels at all the nodes in the case presented in 3.10. The difference between the voltage setpoints, dotted lines, and measured voltage can be seen in the first 100 iterations.

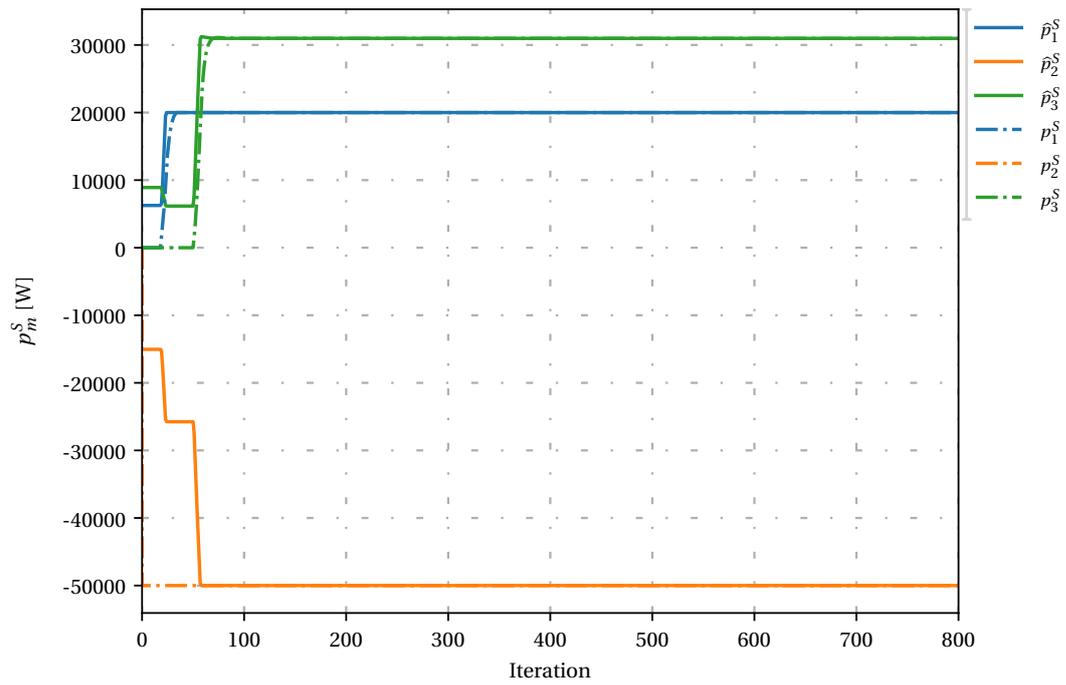


Figure 3.13: Local nodal power at all the nodes in the case presented in 3.10. The difference between the voltage setpoints, dotted lines, and measured voltage can be seen in the first 100 iterations.

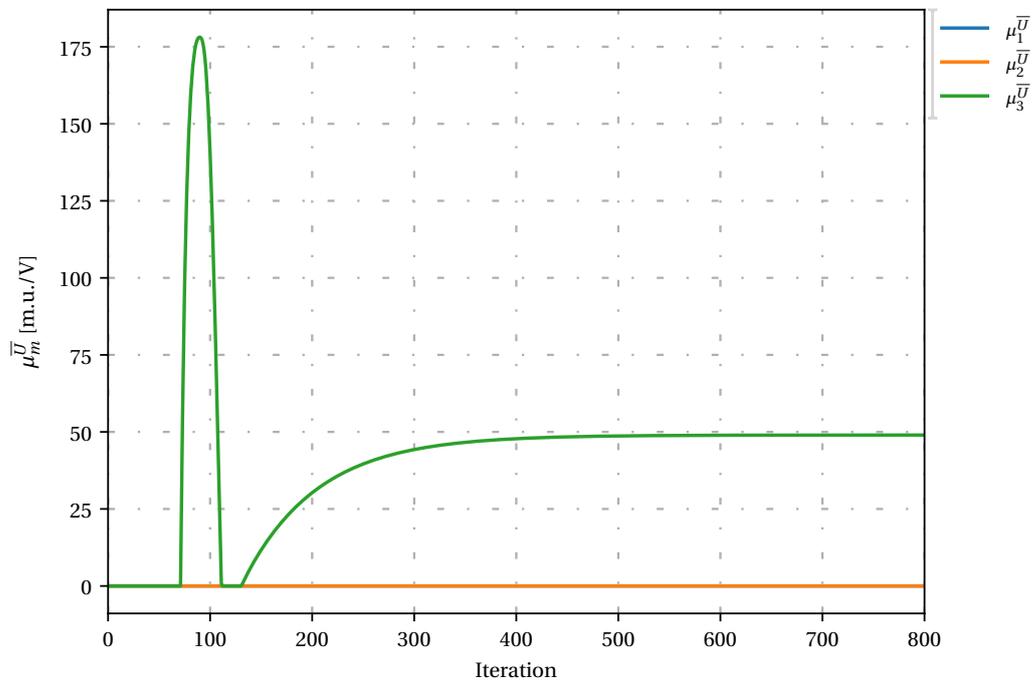


Figure 3.14: Dual variable for the maximum voltage constraint at all the nodes in the case presented in 3.10.

It is important to also note that, while the setpoints, seen as dash-dotted lines in Figure 3.13, are 0 W, the generators 1 and 2 are producing some power in order to account for the demand on the load in node 2. The amount of power generated by nodes 1 and 3 is fully dependant on the droop curves and the on network.

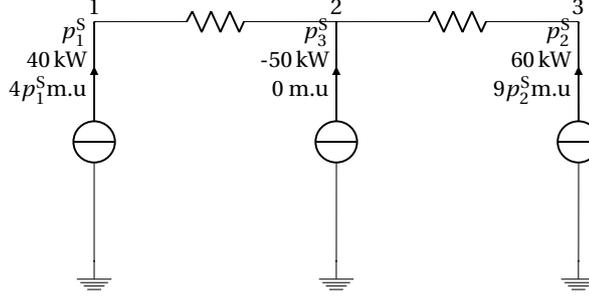


Figure 3.15: Grid in 3.10 with a 20 kW increase power output limit in node 1.

Increasing the output of node 1 by 20 kW will turn the grid in Figure 3.10 into the one in Figure 3.15. In [27], this was the case where oscillations were verified for non-quadratic cost functions and the found solution limited the algorithm to only be able to handle situations where $A_m \neq 0$, as discussed in 3.2.4. Running the same case with the strategy described in this report, the results are very different, as can be seen in Figures 3.11-3.14.

In Figures 3.16 and 3.17 it is shown a comparison between this and the previous implementations, respectively, and it is possible to observe that before the λ_m 's rate of increase is lower, meaning the algorithm will take more time to converge, and, more importantly, there are oscillations around what would be the optimal solution, shown in Figure 3.16. Now, even though the marginal generator is again in node 3, since node 2 cannot accommodate for the whole demand, $\lambda_1 \approx 8.62$ m.u./W, which is lower than the marginal cost of $\lambda_3 \approx 9$ m.u./W. Once more, the LMP in node 2 is the highest since it is the load, but in this case it presents a lower value than in the previous case, $\lambda_3 \approx 9.15$ m.u./W, even though there are more losses on the grid, ≈ 1231 W. This can be explained by the fact that most of the power is now coming from a cheaper source meaning it is less expensive to the grid to have more losses coming a cheaper source than having less losses, but the power coming from a more expensive source. This makes sense from an optimisation point of view since the whole purpose is to minimise the costs, and not the losses, and, although they are connected, sometimes having more losses in the grid is cheaper in the long run, as it was demonstrated in this case.

In Figures 3.18 and 3.19 it is now possible to see that node 1, the one with the highest voltage also has the highest power output. From a physical point of view, these results make sense because the higher the voltage difference the higher the current and power being transmitted in a line and, as such, the generators supplying the highest amount of power will tendentially be the ones with the highest voltage.

Once more, the small voltage and power oscillations are caused by a very steep rise in $\mu_1^{\bar{U}}$, which, as before, causes these oscillations to happen. In this case power also shifts since the voltage responsible for the voltage oscillation is node 1, which is in the maximum power zone, thus having a fixed droop curve. Once the voltage droops in 1, it translates to a voltage drop in other nodes, which triggers the droop in node 3 to act and provide more power to reestablish the previous voltage levels. Since the load cannot accommodate for this increase in power, then the droop in node 1 reduces the power output, meaning the power imbalance between the power setpoint and the actual power increases, and so increase $|\partial \mathcal{L} / \partial \lambda_1|$. This differential term is part of the voltage update and so, once the influence of $\mu_1^{\bar{U}}$ is small enough, the voltage level increases once more to $u_1 = \bar{U}$.

This feedback system, if not properly tuned, can cause oscillations and this is the main reason why $\beta^{\bar{U}} = \beta^{\bar{U}} = 10$ m.u./V².

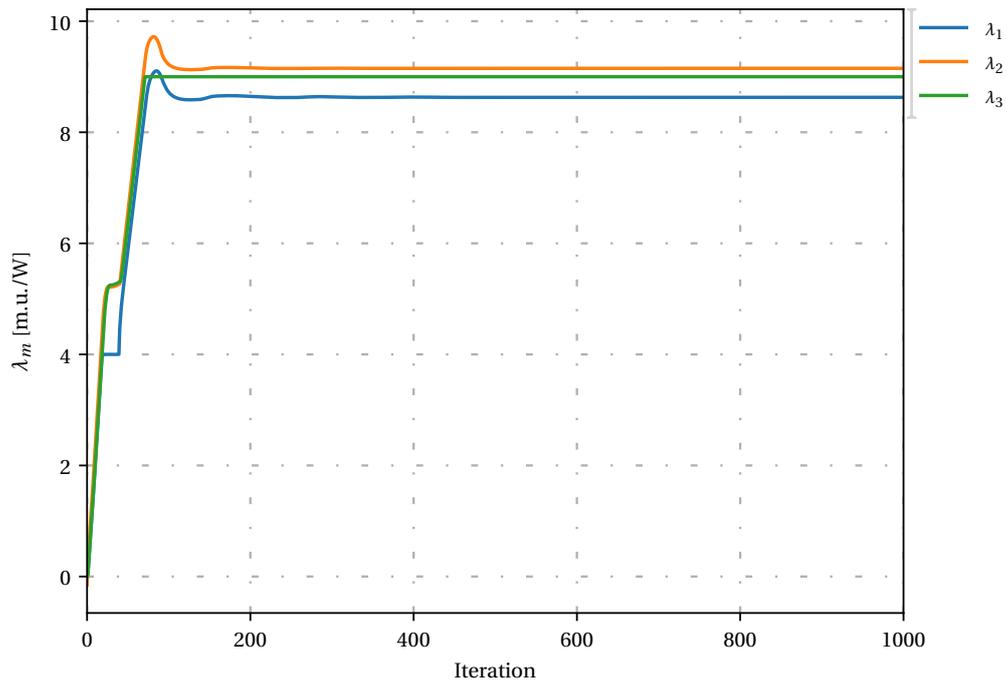


Figure 3.16: LMP at all the nodes in the case presented in 3.15.

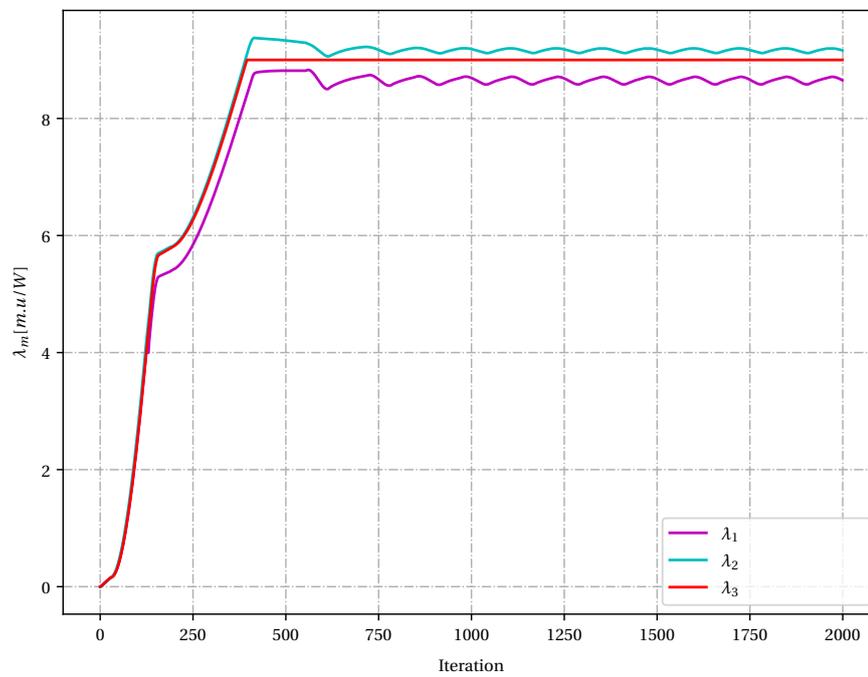


Figure 3.17: LMP at all the nodes in the case presented in 3.15, taken from [27].

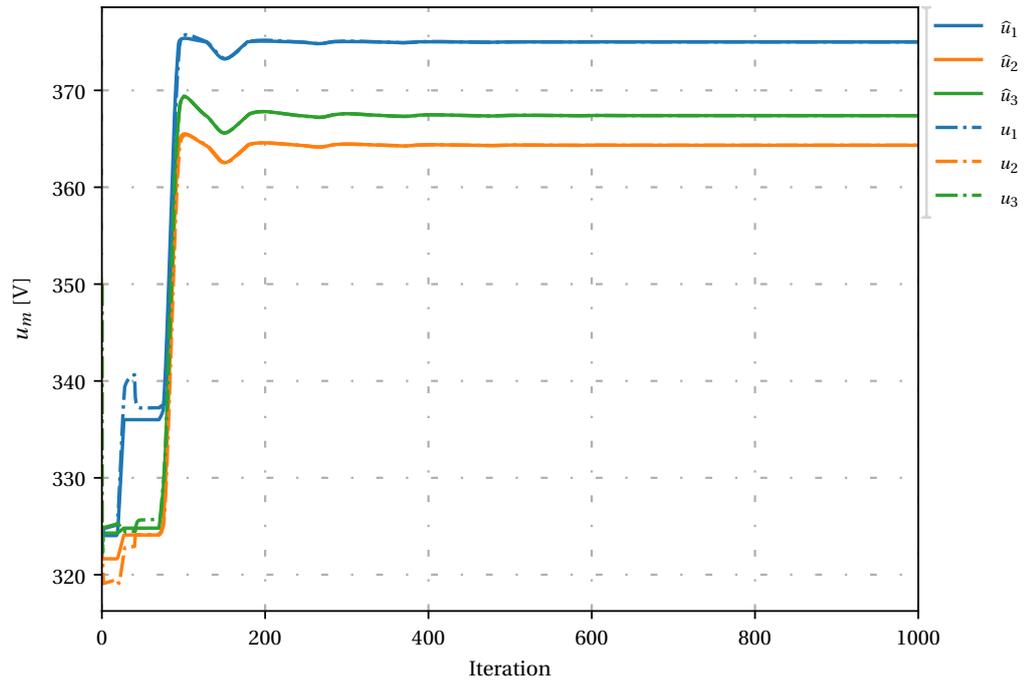


Figure 3.18: Local voltage levels at all the nodes in the case presented in 3.15.

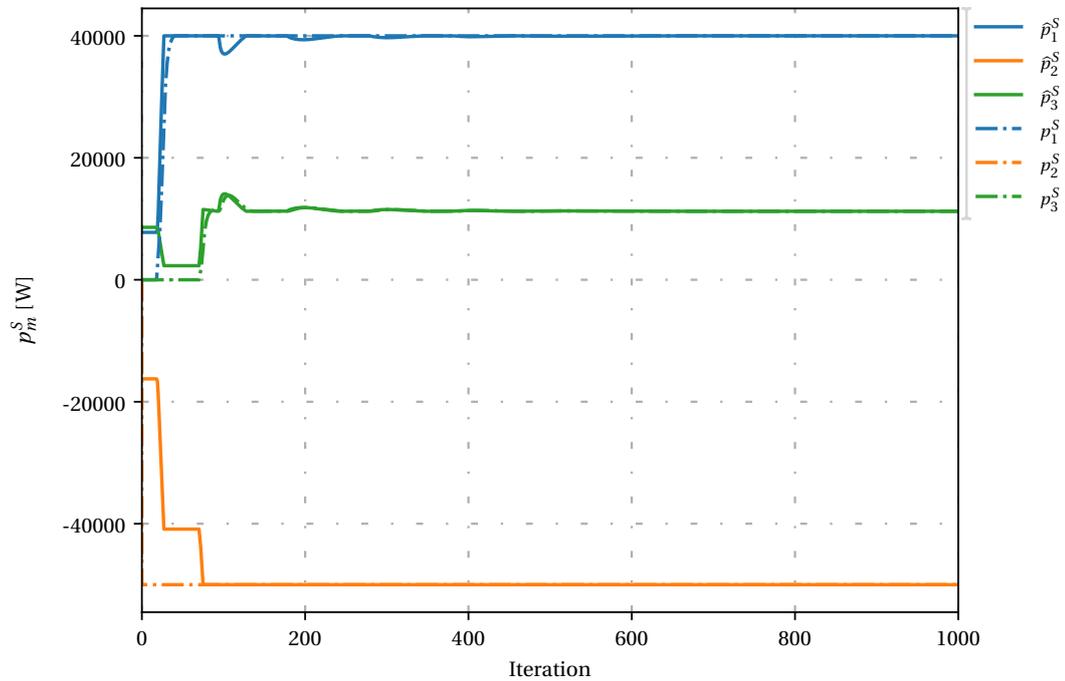


Figure 3.19: Local nodal power at all the nodes in the case presented in 3.15.

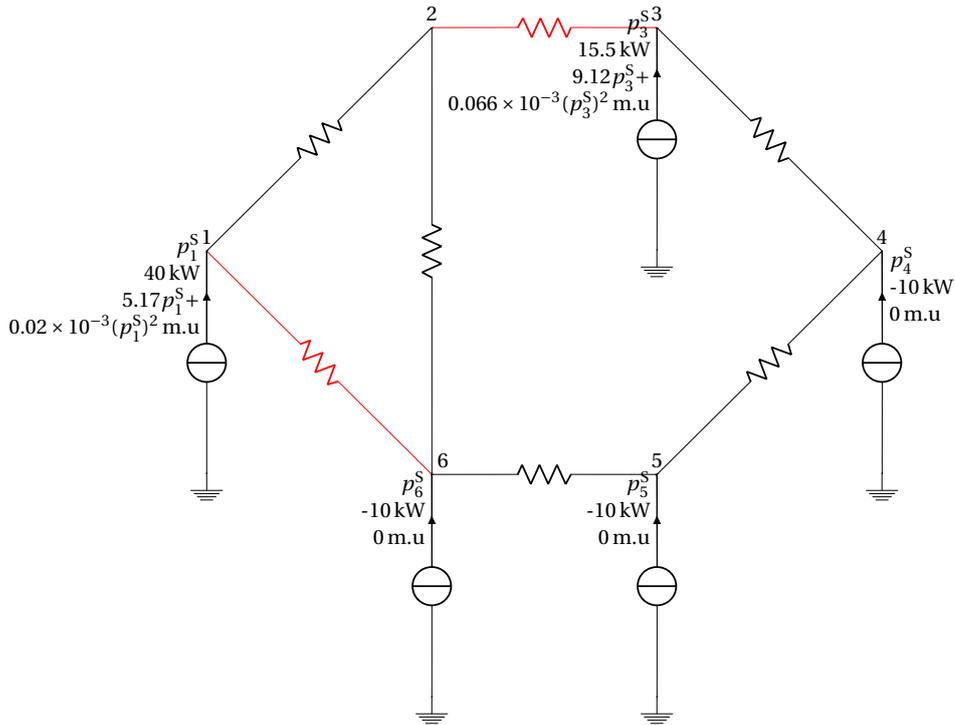


Figure 3.20: A six node meshed grid with line limit (denoted by the red lines). The line connecting node 1 to node 6 has a current limit of 50 A while the current flowing in the line from node 2 to node 3 is limited to 20 A

The optimisation algorithm also has to account for congestion in the grid, both current congestion as well as voltage congestion.

Current congestion, as explained previously, happens when the current that is flowing in a line is higher than the limit that was defined, violating then (3.15), while voltage congestion happens when trying to transmit power over a highly resistive element, like a long line. The higher the resistance in a line, the bigger voltage drop the same current that flows through. If the current increases to much, and since voltage is fixed from the marginal generator, the connected node's voltage can hit the minimum limit, meaning that no more current should be allowed to flow.

In order to test the behaviour of the new optimisation strategy under these conditions, test networks from [27] were used, with the added benefit that, in this way, the convergence speeds could be compared.

The first test was performed in a 6 node grid, seen in Figure 3.20, where some of the lines have current limits set in such a way that it will constraint power flow.

Analysing Figures 3.21-3.25 it is possible to observe the algorithm converges and respects to an optimal point which respects the previously defined line limits.

From the grid layout in Figure 3.20 it is possible to conclude that, if there was no congestion, node 1 would be able to supply all of the loads, since it has a maximum power output of 40 kW and the network is only demanding 30 kW and, at 30 kW, $\lambda_1 = 5.17 + 0.02 \times 30 = 5.77$ [m.u./W] which means there would be no need for node 4 to generate power, since it would be much more expensive.

The problem arises when both the (1,6) and (2,3) exceed the maximum limit for current which can be seen in Figure 3.24, from the increase in the dual variable which starts around the 100th iteration. This line overload is caused by the natural flow of current over the line after the steep increase in supplied power in node 1, visible in Figure 3.23, means the $\mu_{1,6}$ and $\mu_{2,3}$ will begin to increase, as a consequence of the update described in (3.73). This increase in the dual variable will translate in a rise of the LMP of, firstly, nodes 2,3 and 6 as due to $\mu_{2,3}$, for nodes 2 and 3, and $\mu_{1,6}$, for node 6. The λ_1 is not affected since it is a marginal generator and, as such, $\lambda_1 \propto p_q^S$. These 3 nodes will increase then the LMP of the other 2 nodes in the system and it will continue to rise until the minimum marginal cost of the generator in node 3 is reached, at which point it will start to generate power, which can be seen around iteration 150. Once there is some power output at node 3, the current flowing in lines (2,3) and (1,6) decreases to a point where is no longer violating the maximum limit, which can be seen by a slowing rise of $\mu_{2,3}$, meaning this line is at the maximum current limit, and even

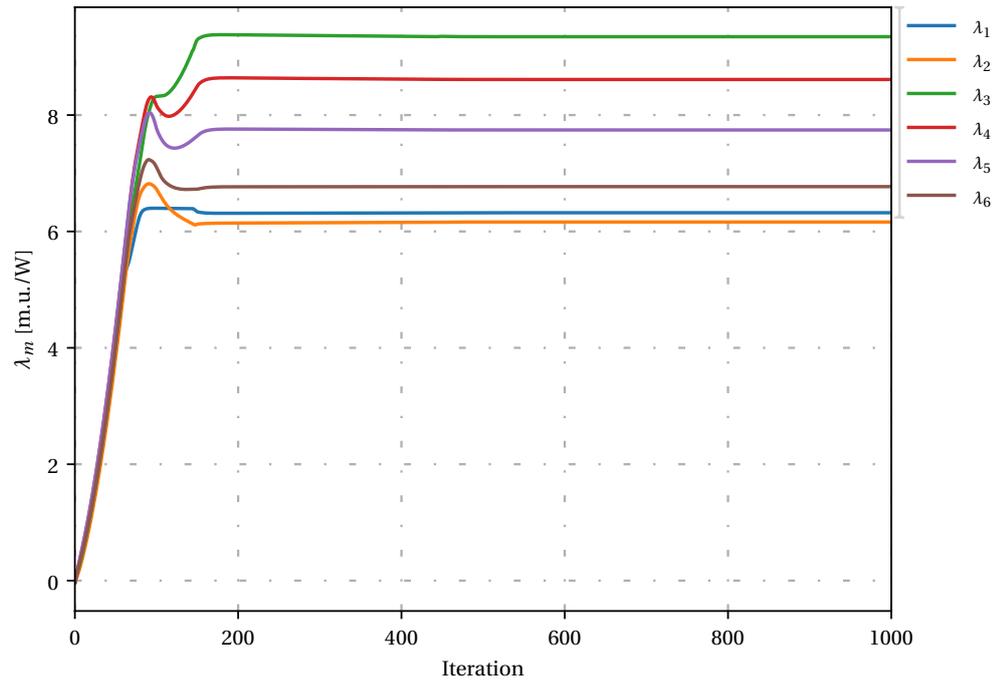


Figure 3.21: LMP at all the nodes in the case presented in 3.20.

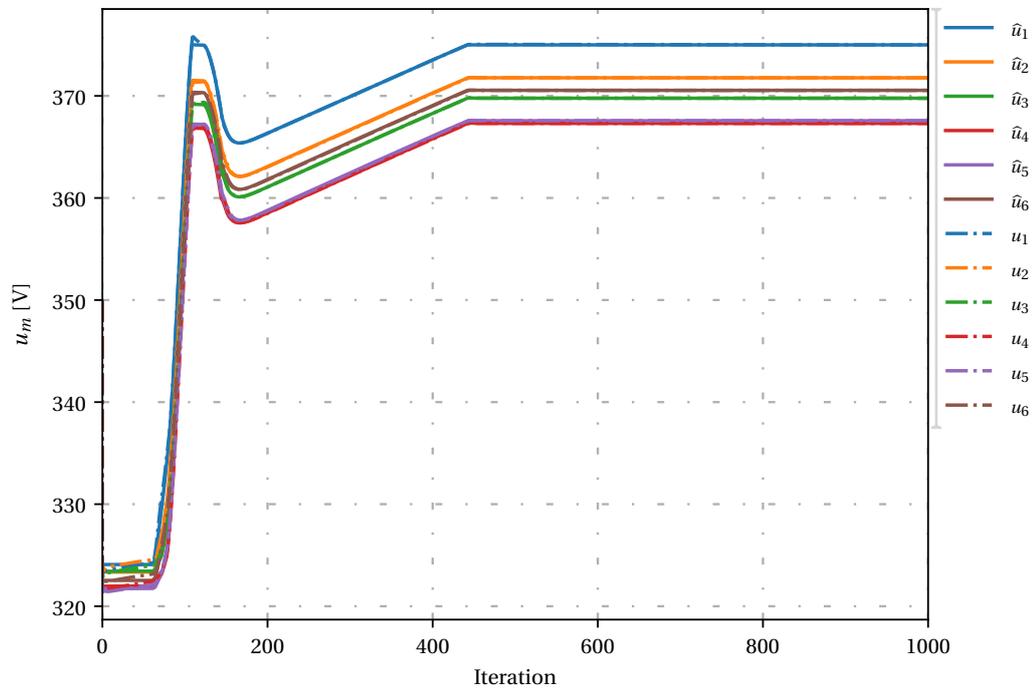


Figure 3.22: Local voltage levels at all the nodes in the case presented in 3.20.

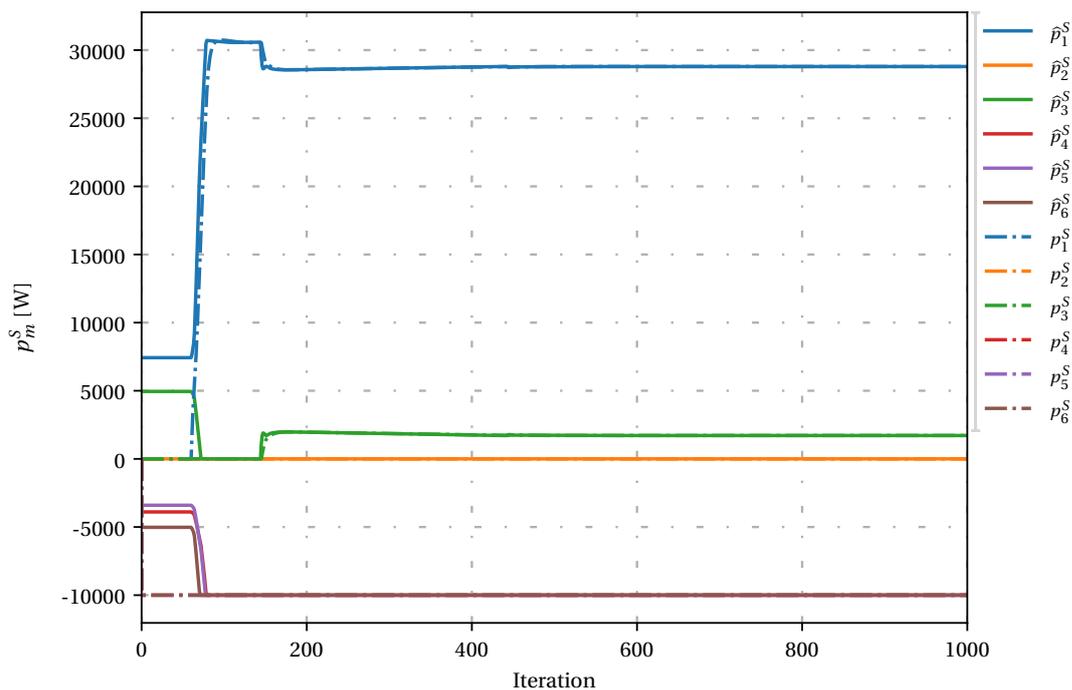


Figure 3.23: Local nodal power at all the nodes in the case presented in 3.20.

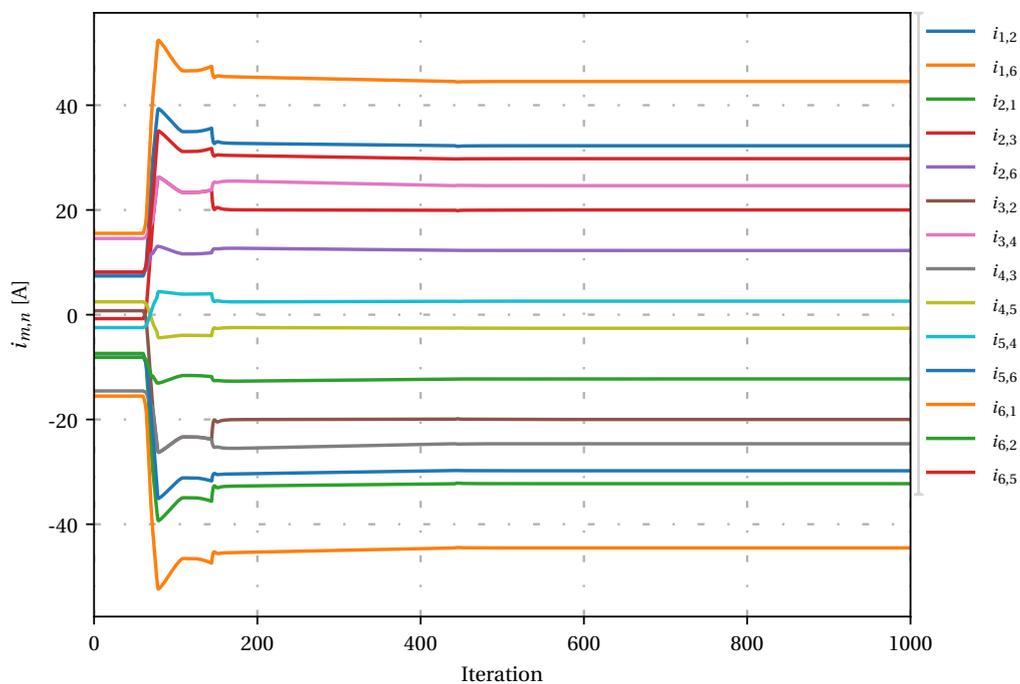


Figure 3.24: Line currents in the case presented in 3.20.

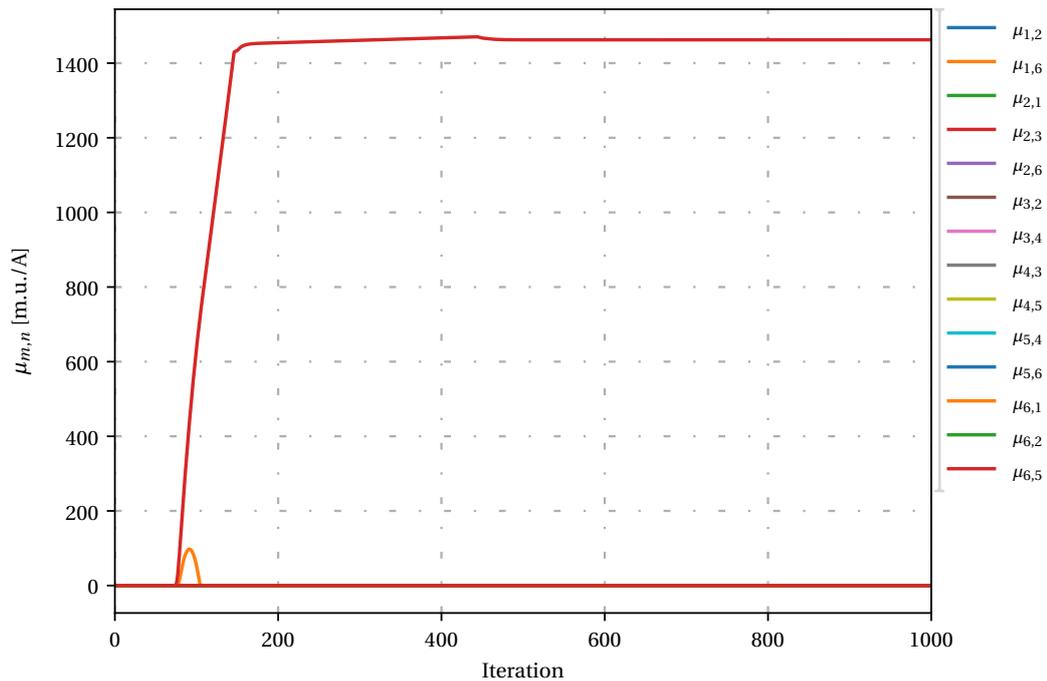


Figure 3.25: Dual variable for the line current limits in the case presented in 3.20.

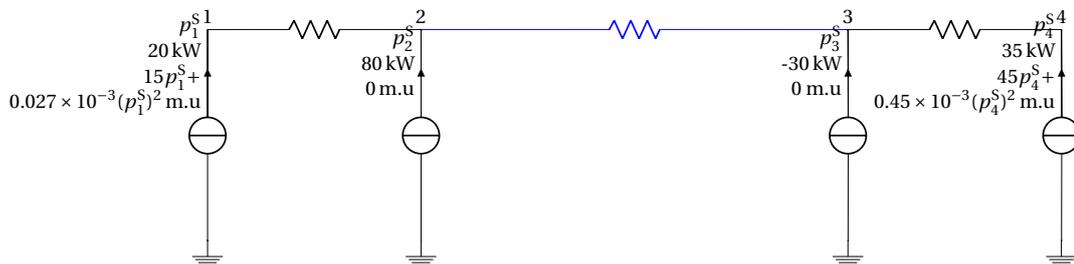


Figure 3.26: Four node serial case with a long line connecting node 2 and 3 (depicted in blue). The line is very long, and due to the voltage limits and losses, all the required power cannot be delivered from the cheapest generator in node 2.

a decrease of $\mu_{1,6} \rightarrow 0$, meaning this line is no longer congested, which can be confirmed in Figure 3.24.

The second congestion test was performed in a 4 node network, shown in Figure 3.26, where there is a long tie line that connects node 2 to node 3. In this case, the cheaper generators are in one side of the line and the loads on the other, which means power will flow from the area (1, 2) to (3, 4), hence creating voltage congestion in line (2, 3).

Initially, it can be seen that the power in node 2 starts to increase, since $\lambda_1 = 0$ m.u./W is the minimum marginal price for that generator. However power and voltage start to climb, there is a big voltage drop on the line that connects both areas. As such, once u_2 reaches its maximum value, there is no more power that can be transferred from the left most area. The voltage over the long line also forces $u_3 < \underline{U}$, which means the dual variable for the voltage limits will rise, as seen in Figure 3.31. This rise will push the LMP of the (3, 4) area up as a consequence of the differential term $\partial \mathcal{L} / \partial u_m$ in (3.65).

Only when $\lambda_4 = 45$ m.u./W, then the generator 4 is allowed to start increasing its power and then the local voltage also increases, although this rise happens very fast and is shown as a peak in Figure 3.28. This rise in voltage means, however, that node 2 will be transferring less power, as seen in Figure 3.29 as a small dip in \hat{p}_2^S , which is not optimal since power from node 4 is much more expensive, and also means $\mu_3^U \rightarrow 0$ very quickly and, consequently, will cause the voltage and LMP to drop at the same rate. The systems shows

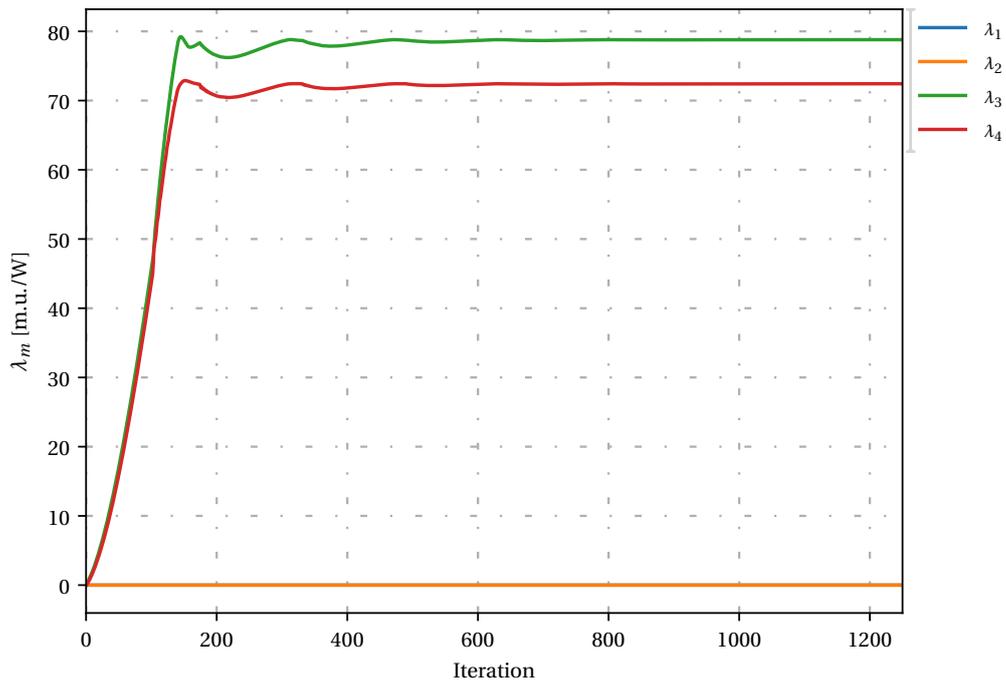


Figure 3.27: LMP at all the nodes in the case presented in 3.26.

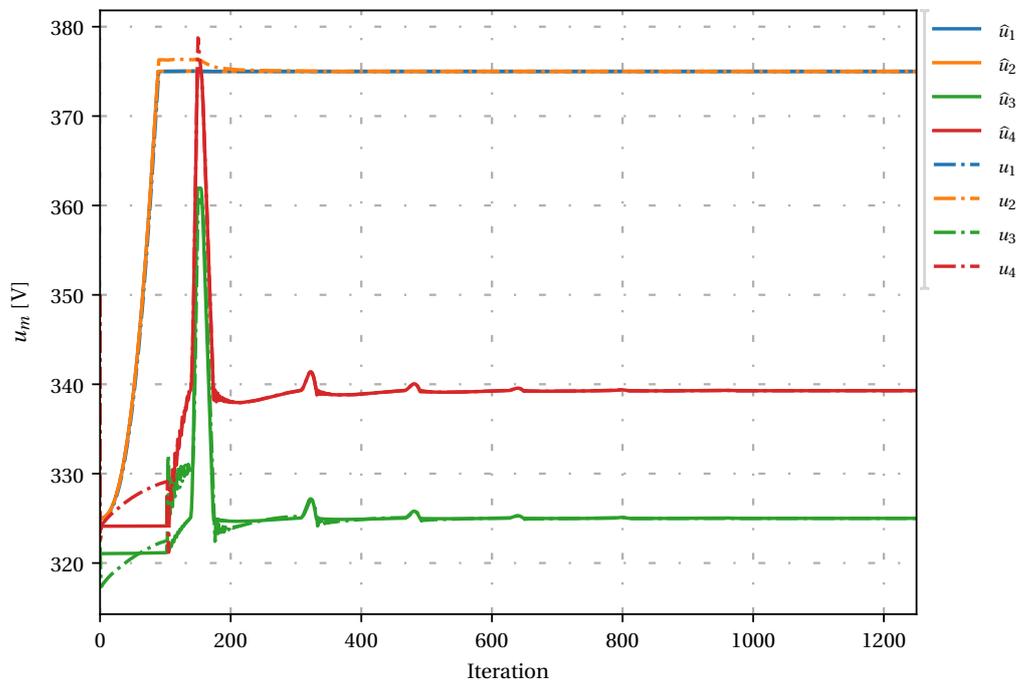


Figure 3.28: Local voltage levels at all the nodes in the case presented in 3.26.

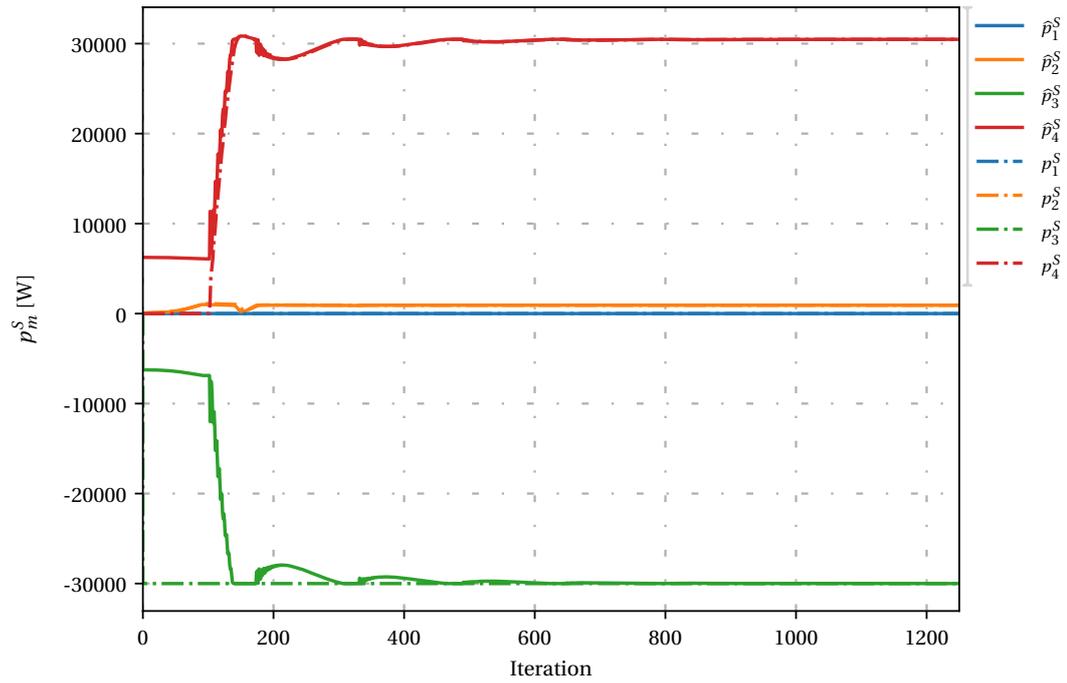


Figure 3.29: Local nodal power at all the nodes in the case presented in 3.26.

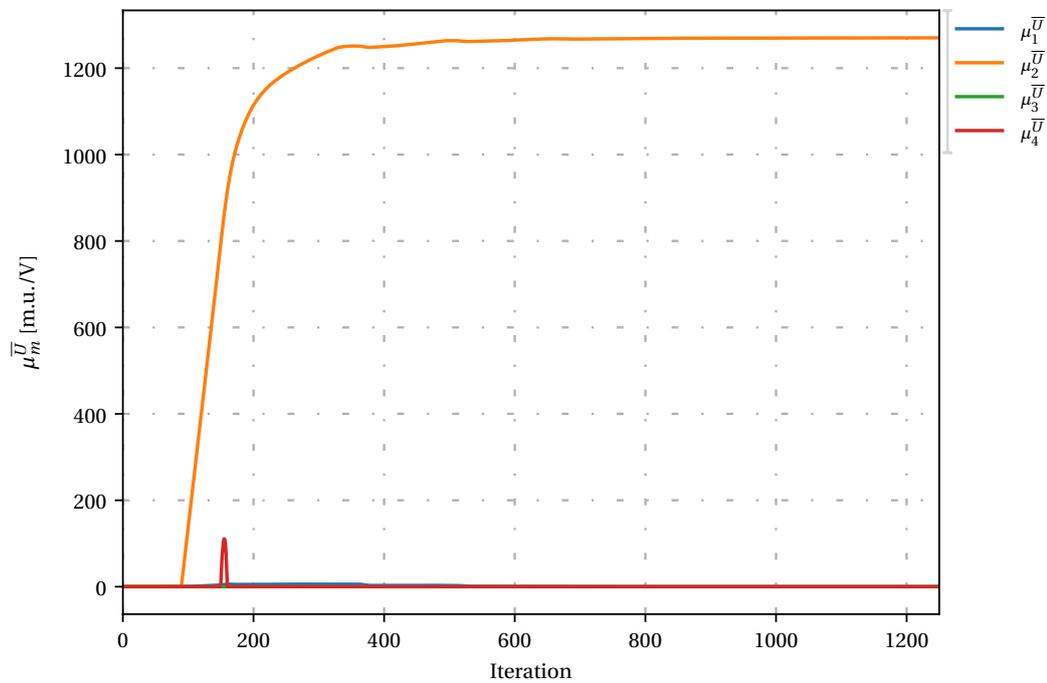


Figure 3.30: Dual variable for the maximum voltage limit at all the nodes in the case presented in 3.26.

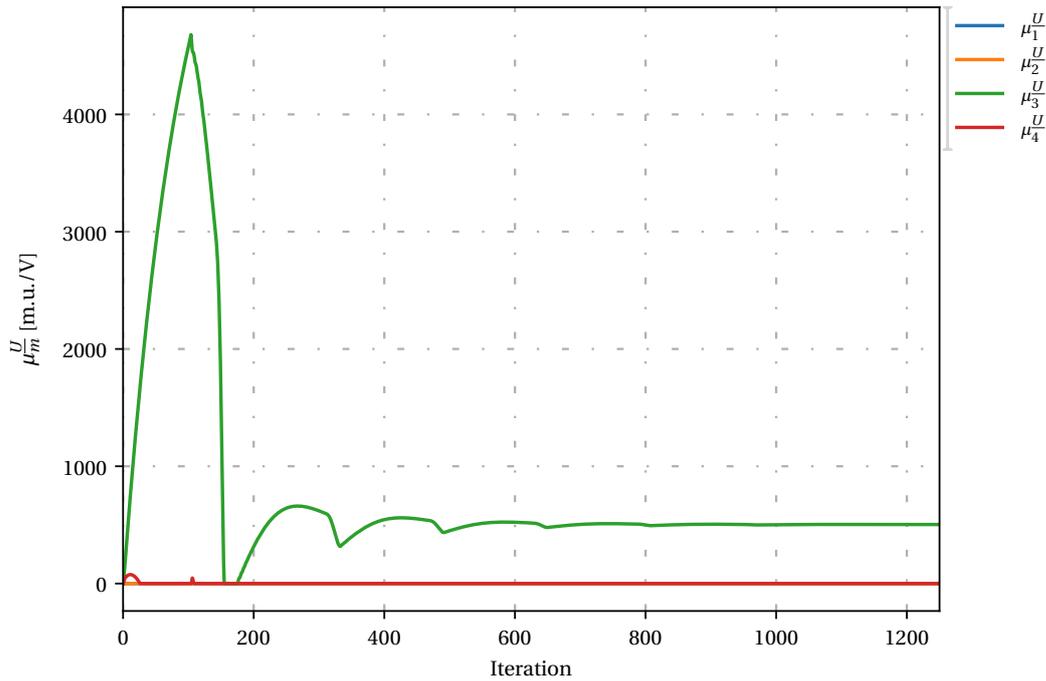


Figure 3.31: Dual variable for the maximum voltage limit at all the nodes in the case presented in 3.26.

damped oscillations as of result of this process, but, after 1000 iterations, it converges to a optimal steady state solution.

Both this cases were tested with the exact same grid as in [27]. In the case of line current congestion, an optimal solution was reached in ≈ 750 iterations which, when compared to the 3000 iterations obtained in [27], mean it takes a quarter of the time. Likewise, for the voltage congestion case, convergence was attained in less than 1000 iterations, which means hat for that case, the new solution is 3 times faster.

In order to test the convergence speed difference for larger networks, a test grid based on the IEEE 9 bus system was used. The layout of the network, as well as the costs and power output of the nodes can be seen in Figure 3.32 and the values for the line resistances are specified in Table 3.1.

Table 3.1: Line data for the IEEE 9 bus system. This values were taken from [27], as double the line resistance values of the grid in [34].

Line Index	Resistance (Ω)
(1,4)	0.1152
(2,7)	0.1250
(3,9)	0.1772
(4,5)	0.184
(4,6)	0.184
(6,9)	0.34
(5,7)	0.322
(7,8)	0.144
(7,9)	0.2016

Initially, this grid was ran with the exact same tuning parameters has all the other cases, but it led to the oscillations seen in Figure 3.33. These oscillations where created by the rapid increase in the dual variable of the maximum voltage in node 2, which led to the decrease in voltage and power, in a similar phenomena as the one explained for the previous cases. However, when the voltage would then again reach the maximum level, the rise and descent in μ_1^U would happened to fast, meaning the oscillations would start again, instead

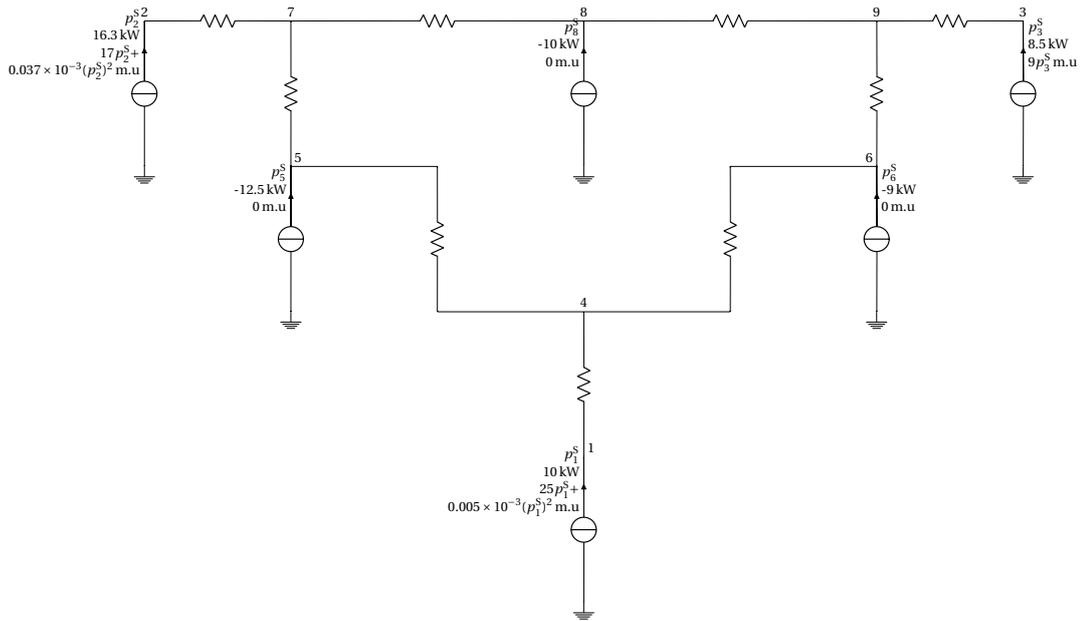


Figure 3.32: The IEEE 9 bus system topology with the modification in the generator and load capacity rating. The resistance values are taken from Table 3.1. Moreover, the generators' cost function are added to create an optimisation problem, as described in [27, 34].

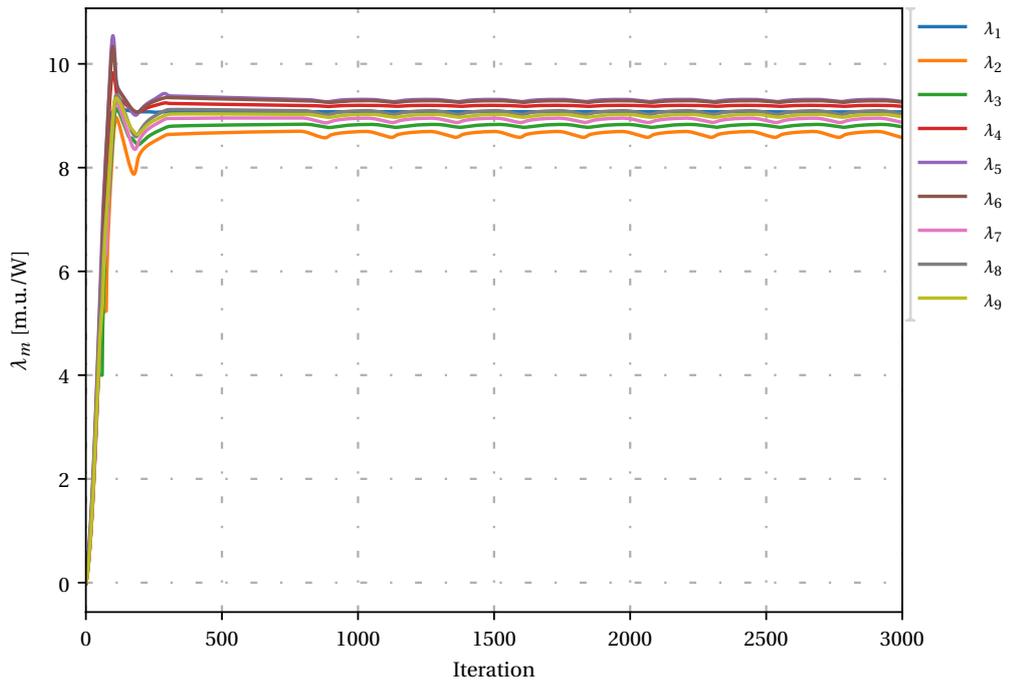


Figure 3.33: LMP at all the nodes in the case presented in 3.32, previously to the change of the tuning parameters.

of dampening throughout the iterations. A more exact explanation of this oscillatory process will be given in Chapter 4.

In order to prevent this oscillation process, $\beta \bar{U}$ was reduced to 0.5 m.u./V², from 10 m.u./V², which meant the rise in $\mu_1 \bar{U}$ would be much slower and oscillations would not happened. After this change, the results in Figures 3.34-3.36 show that the system converges in around 2000 iterations which means it converged 6 times

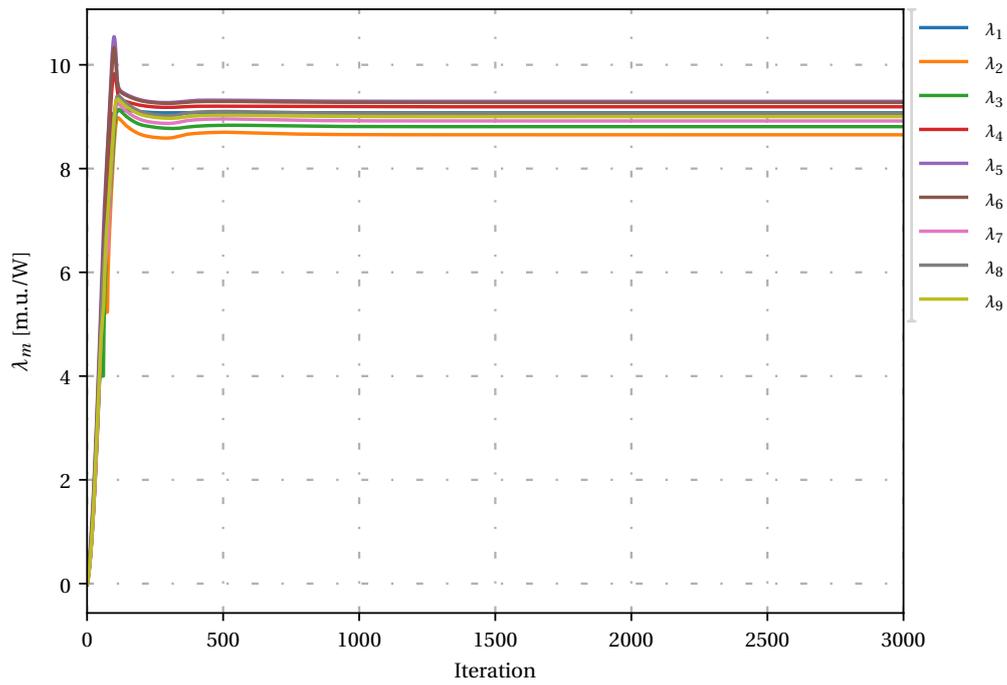


Figure 3.34: LMP at all the nodes in the case presented in 3.32.

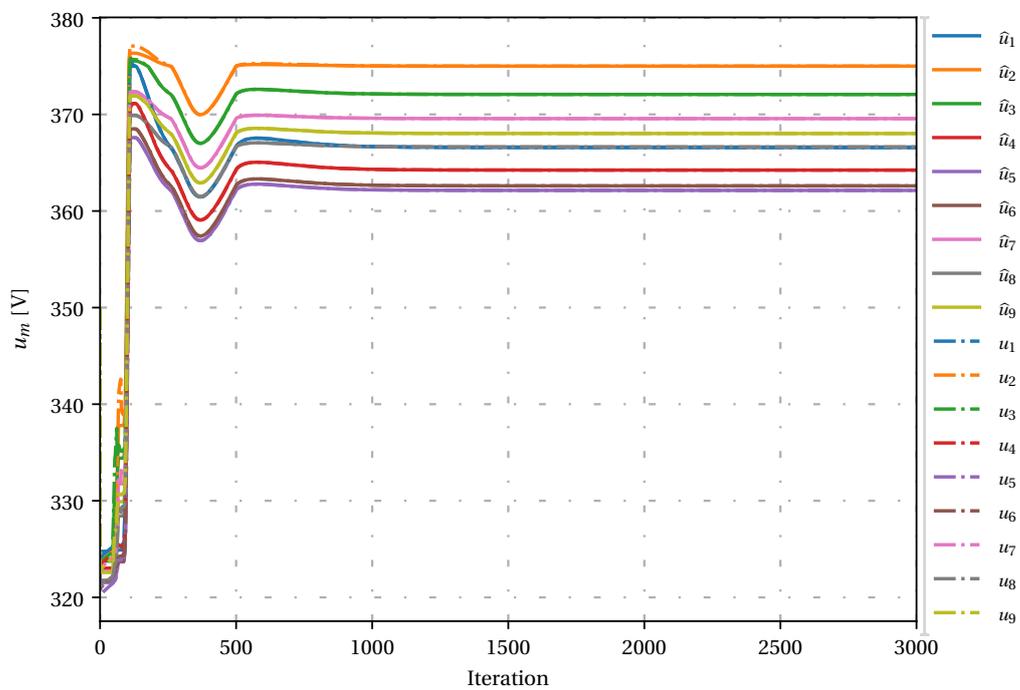


Figure 3.35: Local voltage levels at all the nodes in the case presented in 3.32.

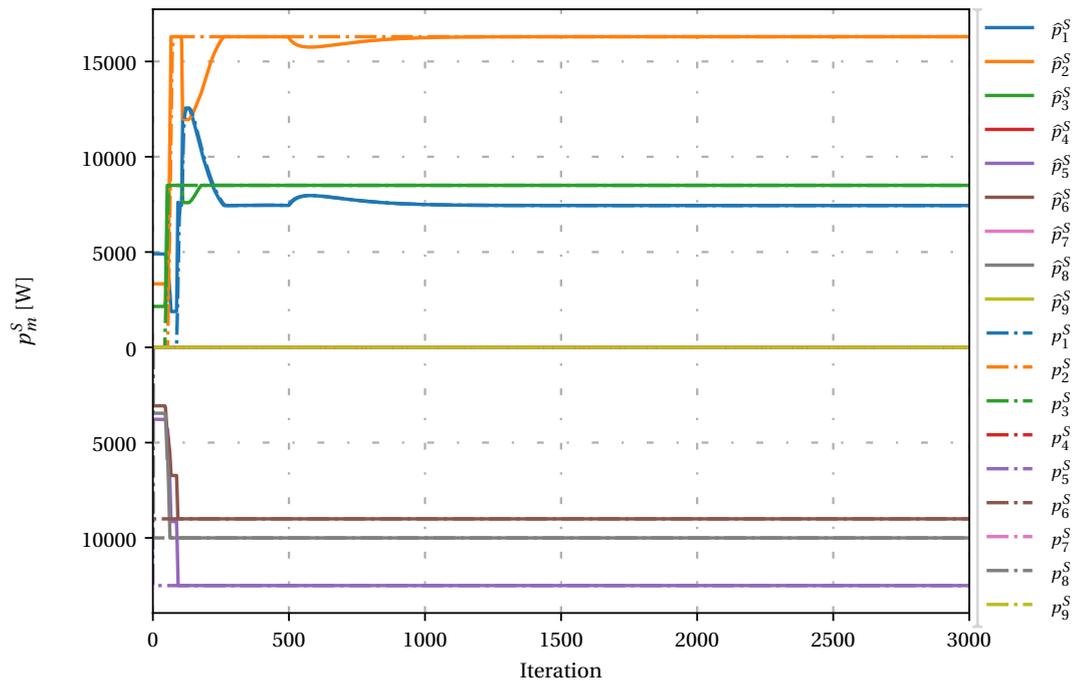


Figure 3.36: Local nodal power at all the nodes in the case presented in 3.32.

as fast as the solution presented in [27].

One thing to take into account is that there are still voltage and power oscillations in the system which are caused by the steep increase in voltage of node 4, as seen in Figure 3.35. This means, that for this case, the β^U was also to high. Tuning both of this parameters, β^U becomes vital to both assure the convergence of the algorithm without many oscillations, but, if set to low, they can mean that the network may take to long to reach an optimal point. How this issue can be tackled is discussed in Chapter 4 as well.

In regards to the 1st research question, the results and discussion presented so far lead to the conclusion that using grid measurements will have a positive effect on the optimisation process, reducing the amount of iterations it takes to converge.

3.6. Asynchronous Behaviour

As mentioned in [27], asynchronous algorithms have different definitions. For example, in [35], the network is divided into areas and those areas preform asynchronous communication, meaning they only send information after a few iterations have passed, while inside the area, it is required the buses exchange information after every iteration, meaning that inside the area, communication is synchronous.

Following the lines of [27], the suggested implementation for an asynchronous algorithm is the following:

$$[\lambda_n(l), \mu_{n,m}(l)] = \begin{cases} [\lambda_n(l), \mu_{n,m}(l)] & \text{if received} \\ [\lambda_n(l-1), \mu_{n,m}(l-1)] & \text{if not received} \end{cases} \quad (3.87)$$

In addition, as previously mentioned in section 3.4, there is also communication between the grid and the node. This means one of two things: either every node has to wait for the same grid measurements after every iteration or they treat the main computer which is running the grid simulation like they would any other node and asynchronous communication would mean if some measurements where not received in time, the last ones would be used.

The first option has a very strong implication which is: all the nodes will wait for central grid simulation to receive the data, process it and resend it, and this can take some time. This means, that if even some nodes

are faster than others, they'll have to wait for each other to send the information to the grid and, as such, its like running a synchronised simulation once more.

The second option means that we are considering that the equipment might have some problems. For example, if an error occurs in the converter and the latest measurements cannot be used, the old ones would then have to suffice, although they might not be the more accurate. As such, the power, current and voltage measurements can also be given as:

$$[p_m(l), u_m(l), i_{n,m}(l)] = \begin{cases} [p_m(l), u_m(l), i_{n,m}(l)] & \text{if received} \\ [p_m(l-1), u_m(l-1), i_{n,m}(l-1)] & \text{if not received} \end{cases} \quad (3.88)$$

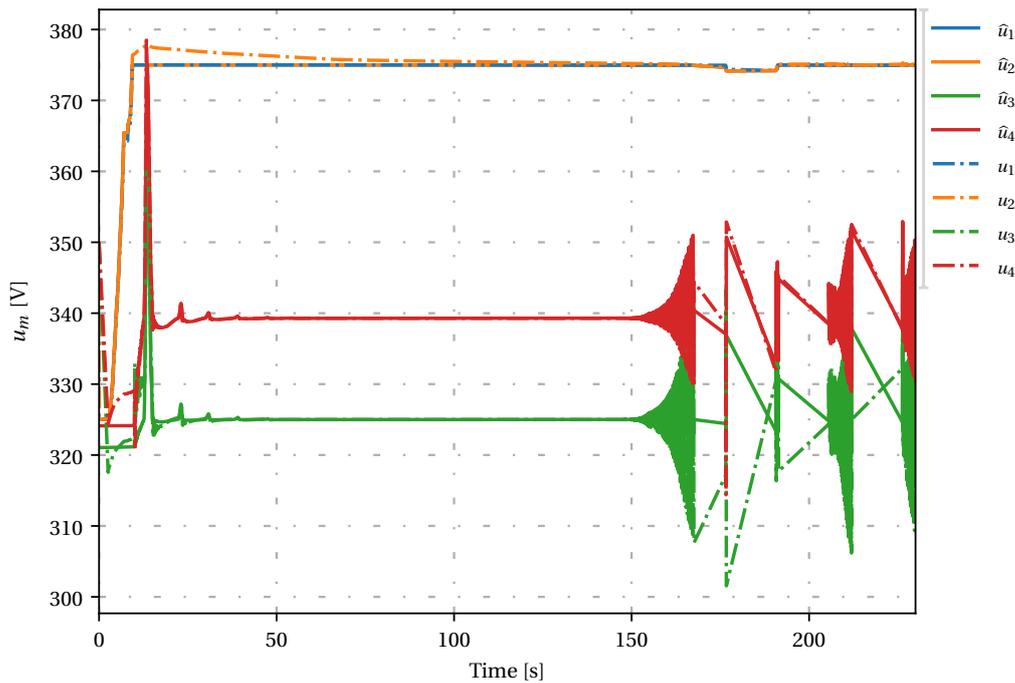


Figure 3.37: Voltage measurements and setpoints for all the nodes, for the grid shown in 3.26, with a 500 ms timeout period.

The second option makes more sense considering the real world implementation, where there would not be a central coordinator and no synchrony between nodes. However, when tested in this simulation environment this implementation did not work properly, as can be seen in Figures 3.37 and 3.38, which are the results of the simulation of the grid in Figure 3.26, using 500 ms and 300 ms of wait time.

Table 3.2: Number of failed communication attempts in the simulation of the grid in 3.26 with a 500 ms timeout, in asynchronous mode, as seen in Figure 3.37. In the columns in the left is identified the origin of he information and the top row is identified what was the target of that information which failed to arrive.

From	To	node 1	node 2	node 3	node 4
node 1		0	---	---	---
node 2		9	2	---	---
node 3		---	1	1	---
node 4		---	---	1	---
Grid		109	102	101	104

In the results of the first simulation, Figure 3.37, it is possible to see that for the first 150 s of simulation the system is converging to the same optimal point as the one in Figure 3.28, but after then the system starts

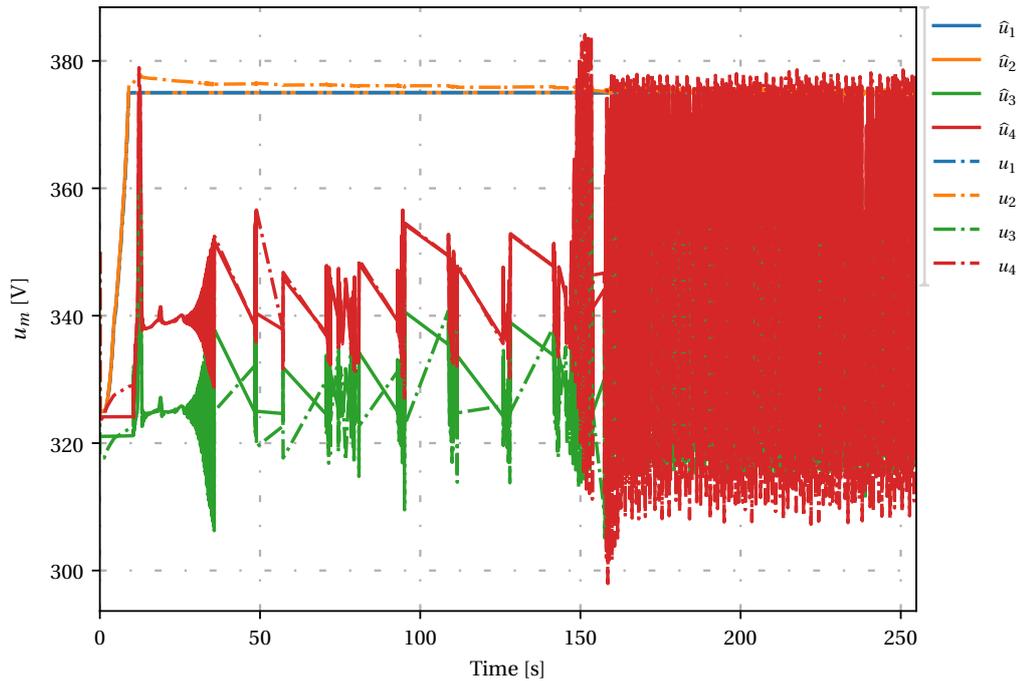


Figure 3.38: Voltage measurements and setpoints for all the nodes, for the grid shown in 3.26, with a 300 ms timeout period.

Table 3.3: Number of failed communication attempts in the simulation of the grid in 3.26 with a 300 ms timeout, in asynchronous mode, as seen in Figure 3.38. In the columns in the left is identified the origin of the information and the top row is identified what was the target of that information which failed to arrive.

From	To	node 1	node 2	node 3	node 4
node 1		0	---	---	---
node 2		31	5	---	---
node 3		---	0	6	---
node 4		---	---	0	---
Grid		331	301	306	310

to oscillate out of control and does not converge anymore. In addition, in Figure 3.38, the results are even worse, since the system is oscillating during the entire simulation time, especially after 150 seconds, when it starts wildly oscillating, with an amplitude of around 60 V. As such, it is clear that the system is not working as intended.

In Tables 3.3 and 3.3 it is possible to see in detail, the amount of information that was not sent on time from the grid, to the nodes. As explained previously in point 2 of Section 3.4, the grid simulation can take up to one second, which means that, in the case of running it with a 300 millisecond timeout, the node would go 3 iterations without updating voltage, power and current, which have vital information for the updates. As such, once the new grid measurements were received, that information would be outdated, which would mean that the new droop setpoints would be very different from what they previously, which would translate to more simulation time since the initial guess, taken as the last result, would not be accurate.

Furthermore, looking at the loss of information between nodes, which in the real world would be much more significant, it is possible to see that is one degree of magnitude lower than the grid, hence the conclusion that this implementation is not the most correct.

As such, in order to simulate the asynchronous optimisation, the algorithm is ran synchronously and, for each connection, in each iteration, there is a defined chance that the information is dropped, simulating

Table 3.4: Number of failed communication attempts in the simulation of the grid in 3.26 with a 25% chance of failure, in synchronous mode, as seen in Figure 3.39. In the columns in the left is identified the origin of the information and the top row is identified what was the target of that information which failed to arrive.

From	To	node 1	node 2	node 3	node 4
node 1		-----	792	---	---
node 2		771	-----	720	---
node 3		---	782	-----	730
node 4		---	---	728	-----
Grid		0	0	0	0

a network communication failure. This way, the node is assured to get all the information necessary and it is easier to control the rate of information lost and analyse how does the algorithm react to that. The results of this new approach can be seen in Figures 3.39 and 3.40, which show the u_m for a loss rate of 25% and 70% respectively, during a 3000 iteration simulation. In Tables 3.3 and 3.3, the exact number of failed communication attempts can be seen.

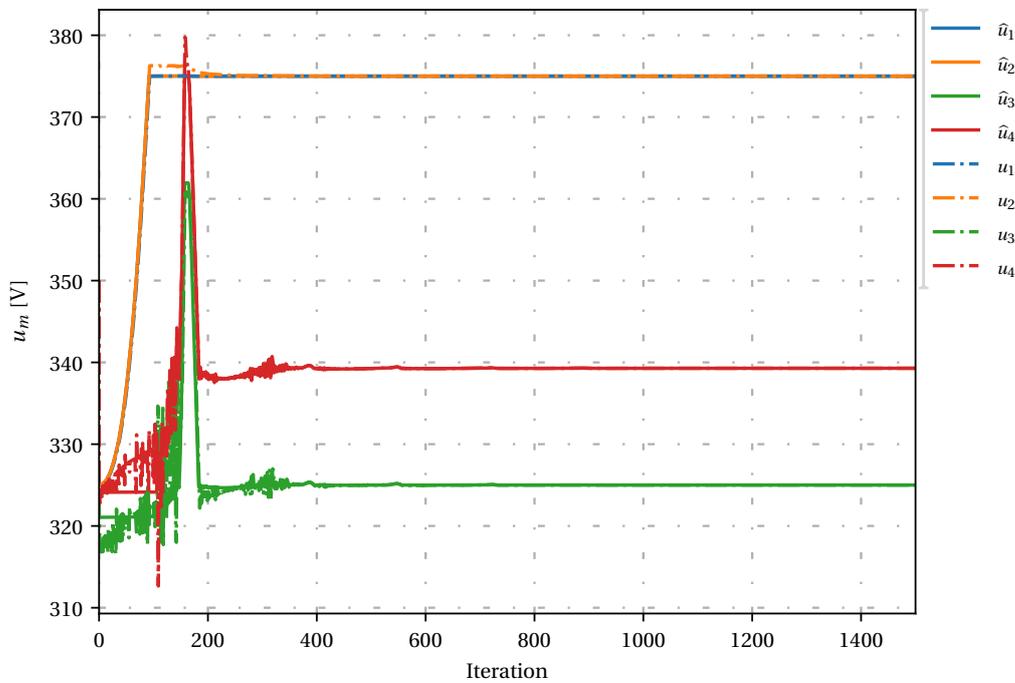


Figure 3.39: Voltage measurements and setpoints for all the nodes, for the grid shown in 3.26, with a 25% failure rate in the communication between nodes.

It is possible to see that, with this simulation strategy the results are very different. Although there are some oscillations in the beginning of the simulation, the whole system ends up converging to the same optimal point as when there were no communication losses.

When the communication losses are at 25%, it is possible to see that the whole system converges in approximately the same amount of iterations as in 3.28, but for a 70% information loss rate, it takes more time than previously. That being said, a possible solution for reducing information loss in a system would then be increasing the time out period, which would work until a certain point, but that would come at the cost of the time to reach convergence and, as such, it would be a matter, of finding a balance between both parameters.

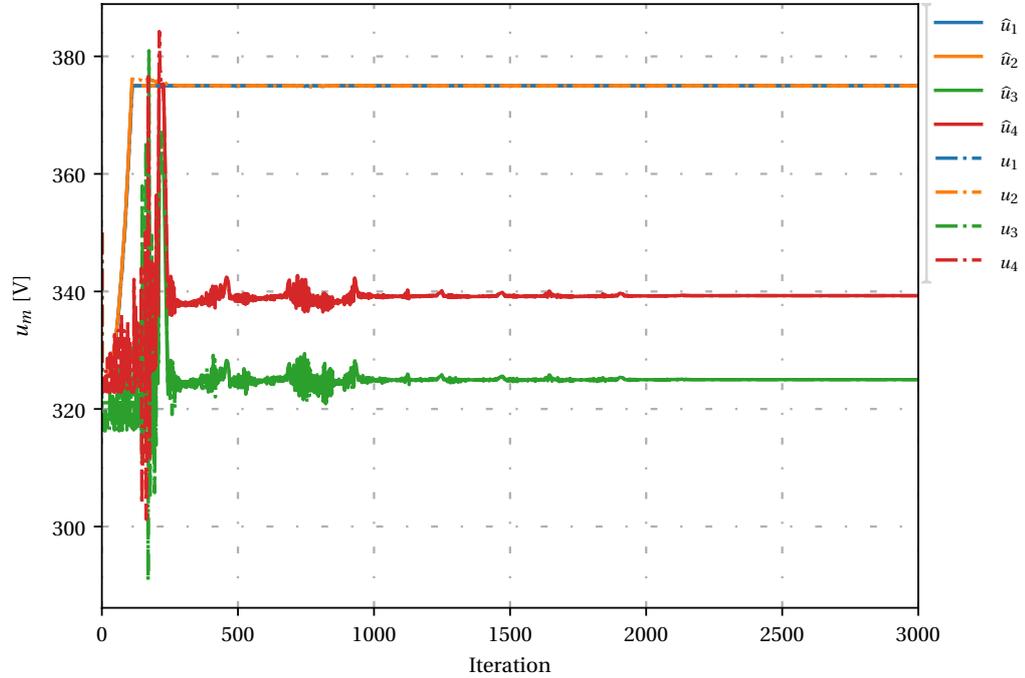


Figure 3.40: Voltage measurements and setpoints for all the nodes, for the grid shown in 3.26, with a 70% failure rate in the communication between nodes

Table 3.5: Number of failed communication attempts in the simulation of the grid in 3.26 with a 70% chance of failure, in asynchronous mode, as seen in Figure 3.40 . In the columns in the left is identified the origin of the information and the top row is identified what was the target of that information which failed to arrive.

From	To	node 1	node 2	node 3	node 4
node 1		_____	2148	---	---
node 2		2069	_____	2086	---
node 3		---	2071	_____	2058
node 4		---	---	2079	_____
Grid		0	0	0	0

4

Adaptive Behaviour

As discussed in the previous chapters and in [16, 27], the convergence of the algorithm is highly dependant on the tuning parameters. If badly tuned, the whole system might not converge and oscillate wildly. This becomes a even bigger problem if online optimisation is to be implemented, since the whole process would have to adapt itself to the changing system and could become badly tuned if the networked changed.

In this chapter, a new adaptive behaviour will be discussed, focusing on the online adaptation of tuning parameters in order to increase convergence speed and flexibility of the algorithm.

For simplicity sake, when voltage is discussed in this chapter, it is referring to the voltage setpoints given by the optimisation layer of the system, mentioned the in the previous chapter, and, if need be, \hat{u}_m will be referred as measured voltage.

4.1. Adaptive Optimisation of tuning parameters

One of the most important variables in this optimisation problems are the dual variables of the local voltage limits $\mu_m^{\bar{U}}$, since they are fundamental to either push the voltage up, or down, during the optimisation. In Figure 3.33, it was clear the influence of $\mu_m^{\bar{U}}$ in the convergence of the system, which was solved by lowering the gain of the update in question from 10 m.u./V² to 0.5 m.u./V². However, in order to understand why this worked and the approach which was used to better update $\mu_m^{\bar{U}}$, it is necessary to understand why do this specific dual variables cause oscillations.

Lets take Figure 4.1 as an example. Here, the voltage can be seen rising until it reaches the maximum level. Once the maximum level is reached, $\mu_m^{\bar{U}}$ starts to rise accordingly to the update given by (3.76), until it reaches a maximum point, which is in the same iteration $u_m(l) < \bar{U}_m$. This decrease in voltage happens when $\mu_m^{\bar{U}}$ becomes the most significant value in $\partial \mathcal{L} / \partial u_m$, meaning $\partial \mathcal{L} / \partial u_m > 0$, and so, will push the voltage down. The higher the value of the dual variable when compared to the other terms in (3.42), the higher the value of the differential term and, therefore, the more u_m will decrease each iteration.

However, as the voltage decreases under the maximum limit, so does $\mu_m^{\bar{U}}$. This means that the relevance it has in $\partial \mathcal{L} / \partial u_m$ also diminishes and voltage starts to decrease less, until it reaches a point where the it starts to rise again. At this point $\mu_m^{\bar{U}}$ is no longer the most relevant term and is no longer pushing the voltage down. If $\partial \mathcal{L} / \partial \lambda_m = 0$, then when the voltage stops decreasing means $\partial \mathcal{L} / \partial u_m = 0$, and, as such, $\mu_m^{\bar{U}}$ is close to the optimal value. However, while the voltage rises and is still not at the limit, the dual variable continues to decrease.

What distinguishes an undamped oscillatory state from the damped one is the following:

1. **Undamped Oscillations** happen when $\mu_m^{\bar{U}} = 0$ before $u_m(l) = \bar{U}_m$, highlighted as a purple point in Figure 4.1. When this happens, the voltage will rise until it reaches the maximum value once more, which will then increase the $\mu_m^{\bar{U}}$ again, and restart the whole process. The system will oscillate around the optimal values, but since the gains are to big, it will never be able to converge.
2. **Damped Oscillations** happen when $u_m(l) = \bar{U}_m$ before $\mu_m^{\bar{U}} = 0$, highlighted as a green point in Figure 4.2. In this case, when the voltage reaches the maximum value again, $\mu_m^{\bar{U}} > 0$ which means $\partial \mathcal{L} / \partial u_m$

has a lower value than the first time $u_m(l) = \bar{U}_m$ and, as such, its rise above the limit will also be lower. A lower rise, translates to a smaller increase in $\mu_m^{\bar{U}}$ and so the system will dampen the oscillations until it reaches the optimal point.

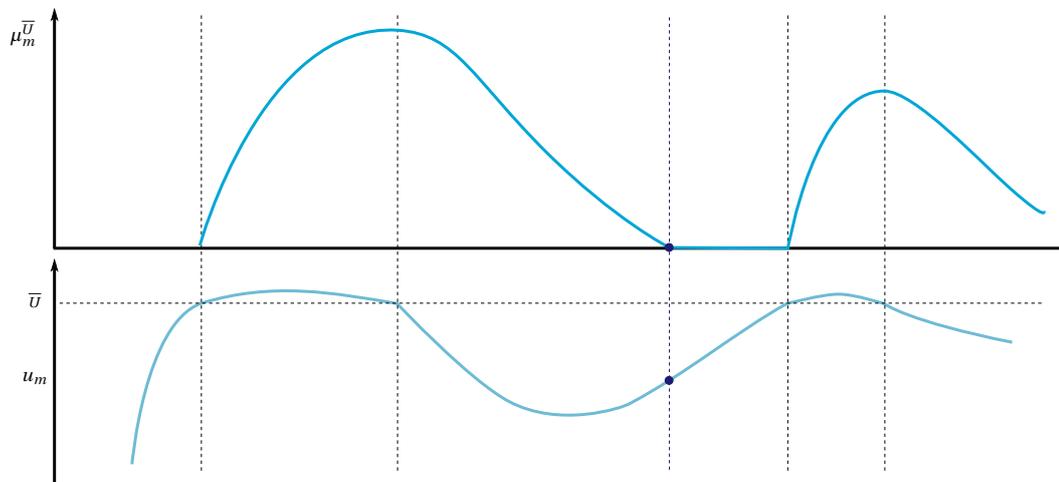


Figure 4.1: Example of the undamped oscillatory process over the iterations, similar to the one in Figure 3.33. Here $\mu_m^{\bar{U}}$ can be seen in the top half and u_m can be seen in the bottom half.

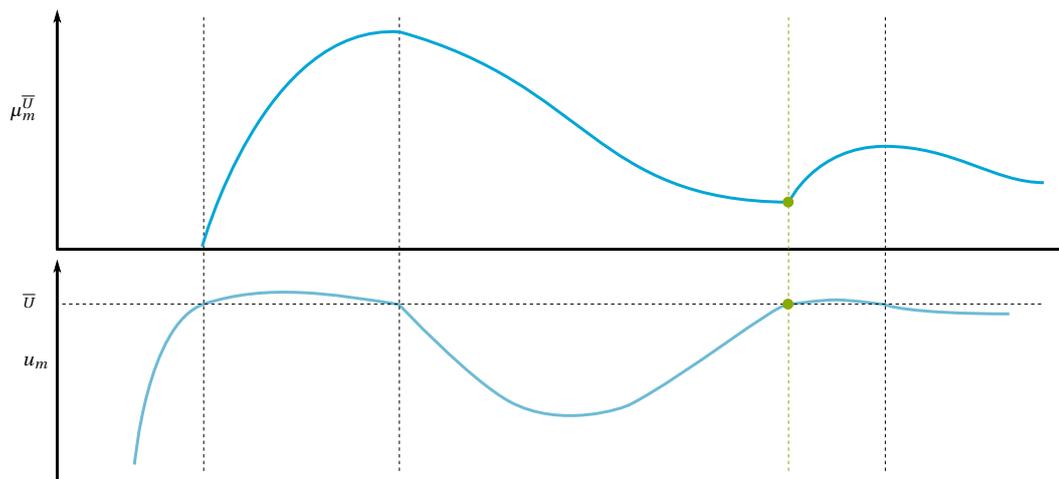


Figure 4.2: Example of the damped oscillatory process over the iterations, similar to the one in Figure 3.34. Here $\mu_m^{\bar{U}}$ can be seen in the top half and u_m can be seen in the bottom half.

Hence, a solution for the oscillation in the system is reached by shifting the $\beta_m^{\bar{U}}$ during the iterative process as to make sure that maximum voltage is reached before the dual variable is null.

The process for $\mu_m^{\underline{U}}$ is the similar and the same reasoning can be applied, but in terms of \underline{U} instead of \bar{U} , and, therefore, it will not be discussed further.

In order to make sure the system does not oscillate, it is better to take a conservative approach for the gains of the system, meaning the gains should start small and, as time advances, adjust them as necessary. This will work towards minimising oscillatory behaviour in the system and make sure it converges to an optimal solution. This is important because, in a real world implementation, the system would run in the background of the users' daily lives and should require minimum input and control from said users. As such, the system should adapt to the current situation and should not oscillate out of control, assuring the safety and integrity of the system.

Following this approach, the initial gain values for the maximum and minimum voltage dual variable

update were reduced to:

$$\beta^{\bar{U}} = \beta^U = 1 \left[\frac{\text{m.u.}}{V^2} \right] \quad (4.1)$$

This approach, however, has the drawback of making the convergence much slower if no extra measures are taken. Because it takes longer the dual variables to rise and lower the system takes more time to converge to the optimal state, which is also not intended.

Therefor, two strategies were implemented in order to adapt the gains during the optimisation process: Gain Adaptation and Voltage Error Integration.

4.1.1. Gain Adaptation

As described before, in order to reduce oscillations, it is necessary to better control the $\mu_m^{\bar{U}}$, and for that, a variable gain was introduced in (3.76) and (3.77), which will be reformulated as such:

$$\mu_m^{\bar{U}}(l+1) = \mathbb{P} \left[\mu_m^{\bar{U}}(l) + \beta^{\bar{U}}(l) \frac{\partial \mathcal{L}}{\partial \mu_m^{\bar{U}}} \right] \quad (4.2)$$

$$\mu_m^U(l+1) = \mathbb{P} \left[\mu_m^U(l) + \beta^U(l) \frac{\partial \mathcal{L}}{\partial \mu_m^U} \right] \quad (4.3)$$

$$\beta^{\bar{U}}(0) = \beta^U(0) = 1 \left[\frac{\text{m.u.}}{V^2} \right] \quad (4.4)$$

where $\beta^{\bar{U}}(0)$ and $\beta^U(0)$ are the initial values of $\beta^{\bar{U}}(l)$ and $\beta^U(l)$ respectively.

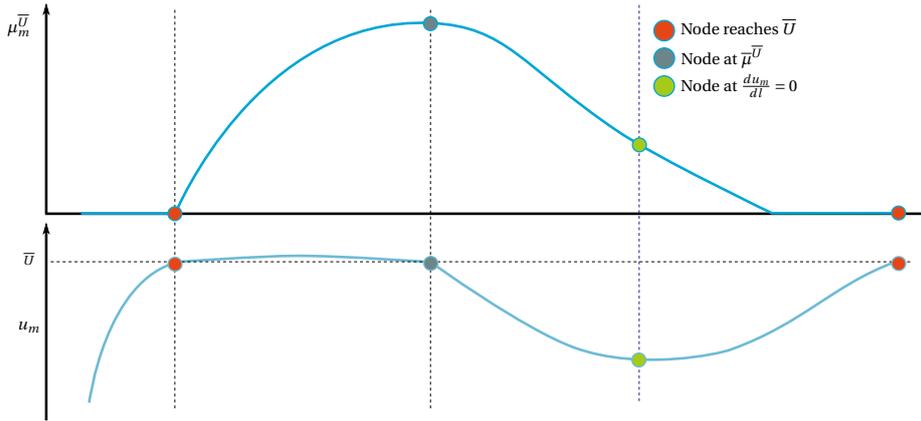


Figure 4.3: Scheme of the 3 zones of operation for the gain adaptation.

Following the previous sections description of the oscillatory behaviour, three zones can be identified which will determine how the current iterations gain is calculated, highlighted in Figure 4.3, which are: the zone where the local voltage level is above \bar{U} , between the leftmost red circle and the grey circle; the zone where $u_m < \bar{U}$ and $du_m/dl < 0$, i.e, when the voltage is decreasing in comparison to the previous iteration, which is the figure, between the grey and green circles; the zone where $u_m < \bar{U}$ and $du_m/dl > 0$, in the figure, between the green and the rightmost red circles.

In the first zone $\beta^{\bar{U}}(l) = \beta^{\bar{U}}(0)$. In this case the Voltage Error Integration will be in charge of increasing, or not, the rate of climb of $\mu_m^{\bar{U}}$, and thus the gain is not changed.

Node with $u_m < \bar{U}$ and $du_m/dl < 0$

When u_m is decreasing and $\mu_m^{\bar{U}} > 0$, this means that the value of the dual variable is too high. As such, it is important that it decreases faster in order to make sure that u_m can rise again. Although this decrease should be proportional with the du_m/dl , i.e., if the voltage is decreasing slowly it is important that the gain is not very high, as not to overshoot the optimal value by much. And so, the tuning parameter is given, when these conditions are met, is given by:

$$\beta_{\bar{U}}(l) = \min \left(1 - k \cdot \frac{du_m(l)}{dl}, 10 \right) \left[\frac{\text{m.u.}}{V^2} \right] \quad (4.5)$$

with: $k > 0$

where k is a positive constant.

It was observed that the maximum $du_m(l)dl$, in these conditions, was in the order of the $10^{-2}V$ and so, k was chosen as 10^3 m.u./V^3 . Using this strategy to calculate the tuning parameter, it is possible to make it proportional to the rate at which the voltage is decreasing, in order to, on one hand, reach the optimal point as fast as possible and, on the other hand, once the voltage starts to level out make sure the gain is small. It is important to note that the minus sign before the differential term comes from the fact that the voltage is dropping and, therefore, the derivatives is negative. As seen by (4.5), the minimum allowed value for the gain while the voltage is decreasing is the initial tuning parameter value, 1 m.u./V^2 , since the differential term will always be negative. Applying $k = 10^3 \text{ m.u./V}^3$ in (4.5) yields:

$$\beta_{\bar{U}}(l) = \max \left(1 - \frac{du_m(l)}{dl} \cdot 10^3, 10 \right) \left[\frac{\text{m.u.}}{V^2} \right] \quad (4.6)$$

(4.7)

Node with $u_m < \bar{U}$ and $du_m/dl > 0$

Once the voltage stops decreasing, the opposite operation is necessary, meaning that the gain must be inversely proportional to the rate of climb, in order to attenuate the decrease in $\mu^{\bar{U}}$ over the iterations and to make sure its value is as close as possible to the optimal value once $u_m = \bar{U}$. And so, when these conditions are met, the tuning parameter is given by:

$$\beta_{\bar{U}}(l) = \max \left(\frac{1}{1 + k \cdot \frac{du_m(l)}{dl}}, 0.1 \right) \left[\frac{\text{m.u.}}{V^2} \right] \quad (4.8)$$

with: $k > 0$

where k is a positive constant.

Using (4.8) to calculate $\beta_{\bar{U}}(l)$ makes it so that the bigger the the voltage rise the smaller will be the gain on the $\mu_{\bar{U}}$, since the minimum value the differential term is allowed to be is 0, which would mean a $\beta_{\bar{U}}(l) = 1 \text{ m.u./V}^2$, and otherwise, as $du_m(l)/dl$ rises, the gain becomes smaller. Using the same rationale of the previous subsection, $k = 10^3 \text{ m.u./V}^3$ was the value chosen for the constant, and so, (4.8) yields:

$$\beta_{\bar{U}}(l) = \max \left(\frac{1}{1 + \frac{du_m(l)}{dl} \cdot 10^3}, 0.1 \right) \left[\frac{\text{m.u.}}{V^2} \right] \quad (4.9)$$

$k > 0$

Equations for $\beta_{\underline{U}}(l)$

The same reasoning can be applied to the tuning parameter calculation of $\beta_{\underline{U}}(l)$. When $u_m < \underline{U}$, $\beta_{\underline{U}}(l) = 1 \text{ m.u./V}^2$. When the voltage is higher than the minimum, and rising, $\beta_{\underline{U}}(l)$ must be $\beta_{\underline{U}}(l) > 1$, for the same motives $\beta_{\bar{U}}(l) > 1$, and when the voltage is decreasing once more, the gain must be $\beta_{\underline{U}}(l) < 1$. Using the same constant k as for the maximum voltage limit case, the update is given by:

$$\beta_{\underline{U}}(l) = 1 \left[\frac{\text{m.u.}}{V^2} \right], \quad u_m < \underline{U} \quad (4.10)$$

$$\beta_{\underline{U}}(l) = \min \left(1 + \frac{du_m(l)}{dl} \cdot 10^3, 10 \right) \left[\frac{\text{m.u.}}{V^2} \right], \quad u_m > \underline{U}, du_m/dl > 0 \quad (4.11)$$

$$\beta_{\underline{U}}(l) = \max \left(\frac{1}{1 - \frac{du_m(l)}{dl} \cdot 10^3}, 0.1 \right) \left[\frac{\text{m.u.}}{V^2} \right], \quad u_m > \underline{U}, du_m/dl < 0 \quad (4.12)$$

4.1.2. Voltage Error Integration

While the adaptive tuning parameter works on preventing oscillations caused by the voltage limit dual variable, the Voltage Error Integration is a strategy used to increase those dual variables faster, once the voltage is at the limit. This is necessary because, when the tuning parameters have lower values, the $\mu^{\bar{U}}$ update can be very small, meaning that the system will take a more time to converge to an optimal value.

In order to tackle this problem, a solution was devised where the integrated difference between u_m and \bar{U} would be an additional factor in $\partial\mathcal{L}/\partial\mu^{\bar{U}}$, as to increase the rate at which $\mu^{\bar{U}}$ increases. The new derivative terms can then be written as:

$$\frac{\partial\mathcal{L}}{\partial\mu_m^{\bar{U}}} = (u_m + \epsilon_m) - \bar{U}_m \quad (4.13)$$

$$(4.14)$$

where ϵ_m is the integrated voltage error.

Since this optimisation process has discrete iterations, the error integration was calculated as follows:

$$\epsilon_m(l+1) = \epsilon_m(l) + u_m(l) - \bar{U}_m, \quad \text{if } \frac{\partial\mathcal{L}}{\partial u_m} < 0 \cap \frac{\partial\mathcal{L}}{\partial u_m} > 0.01 \cdot \frac{\partial\mathcal{L}}{\partial u_m} \quad (4.15)$$

$$\epsilon_m(l+1) = 0, \quad \text{if } \frac{\partial\mathcal{L}}{\partial u_m} > 0 \cup \frac{\partial\mathcal{L}}{\partial u_m} < 0.01 \cdot \frac{\partial\mathcal{L}}{\partial u_m} \quad (4.16)$$

where $\frac{\partial\mathcal{L}}{\partial u_m}$ is the minimum value of this differential term has taken.

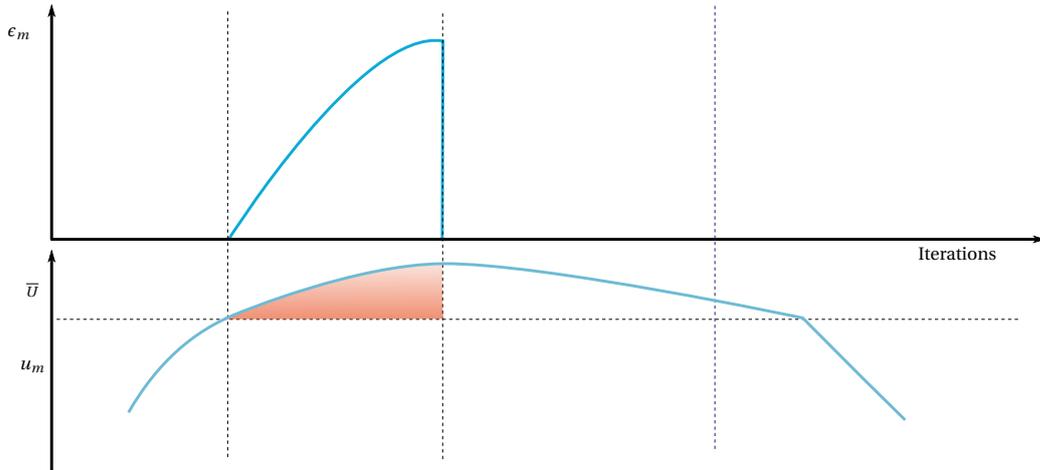


Figure 4.4: Illustration on how the error integration is calculated. The integral of the voltage error, shown as a line on the top plot is also highlighted as the red area on the bottom plot.

The two conditions in this calculation strategy are present to avoid creating oscillations on the system by increasing the $\mu_m^{\bar{U}}$ to much, something that was avoided by reducing the gains. Such balance is then obtained by, firstly, making sure that this added error is only active while the voltage is increasing, i.e., while $\partial\mathcal{L}/\partial u_m$ is negative which, as discussed previously, means $\mu_m^{\bar{U}}$ is not yet higher than the optimal point. To control the rate at which $\mu_m^{\bar{U}}$ is rising, the second condition exists, and it means that the rate at which the $\mu_m^{\bar{U}}$ is changing cannot make it so $\mu_m^{\bar{U}}$ changes more than 1% of the highest registered peak in a near past. In Figure 4.4, it is possible to better visualise the zone of operation of this strategy.

This solution was not applied to the minimum voltage because the system has more difficulty in increasing $\mu_m^{\bar{U}}$ than μ_m^U , since there are usually more elements pushing the voltage down, loads, than trying to increase the voltage, the marginal generator, and, as such, μ_m^U increases faster than $\mu_m^{\bar{U}}$. Implementing the same strategy for lower voltage limit dual variable could introduce oscillations, something which this strategies are working to avoid.

For the same reason, this strategy only works when applied on the marginal node. As discussed in the previous chapter, the marginal node is the one pushing up the voltage and, as such, has more control over this parameter than the nodes at maximum generation capacity. If this was to be applied to every node, the information about the $\mu_m^{\bar{U}}$ would have to be transferred from the maximum voltage node to the marginal node, through the λ_m , and only then the voltage would stop rising. This, however, has some inherent associated delays in the information transmission, and thus creating more oscillations in the system.

In order to test this new strategies, the two grids where the system was most sensitive to the dual variables were tested: the : four node serial case with a long line, presented in Figure 3.26, and the adapted IEEE 9 node case, presented in Figure 3.32. This simulations were ran in synchronous mode in order to have make it easier to understand and to reduce the number of variables that might affect the system.

Firstly, the IEEE 9 bus case will be presented in order to demonstrate the oscillation attenuation and then the 4 node case, where the benefits of using the error integrator are more evident.

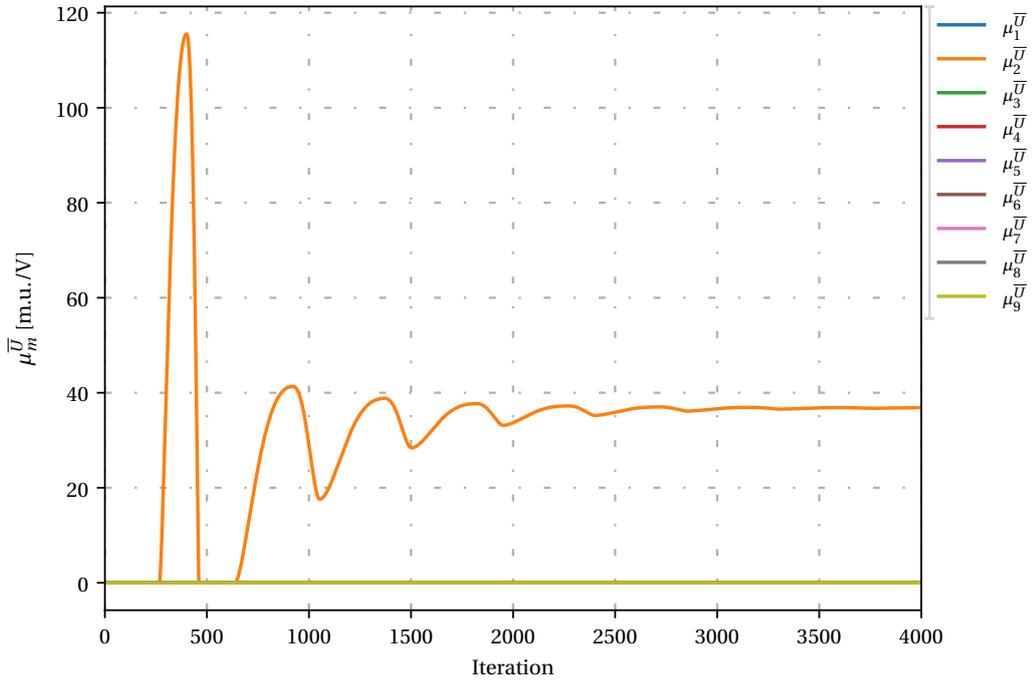


Figure 4.5: Nodal values for $\mu^{\bar{U}}$, over 4000 iterations, without the adaptive behaviour, for the network presented in Figure 3.32.

As can be seen in Figures 4.5 and 4.6, before and after the adaptive behaviour was implemented respectively, when the adaptive gain is working the amplitude of the oscillations is reduced, meaning the system can converge faster. Although the first two peaks have the same amplitude in both simulations, the rise of $\mu_m^{\bar{U}}$ is not affected by the adaptive gain, after the second peak, the oscillations starts to dampen, from a $\Delta\mu_m^{\bar{U}} \approx 22$ m.u./V to a $\Delta\mu_m^{\bar{U}} \approx 17$ m.u./V, around the 1000th iteration mark. This process will continue over the simulation and, in the end, will result in a reduction of 1000 iterations to reach convergence.

In the four node case, Figures 4.7-4.13, the benefits of the adaptive gain on μ^U and the voltage error integration on $\mu^{\bar{U}}$ are more evident. In Figures 4.7 and 4.8, there is a clear difference between the rise of $\mu^{\bar{U}}$, which in the first figure is clearly slower than on the second one. This fast increase can be due to the addition of the integrated error, which can be seen in Figure 4.9 as the orange spikes between the 100th and the 500th iteration. It is worth noting that if the conditions for the integrated error were not included, especially the cap on the rate increase rate, $\partial\mathcal{L}/\partial\mu^{\bar{U}_2}$ would present a much higher amplitude, meaning $\mu^{\bar{U}}$ would rise to fast and cause oscillations.

The effect of the gain adaptation on the μ^U is very evident in Figures 4.10-4.13. When comparing both dual variables, it is visible that the rate at which both increase is the same, however, around the 375th iteration, the

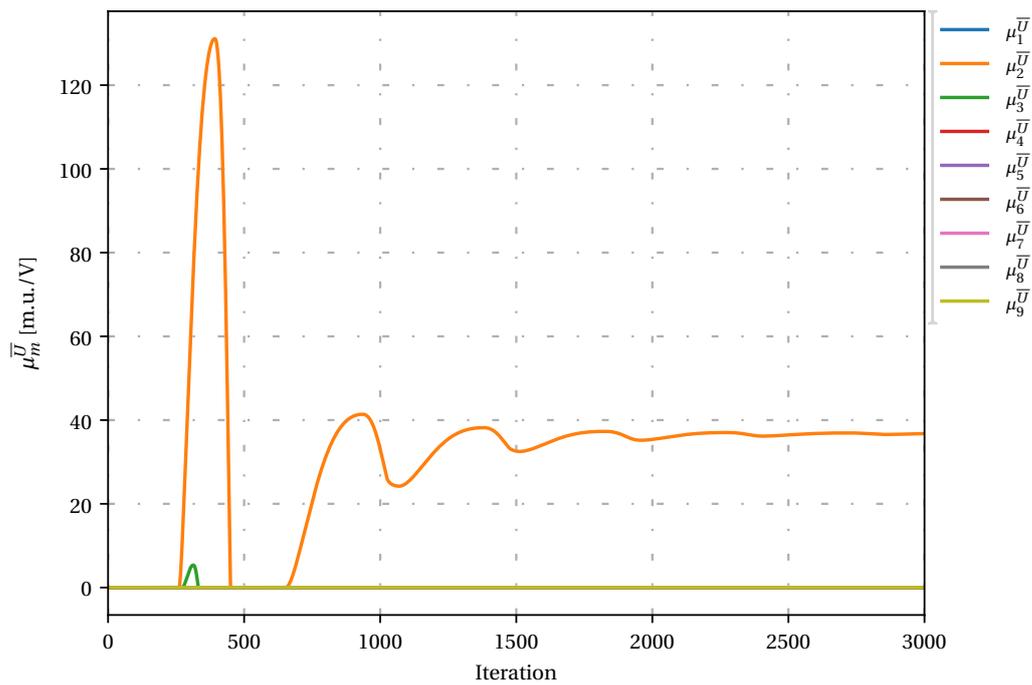


Figure 4.6: Nodal values for $\mu^{\bar{U}}$, over 3000 iterations, with adaptive behaviour, for the network presented in Figure 3.32.

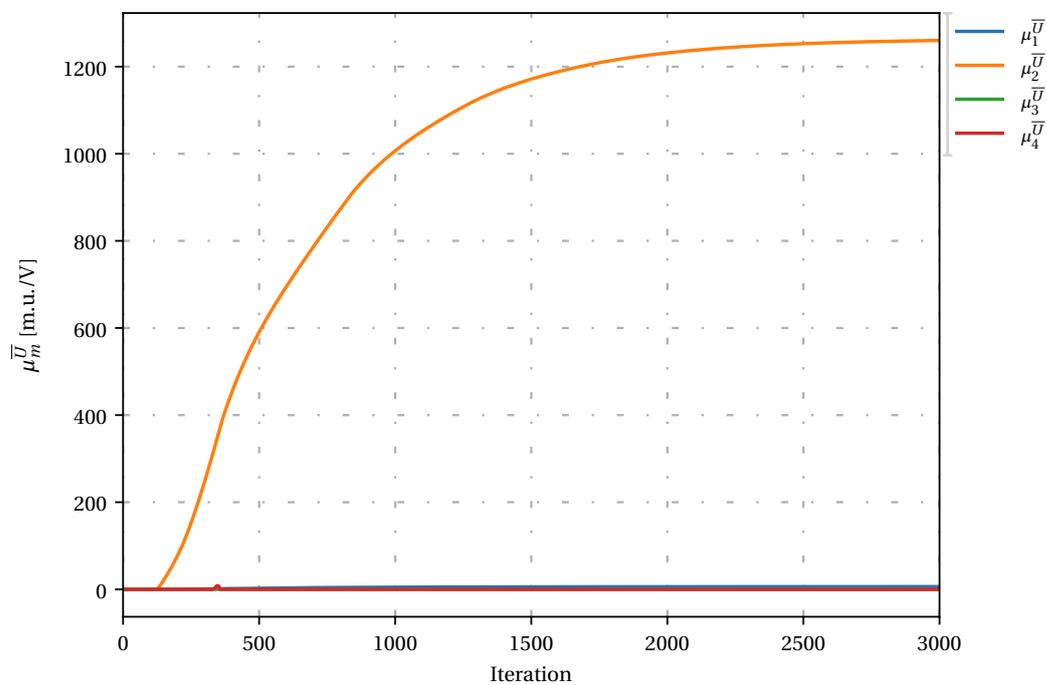


Figure 4.7: Nodal values for $\mu^{\bar{U}}$, over 3000 iterations, without adaptive behaviour, for the network presented in Figure 3.26.

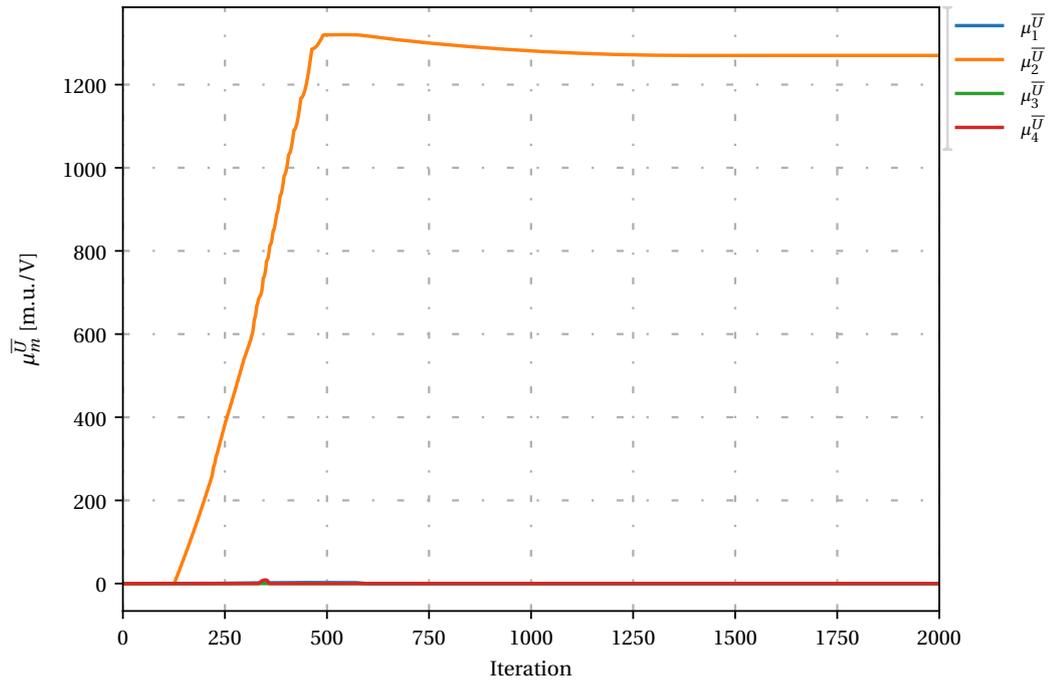


Figure 4.8: Nodal values for $\mu^{\bar{U}}$, over 2000 iterations, with adaptive behaviour, for the network presented in Figure 3.26.

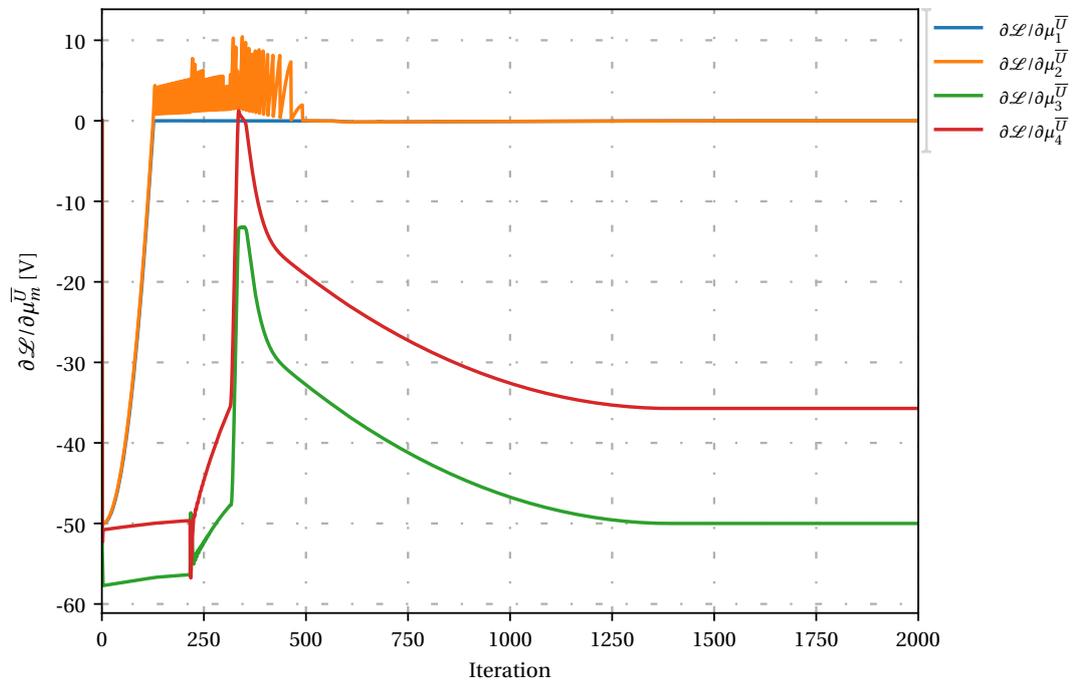


Figure 4.9: Nodal values for $\frac{\partial \mathcal{L}}{\partial \mu^{\bar{U}}}$, over 2000 iterations, with adaptive behaviour, for the network presented in Figure 3.26.

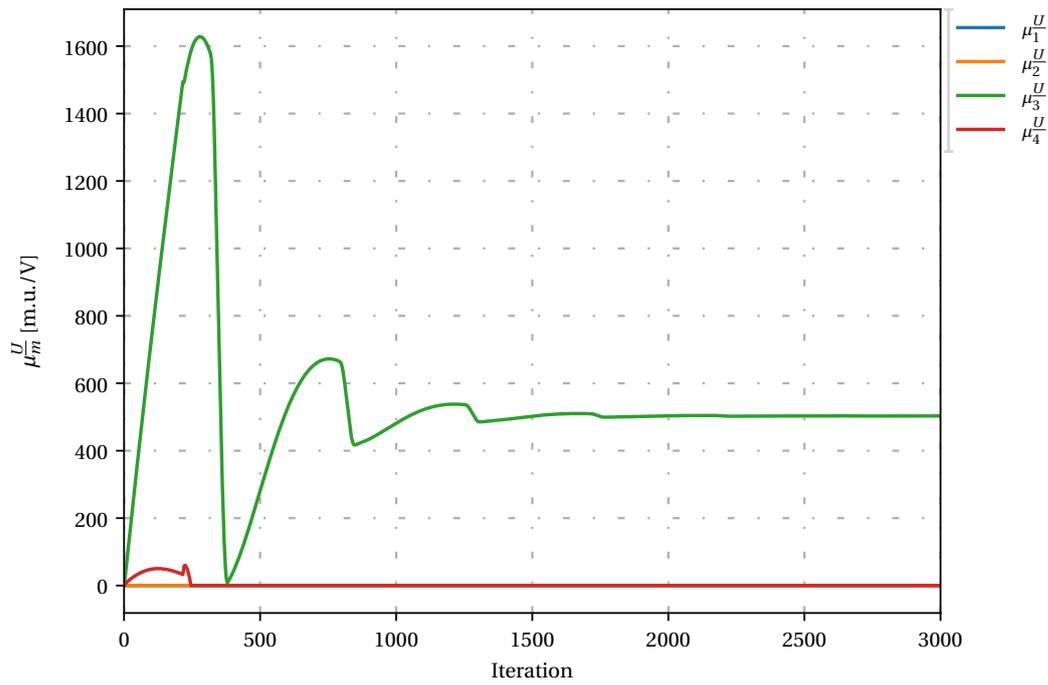


Figure 4.10: Nodal values for μ_m^U , over 3000 iterations, without adaptive behaviour, for the network presented in Figure 3.26.

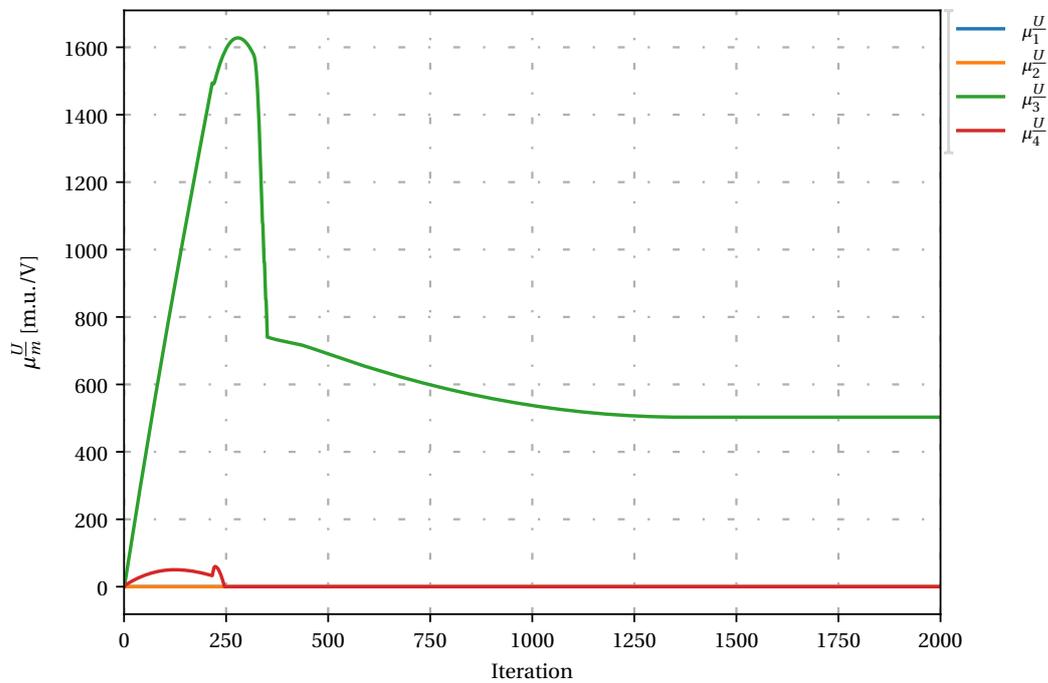


Figure 4.11: Nodal values for μ_m^U , over 2000 iterations, with adaptive behaviour, for the network presented in Figure 3.26.

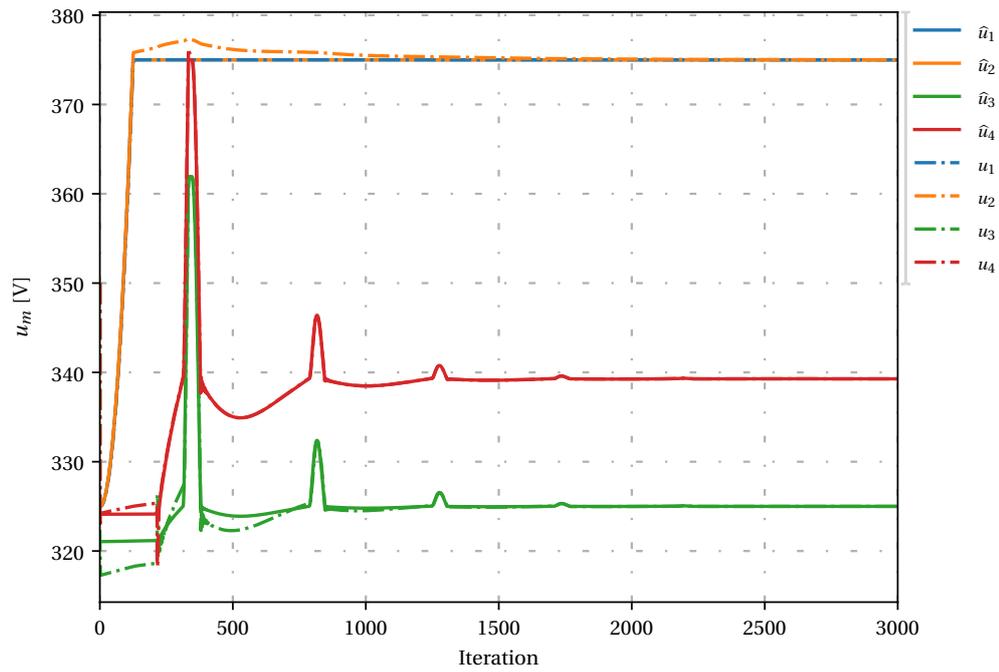


Figure 4.12: Setpoints and measured values of u_m , over 3000 iterations, without adaptive behaviour, for the network presented in Figure 3.26.

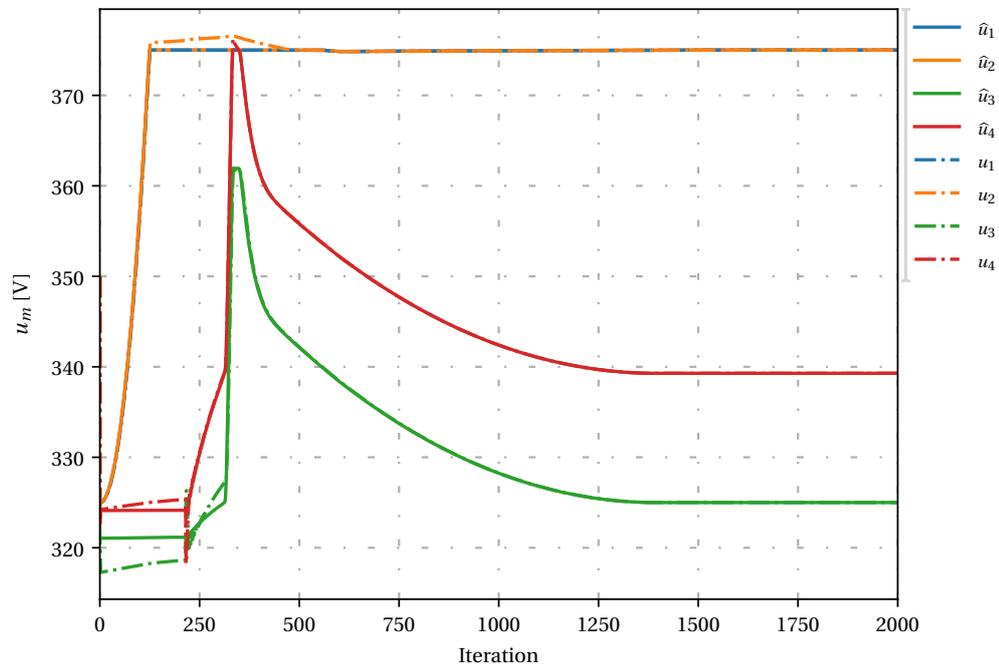


Figure 4.13: Setpoints and measured values of u_m , over 2000 iterations, with adaptive behaviour, for the network presented in Figure 3.26.

β_3^U is reduced in such a way that the steep decrease in μ^{U_3} is severely halted, and it becomes a much smoother decrease until it reaches the convergence point. This modified behaviour of the dual variable translates to a decrease in u_3 and u_4 which is much less steep and avoids the voltage oscillations seen in 4.12.

4.2. Power Supply and Demand Changes

Until this point, it has been assumed that power demand and supply did not change during the extent of the simulation. However, this is not true for a real world implementation, where the system would be subject to power supply and demand changes. As such, the system needs to account for those changes.

There are two different scenarios that may happen during the normal operation of the system, which have different consequences:

1. One of the generators increases its maximum power output.
2. One of the generators decreases its maximum power output or there is a shift in the demanded power.

Scenario 1 is important because it might lead to a shift in prices of the network. If the generator which sees its maximum power increasing was in the maximum power region, i.e. the LMP of that node is higher than the power generation cost of the generator, there is an opportunity for that generator to increase its production. Once this shift occurs, the power setpoint will increase, and the $\partial\mathcal{L}/\partial\lambda$ will become negative and as such, reducing λ_m . This will then LMP of other nodes until node m is the Marginal Generator Zone. This phenomena can be seen in Figures 4.14 to 4.16.

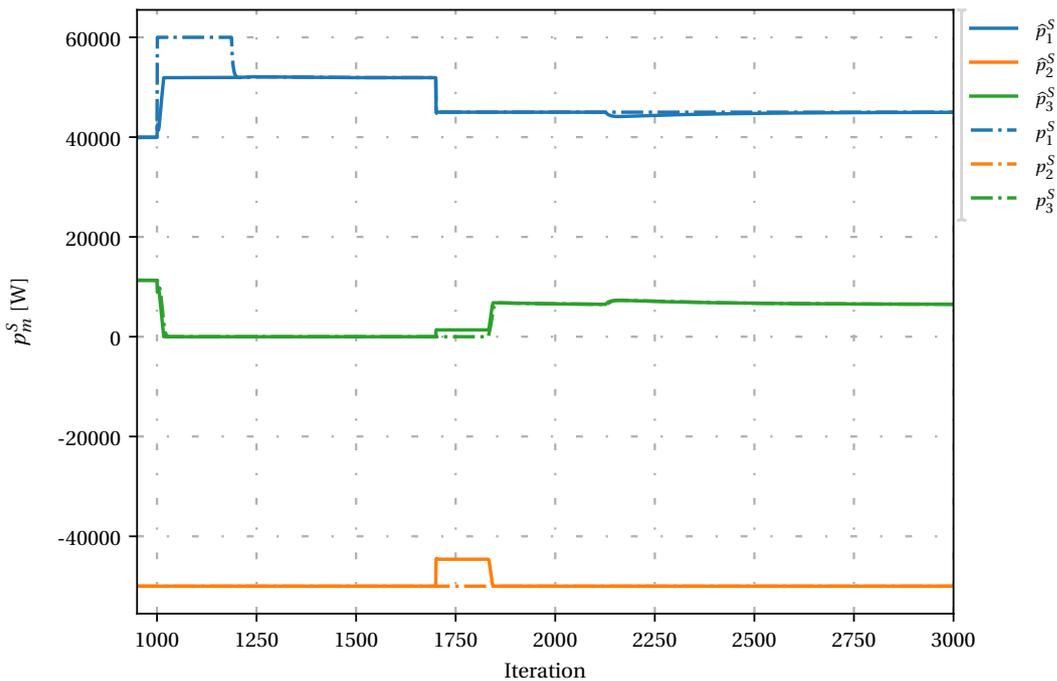


Figure 4.14: Nodal values of local power measurements and setpoints for the network presented in Figure 3.15, starting at iteration 950.

Initially, the system was at the optimal point shown in Figures 3.16 to 3.19. At the 1000th iteration, the maximum power limit of the generator in node 1 is raised from 40 kW to 60 kW, meaning that this node is no longer at the limit, which can be seen as the shift in the power setpoint for node 1 in Figure 4.14. Consequently, the LMP of the other nodes starts to decrease, as seen in Figure 4.15 and, as such, the node 3 will decrease its power production, since it enters the Minimum Power Region. The LMP will continue to decrease until node 1 enters the Marginal Generation Zone, around the 1200th iteration. Here, according to the update strategy shown in 3.2.4, p_1^S is free to change, and it will converge to the optimal point, around 51 kW.

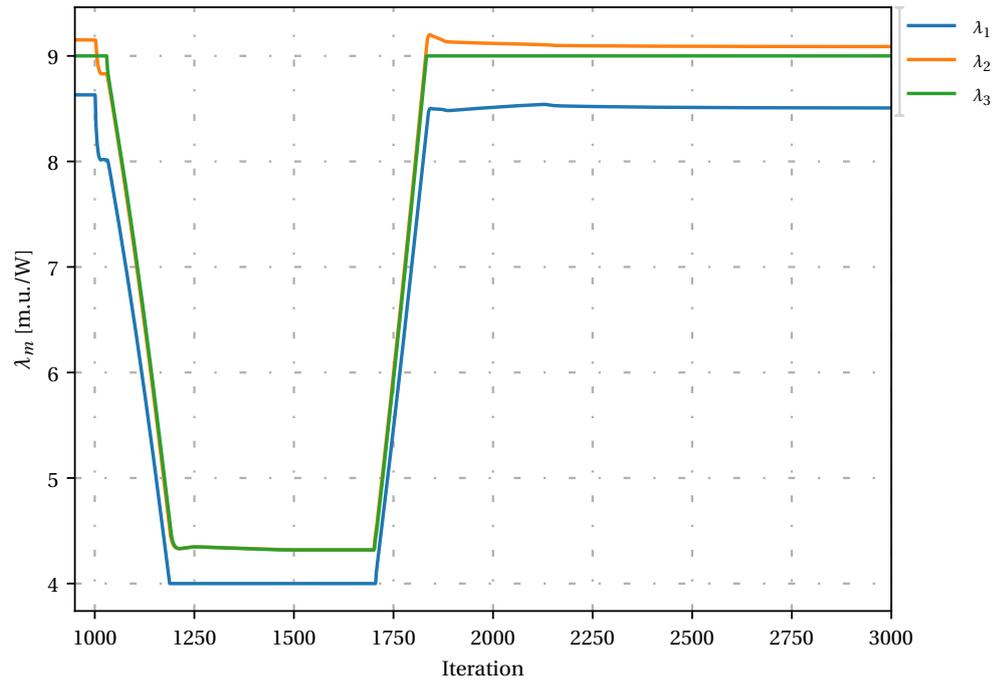


Figure 4.15: Nodal values of LMP for the network presented in Figure 3.15, starting at iteration 950.

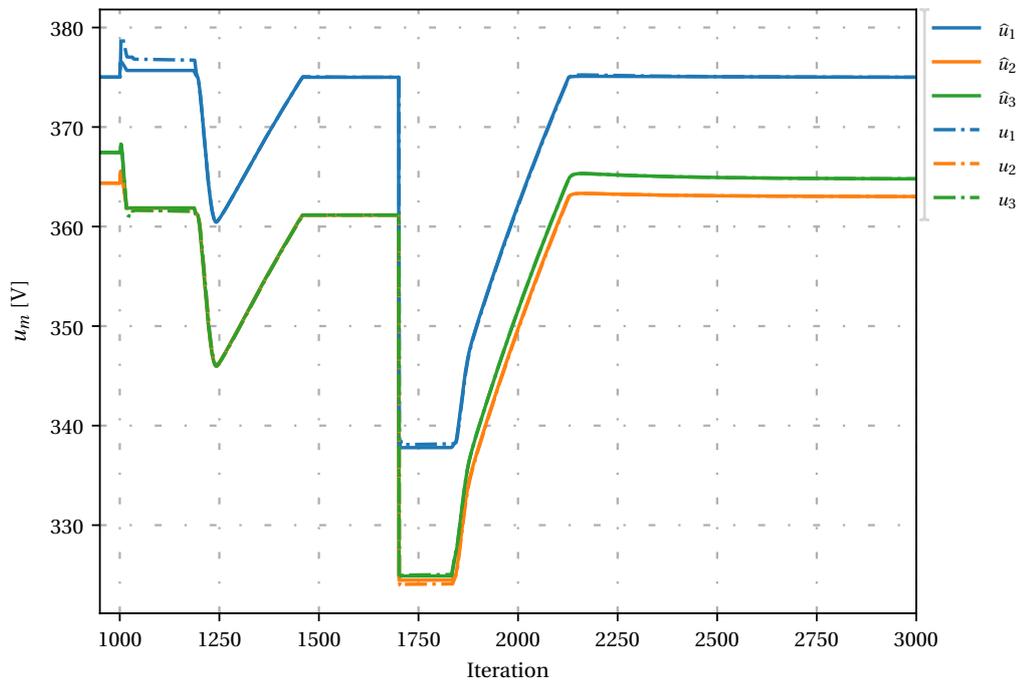


Figure 4.16: Nodal values of the local voltage measurements and setpoints for the network presented in Figure 3.15, starting at iteration 950.

The voltage drop seen in 4.16 around iteration 1200 is a consequence of u_1 being slightly over \bar{U}_1 while the power converges to the optimal point, which increases the $\mu_1^{\bar{U}}$, hence reducing the voltage level of node 1 and, consequently, all other nodes.

In the same simulation, at iteration 1700, \bar{P}_1 is reduced back down to 45 kW, meaning that the generator in node 1 will be unable to fully supply the load in node 2. This means that both the load and the generator in node 3 will react to this power mismatch, the scenario presented in 2.

In this case, the physical system reacts immediately to the supply/demand mismatch according to the defined droop curves. This can be verified by the steep decrease in voltage shown in Figure 4.16. When \bar{P}_1 is decreased, the system doesn't have any marginal generator, which means that the voltage must reduce all the way to the lower limit in order to be able to reduce the power demand of the load, as a consequence of the way the droop curves are created, explained in section 3.3.2. Ideally, the generators and loads should react to the power mismatch with as little voltage droop as possible.

Hence, in theory, the way the droop curves are generated could be changed in a way as to make sure that $p_m^S = \underline{P}_m$ for u_m and $p_m^S = \bar{P}_m$ for $u_m + 5$. However, this hypothesis could not be tested because the grid simulation could not find the mathematical solution for the grid state when the droop curves were modified in this way, making it impossible to get the optimisation results.

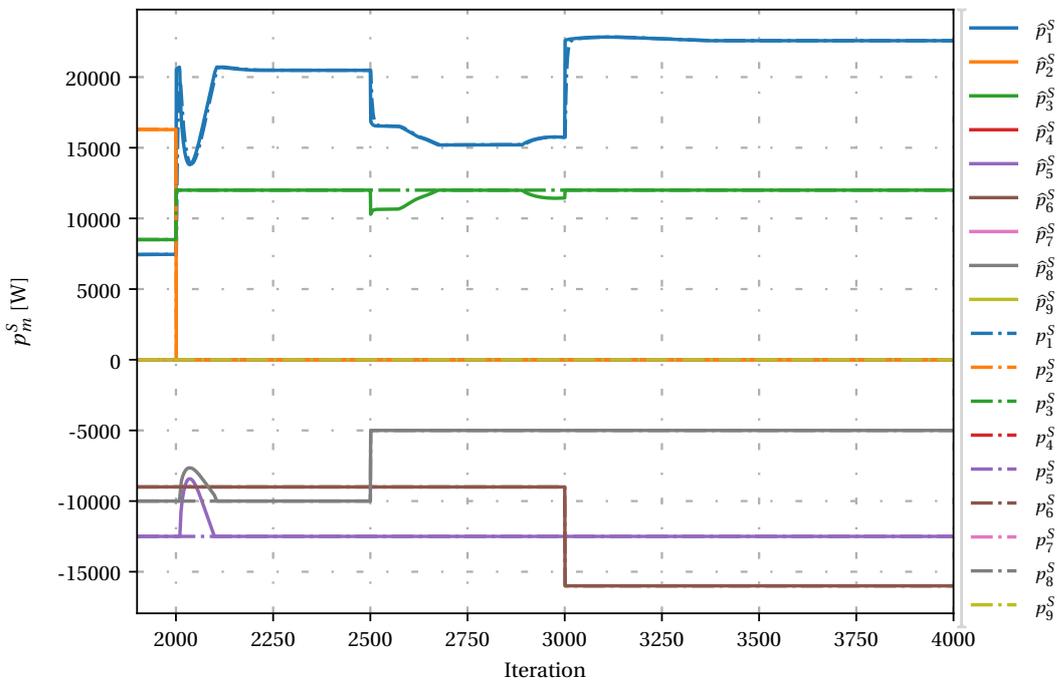


Figure 4.17: Nodal values of the local power measurements and setpoints, for the network presented in Figure 3.32, starting at iteration 1950.

In Figures 4.17 to 4.19 the adapted IEEE 9 bus grid, shown in Figure 3.32, was also subject to power shifts. In this case, at iteration 2000, the generator in node 2 is shutdown and the generators in node 1 and 3 increase their maximum power output, in order to accommodate for the loss of generation power. Then, at the 2500th iteration, the load in node 8 is reduced to 5 kW. Finally, at the 3000th iteration, the power demand at node 6 is increased to 16 kW.

It is possible to observe that the system, in the end, converges to a new optimal state which is very different from the initial one.

Then, similarly to what happened in the previous simulation, the way the droop curves are set means that there is an abrupt fall in voltage when generator 2 is shut down. However, 500 iterations later the system has already converged to a new optimal point, where the generator in node 3 is at maximum power output and the one in node 1 is a marginal generator.

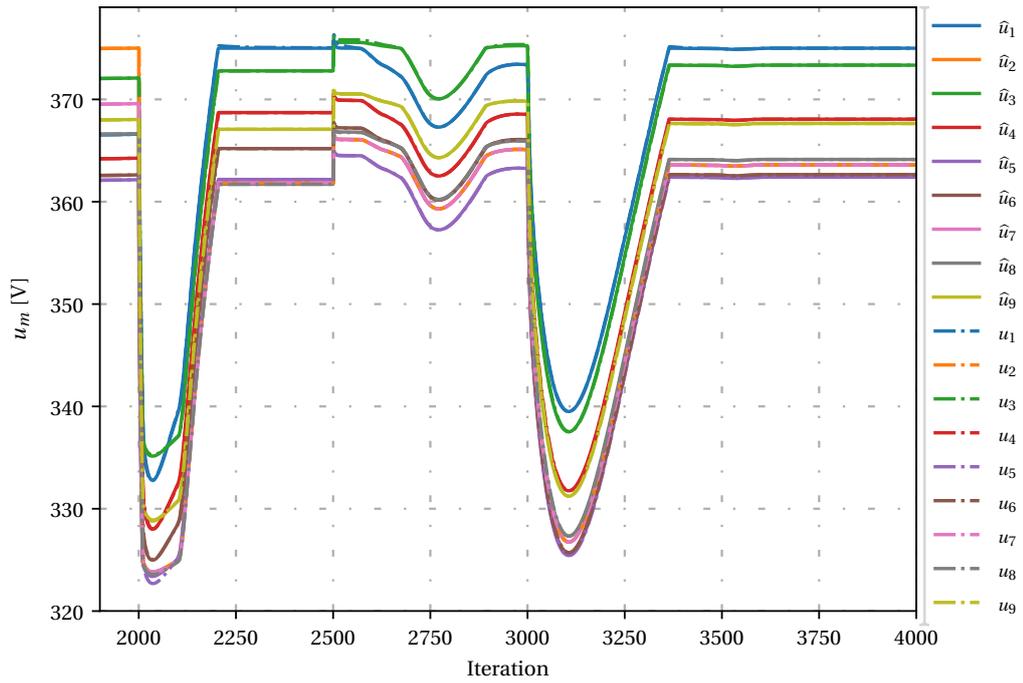


Figure 4.18: Nodal values of the local voltage measurements and setpoints, for the network presented in Figure 3.32, starting at iteration 1950.

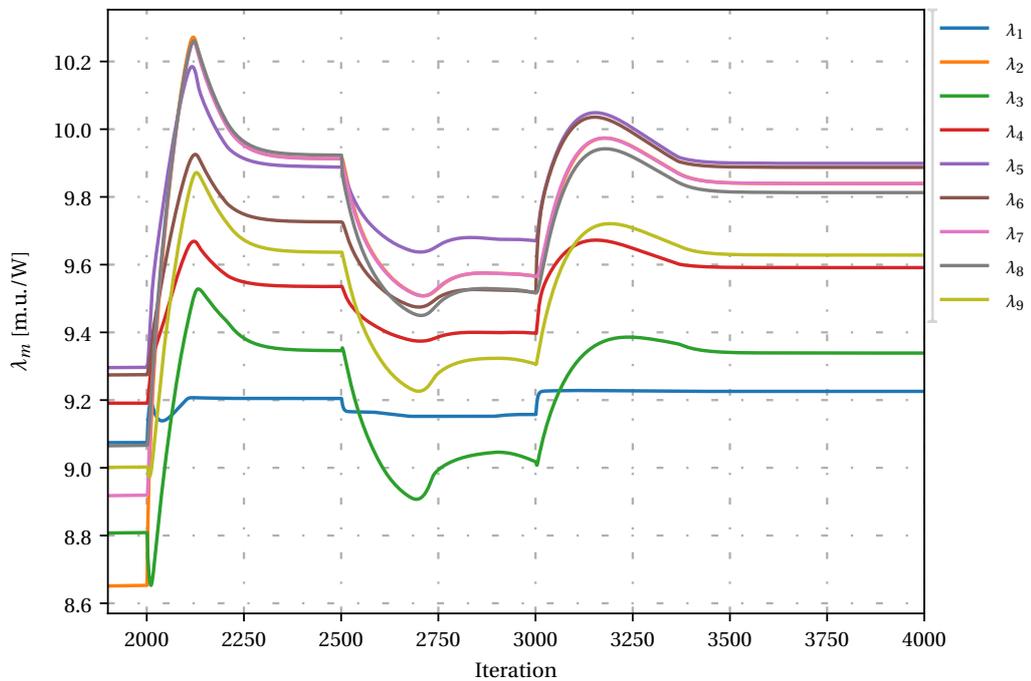


Figure 4.19: Nodal values of LMP, for the network presented in Figure 3.32, starting at iteration 1950.

Once the load decreases, both generators reduce its power while the voltage increases, once more, due to the droop. Afterwards, the value of all LMP starts to lower as a consequence of the power in node 1 decreasing as well, meaning that node 3 will increase its power output to the maximum once more. While the system is still converging, there is an increase in the load. This causes a voltage drop once more which is now smoother than the previous one, as the marginal generator is able to account for the power imbalance in the grid. As a consequence of the update strategy, the values of u_m continue to decrease since, after the increase in power, $\partial \mathcal{L} / \partial u_1 > 0$ which drives the voltage down. In the other nodes, this differential term takes negative values driving the LMP up. This difference in positive and negative terms can be explained by the consensus part of the updates which is dependent on the line currents and voltage, as explained previously.

It is important to note that every simulation, the power shift accounts for a big percentage of total power of the grid. E.g. in the last case, an increase of 7 kW accounts for a $\approx 26\%$ increase in the total energy demand. In a bigger system, this changes would be less significant and, as such, its effects would also result in lesser voltage swings.

4.3. Price Changes

The same way there might be power shifts on the network, the online optimisation must also account for price changes. This changes might occur because a generation unit is not fully independent for external factors and so, the price at which it operates is subject to change. As an example, if we consider a diesel generator, the operation cost is dependent on fuel prices and, as such, is subject to change.

In order to test the algorithm to price changes, the grid in 3.15 was taken as starting point. Once the system was running for 1500 iterations and the optimal point had been reached the cost of for the generator in node 1 was increase from 4 m.u./W to 9.16 m.u./W. Then, at the 2500th iteration, the price for the generator in node 3 shifts from being a linear cost, given $C_3 = 9 \cdot p_3^S$ m.u., and becomes a quadratic cost function given by $C_3 = 0.003 \cdot (p_3^S)^2 + 9 \cdot p_3^S$ m.u. The results for this test can be seen in Figures 4.20 to 4.22.

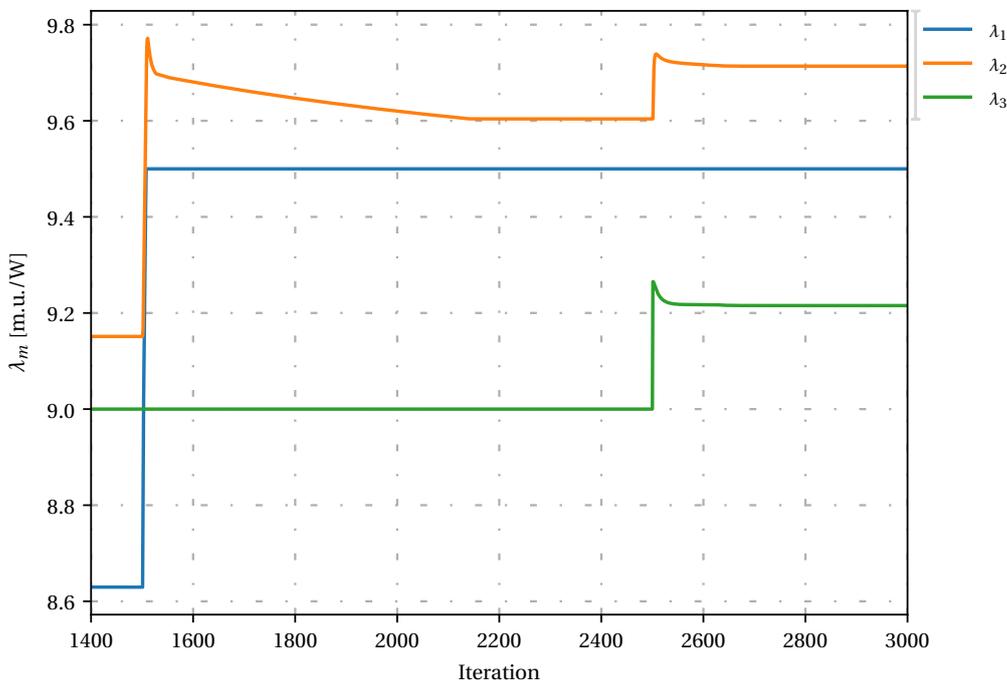


Figure 4.20: Nodal values of LMP for the network presented in Figure 3.15, starting at iteration 1400.

Initially, the node 1 is at maximum power but once the cost increases, the LMP becomes lower than the Minimum Marginal Cost, meaning the generator should shut down. This creates a spike in the power of node 3, and the demand is curtailed in $\approx 20\%$, which causes the voltage to drop significantly. The reason why there

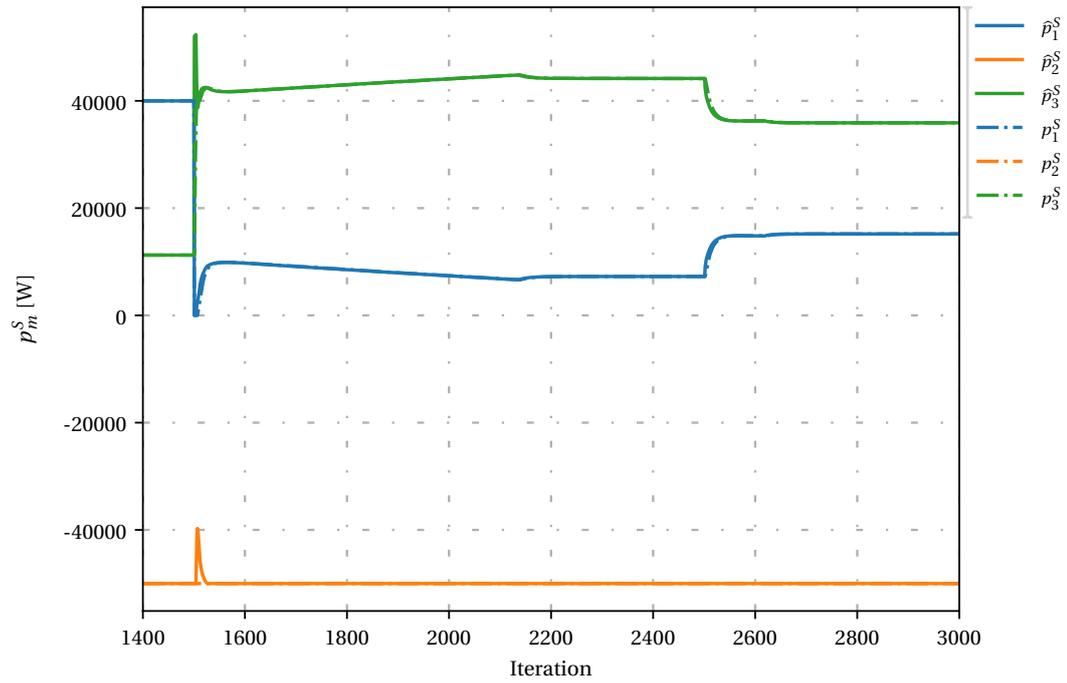


Figure 4.21: Nodal values of local power measurements and setpoints for the network presented in Figure 3.15, starting at iteration 1400.

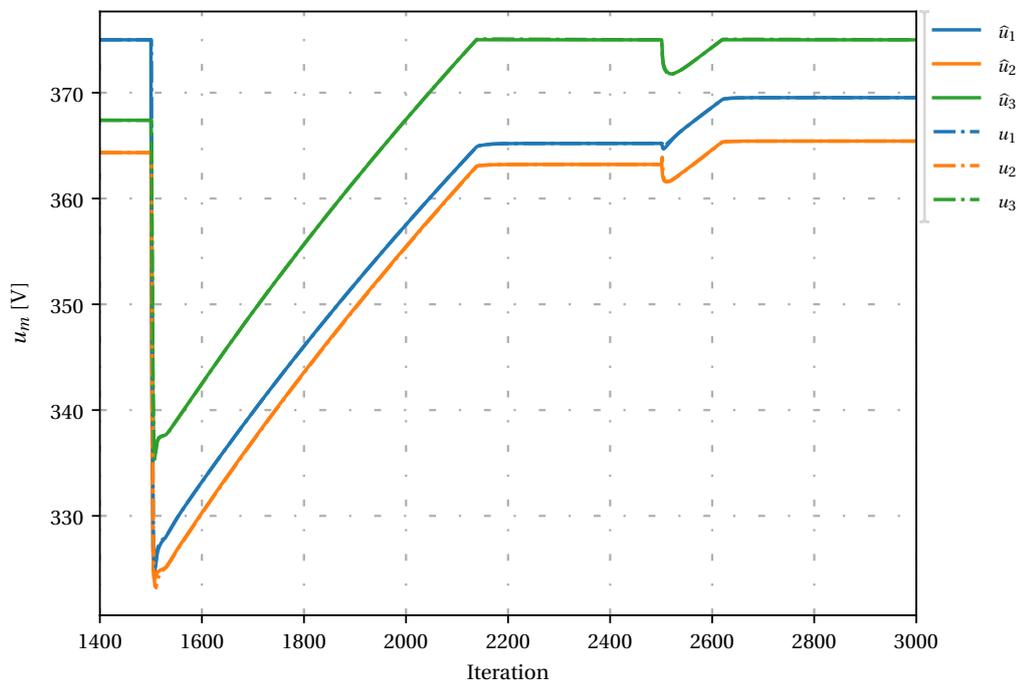


Figure 4.22: Nodal values of local voltage measurements and setpoints for the network presented in Figure 3.15, starting at iteration 1400.

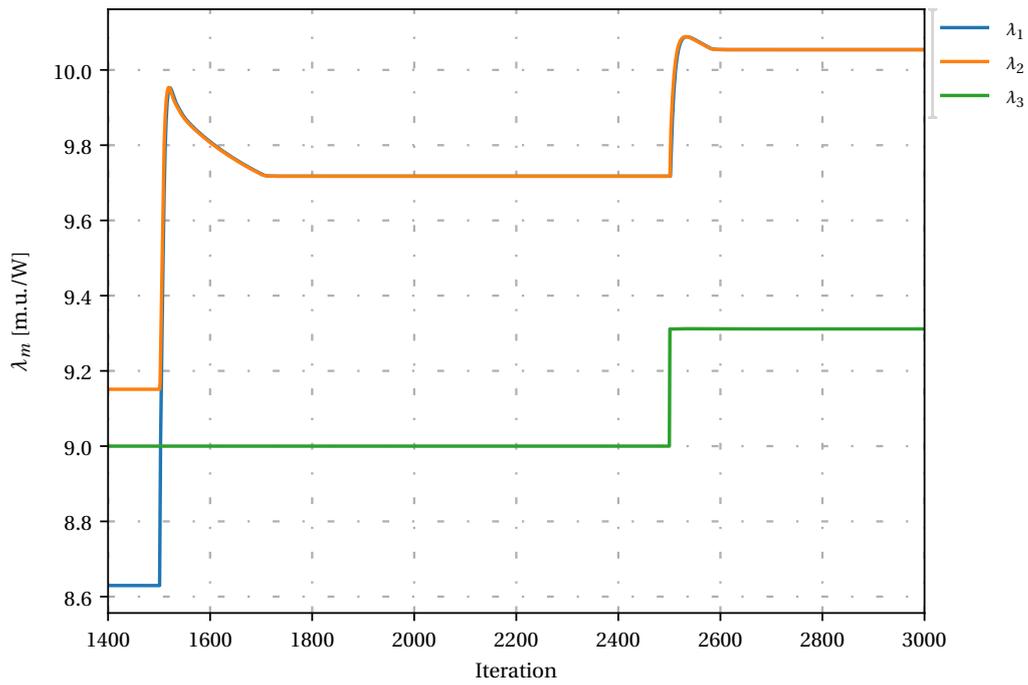


Figure 4.23: Nodal values of LMP for the network presented in Figure 3.15, starting at iteration 1400, with an increased price for node 1.

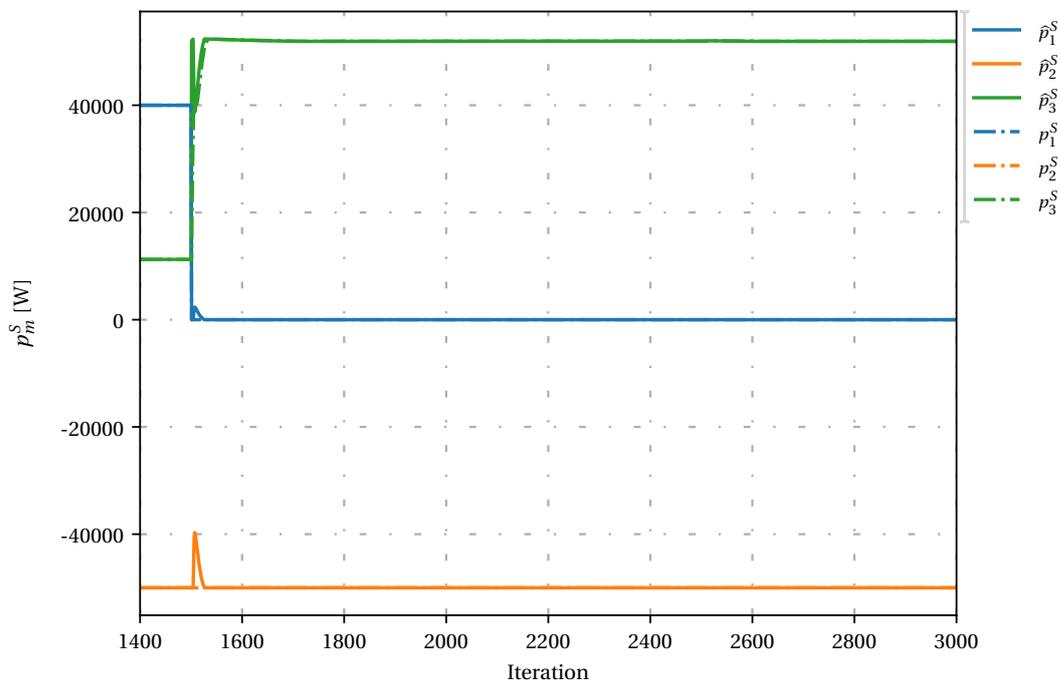


Figure 4.24: Nodal values of local power measurements and setpoints for the network presented in Figure 3.15, starting at iteration 1400, with an increased price for node 1.

is such big voltage drop is the same as explained in the previous section. However, λ_2 and λ_3 are quick to rise, meaning the generator in node 1 becomes marginal as well. Then, the power in node 1 can increase and around 2100th iteration, a new optimal point is reached. Once the cost function of node 3 is changed, it becomes more expensive to generate power in that node and, as such, the λ_1 rises, leading to a rise in λ_2 , and the p_2^S drops, until optimally is once again reached.

It is important to note that, even though node 3 is not at the maximum limit and has a LMP lower than the Minimum Marginal Cost of node 1, the latter is also generating power. This might seem counter intuitive and give the impression that the optimisation process is incorrect.

However, the losses, which are paid by the loads, have also to be taken into account. This means that, although it might be cheaper to produce more power in node 3, the load in node 2 would have an increased price when compared to the optimal solution in Figure, meaning that, overall, having node 1 supplying all the power would be less optimal. In fact, observing Figures 4.23 and 4.24, it is possible to confirm the previous statement. In this case, the generation cost for node 1 was set at such a high value that the generator never reaches marginal operation, while the rest of the parameters were kept the same, and so, node 1 is fully supplying the demand in node 2.

As can be seen at the end of the simulation, all the LMP values are higher, meaning that the overall system operation is more expensive.

4.4. Line Parameter Estimation

One assumption that has been made so far is that, when the system is initialised, the exact line resistances are known, i.e. that the values for $G_{m,n}$ are known. This assumption is, in a field application, not correct. In order to know the exact resistance of a line, it is necessary to know its exact length and value of Ω/km , or test every line before installing them. However, neither of the options before mention are applicable to the implemented distribution systems. Sometimes, the cables used might be old enough that there might not be accurate information and even in newer systems, the length of the cables registered in the databases is just an approximation of the real value. Furthermore, even if the information was accurate, the temperature of the lines changes during the operation, either by influence of external factors and due to heating by Joule's effect, which means the resistance of the line will inevitably change during operation of the system.

As such, in order to make sure the algorithm is tuned accordingly to the physical system, it becomes necessary to estimate the value of the $G_{m,n}$ at the same time the system is operating, represented by $\hat{G}_{m,n}$.

Initially, the system has no information and a conservative approach for the updates must be followed in order to make sure the system does not oscillate.

From Chapter 3, it is possible to observe that most of the tuning parameters are inversely proportional to $G_{m,n}$, $\alpha \propto 1/G_{m,n}$. In order to keep the updates small, α must also have smaller values, meaning $\hat{G}_{m,n}$ is initialised with a very high value, $\approx 10^7$. As time progresses, it becomes then necessary to approximate $\hat{G}_{m,n}$ to its real value. Recurring to Ohm's law, it is possible to get an estimation of the line conductance by using:

$$\hat{G}_{m,n}(l) = \frac{\hat{i}_{m,n}(l)}{\hat{u}_m(l) - \hat{u}_n(l)} \quad [\text{S}] \quad (4.17)$$

where $\hat{i}_{m,n}$ and \hat{u}_m are the line current and voltage measurements, respectively.

This estimation does not need to be performed every iteration because the shift in the line conductance is slow compared to the rate at which the iterations are performed, meaning that the value neighbours voltage level does not need to be received every time the updates are performed.

In asynchronous operation, the time at which the measurements are taken is different and, as such, when the local voltage level is sent, it also needs a time stamp. Once u_n is received, it is compared to a stored value of u_m which has the closer timestamp to the received value, making the approximation as close as possible.

In the simulations that were performed until now, the measured values for voltage and currents were the exact results from the grid simulation. In a real world implementation, however, perfect measurements are not possible, and as such, getting the exact values of voltage and currents from the physical layer becomes an increasingly invalid assumption. This is important to test because, in this case of imperfect measurements, the line resistance approximation will never take its exact value, but the system needs to converge, non the less. To reduce the effect the noise might have on the line conductance calculations, every new value was updated using the following expression:

$$\hat{G}_{m,n}(l) = \frac{\hat{G}_{m,n,\text{meas}}(l) + \hat{G}_{m,n,\text{old}}}{2} \quad (4.18)$$

This update strategy means that the new approximated value of $G_{m,n}$ was averaged between the old value and the one obtained by using the expression in (4.17).

In order to test performance under non-ideal conditions, white noise was introduced in all the measurements, with a maximum amplitude of 0.1% of the real value. This means, for example, that for a voltage of 375 V, the measured value could be $\hat{u}_m = 375 \pm 0.375$ V.

In Figures 4.27 to 4.25, it is possible to see the results for the 4 node linear network, shown previously in Figure 3.26.

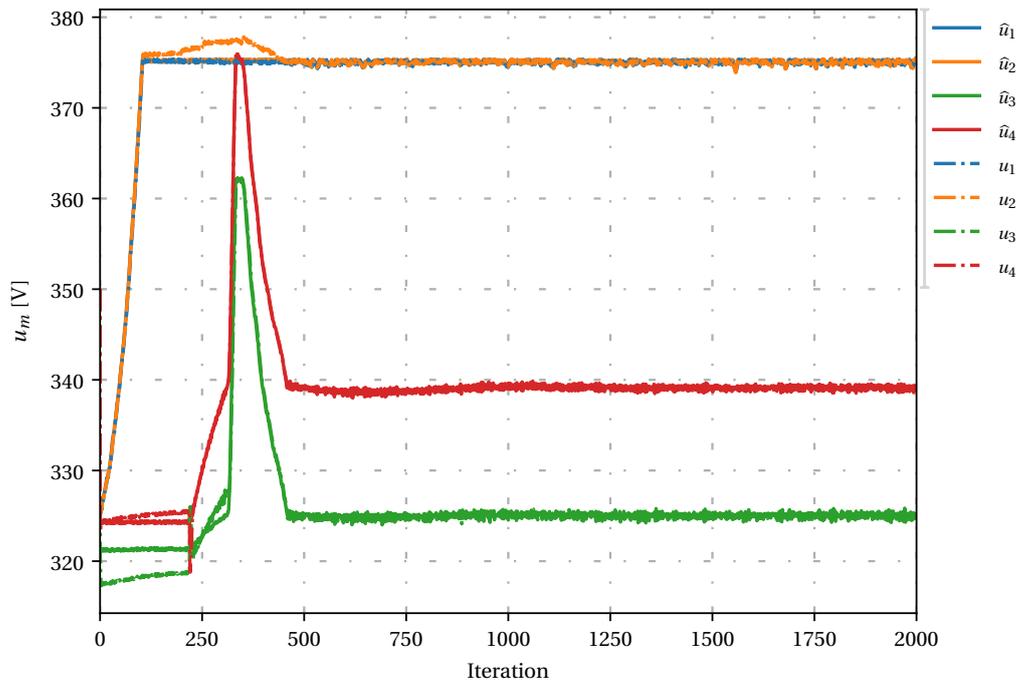


Figure 4.25: Nodal values of local voltage measurements and setpoints for the network presented in Figure 3.26, with an introduced noise of 0.1%.

When looking at the results it is possible to observe the small amplitude oscillation caused by the the introduced white noise, especially if Figure 4.25 is compared to the previous results, shown in Figure 4.13, for example. The introduced noise means that LMP will never converge to a stationary point, since the local voltage and power is never constant. However, the oscillatory behaviour is centred around the solution and has a very small amplitude, meaning that the algorithm is working as intended, as the system is in the most optimal state possible with the measurements.

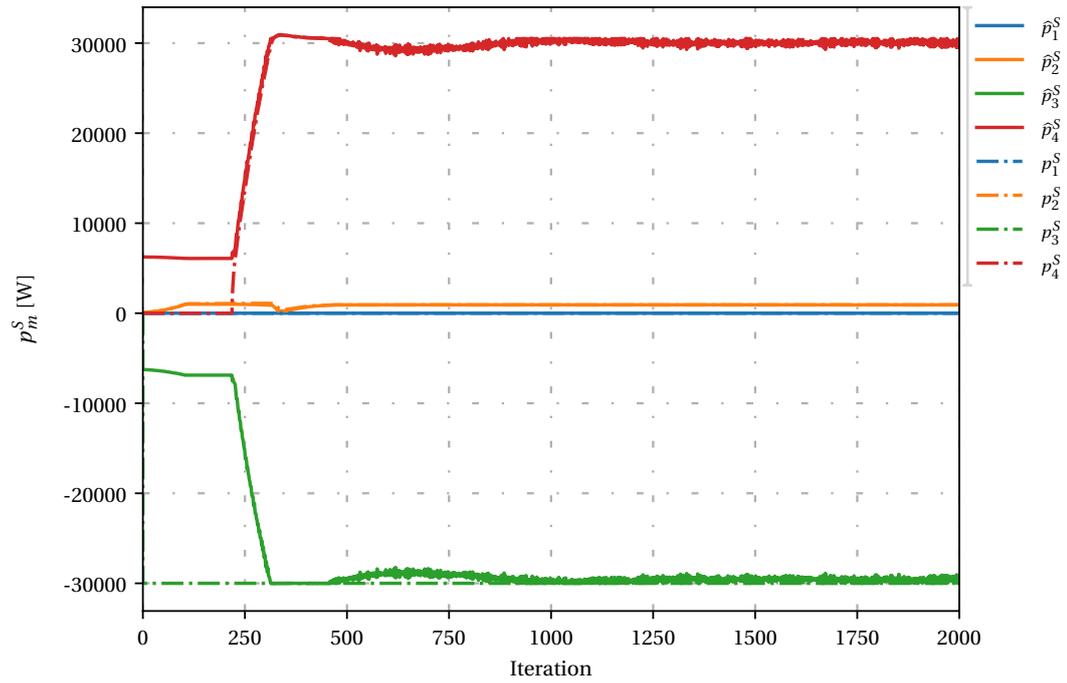


Figure 4.26: Nodal values of local power measurements and setpoints for the network presented in Figure 3.26, with an introduced noise of 0.1%.

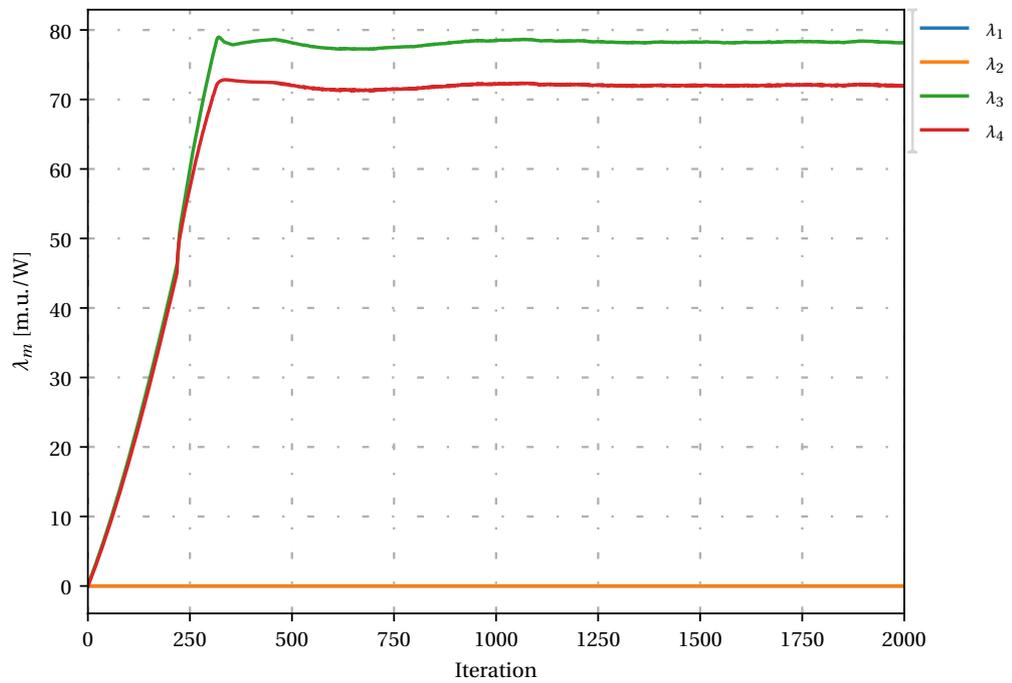


Figure 4.27: Nodal values of LMP for the network presented in Figure 3.26, with an introduced noise of 0.1%.

5

Case Study

Until now, every test was done in relatively small synthetic grids. And although these tests give good insight on how is the algorithm performing, it is also important to run this algorithm in conditions that better resemble the real world. In order to do so, a test case was devised, base on a section of the street lighting grid at Zoetermeer's municipality, in the Netherlands.

5.1. Description of the Test Case

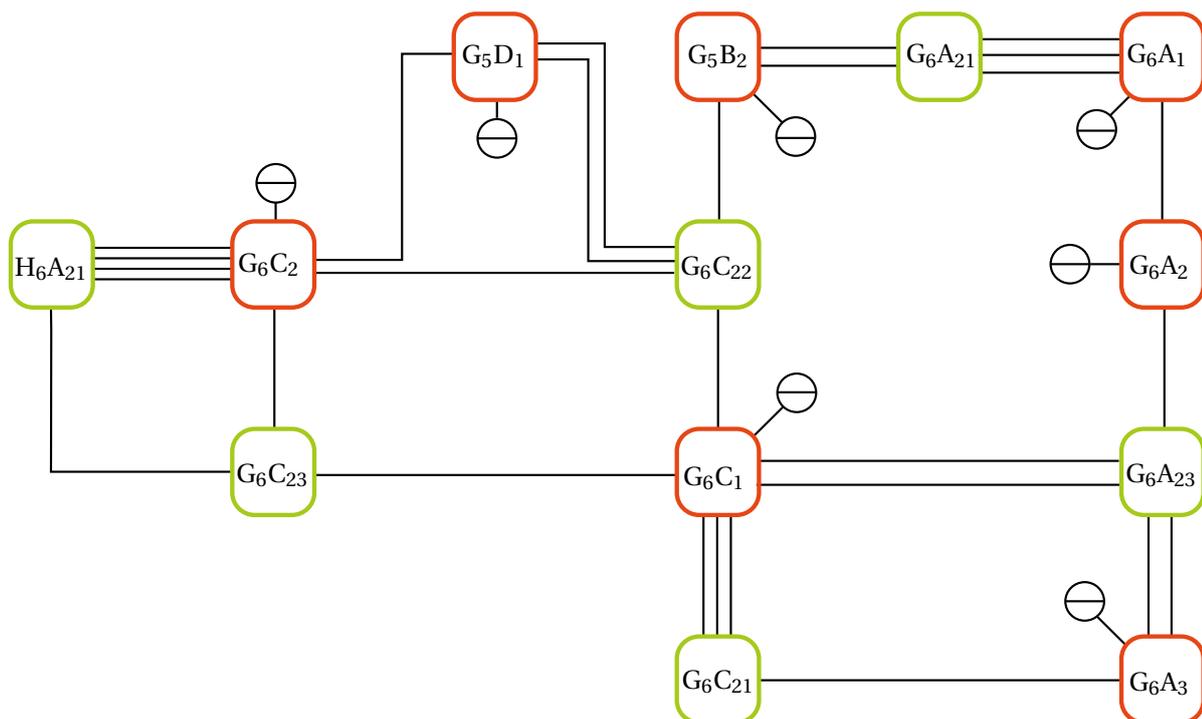


Figure 5.1: Scheme of the selected section of Zoetermeer's street lighting grid. The junction boxes where the lines are connected to each other can be seen in red and green, with the difference being that the red boxes also serve as the feeding points for the grid.

In order to perform the simulations, it was necessary to, firstly, define the grid.

Initially, a section with an area of approximately 47 Hectares was chosen, mainly because it had few connections to the rest of the lighting network and, as such was easier to isolate, and hence, easier to test. Within that zone, there were two identified types of junction boxes: the ones connected to the power grid, from where power was being supplied from, which will be called Feeding Boxes, and the ones which served as end-points for the cabling, which will be named Connection Boxes. A list of all the boxes can be seen in Table 5.1.

Table 5.1: List of boxes from the selected section of Zoetermeer's street lighting grid.

Feeding Boxes			Connection Boxes	
Box Name	Bus Number	Maximum Power Limit [kW]	Box Name	Bus Number
G5B2	1	1000	G6A21	5
G5D1	2	1000	G6A23	6
G6A1	3	1000	G6C21	10
G6A2	4	1000	G6C22	11
G6A3	7	1000	G6C23	12
G6C1	8	1000	H6A21	13
G6C2	9	1000	_____	_____

As it currently stands, the system runs in AC and is arranged radially, from the feeding boxes and ending up in the connection boxes. Both boxes can house multiple cables, however, as it stands, they are not interconnected between them. For this test, the whole grid was meshed and, as such, there is copper connection between all the lines inside a junction box. A scheme of the grid can be seen in Figure 5.1, where the different box types are in different colours, the generators, which model the connection to the external grid, are shown as a circle with an line in the middle, and every line connecting two boxes is a underground cable.

Currently, the network serves only as a lighting grid, meaning that there is a constant load at every light pole. Since the system would be switched so it operates in DC, the chosen illumination technology is LED, which translates to a consumption of 25W [36] per light pole. However it was conceived that every pole would also have a electric car charging station attached, which would add an extra 5 kW of load at every point. Finally, it was assumed that there was a light pole every 25 meters, meaning that every 25 meters of cable, there would be a total load of 5.025 kW. However, although the 25 W lighting is fixed, the power fed to the converters is variable and, as such, a price was given to these loads, in order to make demand response possible. I.e., setting a cost function on the loads meant that if the LMP was higher than the assign cost, the local power demand would reduce.

For the feeding boxes, a cost function of $C_m = 0.001p_m^S{}^2 + 2p_m^S$ m.u. was given and, for the light poles, a cost function was given as $C_m = -0.001p_m^S{}^2 - 10|p_m^S|$, where $p_m^S \leq 0$.

Although the exact installed cable type is unknown, some information was available: it had a 4 core copper conductor, each with 10 mm² cross section. As such, a similar cable was chosen: the VG-YMvKas Dca, from Nexans [37]. A list of all lines shown in Figure 5.1 can be seen in Table 5.2.

Contrary to previous cases, the voltage range of the system is from 650V - 750V with an additional 5V margin for the droop, meaning that the full range of voltage is set from 645V to 755V.

There was no induced communication failure or measurements errors because the system is already very complex and adding more uncertainty to the system would mean it could become to complicated to analyse.

5.2. Test Results

Although testing the grid with a node every 25 m of line would be ideal, it made it so the grid had more than 400 nodes. This made it very computationally complex to test, and a new solution was found. In order to reduce the number of nodes, but maintain the characteristics of the test, loads were aggregated into sets of 6 or 8, i.e, instead of having a pole with a 5.025 kW demand every 25 m, there would be a pole with a 40.2 kW demand every 200 m, in case of agregating 8 loads into 1.

As such, two tests were preformed: one with a load every 200 m, which totalled 53 nodes, and another with a load every 150 m, which totalled 67 nodes.

The results for the first set of simulations can be seen in Figures 5.2 to 5.5.

From the analysis of the simulations, the first conclusion that can be drawn is the fact that the system converges in less than 4000 iterations. Also, from Figure 5.3, it is possible to observe that feeding boxes with more connections, node 8 and node 9, are supplying more power than the other ones and that, overall, the nodes which are supplying power are all at maximum voltage, as seen in Figure 5.4.

What is also important to highlight is the fact that the loads, due to the fact that they were given a cost function, are performing demand side response to line congestion. This is possible to observe by the many different levels of load consumption observed as grey lines in Figure 5.3. Every grey line in this figure represent the agglomeration of 8 light poles with EV charging.

Table 5.2: List power cables in the selected section of Zoetermeer's street lighting grid.

Line Index	Line resistance [Ω/m]	Length [m]	Maximum Current [A]
(1,5)	0.00168	232	61
(1,5)	0.00168	275	61
(1,11)	0.00168	429	61
(1,11)	0.00168	444	61
(2,9)	0.00168	591	61
(2,11)	0.00168	327	61
(2,11)	0.00168	464	61
(3,4)	0.00168	546	61
(3,5)	0.00168	241	61
(3,5)	0.00168	557	61
(3,5)	0.00168	607	61
(4,6)	0.00168	257	61
(7,6)	0.00168	314	61
(7,6)	0.00168	209	61
(7,10)	0.00168	259	61
(8,6)	0.00168	301	61
(8,6)	0.00168	423	61
(8,10)	0.00168	371	61
(8,10)	0.00168	204	61
(8,10)	0.00168	316	61
(8,11)	0.00168	286	61
(8,12)	0.00168	397	61
(9,11)	0.00168	227	61
(9,12)	0.00168	235	61
(9,13)	0.00168	837	61
(9,13)	0.00168	374	61
(9,13)	0.00168	262	61
(9,13)	0.00168	245	61
(12,13)	0.00168	179	61

Due to current congestion near the feeding boxes, seen in Figure 5.5, the LMP of the non-power supplying nodes increases and, due to the losses in the network, the further away a node is from a box, the more expensive it is to consume power and, hence, the demand is reduced. It is also possible to observe that some loads have a LMP lower than the minimum marginal cost, which means they are at maximum power consumption and there is no line congestion in the connecting lines.

When testing for 68 nodes, meaning a load every 150 m, the convergence time was approximately the same, 4000 iteration. This means that for a system 25 % larger, there was no significant increase in the convergence time. This gives a good indication that this algorithm is scalable and can work well under increasingly bigger networks, which is one of the advantages of fully decentralised systems.

Comparing this result with the ones in previous chapters, it suggests that the algorithms speed of convergence might be more heavily dependent on the network characteristics and topology and less on the actual number of nodes of a network, since previous tested networks were very different between them and the results showed a big difference in the number of iterations it took to reach convergence, while the main variable that changed in this test scenario was the number of nodes of the network, which had little impact on the convergence speed.

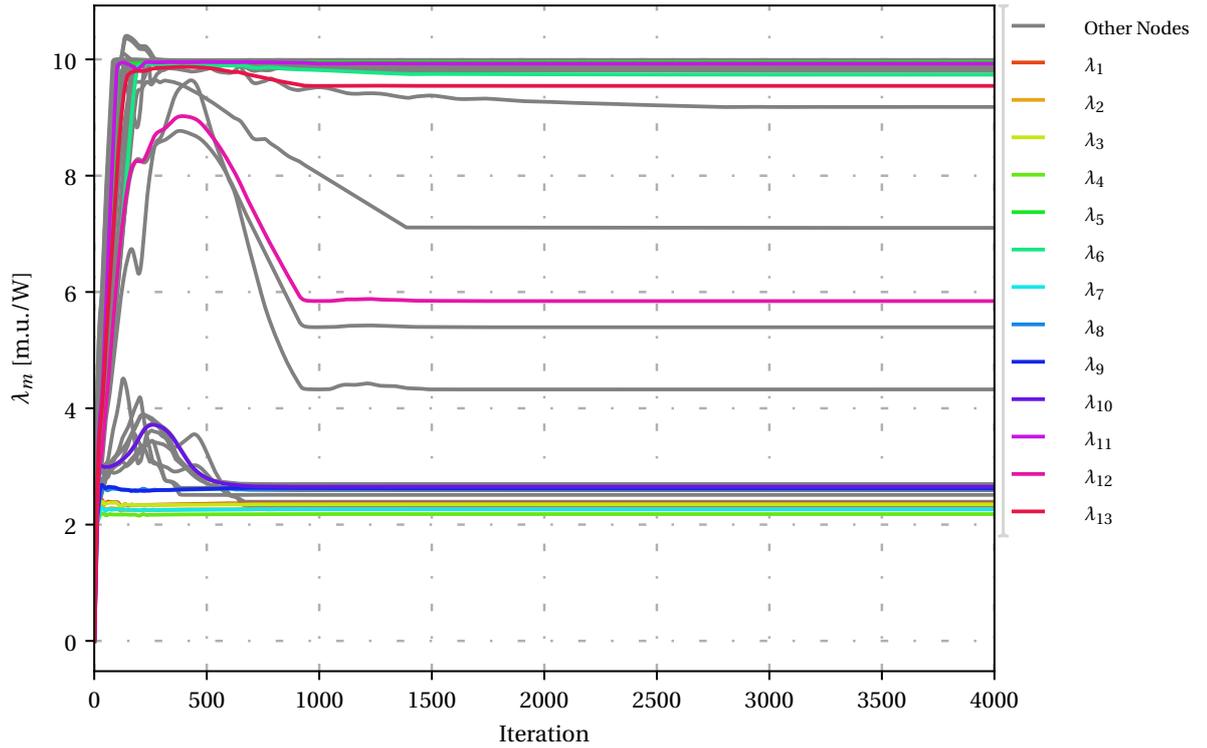


Figure 5.2: Nodal values of LMP for the network presented in Figure 5.1, with 53 nodes. In coloured lines is possible to observe the feeding and connection boxes, while the light poles present grey lines.

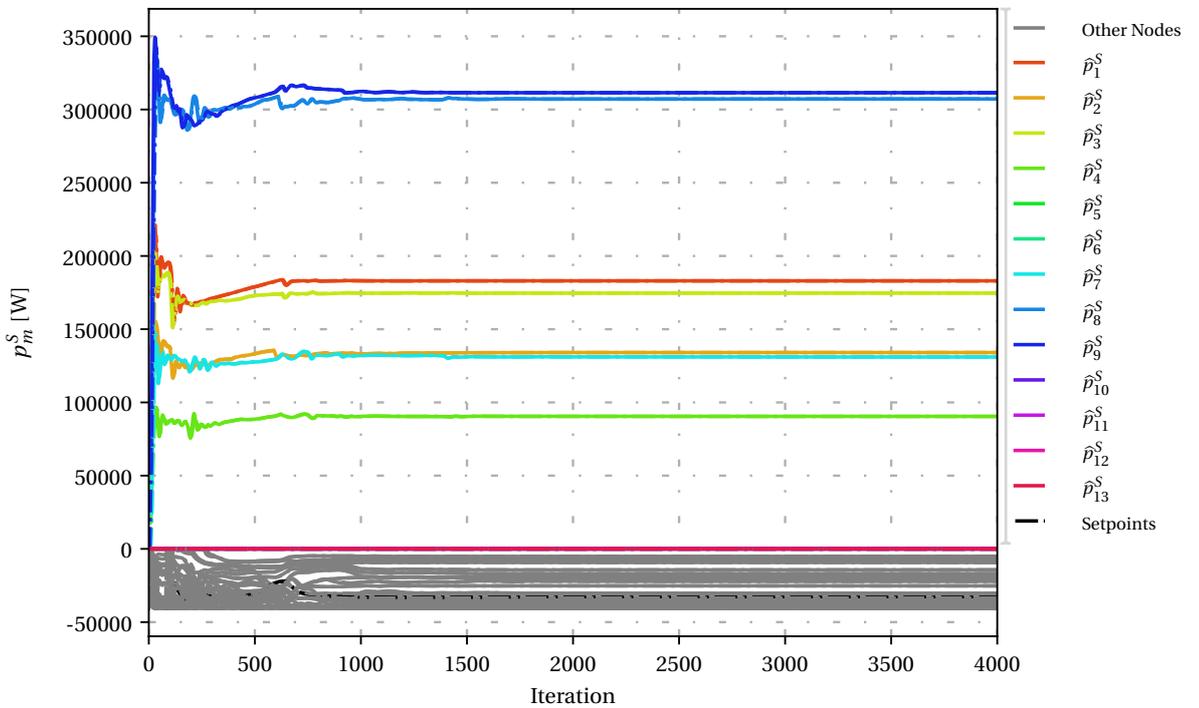


Figure 5.3: Nodal values of local power measurements and setpoints for the network presented in Figure 5.1, with 53 nodes. In coloured lines is possible to observe the feeding and connection boxes, while the light poles present grey lines. The setpoints are presented as dash-dotted lines.

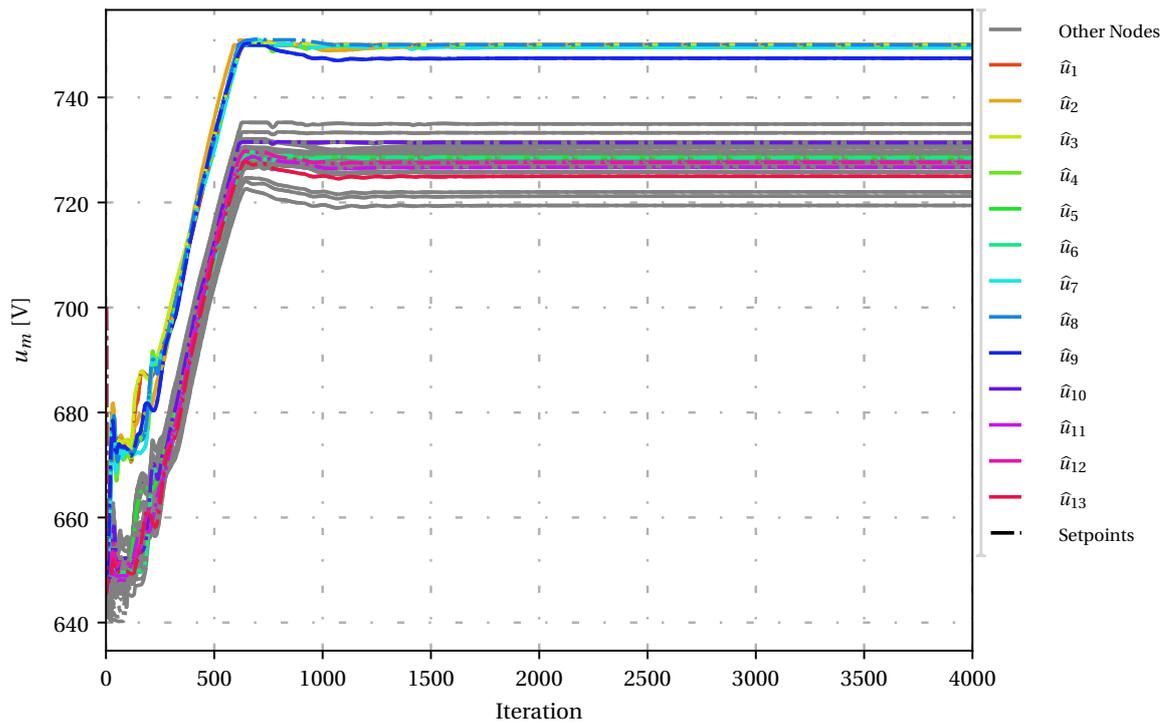


Figure 5.4: Nodal values of local voltage measurements and setpoints for the network presented in Figure 5.1, with 53 nodes. In coloured lines is possible to observe the feeding and connection boxes, while the light poles present grey lines. The setpoints are presented as dash-dotted lines.

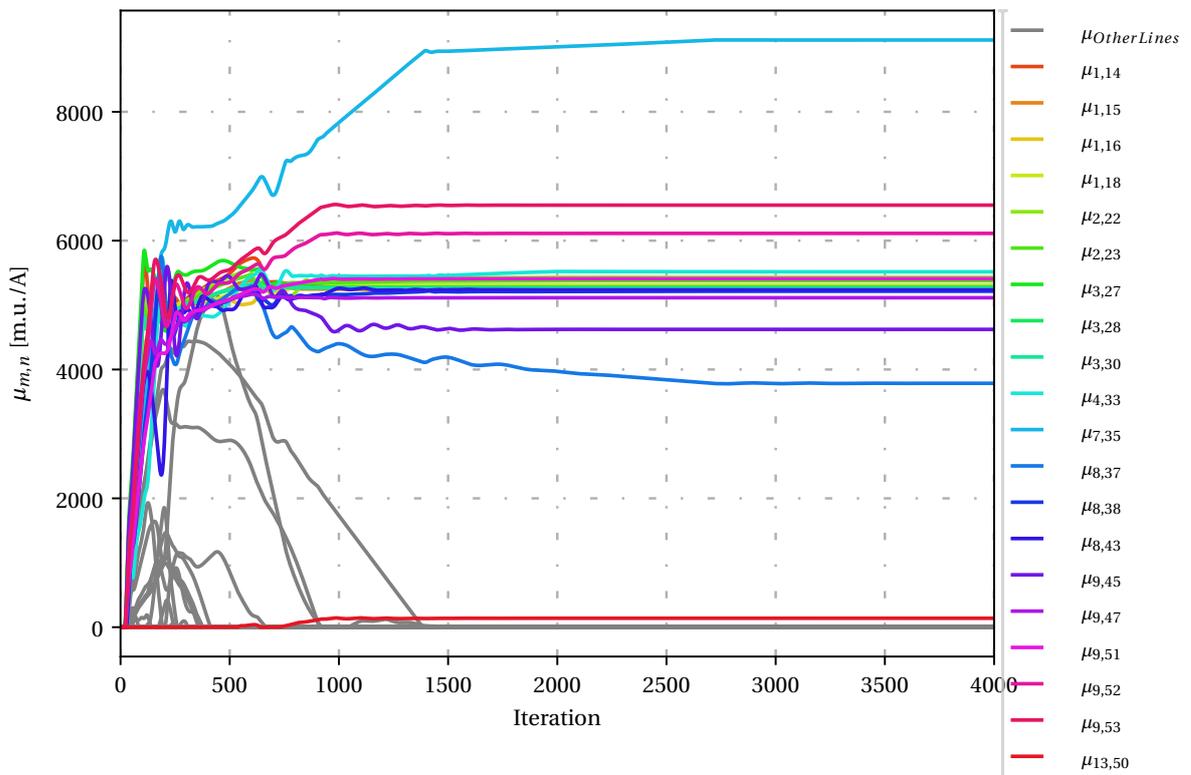


Figure 5.5: Nodal values of local current measurements and setpoints for the network presented in Figure 5.1, with 53 nodes. In coloured lines is possible to observe the feeding and connection boxes, while the light poles present grey lines. The setpoints are presented as dash-dotted lines.

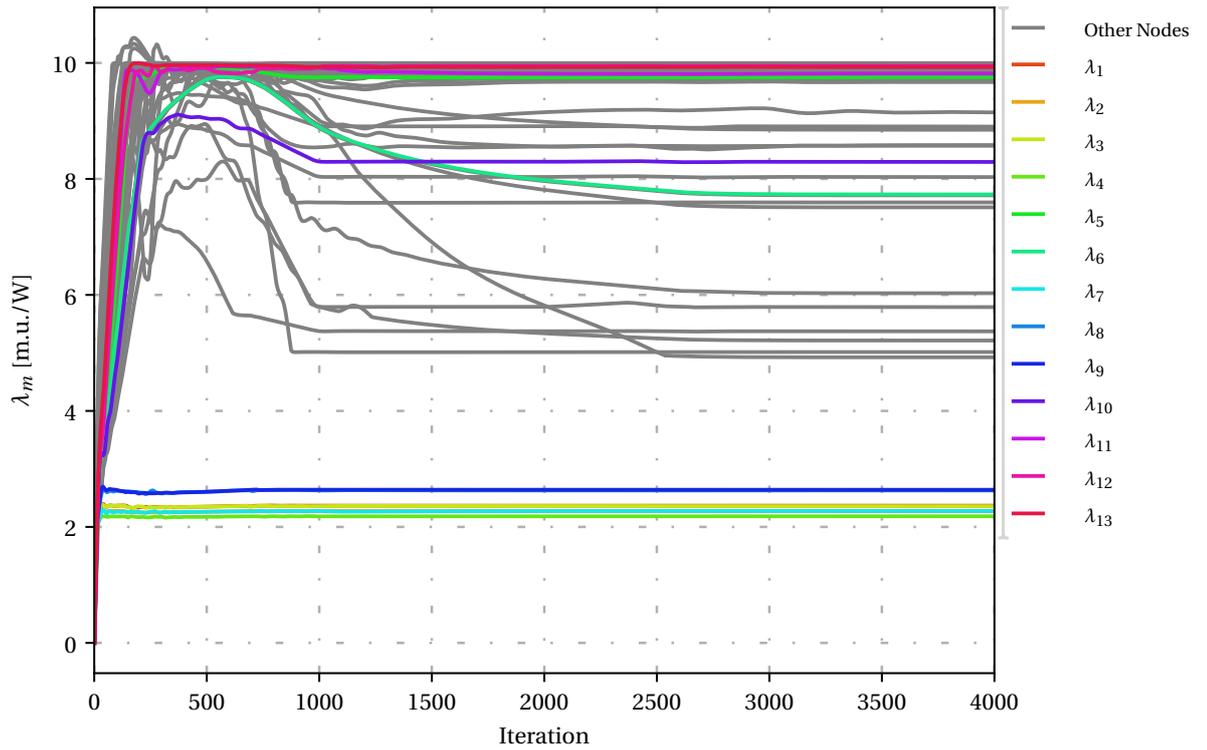


Figure 5.6: Nodal values of LMP for the network presented in Figure 5.1, with 67 nodes. In coloured lines is possible to observe the feeding and connection boxes, while the light poles present grey lines.

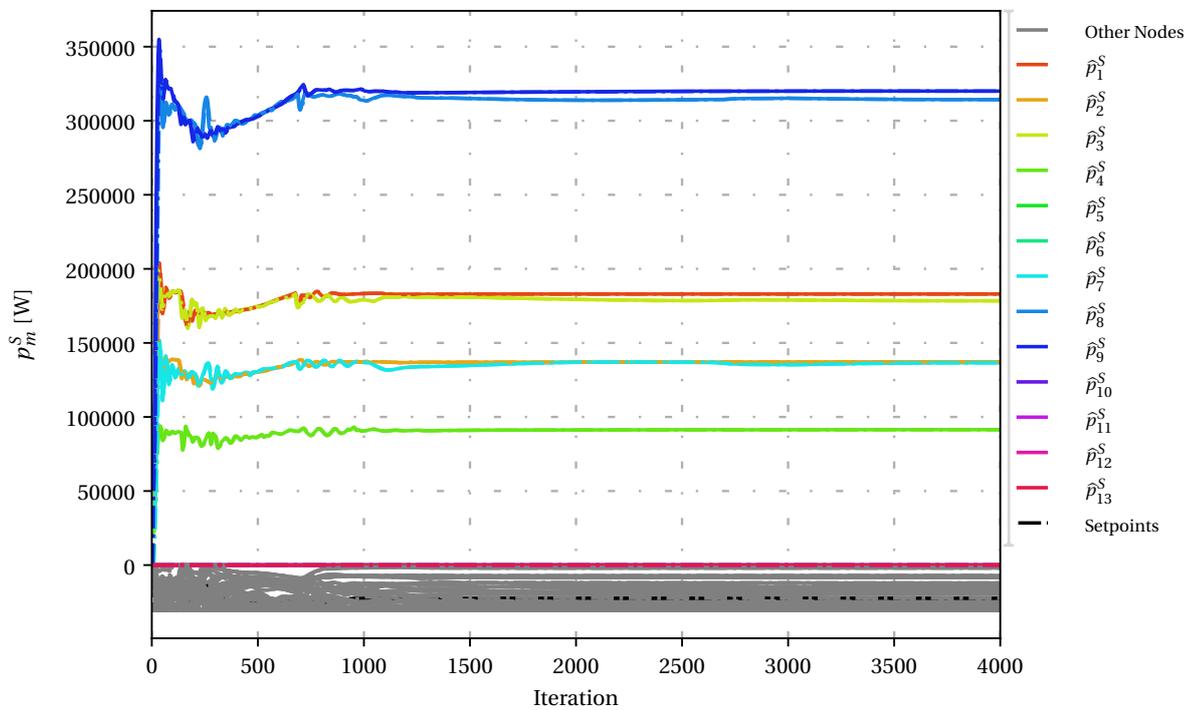


Figure 5.7: Nodal values of local power measurements and setpoints for the network presented in Figure 5.1, with 67 nodes. In coloured lines is possible to observe the feeding and connection boxes, while the light poles present grey lines. The setpoints are presented as dash-dotted lines.

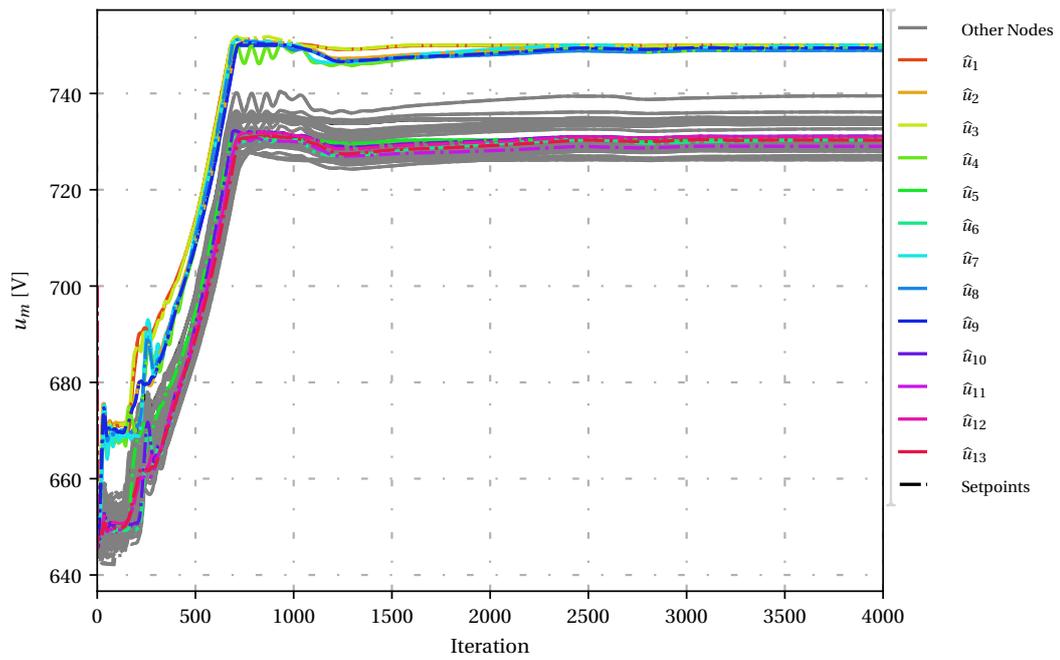


Figure 5.8: Nodal values of local voltage measurements and setpoints for the network presented in Figure 5.1, with 67 nodes. In coloured lines is possible to observe the feeding and connection boxes, while the light poles present grey lines. The setpoints are presented as dash-dotted lines.

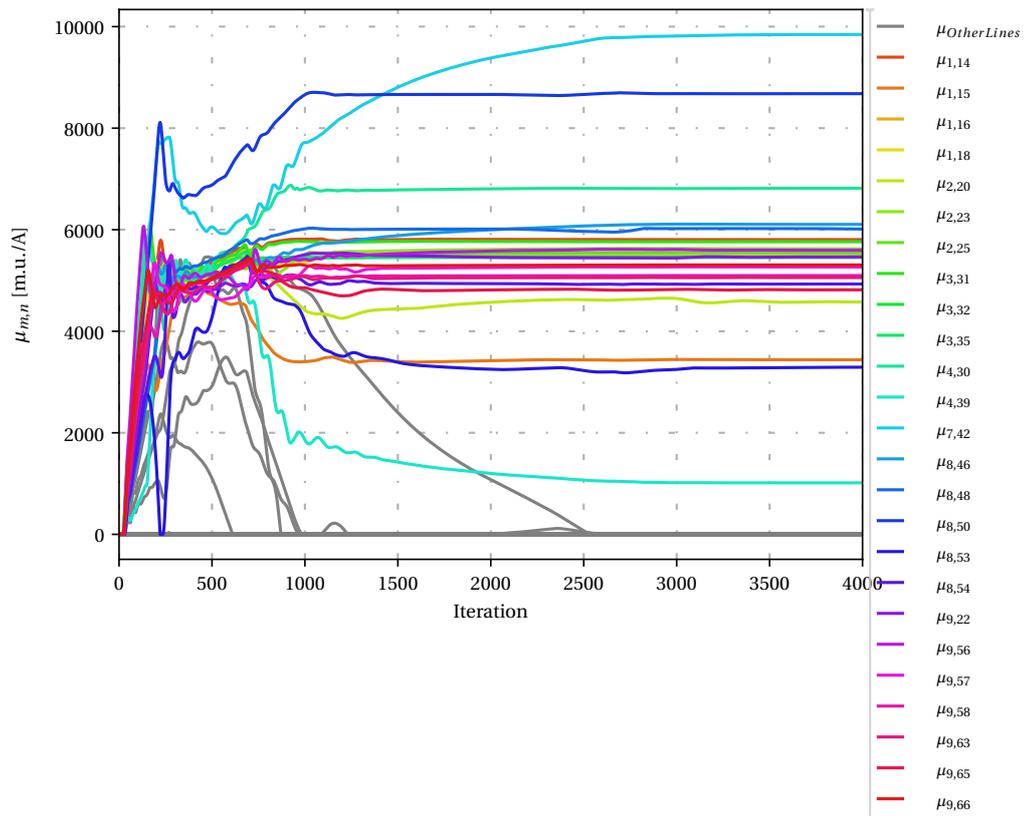


Figure 5.9: Nodal values of local current measurements and setpoints for the network presented in Figure 5.1, with 67 nodes. In coloured lines is possible to observe the feeding and connection boxes, while the light poles present grey lines. The setpoints are presented as dash-dotted lines.

6

Conclusion and Future Research

6.1. Conclusion

As previously stated in the introduction chapter, the main objective of this thesis report is to improve the speed and flexibility of the C+I fully decentralised online optimisation algorithm of DC Microgrids using local physical system measurements. The discussion of this objective will be done by answering the research questions presented in 1.5.2.

How can physical measurements be used to improve the convergence rate?

In Chapter 3 physical measurements were included in the optimisation process. This was possible by measuring the current that was flowing in or out of the node and the local voltage. The local power was then obtained as a product of the local voltage and current measurements. Using these new variables, the updates were changed in order to use the line currents and local voltage instead of needing the neighbours voltage level. This meant less information had to be shared between nodes. This new update strategy also meant that the optimisation layer had to control the droop control curve, in order to try to make the physical node follow the optimal setpoints.

The inclusion of physical measurements had 2 main objectives: to reduce the number of iterations necessary for the system to converge to an optimal point and to take steps into making this an online optimisation strategy.

As it was previously implemented, the algorithm had to solve two problems while running the optimisation problem: the nodes had to agree on a feasible physical solution and, on top of that, it should be the less costly solution. By including the physical network, the first problem is solved automatically by the grid, meaning that the optimisation layer only has to work on making the network converge to the most cost effective solution. This also included a feedback system, where the cyber layer would introduce the setpoints for power and voltage and would get a reaction from the grid, meaning it could react to the actual grid state as the optimisation process develops, achieving online optimisation.

As a matter of fact, it was possible to see that the developed implementation, when compared to previous ones in [27], led to an increase in convergence speed, especially on bigger networks. In comparative terms, the online optimisation converged to the same solution 6 times faster in the best case, and at the same speed in the worst tested case.

What is the impact of communication loss on the convergence?

In Section 3.6, the effects of communication were tested in the system.

Initially, the same approach as [27] was taken to run the algorithm asynchronously: set a timeout and if the nodes didn't receive the necessary information within that time period, the update would be run with old information. This turned out not to be the best option because the same timeout period would apply to the grid simulations. This meant that the nodes would not receive measurements every iterations and the overall system would oscillate wildly.

Therefore, a new solution was implemented where a communication failure rate was set for each simulation and, every iteration, the nodes would either use previous information or use new received information

depending on a predefined chance. This way, it was also possible to have having consistent data over multiple simulations, since every consecutive run of the algorithm, the nodes would use newly received values at the same iterations.

When tested for both 25% and 70% communication loss, in both occasions the system successfully converged to an optimal point, although a 70% failure rate meant that there was an increased oscillatory behaviour. This was possible because the only exchanged variables in the system are the LMP and the dual variables for the line limits, with every other variable being obtained locally. This means that, compared to previous implementations, there is a bigger autonomy from other the nodes and, therefore, a failure in communication is not as critical.

It is important to note that it was not possible, because of the before mentioned limitations, to test if this application was faster in terms of time. However, from research question 1 it was concluded that this application took less iterations to converge and from this section it was concluded that this algorithm is relatively resistant to communication failures, meaning that, in a real world application, time out periods could be reduced, which is a very strong indication that, overall, this new implementation will take less time to converge to optimal values than previously developed work.

How can the optimisation parameters be adapted online to improve convergence rate?

In Chapter 4, a new solution in order to adapt the tuning parameters of $\mu_m^{\bar{U}}$ was devised, in order to increase convergence rate and reduce oscillations in the online optimisation process.

One of the proposed solutions was to make $\beta^{\bar{U}}$ inversely proportional to the rate at which the local voltage was varying. This meant that if the rate was low, the gain would increase but in case the variation rate was high the gain would reduce in an attempt to reduce the oscillations in the system. Simulations showed that this approach was successful in reducing both oscillations in the maximum and minimum voltage, which helped convergence time.

Another solution that was implemented was the addition of a voltage error integrator, which increased the rise on the maximum voltage limit dual variable under certain conditions. The use of this strategy was shown to have a great effect in reducing convergence time, when applied to the marginal generator, reducing the number of iterations, in some cases, in 25 %.

One other parameter that was changed during the optimisation process was the value of the line conductance, $G_{m,n}$. Although it didn't change the convergence speed directly, it made the algorithm more adaptable to changes in the network, since it would approximate the line parameters on its own, without input from the user. In case there are changes in the network, the algorithm would adapt automatically to those changes, and, in that sense, make the algorithm more flexible.

What is the impact of changing supply and demand on the convergence rate?

In Chapter 4, in Sections 4.2 and 4.3, the effects of modifying the power of loads and generators, as well as the cost of production of said power.

From the simulation results, two instances could be differentiated: the first is the instantaneous response of the physical system to an event, e.g. an increase in power demand, and the second is the following optimisation process to the new optimal point.

The first instance is dependent on the droop and how the curves are drawn. It was possible to observe that the strategy that was used in this project meant that, if the marginal nodes didn't have enough margin to accommodate for the shift in power, then the voltage could decrease very rapidly until the local voltage was low enough that some loads could reduce their demand. It was suggested that the droop curves were created differently, however, this new approach could not be tested due to computational limitations.

Then, it was up for the optimisation process to make sure the system converged to a new optimal state. The previously described behaviour meant the operating point after the power shift could be further away from the optimal point, meaning it spent more iterations converging to the optimal values than it would be optimally be required.

That being said, the systems were always able to converge to a new optimal values, in less or around the same number of iterations it took to initialise the system, which are positive results.

For the price changes, the results were similar. It was possible to change the characteristics of the nodal cost functions by changing the values of both A_m and B_m . If the price change meant there would be no marginal generator which would be able to account for the power shift caused by the changing cost, then the same voltage decrease was observed. Otherwise, the system was fast to converge to a new optimal state.

6.2. Future Research

As it stands, the algorithm can only run in a unipolar DC grid. Although they are simpler, unipolar DC grids are not as efficient and flexible as bipolar DC grids for power transmission. As such, investigative work will have to go into researching and developing a way of turning the developed solution into an algorithm that is able to optimise bipolar DC grids.

Another point for further research is to test how the algorithm behaves in a truly asynchronous mode. For that, it is necessary to remove the grid simulation constraints, both in terms of simulation time and communication with the nodes, which limited the ability to test the full potential of this solution.

One last point for research in the future would be how to improve on the adaptive behaviour of the algorithm during the optimisation process. Although the presented solutions worked, more thought and new ways to better tune the updates can be developed in order further reduce oscillations on the system and increase the convergence speed. Furthermore, it is necessary to test some of the suggested modifications on the droop in order to reduce the large voltage drops which happen when changing power and cost during the optimisation process.



Simulation Results

A.1. Results for the simulation of the 6 node grid with congestion
The following plots are the results of the simulation of the network shown in 3.20.

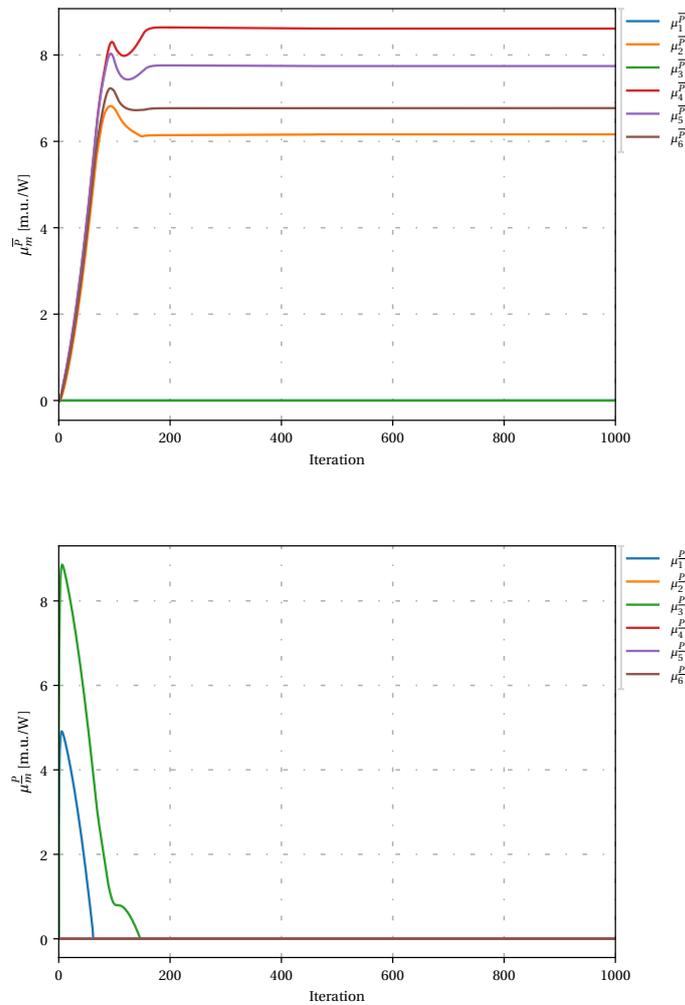


Figure A.1: From top to bottom: nodal values of maximum power limit dual variable and nodal values of minimum power limit dual variable.

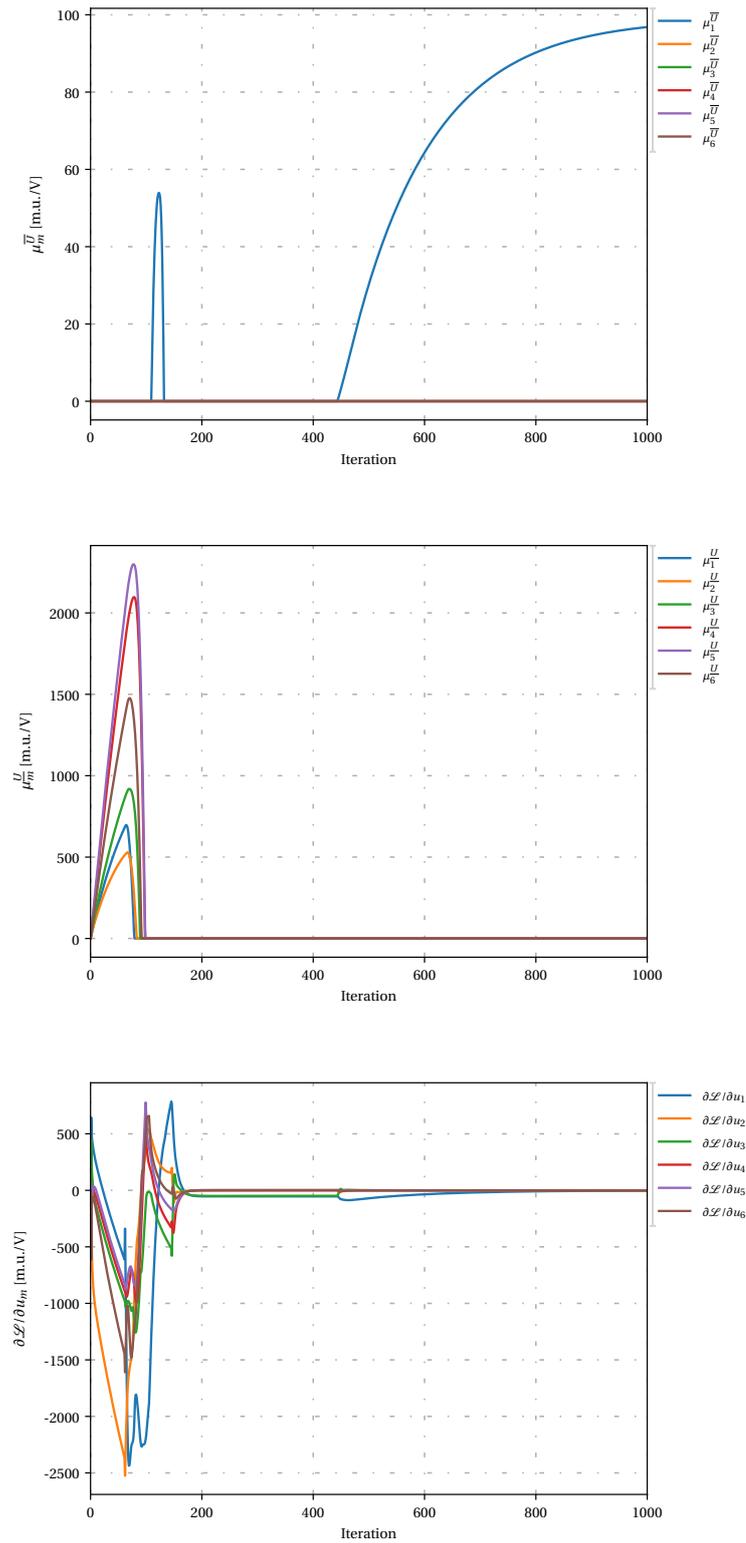


Figure A.2: From top to bottom: nodal values of maximum voltage limit dual variable, nodal values of minimum voltage limit dual variable and local derivative of the Lagrangian function in terms of voltage.

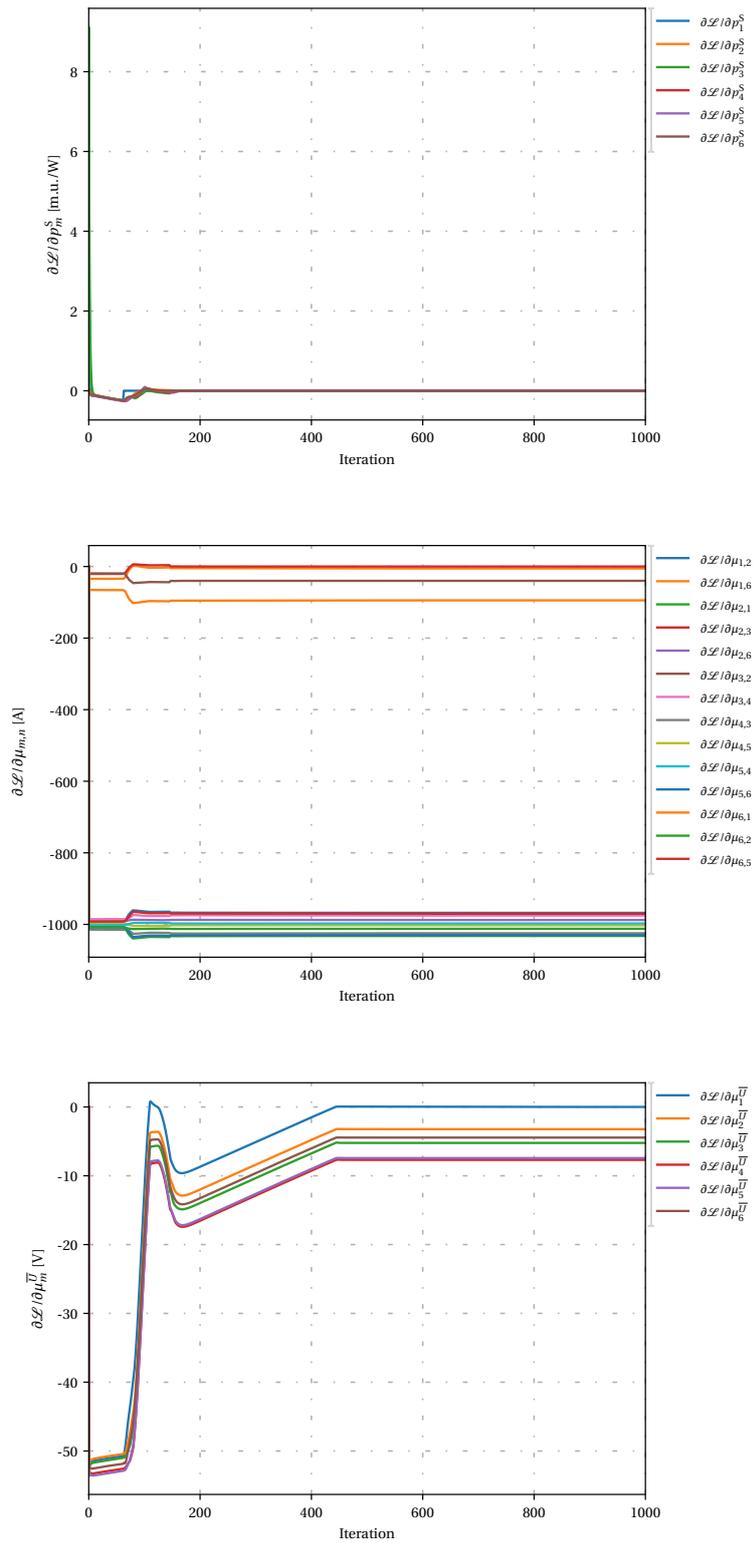


Figure A.3: From top to bottom: local derivative of the Lagrangian function in terms of power, local derivative of the Lagrangian function in terms of the dual variable for the line current limits and local derivative of the Lagrangian function in terms of the dual variable for the maximum voltage limit.

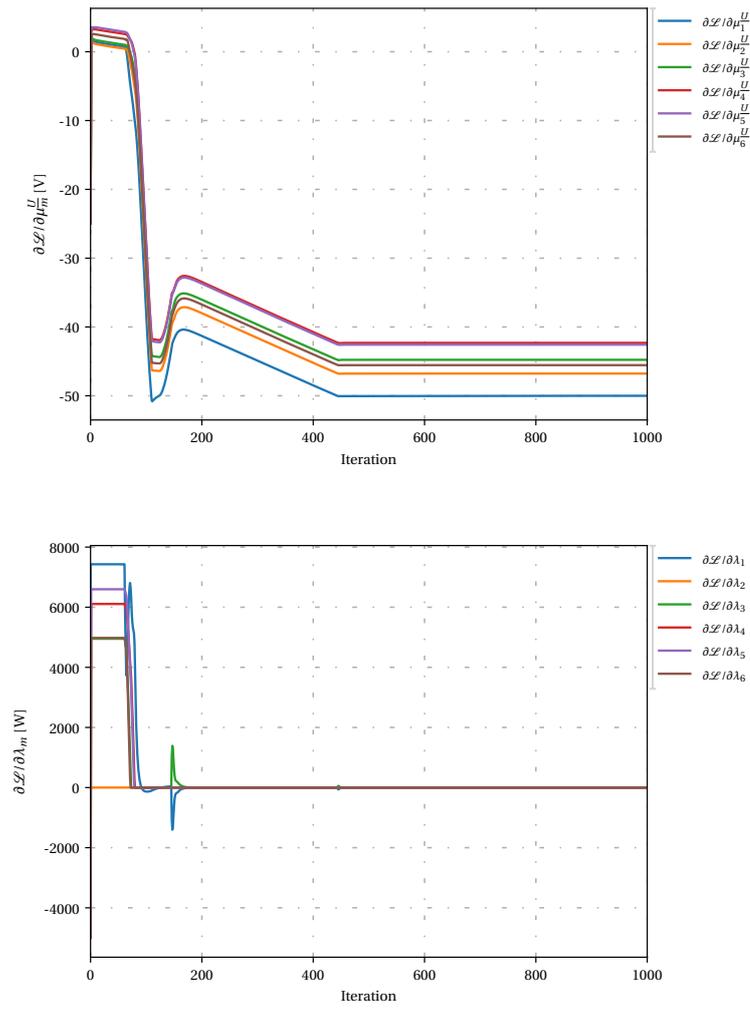


Figure A.4: From top to bottom: local derivative of the Lagrangian function in terms of the dual variable for the maximum voltage limit and local derivative of the Lagrangian function in terms of LMP.

A.2. Results for the simulation of the 4 node grid, with voltage congestion, after tuning parameters were made adaptive.

The following plots are the results of the simulation of the network shown in 3.26.

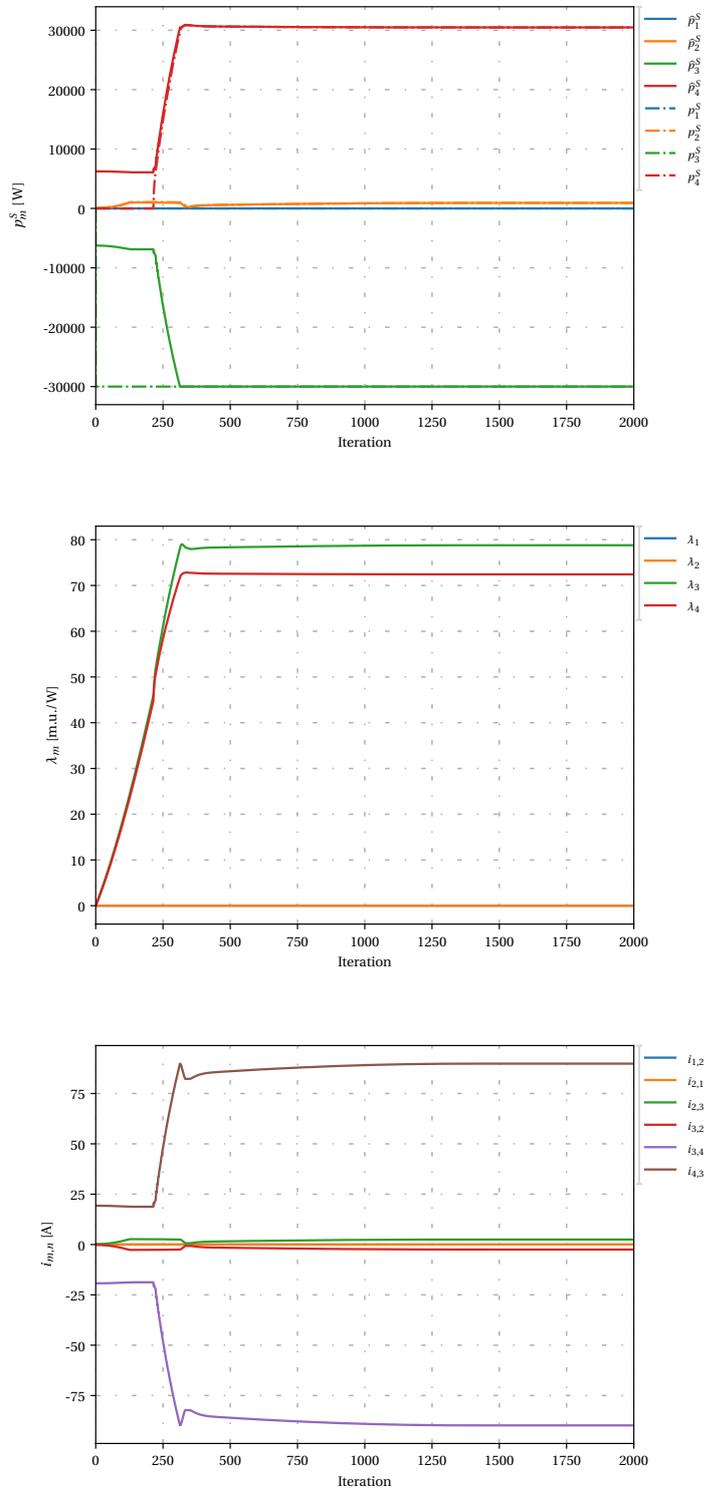


Figure A.5: From top to bottom: nodal values of local power measurements and setpoints, nodal values of LMP variable and line current measurements.

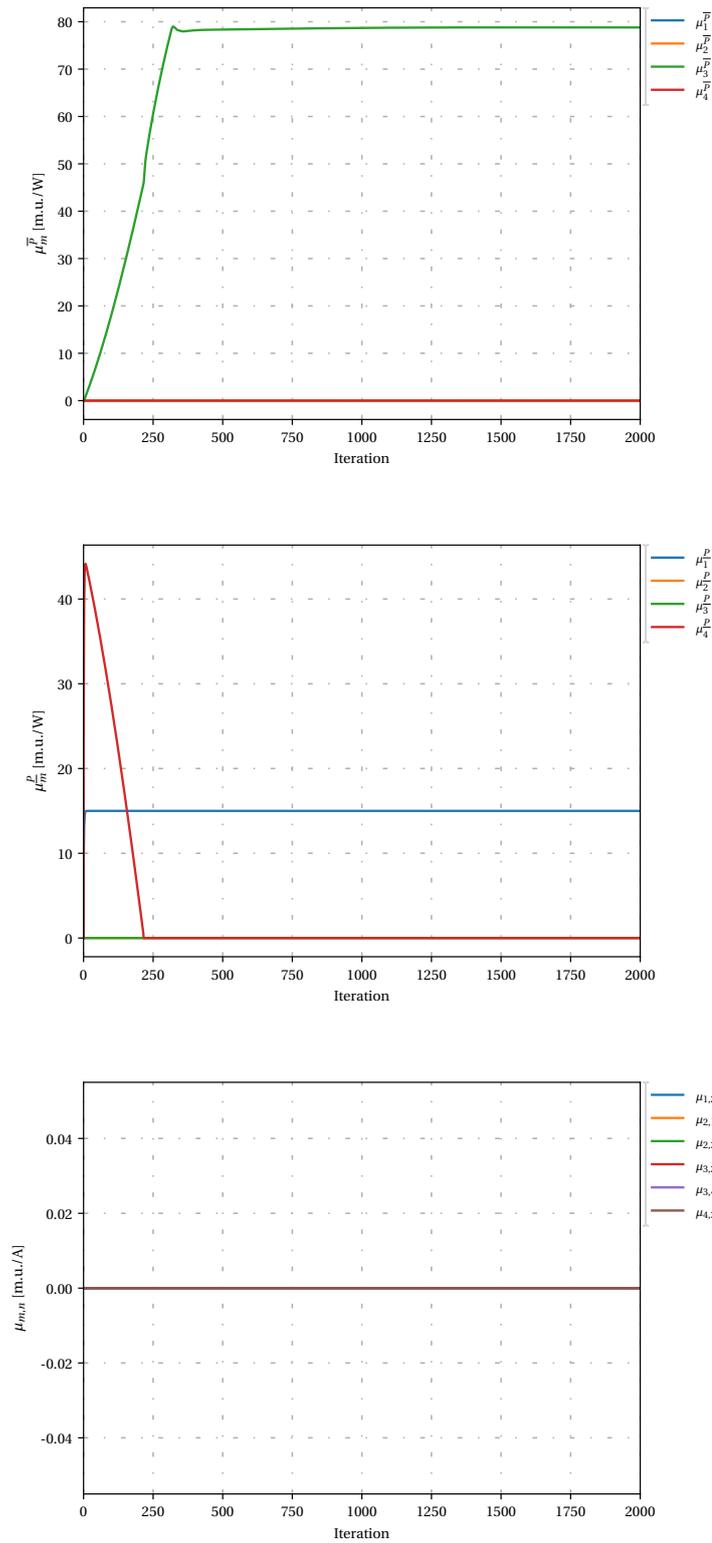


Figure A.6: From top to bottom: nodal values of maximum power limit dual variable, nodal values of minimum power limit dual variable and maximum current limit dual variables.

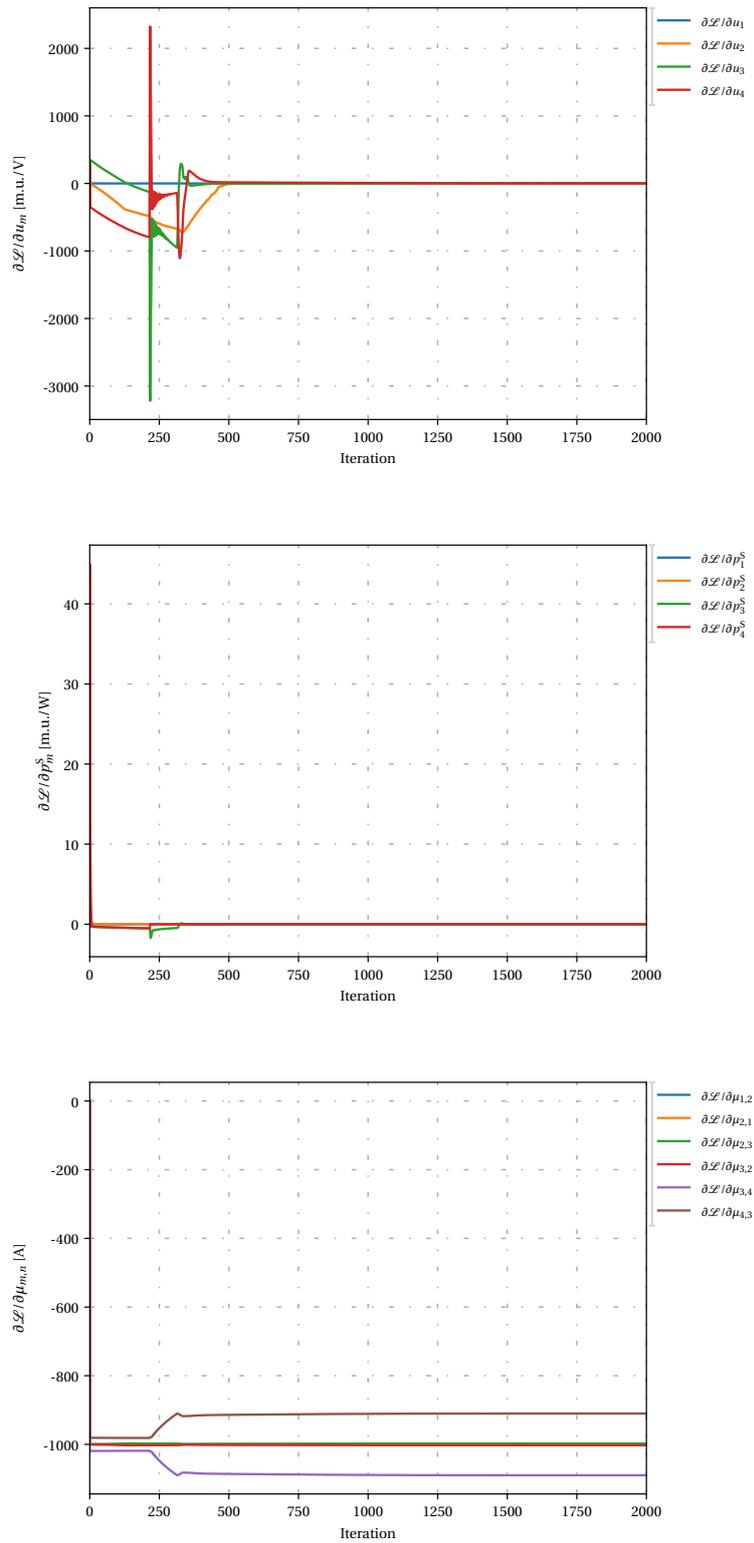


Figure A.7: From top to bottom: local derivative of the Lagrangian function in terms of voltage, local derivative of the Lagrangian function in terms of power and local derivative of the Lagrangian function in terms of the dual variable for the line current limits.

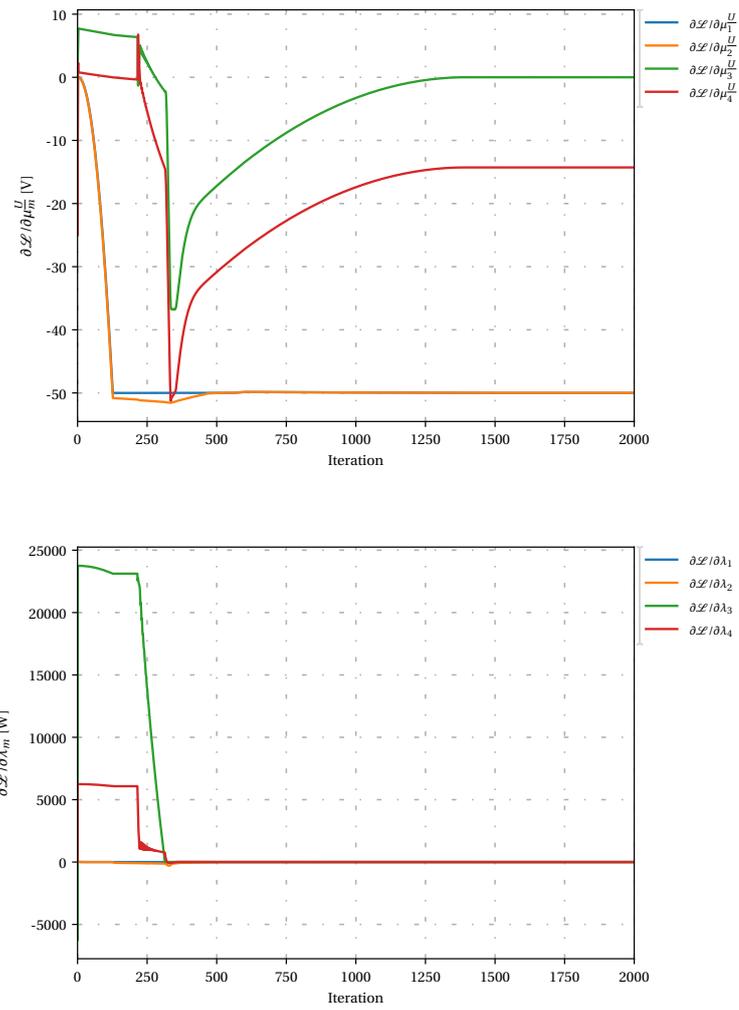


Figure A.8: From top to bottom: local derivative of the Lagrangian function in terms of the dual variable for the minimum voltage limit and local derivative of the Lagrangian function in terms of LMP.

A.3. Complete simulation 3 node grid subject to changes in power

The following plots are the results of the simulation of the network shown in 3.15.

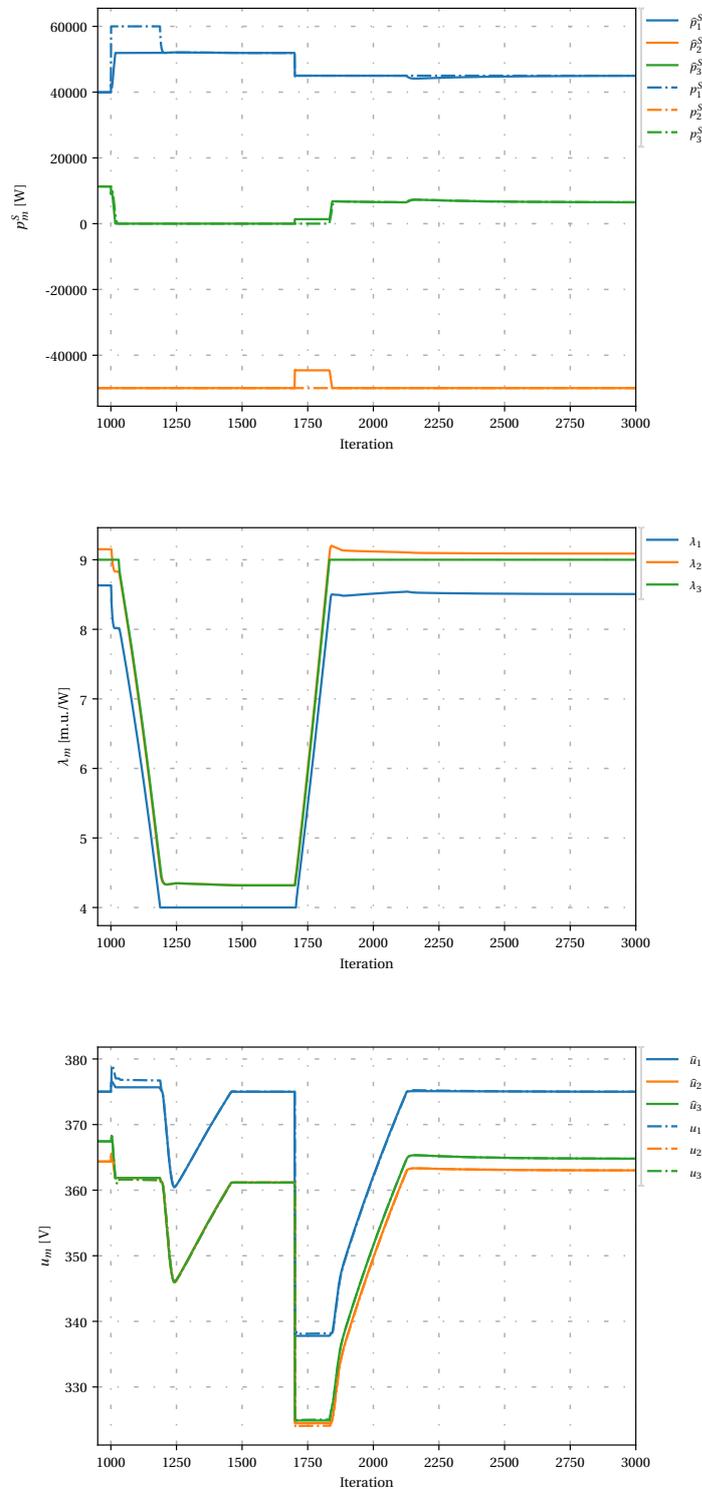


Figure A.9: From top to bottom: nodal values of local power measurements and setpoints, nodal values of LMP variable and nodal values of local voltage measurements and setpoints.

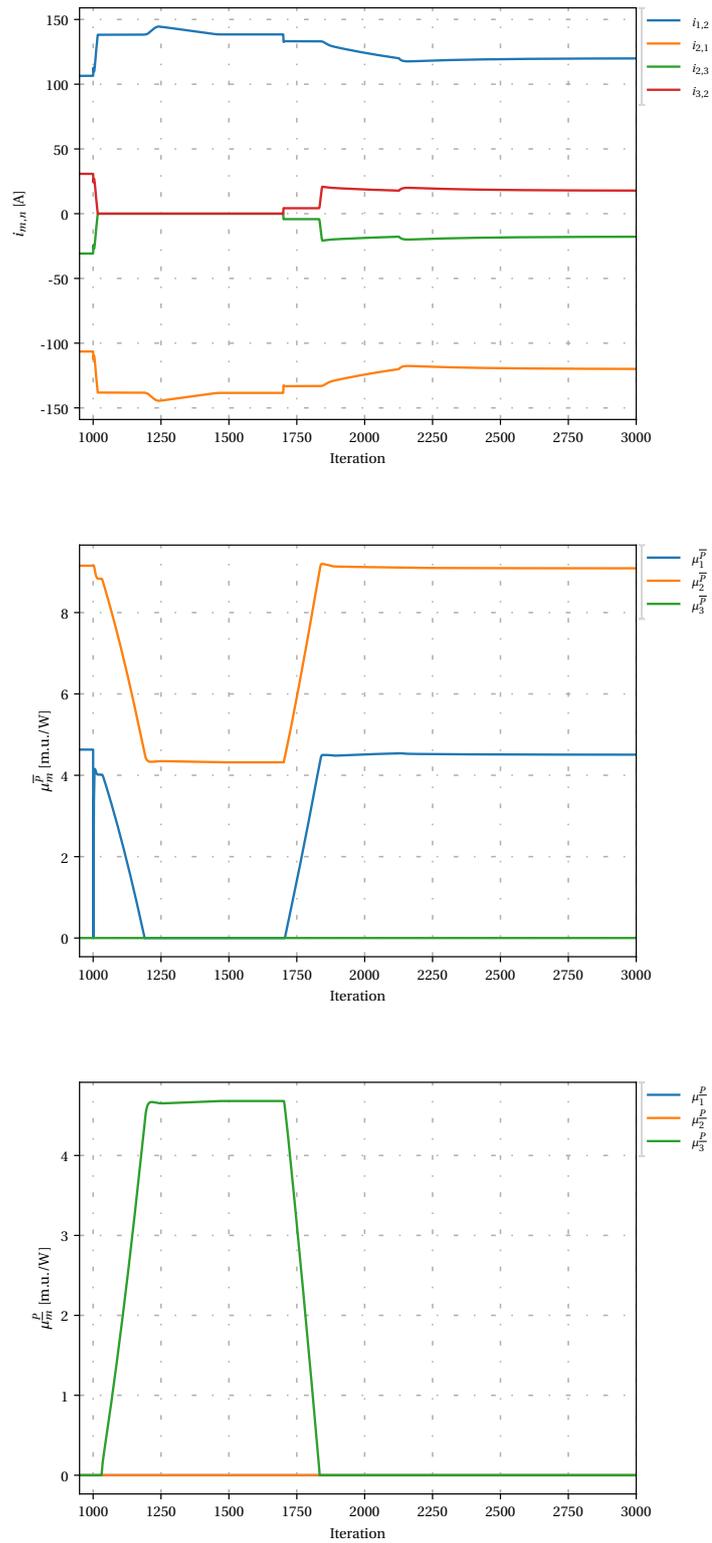


Figure A.10: From top to bottom: line current measurements, nodal values of maximum power limit dual variable and nodal values of minimum power limit dual variable.

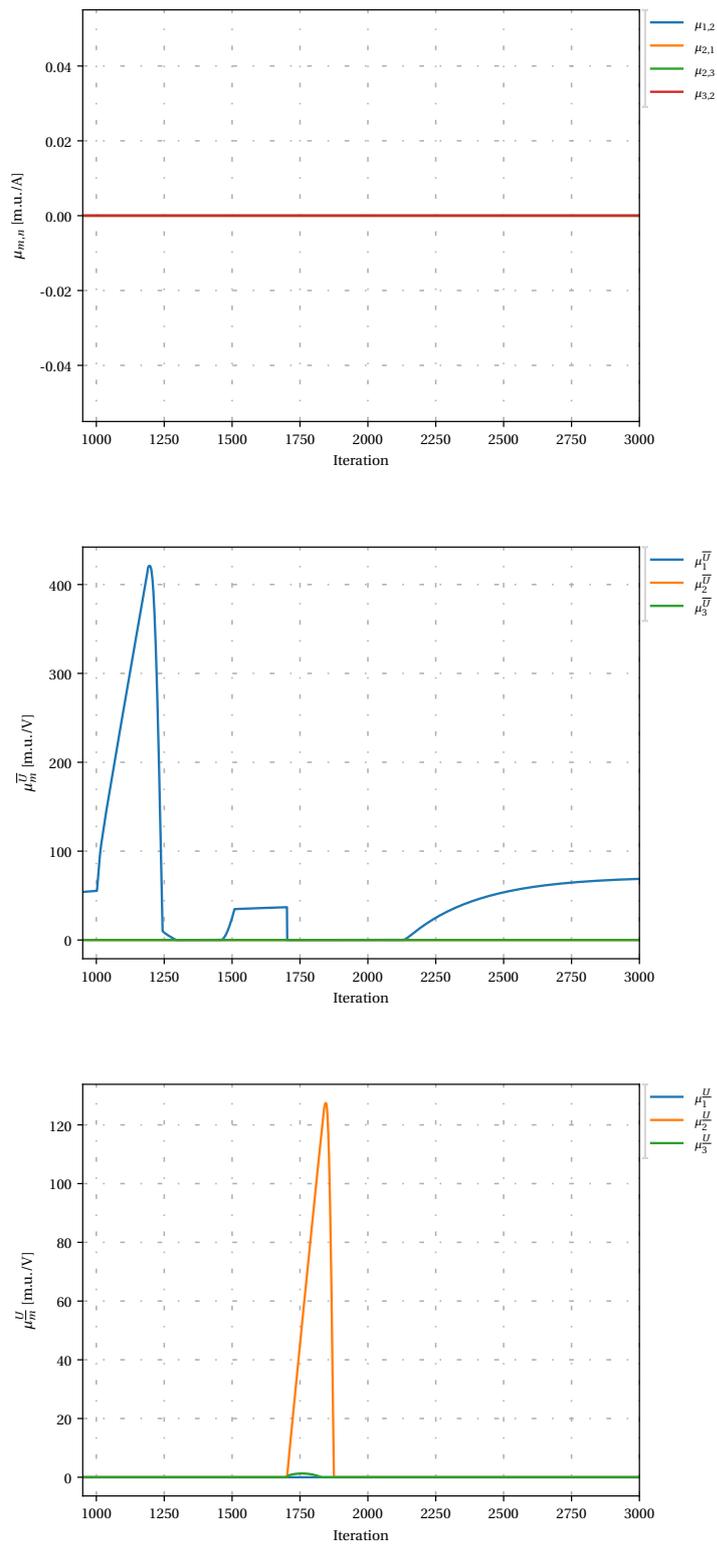


Figure A.11: From top to bottom: maximum current limit dual variables, nodal values of maximum voltage limit dual variable and nodal values of minimum voltage limit dual variable.

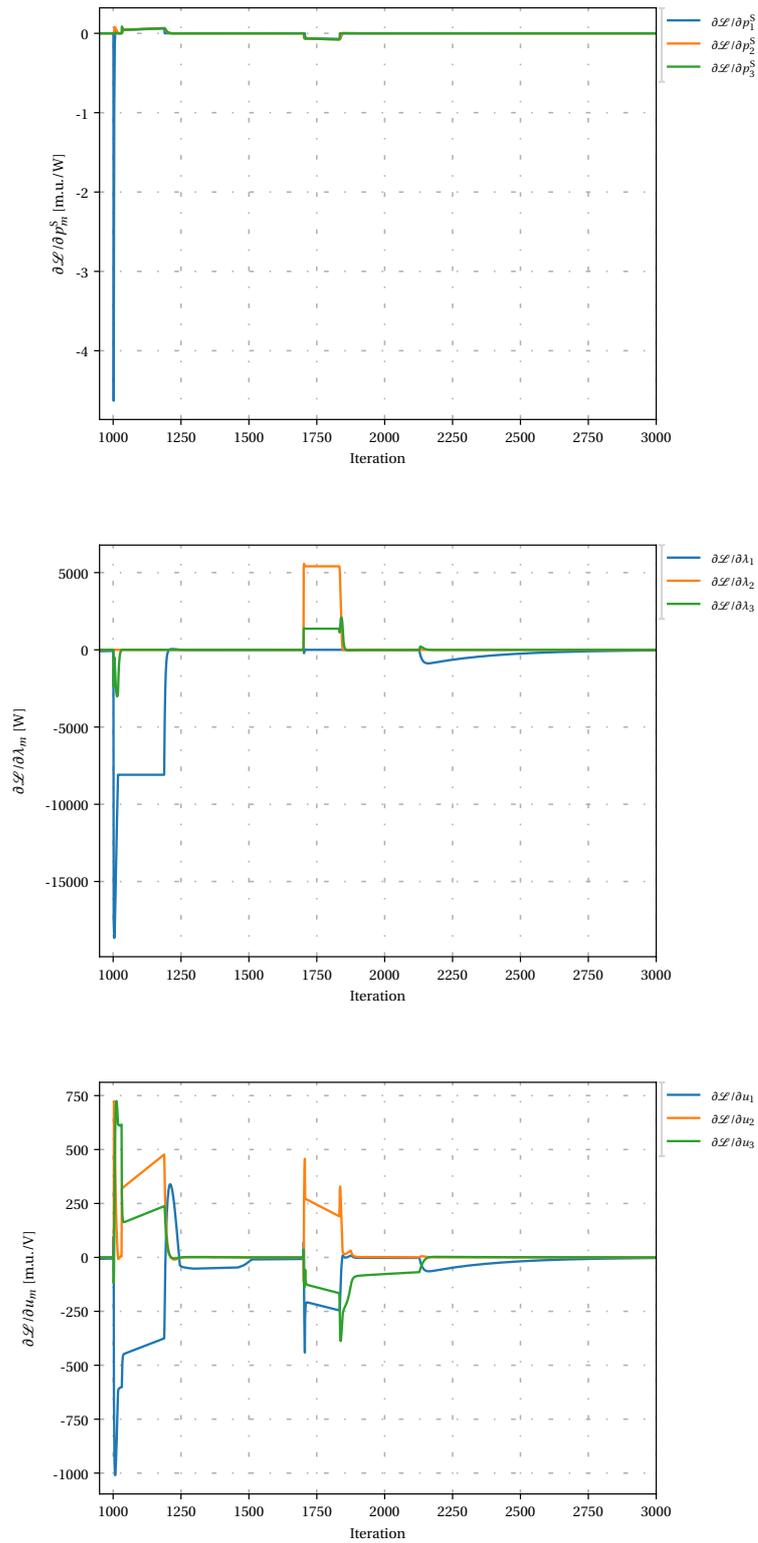


Figure A.12: From top to bottom: local derivative of the Lagrangian function in terms of power, local derivative of the Lagrangian function in terms of LMP and local derivative of the Lagrangian function in terms of voltage.

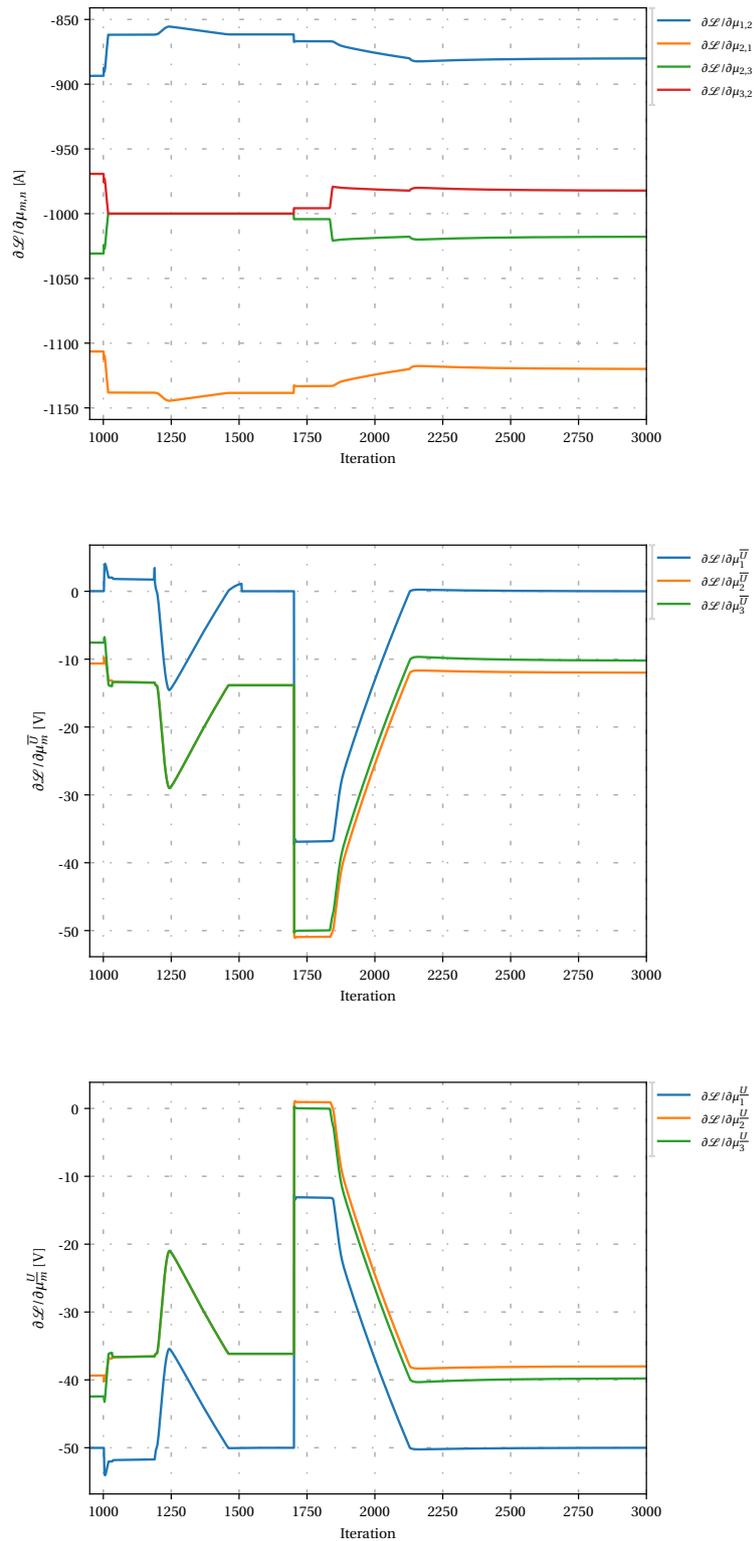


Figure A.13: From top to bottom: local derivative of the Lagrangian function in terms of the dual variable for the line current limits, local derivative of the Lagrangian function in terms of the dual variable for the maximum voltage limit and local derivative of the Lagrangian function in terms of the dual variable for the minimum voltage limit.

B

Schematics of Zoetermeer Lighting Grid



Figure B.1: Net schematics of the Zoetermeer grid and its cable layout, provided by the municipality of Zoetermeer

Bibliography

- [1] IEA, *Oecd-Iea*, Tech. Rep. March (International Energy Agency Global, 2018).
- [2] D. S. Philips and W. Warmuth, *Photovoltaics Report*, Tech. Rep. March (Fraunhofer Institute for Solar Energy Systems, 2019).
- [3] L. Mackay, T. Hailu, L. Ramirez-Elizondo, and P. Bauer, *Decentralized current limiting in meshed DC distribution grids*, *2015 IEEE 1st International Conference on Direct Current Microgrids, ICDCM 2015*, 234 (2015).
- [4] T. Faulwasser, A. Engelmann, T. Mühlpfordt, and V. Hagenmeyer, *Optimal power flow: An introduction to predictive, distributed and stochastic control challenges*, *At-Automatisierungstechnik* **66**, 573 (2018), [arXiv:arXiv:1811.01163v1](#).
- [5] D. Kirschen and G. Strbac, *Fundamentals of Power System Economics*, 1st ed. (John Wiley & Sons, Inc., Chichester, West Sussex, England, 2005).
- [6] W. T. Huang, K. C. Yao, and C. C. Wu, *Using the direct search method for optimal dispatch of distributed generation in a medium-voltage microgrid*, *Energies* **7**, 8355 (2014).
- [7] M. Godoy Simões, C. S. Uriarte, S. Chakraborty, and F. A. Farret, *Cost considerations on fuel cell renewable energy systems*, *Conference Record - IAS Annual Meeting (IEEE Industry Applications Society)* **5**, 2169 (2006).
- [8] S. A. Alavi, A. Ahmadian, and M. Aliakbar-Golkar, *Optimal probabilistic energy management in a typical micro-grid based-on robust optimization and point estimate method*, *Energy Conversion and Management* **95**, 314 (2015).
- [9] G. Hamoud and I. Bradley, *Assessment of transmission congestion cost and locational marginal pricing in a competitive electricity market*, *IEEE Transactions on Power Systems* **19**, 769 (2004).
- [10] P. L. Lewin, J. Altus, G. Amaratunga, R. Belmans, J. Blom, H. Frank, L. Haarla, M. O'Malley, V. Radziukynas, M. Sterling, V. Trenev, H. J. Wagner, and J. Holmes, *European Academies Science Advisory Council*, Tech. Rep. May (European Academies Science Advisory Council, 2009).
- [11] J. J. Grainger and W. D. Stevenson, *Power System Analysis* (McGraw-Hill, Inc., 1994) p. 787.
- [12] H. Yuan, F. Li, Y. Wei, and J. Zhu, *Novel linearized power flow and linearized OPF models for active distribution networks with application in distribution LMP*, *IEEE Transactions on Smart Grid* **9**, 438 (2018), [arXiv:1705.04118](#).
- [13] L. Mackay, R. Guarnotta, A. Dimou, G. Morales-España, L. Ramirez-Elizondo, and P. Bauer, *Optimal power flow for unbalanced bipolar DC distribution grids*, *IEEE Access* **6**, 5199 (2018).
- [14] L. Mackay, N. H. der Blij, L. Ramirez-Elizondo, and P. Bauer, *Toward the Universal DC Distribution System*, *Electric Power Components and Systems* **45**, 1032 (2017).
- [15] A. Kargarian, J. Mohammadi, J. Guo, S. Chakrabarti, M. Barati, G. Hug, S. Kar, and R. Baldick, *Toward Distributed/Decentralized DC Optimal Power Flow Implementation in Future Electric Power Systems*, *IEEE Transactions on Smart Grid* **9**, 2574 (2018).
- [16] S. Karambelkar, *Distributed Optimal Power Flow in a DC Distribution System*, *Master's thesis*, Delft University of Technology (2017).
- [17] H. Wang and J. Huang, *Incentivizing Energy Trading for Interconnected Microgrids*, *IEEE Transactions on Smart Grid* **9**, 2647 (2018).

- [18] S. Chakrabarti, M. Kraning, E. Chu, R. Baldick, and S. Boyd, *Security Constrained Optimal Power Flow via proximal message passing*, [2014 Clemson University Power Systems Conference, PSC 2014](#), 1 (2014).
- [19] W. Karush, *Minima of Functions of Several Variables with Inequalities as Side Conditions*, Master's thesis, University of Chicago (1939).
- [20] H. W. Kuhn and A. W. Tucker, *Nonlinear programming*, in [Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability](#) (University of California Press, Berkeley, Calif., 1951) pp. 481–492.
- [21] J. Huang, Z. Li, and Q. H. Wu, *Fully decentralized multiarea reactive power optimization considering practical regulation constraints of devices*, [International Journal of Electrical Power and Energy Systems](#) **105**, 351 (2019).
- [22] W. Lu, M. Liu, S. Lin, and L. Li, *Fully Decentralized Optimal Power Flow of Multi-Area Interconnected Power Systems Based on Distributed Interior Point Method*, [IEEE Transactions on Power Systems](#) **33**, 901 (2017).
- [23] S. Kar and J. M. Moura, *Consensus + innovations distributed inference over networks: cooperation and sensing in networked systems*, [IEEE Signal Processing Magazine](#) **30**, 99 (2013).
- [24] J. Mohammadi, G. Hug, and S. Kar, *Fully distributed DC-OPF approach for power flow control*, [IEEE Power and Energy Society General Meeting 2015-Septe](#) (2015), 10.1109/PESGM.2015.7285770.
- [25] J. Mohammadi, S. Kar, and G. Hug, *Distributed Approach for DC Optimal Power Flow Calculations*, [2015 IEEE Global Conference on Signal and Information Processing \(GlobalSIP\)](#), 1121 (2014), arXiv:1410.4236 .
- [26] S. Karambelkar, L. Mackay, S. Chakraborty, L. Ramirez-Elizondo, and P. Bauer, *Distributed Optimal Power Flow for DC Distribution Grids*, in [IEEE Power and Energy Society General Meeting](#), Vol. 2018-Augus (2018).
- [27] D. M. Dolaputra, *Fully Distributed Optimal Power Flow*, Master's thesis, Delft University of Technology (2019).
- [28] K. Van Den Bergh, E. Delarue, and W. D'haeseleer, *DC power flow in unit commitment models*, (2014), working paper.
- [29] G. Van Rossum and F. L. Drake Jr, *Python tutorial* (Centrum voor Wiskunde en Informatica Amsterdam, The Netherlands, 1995).
- [30] L. Thurner, A. Scheidler, F. Schäfer, J. Menke, J. Dollichon, F. Meier, S. Meinecke, and M. Braun, *pan-dapower an open-source python tool for convenient modeling, analysis, and optimization of electric power systems*, [IEEE Transactions on Power Systems](#) **33**, 6510 (2018).
- [31] E. Jones, T. Oliphant, P. Peterson, et al., *SciPy: Open source scientific tools for Python*, (2001–), [Online].
- [32] W. McKinney et al., *Data structures for statistical computing in python*, in [Proceedings of the 9th Python in Science Conference](#), Vol. 445 (Austin, TX, 2010) pp. 51–56.
- [33] J. D. Hunter, *Matplotlib: A 2d graphics environment*, [Computing in science & engineering](#) **9**, 90 (2007).
- [34] D. Asija, P. Choudekar, K. M. Soni, and S. K. Sinha, *Power flow study and contingency status of wscs 9 bus test system using matlab*, [2015 International Conference on Recent Developments in Control, Automation and Power Engineering \(RDCAPE\)](#) (2015), 10.1109/rdcape.2015.7281420.
- [35] J. Mohammadi, G. Hug, and S. Kar, *Asynchronous distributed approach for dc optimal power flow*, [2015 IEEE Eindhoven PowerTech](#), 16 (2015).
- [36] Philips, *Mithra LED*, (2019).
- [37] Nexans, *VG-YMvKas Dca / VG-YMvKas Dca Flex*, (2019).