

DECENTRALIZED OPTIMAL POWER FLOW FOR LOW VOLTAGE DC GRIDS

AN ALGORITHM FOR ONLINE OPTIMIZATION ON A PHYSICAL ENVIRONMENT

by

Thalis Papakyriakou



DECENTRALIZED OPTIMAL POWER FLOW FOR LOW VOLTAGE DC GRIDS

AN ALGORITHM FOR ONLINE OPTIMIZATION ON A PHYSICAL ENVIRONMENT

by

Thalis Papakyriakou

in partial fulfillment of the requirements for the degree of

Master of Science

in Computer Science

at the Delft University of Technology,

to be defended publicly on Tuesday August 31, 2021 at 03:00 PM.

Supervisor:	prof. Dr. Mathijs de Weerd, TU Delft
Advisor:	Dr. Ir. Laurens Mackay, DC Opportunities R&D BV
Thesis Committee:	prof. Dr. Mathijs de Weerd, TU Delft prof. Dr. Laura Ramirez Elizondo, TU Delft Dr. Ir. Laurens Mackay, DC Opportunities R&D BV

This thesis is confidential and cannot be made public until August 31, 2023.

An electronic version of this dissertation is available at

<http://repository.tudelft.nl/>.

ACKNOWLEDGEMENTS

I would like to thank my Thesis supervisor, Dr. Mathijs, for the consistent and in-depth support he has been offering me throughout my whole Thesis. I wouldn't switch that amount of devotion with any less-strict or easy-going professor. Speaking of devotion, I would also like to thank my advisor and CEO of DC Opportunities, Dr. Laurens. I often heard of CEOs being distant to their employees because of the workload, and Dr. Laurens was, surprisingly, the exact opposite, always having time for whatever you needed of him. Last but not least, I would like to thank all the DC Opportunities employees, with Faik and Samad in particular. While previous colleagues of mine often tried to avoid responsibility and focus on their own personal matters, the buddies at DC Opportunities were always inexplicably willing to sacrifice their own personal time and progress to help me, and each other, with whatever help was needed. I left that company with a completely new definition for the term "working environment".

I wish to express my most appreciation and deepest of gratitude to my parents. I would have never made it to this point without their support. I never quite understood how it is possible for one person to sacrifice almost their entire life for another, but I know where to look if I ever truly want to understand that.

Shout-out to my Otouto-san, the man of few words, always willing to physically support me on my journey. Shout-out to my Imouto-san, the creative genius that never loses her cool with me. Shout-out to my friends, Andreas, Markos, Xristos, Frixos, Sotiris, Milena, and a few others, for always being there for me, and for constantly reminding me what it really feels like to be amongst good company, and what there is to miss when they are not there. The final shout-out goes to my student-friends, Andrea, Aditya and Francisco, who made my time in the Netherlands so much more enjoyable, and any struggle I was going through so much more manageable.

Closing, I want to honourably mention my biggest inspiration, that made me who I am, and allowed me to get to this point in my life: Solid Snake, Naruto, the heroes who died so that I can live free, and David Eving Goggins. In the words of Goggins:

"I have lived a life of facing discomfort on a daily basis, and by doing that, I no longer want for anything; no material thing can give me the peace I found within myself. [...] My biggest fear was not living up to my full potential. Suppose I did not. I visualized that, when I died, there is this long line to get into Heaven, and God is interviewing all of us, having a chart with our names on it. So my turn comes, I sit down, and God says 'take a look at this chart'. And it's the chart of the life I was supposed to live. On that chart, it has me at 185 pounds, it has me being a Navy Seal, it has me being an Army Ranger, ran these races, broken these records, being a motivator and changing people's lives. And I look at this chart, and God says 'This is the life that you were supposed to have'. And I missed the mark, by a lot. I didn't have that life because I was not willing to suffer, to go into the

dungeon of my own soul to find more of myself. So, when you get to Heaven, are you really in Heaven? Or are you in Hell?..."

-Thalis Papakyriakou

August 2021

ABSTRACT

The DC Optimal Power Flow (DC-OPF) problem is a widely-studied topic in the field of power systems. A solution to the problem consists of minimizing the running costs of the power system, through defining the optimal operating state for each entity in the system, while adhering to a set of physical constraints. A lot of research has been conducted on decentralized and distributed solutions to this problem, which, when compared to centralized solutions, offer benefits such as adaptability, reliability and scalability. Nevertheless, most of these solutions have only been evaluated through simulations, while physical applications of these algorithms introduce new challenges, such as noise, delays, and the regulation of physical variables like voltage and current. In this thesis, we focus on a decentralized DC-OPF algorithm based on the Consensus and Innovation method, where system entities utilize their physical measurements and communicate with their neighbouring entities in order to reach consensus on a solution. While previous implementations of the algorithm were tested on simulated environments, this thesis explores and proves the effectiveness of the algorithm implemented for a real DC unipolar microgrid, consisting of power supplies, loads and a Power Circuit Board attached to each, where each device's behavior is governed by its own, exclusive entity. The newly-introduced challenges of a physical environment are accounted for, and any negative effects of them are mitigated as much as possible. Furthermore, the algorithm is successfully modified and extended to handle region DC-OPF, something that has not been attempted before, where a single entity of the algorithm could be responsible for many devices in the network. Finally, it is known that the original algorithm has not been experimented on before on scenarios of islanding and de-islanding, which are of importance in OPF, because a power fault may occur and one area may wish to isolate itself from the faulty area, or perhaps a distributed energy resource should only be connected to the network during certain periods, e.g., during sunlight for solar panels. Hence, this thesis also proves the effectiveness of the algorithm on scenarios of islanding and de-islanding, in an innovation site called the Green Village, where technologies in the field of sustainable energy provision are tested and applied in a real-life environment.

CONTENTS

Acknowledgements	v
Abstract	vii
1 Introduction	1
1.1 Renewable Energy and Distributed Energy Resources	1
1.2 Meshed Low Voltage Grids (microgrids)	1
1.3 Alternating and Direct Current	2
1.4 Research Motivation	2
1.5 Research Questions and Objectives	3
1.6 Contributions	4
1.7 Structure of the Thesis	4
2 Literature Review	7
2.1 The Optimal Power Flow problem	8
2.2 Decentralized solutions to the OPF problem	10
2.3 The fundamentals of the Consensus and Innovation algorithm	17
2.3.1 Karush-Kuhn-Tucker (KKT) conditions	17
2.3.2 Consensus and Innovation: the fundamentals	18
3 The Consensus and Innovation algorithm for Unipolar DC microgrids: Theory	21
3.1 Overview of the structure: entities and communications	21
3.2 Adapting the C+I algorithm for the DC-OPF problem	23
3.2.1 Formulating the KKT Optimality Conditions	23
3.2.2 Formulating the updates per iteration	27
3.3 Droop Control	35
3.3.1 The objective of droop control	36
3.3.2 Droop control per operation region	37

4	The Consensus and Innovation algorithm for Unipolar DC microgrids: Implementation and technical details	41
4.1	Physical Setup: Nodes and grid components	42
4.2	Inter-nodal communication	42
4.2.1	Neighbour discovery	42
4.3	Inter-layer communication	43
4.4	Diagram of implementation	43
5	Case Studies	47
5.1	One generator and one load.	48
5.2	Two generators and one load	56
5.3	Concluding Remarks	58
6	The Consensus and Innovation algorithm: A modified algorithm for region-based OPF	67
6.1	The modified algorithm.	68
6.1.1	The new optimization problem	68
6.1.2	The new strategy	71
6.2	Case Studies	73
6.3	Concluding Remarks	75
7	The Consensus and Innovation algorithm: Islanding and de-Islanding	81
8	Conclusion	91
	Epilogue	95
	Appendices	97
A	Fundamentals of Electricity.	97
A.1	Fundamental Laws of Electricity	97
A.2	The interaction between a load and a generator	97
	Bibliography	99

1

INTRODUCTION

1.1. RENEWABLE ENERGY AND DISTRIBUTED ENERGY RESOURCES

Renewable Energy (RE) refers to sources of energy that are not depletable, like wind and solar power. These energy sources have nowadays become increasingly popular, mostly because of the dangers that lie with the extended use of non-renewable ones, such as pollution and the phenomenon of Climate Change. The Paris Agreement of 2015 is one of the latest and most significant indicators that these issues are starting to receive global attention and consideration. Regardless of the dangers involved with the alternative, the use of RE also provides some direct benefits: instead of relying on a centralized power source controlled by the government, Renewable Energy Sources (RES) such as photovoltaic (PV) panels can be installed in and owned by private households, essentially turning had-been clients into producers themselves. PV panels are also becoming increasingly cheaper and more effective[19]. This shift gave rise to the phenomenon of Distributed Energy Resources (DER), bringing new capabilities and challenges to the table, that are expected to significantly affect the development of power grids[15]. Drawbacks related to DER usage include the lack of constant availability of certain power sources, line congestion, and a low load factor[15], which is a measure of the utilization rate or efficiency of electrical energy usage. These problems were to be expected, since DERs are privately owned and decentralized, thus harder to coordinate, and power generation is often dependent on weather conditions.

1.2. MESHED LOW VOLTAGE GRIDS (MICROGRIDS)

Some of the challenges introduced by the use of DERs are directly related to the infrastructure of Low Voltage (LV) distribution grids: traditionally, LV grids are arranged in a top-to-bottom fashion, meaning that power flows from high power areas to decreasingly lower power areas, i.e., from neighbourhoods to apartment blocks, and then to private households. This setups makes it easier to isolate certain areas in case of faults, or coor-

dinate the whole flow of power and congestion. By introducing DERs into the equation, we observe a shift in the "power dynamic" (pun-intended), as power can now also be generated in low power areas such as households, disrupting the original top-to-bottom model, and can flow in both directions, something that the traditional LV grid is not built to handle. Furthermore, the traditional LV grid is susceptible to the drawbacks mentioned regarding the use of DERs, such as line congestion and the availability of power sources.

A solution suggested towards resolving these issues is meshing the LV grid[12], i.e., change the infrastructure so that nodes connect directly, dynamically and non-hierarchically to as many other nodes as possible, cooperating with one another efficiently. Naturally, a meshed LV grid, also known as a microgrid, offers redundancy, which helps tackle the issues of line congestion, power availability and fault recovery. Nonetheless, the usage of a meshed LV grid presents its own challenges, due to the increased amount of control variables and complexity, especially when deciding between the use of Alternating or Direct Current.

1.3. ALTERNATING AND DIRECT CURRENT

While Alternating Current (AC) has been the traditional standard of power transmission, especially considering long-distance transmission, Direct Current (DC) systems have lately seen an increase in popularity and usage, largely due to the introduction of DERs into the grid. For instance, a lot of RES such as PV panels and wind turbines already operate on DC, and a DC grid would eliminate the need for current conversion and the related losses. Furthermore, in DC, current flow only depends on the voltage level, in contrast to AC where the voltage angles also matter, with the latter making the whole meshing process more of a challenge. Finally, DC offers benefits such as increased power-transfer capacity and the elimination of reactive power considerations. Thus, DC proves to be a considerable and viable choice for the design of meshed LV grids[12].

1.4. RESEARCH MOTIVATION

As mentioned in Section 1.2, meshed LV grids present certain challenges with coordinating the power flow, due to their increased complexity and the fact that power flows in both directions. Formally, this problem is referred to as the Optimal Power Flow problem (OPF), or DC-OPF, if specifically referring to DC grids or DC approximations of AC grids, and it is a widely-studied topic in the field of power systems. A solution to the problem consists of minimizing the running costs of the power system, through defining the optimal operating state for each entity in the system, while adhering to a set of physical constraints. A lot of this research has been conducted on decentralized and distributed solutions to this problem[2], which, when compared to centralized solutions, offer benefits such as adaptability, reliability and scalability. One of these decentralized solutions is called the Consensus and Innovation (C+I) algorithm, which has been studied by various scientists under different assumptions and problem formulations[21][10][20]. In its essence, the C+I algorithm utilizes the Karush-Kuhn-Tucker Theorem to establish a set

of conditions that each sub-problem needs to satisfy, in order for optimality and convergence to be achieved.

While the C+I algorithm is relatively simple to apply in practise, there exists no work of it being tested in a real, physical microgrid. The results of prior work on the algorithm have been demonstrated through simulations, however, the transition from a simulated to a physical environment is not trivial, as it involves factors that are not accounted for in the simulated one, such as noise, delays, and the regulation of physical variables like voltage and current. An interesting version of the simulated algorithm can be found in the work of Pedro Parreira[17], with its innovative aspect being that it incorporates physical measurements and droop control, effectively eliminating a lot of computational requirements and boosting performance. Thus, it would be interesting to see how this version of the algorithm can be adjusted for a physical microgrid, and how it performs on it.

The C+I algorithm can hypothetically be implemented for both nodal and region OPF[2], with nodal OPF being the decentralized solution where each sub-problem is concerned with exclusively one device, and region OPF being the decentralized solution where each sub-problem can be concerned with multiple devices. Parreira's version of the algorithm is exclusively applicable to nodal OPF, and no attempts have been made to extend the solution, along with its performance benefits, towards region OPF. Hence, it would be of significance to see how the algorithm can be extended to be applicable to region OPF, and how it performs at it.

Finally, the C+I algorithm has not been experimented on before on scenarios of islanding and de-islanding, i.e., in the case a connection between two devices is lost/removed and their number of neighbours is reduced by one, or the opposite. Scenarios like these are of importance in OPF, because a power fault may occur and one area may wish to isolate itself from the faulty area. It might also be the case that a DER should only be connected to the network during certain periods, e.g., during sunlight for solar DERs. Hence, ensuring that the algorithm can handle such scenarios of islanding and de-islanding can prove of great utility to various physical setups running on the specific algorithm.

1.5. RESEARCH QUESTIONS AND OBJECTIVES

The aforementioned goals can be summarized and divided into the following research questions:

1. How is the C+I algorithm to be implemented for a physical environment where, compared to the simulated one, various additional factors like noise, delays, and voltage/current regulation come into play? What design choices and improvements can be performed in order to eliminate the negative effects of these additional factors?
2. How does the algorithm perform, given these extra factors, in terms of convergence time and overall execution speed, which are the traditional metrics the algorithm is evaluated on?

3. How can the original, nodal-OPF C+I algorithm be extended for region-OPF, while maintaining its performance regarding metrics such as optimality, convergence time and execution speed?
4. How does the C+I algorithm perform on scenarios of islanding and de-islanding?

1.6. CONTRIBUTIONS

In this Thesis, we present our implementation of the C+I algorithm for a physical microgrid, which we evaluate on small-scale networks of interconnected devices: generators and loads. The theoretical aspect of our implementation is based on the work of Pedro Parreira[17], performing any adjustments where deemed necessary. Our design choices ensure that we achieve our expectations of a fast, effective and efficient algorithm, across all the aforementioned performance metrics and eliminating a lot of the negative effects that burden a physical operational environment.

We also present and evaluate a modified version of the C+I algorithm that is appropriate for region OPF, while maintaining its performance regarding metrics such as optimality, convergence time and execution speed.

Finally, by using the appropriate components and by extending the capabilities of the algorithm, we verify that our C+I algorithm can handle scenarios of islanding and de-islanding. The grander experiment takes place in the Green Village, an innovation site where technologies in the field of sustainable energy provision, such as ours, are tested and applied in a real-life environment.

1.7. STRUCTURE OF THE THESIS

This Thesis is divided into 8 chapters.

Chapter 2 comprises the literature review, where the OPF problem is formulated and various decentralized solutions to it are presented. The fundamentals of our C+I algorithm are also presented, for the purposes of theoretical comparison with the other solutions.

Chapter 3 describes the C+I algorithm in detail, focusing only on its theoretical aspect, while Chapter 4 dives into the technical and implementation details in order for this algorithm to be applied to a physical environment.

Chapter 5 presents the case studies performed with our algorithm, along with the resulting observations.

Chapter 6 describes the modified version of the C+I algorithm, for handling region-OPF. Again, case studies are performed, and the effectiveness of the algorithm is evaluated.

Chapter 7 evaluates the effectiveness of the C+I algorithm on scenarios of islanding and de-islanding, with the grander of experiments taking place at the Green Village.

Finally, Chapter 8 summarizes the conclusions of our work and the answers to our

established research questions. Future work is also discussed.

2

LITERATURE REVIEW

As part of the literature review, previously established solutions to the OPF problem will be examined. This allows for comparisons to be made between our implemented C+I algorithm and the presented algorithms, in terms of methodology, effectiveness and advantages. A comparison between physical implementations of the algorithms would be preferred, but since there is a very limited and insufficient amount of physical experiments to be found in the literature, we settle for a comparison between the structure of each of the algorithms.

Section 2.1 begins by explaining the physical factors that govern the formulation of the OPF problem, and specifies the assumptions made for the purposes of this thesis. Having provided the generic problem definition, Section 2.2 presents a wide variety of decentralized and distributed approaches, which are closely related to the Consensus and Innovation (C+I) algorithm used in this Thesis. Finally, in Section 2.3, the fundamentals of the C+I method are presented, serving as a backbone for the upcoming description of the adapted C+I algorithm in the next chapter. A comparison between the C+I algorithm and the presented ones, based on the aforementioned set of criteria, takes place as the concluding part of this section.

2.1. THE OPTIMAL POWER FLOW PROBLEM

Assume an electric network of devices, connected to each other through transmission buses. These devices can be either power generators, e.g. power supplies and photovoltaics, or power consumers, also referred to as loads, e.g. light bulbs. A visual representation can be seen in Figure 2.1. In its essence, the goal of the Optimal Power Flow (OPF) problem is to minimize the running costs of the network while making sure that the power demand of the loads is satisfied, and while adhering to the physical restrictions of the network. Hence, OPF can be formulated as an optimization problem where the objective function represents the minimization of costs, while the constraints constitute the physical network restrictions and the problem requirements.

The literature has an abundance of OPF model formulations, based on both physical factors and modelling factors. A few examples of physical factors include the consideration (or not) of transmission losses, the consideration (or not) of bus voltage angles, system polarity, and so on. The most prevalent physical factor amongst them all is the type of power flow adopted: alternating current (AC) or direct current (DC). Let us first determine the type of OPF problem this thesis is focused on.

As mentioned in section 1.3, AC has been the dominant form of electrical power transmission and distribution, thus, a lot of the existing literature focuses on AC grids[7][9][3]. In this thesis, however, we focus on DC grids. One of the advantages of focusing on DC grids compared to AC grids, as presented in section 1.3, is the absence of reactive power, and consequently of any related equations in the model. DC grids can also neglect bus voltage angles, which affect the amount of outgoing power in AC grids, whereas in DC grids the current flow depends only on the voltage level[17]. Hence, bus voltage angles are also ignored in our problem definition. Furthermore, we assume a unipolar grid, as this offers less implementation complexity, while suffering drawbacks such as higher current flow requirements. This trade-off is appropriate, given the nature of this

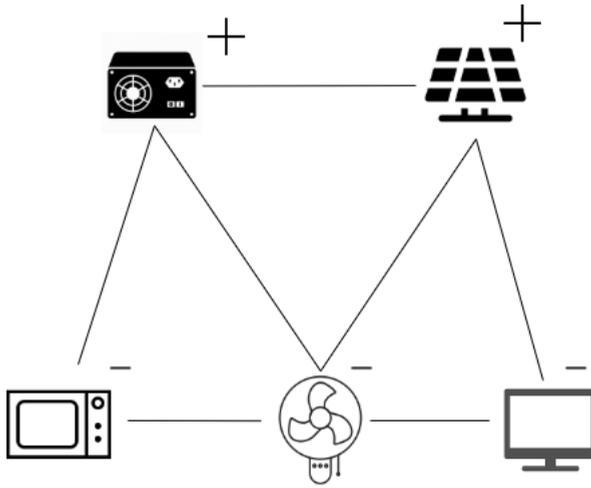


Figure 2.1: An abstract representation of an electric network

implementation of the algorithm and the research questions. Finally, we employ Exact-OPF[18], meaning that the transmission losses are taken into consideration, as opposed to Lossless-OPF[10][25]. From this point onward, whenever we refer to the OPF problem, it will be under the above assumptions, unless stated otherwise. Having defined the type of OPF problem we will be focusing on, let us examine the aspects of the optimization problem in more detail.

The running costs of the network can be described as the summation of the costs for each device in the network to produce/consume a certain amount of power. Each device has its own cost function, which determines the costs given the required amount of power. The form of this function depends on the device itself, for example, gas turbine and diesel-fuelled generators usually have a function of quadratic nature[24], while fuel-cell devices tend to have linear functions[13]. Hence, the objective function of the OPF problem can be defined as follows

$$\min_p \sum_{m \in N} A_m p_m^2 + B_m p_m \quad (2.1)$$

where parameters A_m and B_m are, respectively, the quadratic and linear polynomial coefficients of the cost function of each device m , out of the total number of devices N . Decision variable p_m is the power generated/consumed by the generator/load m .

Regardless of the specific type of OPF problem investigated, the constraints representing the physical network restrictions always need to preserve some fundamental properties and laws of electric networks. Namely, they need to satisfy Ohm's law for the current-voltage proportional relationship, Kirchhoff's laws for power balance at the buses, current flow limits for transmission buses, and power and voltage generation limits for generators. Given the type of OPF this thesis is focused on, the above constraints

can be formulated as follows

$$p_m = u_m \cdot i_m^s = u_m \sum_{n \in \Omega_m} i_{mn} = u_m \sum_{n \in \Omega_m} G_{mn}(u_m - u_n), \quad \forall m, n \in N \quad (2.2)$$

$$i_{mn} = \sum_{n \in \Omega_m} G_{mn}(u_m - u_n) \leq \bar{I}_{mn}, \quad \forall m, n \in N \quad (2.3)$$

$$\underline{P}_m \leq p_m \leq \bar{P}_m, \quad \forall m \in N \quad (2.4)$$

$$\underline{U}_m \leq u_m \leq \bar{U}_m, \quad \forall m \in N \quad (2.5)$$

where N is the set of all devices on the network. Notations p, u and i are the standard notations for power, voltage and current. While p_m would denote the power generated at device m , notation i_m^s denotes the current passing through device m , with s making a clearer distinction between this notation and the current i_{mn} passing through bus (m, n) . The power and voltage upper and lower limits of device m are denoted by $\bar{P}_m, \underline{P}_m, \bar{U}_m, \underline{U}_m$, and the maximum current limit for bus (m, n) is denoted by \bar{I}_{mn} . Finally, Ω_m represents the set of the physically connected neighbours of device m , and G_{mn} represents the conductance of bus (m, n) . For a better insight on how the above equations are derived from Ohm's and Kirchhoff's laws, please refer to Appendix Section A.1. Additionally, a visual representation of the relationship between the above variables can be seen in Figure 2.2.

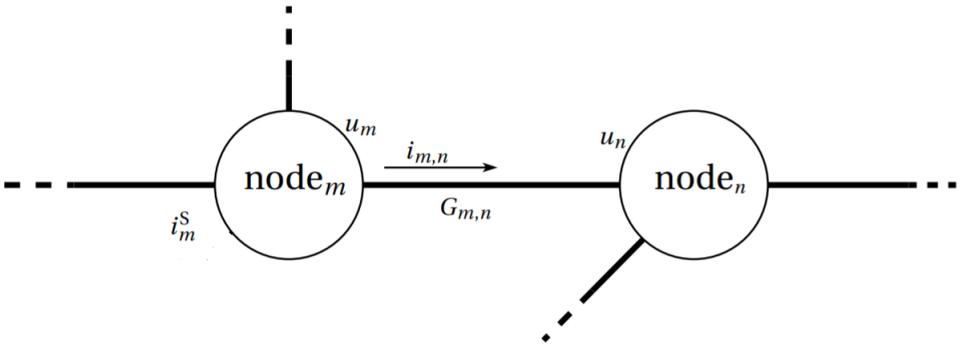


Figure 2.2: A visual representation of the relationship between variables u , i and G , as presented in the generic problem definition.

2.2. DECENTRALIZED SOLUTIONS TO THE OPF PROBLEM

Compared to centralized solutions, decentralized OPF solutions offer various benefits: they allow for different handling of each device in the network, and thus allow for diversity amongst the selected devices. They also allow for the computational requirements

to be divided and distributed. Yet another benefit of decentralized OPF solutions is the safety they provide against cyber-attacks, and the possibility of single-area-failure without having the whole algorithm collapse. The decentralized solutions presented in this section serve as a great framework for comparison between our decentralized implementation and them.

To the best of our knowledge, the most recent, inclusive and comprehensive survey of distributed and decentralized OPF solutions is provided by Kargarian et. al[2]. The authors present six methods for OPF algorithms that are prevalent in the distributed and decentralized domain, namely analytical target cascading (ATC), optimality condition decomposition (OCD), alternating direction method of multipliers (ADMM), auxiliary problem principle (APP), consensus+innovation (C+I), and proximal message passing (PMP). They nicely present, in a hierarchical diagram, the way each method decomposes the problem, and they also present a table with the most important attributes of each method. Both the diagram and the table can be seen below in Figures 2.3 and 2.4. Before proceeding to describe the methods, let us first mention a few things about the important attributes and properties that define them.

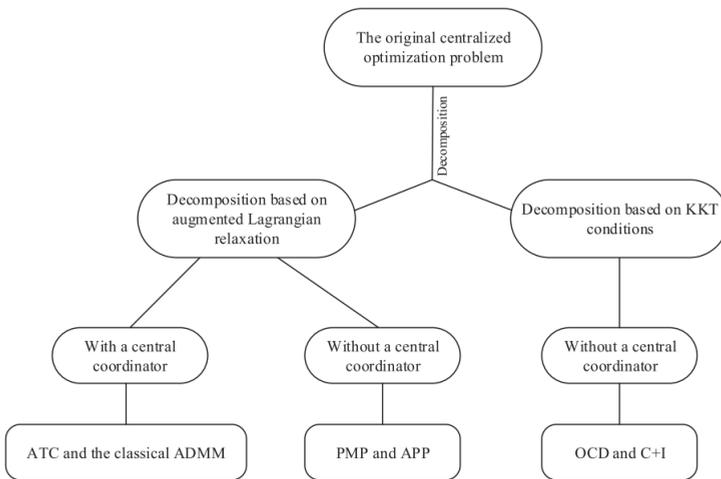


Figure 2.3: A diagram demonstrating how each distributed/decentralized OPF method decomposes the original problem[2]

The main difference between the OPF problem, as defined in section 2.1, and the OPF problem as approached by the aforementioned methods, is the consideration of bus voltage angles θ by the latter. These variables are included as part of the decision variables, which, in the original problem, consist of only the power variables (2.1).

Regarding the distinction between distributed and decentralized methods, the following is assumed[2]: In distributed approaches, there exists a central coordinator that coordinates the otherwise independent entities. Each entity only communicates with the coordinator, meaning there is no communication between the entities. In contrast,

TABLE II: Summary of the six algorithms for the DC-OPF application

Algorithm	Category	Foundation	Central coordinator	Shared information	Computational effort per iteration	Amount of data exchange per iteration	Stopping criteria
ATC	Distributed	Augmented Lagrangian relaxation	yes	voltage angles of border buses	high	low	mismatch between angles of border buses
ADMM	Distributed	Augmented Lagrangian relaxation	yes	voltage angles of border buses	high	low	mismatch between angles of border buses
PMP	Decentralized	Augmented Lagrangian relaxation	no	average node power mismatch and voltage angles, Lagrange Multipliers of node power mismatch and voltage angle inconsistency	low	high	Primal and dual residuals
APP	Decentralized	Augmented Lagrangian relaxation	no	voltage angles of border buses	high	low	mismatch between angles of border buses
OCD	Decentralized	KKT conditions	no	voltage angles and Lagrangian Multipliers of power balance constraints at border buses	high	medium	mismatch of KKT conditions
C+I	Decentralized	KKT conditions	no	voltage angles and Lagrangian Multipliers of power balance constraints at border buses	low	high	mismatch of power balance

Figure 2.4: A table providing information on the most important attributes of each distributed/decentralized OPF method[2]

decentralized methods do not involve a central coordinator, and the entities directly communicate and exchange information with their neighbouring entities.

As for the decomposition strategies, we have the "Augmented Lagrangian relaxation" (ALR) strategy and the "KKT conditions" strategy. ALR is used for solving constrained optimization problems, by decomposing the original problem into a series of unconstrained sub-problems, and adds a penalty term to their objective function in order to implicitly enforce constraints between the sub-problems. This penalty term includes the Lagrangian term λ , also known as the dual variable of the problem, as found in the derived Lagrangian function used to decompose the problem. Figure 2.5 demonstrates this decomposition. The "KKT conditions" strategy takes the Lagrangian function of the original (nonlinear) optimization problem and applies to it the "Karush–Kuhn–Tucker" theorem, which states that an optimal solution is obtained by identifying the saddle points in the Lagrangian function, i.e., the (non-local-extremum) points where the slopes/ derivatives in orthogonal directions are all zero. A more concrete presentation of the latter strategy is provided in section 2.3, as it is the decomposition strategy of the algorithm explored in this Thesis.

There are two strategies through which the OPF methods coordinate the sub-problems after the decomposition: coupling variables and coupling constrains. Under the former strategy, different sub-problems contain common variables but the constraint sets are separated, while in the latter strategy, the variable sets are separated but there are constraints that contain variables from two or more sub-problems.

The following subsections give an overview of the aforementioned methods and the way they relate to each other, in terms of approach, advantages, limitations etc. It is suggested that the subsections are read in order, as the first subsections introduce concepts that are briefly referred to in the following ones.

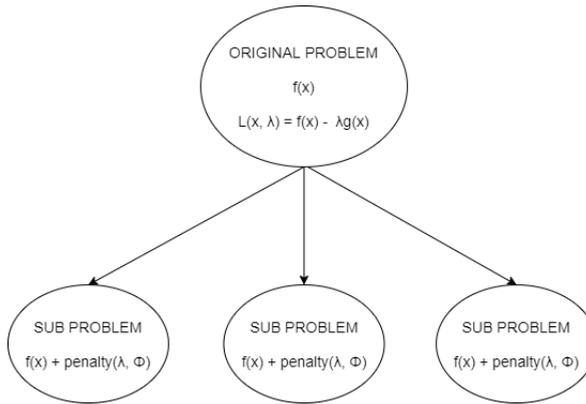


Figure 2.5: A demonstration of the idea of decomposition behind Augmented Lagrangian Relaxation, where we can observe each sub-problem containing a penalty term depended on the Lagrangian term λ and the coupling variables Φ

ANALYTICAL TARGET CASCADING AND ALTERNATING DIRECTION METHOD OF MULTIPLIERS

The ATC and ADMM are distributed methods involving a central coordinator. We start off by defining a region as a set of entities, that is connected to other regions through buses. Each region solves its own "slave" sub-problem (sub-problems are solved in parallel) and then informs the coordinator about the obtained values of some variables they have in common. The coordinator then solves the "master" problem, and reports back to the slave on the value of their common variables. Finally, the slave uses the reported values to perform its variable updates (note that this is a different step from the one of solving its sub-problem). The above process repeats iteratively, until convergence is achieved, i.e., until all pairs of master and slave reach consensus, to a certain degree, on the values of their common variables.

In the case of ATC, which is based on a multilevel hierarchical model, each entity can be both a slave and a master, depending on its level in the hierarchy, with any connected entities of lower levels being its slaves. In the case of ADMM, the problem is divided into "slave" regions, and all regions respond to a common "master" coordinator. Thus, ADMM can be seen as a two-level hierarchy. Figure 2.6 provides an indicative schematic.

For both methodologies, the common variables between master and slave, a.k.a the coupling variables, can be defined as the voltage angles of the buses connecting the two regions. These coupling variables are used within consistency constrains, which enforce equal values for these variables between master and slave, in order for convergence to be achieved. The consistency constrains are then applied to the objective function as the penalty term, previously defined in the context of ALR. Figure 2.7 gives a visual representation of the voltage angles and their role as coupling variables.

It is interesting to note that there is actually a whole family of ADMM-based methods, such as the classical ADMM, Global Forms Consensus Optimization, Proximal Message

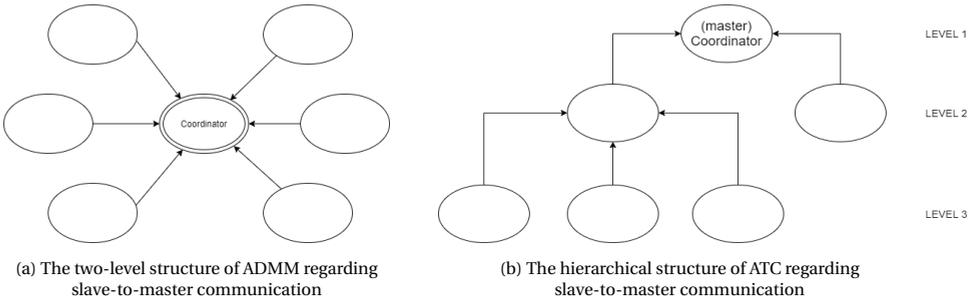
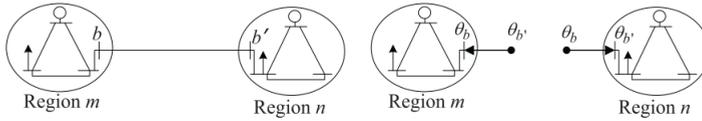


Figure 2.6

Figure 2.7: For two regions connected through bus (b, b') , the common variables θ_b and $\theta_{b'}$ are as depicted per region[2]

Passing etc. A peculiarity of the classical ADMM method is that it decomposes the originally obtained Lagrangian function, into minimized augmented Lagrangians per sub-problem, using the Gauss-Seidel method. At each Gauss-Seidel iteration, the minimization occurs with respect to the relevant primal variables used in the decomposition, and the dual variables of the problem are updated using the updated primal variables. On the other hand, the ATC method follows the traditional approach for decomposing the original problem using the Lagrangian function.

One advantage of the ATC and ADMM methods is that they exchange very little data per iteration, and this communication only happens between master and slave. More sub-problems equates more shared variables and subsequently higher computational effort, and thus, ATC and ADMM are well-suited for region-based OPF rather than nodal OPF. Regardless, for convex optimization problems, ATC and ADMM have been proven to converge to the optimal point[14][8]. An advantage of ATC over ADMM is that, because of its multilevel hierarchical structure, it is well-suited for power systems with multiple voltage or management levels. Additionally, it is a more flexible method than ADMM, the latter only allowing for a second-order penalty function. However, ADMM is less complex to implement.

Tarek Alskaf and Gijs van Leeuwen[1] propose a novel approach for replacing the central coordinator required by the aforementioned methods. They resort to using a blockchain technology known as smart-contracts. A smart contract is a piece of computer code that is deployed on the blockchain and can execute certain functions when called upon by other agents. In their implementation, parts of the algorithm are moved to the smart contract, and essential data that is needed by each region for optimization

is retrieved from the contract. The contract also ensures that all nodes operate simultaneously by giving permission to proceed with the next operation only after all regions have declared completion of the previous section. A drawback to this approach is created by the inherent built-in delay in the blockchain verification process, and therefore the authors suggest use of their algorithm for forecasting purposes, and not real-time optimization.

PROXIMAL MESSAGE PASSING

PMP is an ADMM-based decentralized OPF method, thus erasing the need for a central coordinator through message exchange between neighbours. Additionally, an entity in the network is no longer identified as a region, like in the previous section, but as a device itself, along with the node that controls it. Hence, PMP applies well to nodal OPF, as opposed to region-based OPF.

The original objective function, in the case of PMP, is referred to as a prox function, shown below:

$$\text{prox}_{f,\rho}(v) = \underset{x}{\text{argmin}} (f(x) + (\rho/2) \|x - v\|_2^2) \quad (2.6)$$

Just like in the case of classical ADMM, the prox function is decomposed, using the Gauss-Seidel method, into one prox function per sub-problem, but for clarity purposes, we will be describing the original prox function. Function $f(x)$ is the device's cost function, with respect to decision vector x , a vector containing both the primal and dual variables. Parameter ρ is the tuning parameter for the augmented Lagrangian. The term v represents the average value of vector x in the other nodes.

The objective of PMP is to minimize the summation of the devices' objective functions. This is actually a summation of power generation costs and augmented Lagrangian penalty terms, the latter focused on satisfying the constraints corresponding to the Kirchoff's laws for power balance at the buses and the equality of bus voltage phase angles for the different devices connected to a particular bus. The method proceeds as indicated in pseudocode Algorithm 1: Each node solves its local sub-problem (prox function), and the resulting values for power generation and bus voltage angles are sent to the neighbours. Once all neighbouring messages have been received, the node uses them to perform its penalty-multiplier updates, and checks for convergence based on the consensus status for the decision variables. If convergence has not been achieved, the process repeats iteratively, with the penalty-multiplier updates sent to the neighbouring nodes, so that they can be used in solving the local sub-problems again.

In the context of the DC-OPF problem, the decision vector is fully decomposable across the devices, which renders PMP well-suited for nodal OPF, as mentioned. The fact that PMP is fully decentralized helps a lot with fine tuning and modification down to individual-device level, and works well for large-scale, multi-time step dispatch problems. However, convergence takes a large number of iterations, a limitation mitigated by the reduced time to solve each sub-problem. Finally, it is worth noting that convergence rate is highly dependent on the value of the tuning parameter ρ .

Algorithm 1 Proximal Message Passing OPF algorithm

- 1: Initialize decision variables, penalty-multiplier variables, power/voltage angle variables.
- 2: **while** not convergence **do**
- 3: Solve local prox function using penalty-multiplier variables
- 4: Broadcast updated power/voltage angle variables to neighbours
- 5: Gather power/voltage angle variables sent to you
- 6: Calculate updates for decision variables and penalty-multiplier variables
- 7: Check decision variables for convergence
- 8: Broadcast updated penalty-multiplier variables
- 9: Gather penalty-multiplier variables sent to you
- 10: **end while**

AUXILIARY PROBLEM PRINCIPLE

APP is a method that can be seen as a mixture of the attributes the previously presented methods had: It is decentralized, just like PMP, but at the level of regions, not individual devices, just like ATC and ADMM. Another commonality with ATC and ADMM is that the messages exchanged contain the bus voltage angles of the connected regions. However, in this case these messages are exchanged between neighbours, since APP is a decentralized method. Below are three differences/peculiarities of APP compared to previously introduced methods:

- APP decomposes the original problem into auxiliary sub-problems through linearization of the cross-terms in the Augmented Lagrangian.
- The bus voltage angle variables of each region are not only those connected to its end of the bus, θ , but also those at the other end, θ' : in a sense, the coupling variables θ and θ' are duplicated. The local updates and convergence checks after message retrieval now include these additional variables, please refer to the original paper[4] for more information on the specifics of this method.
- In contrast to ATC and ADMM, at iteration t , APP uses the values of the coupling variables received through messages at iteration $t - 1$, and not those of the current iteration t .

APP has similar advantages and limitations to ATC and ADMM (high computational effort based on the amount of shared variables, exchange of few data, good for region-based OPF etc.), but with the difference of not requiring a central coordinator, as it is a decentralized single-level method.

OPTIMALITY CONDITION DECOMPOSITION

In contrast to the aforementioned solutions, Optimal Condition Decomposition (OCD) uses the "KKT conditions" decomposition strategy, instead of the "Augmented Lagrangian Relaxation" strategy. OCD is a decentralized algorithm, operating at the level of regions.

After the KKT conditions are established, the algorithm works iteratively, and in each iteration the variables are updated/each subproblem is solved using a Newton-Raphson step. Part of each subproblem m consists of coupling constraints, which constraints are also part of the objective function of each neighbouring region n , with Lagrange multipliers λ_m for each. All variables associated with boundary buses are also exchanged with each neighbouring region, and their values are used at the updates of the next iteration for the receiving region.

Compared to the other solutions, OCD offers a balance of computational effort per iteration and data exchange with neighbours, but doesn't shine anywhere in particular, and can only be used at the level of regions, not individual devices. One major advantage of this method is the fact that it can be generalized to solve the AC-OPF problem, which is non-convex.

2.3. THE FUNDAMENTALS OF THE CONSENSUS AND INNOVATION ALGORITHM

In this section, the fundamentals of the C+I method are presented, serving as a backbone for the upcoming description of the adapted C+I algorithm in the next chapter. Firstly, the Karush-Kuhn-Tucker conditions are presented, upon which the C+I method is based. The description of the C+I method fundamentals follows next, along with a comparison between this algorithm and the previously presented ones, based on a set of certain criteria.

2.3.1. KARUSH-KUHN-TUCKER (KKT) CONDITIONS

The KKT conditions strategy is used for optimising non-linear constrained problems. As mentioned at the beginning of Section 2.2, the strategy takes the Lagrangian function of the original (nonlinear) optimization problem and applies to it the "Karush-Kuhn-Tucker" theorem, which states that an optimal solution is obtained by identifying the saddle points in the Lagrangian function, i.e., the (non-local-extremum) points where the slopes/derivatives in orthogonal directions are all zero. We can abstractly view the original optimization problem as follows:

$$\min f(x) \tag{2.7}$$

subject to:

$$h_i(x) = 0, \quad \forall i \in N \tag{2.8}$$

$$g_j(x) \leq \bar{G}_j, \quad \forall j \in M \tag{2.9}$$

$$g_j(x) \geq \underline{G}_j, \quad \forall j \in M \tag{2.10}$$

The Lagrangian function of this problem would then be defined as follows:

$$\mathcal{L}(x, \lambda, \mu) = f(x) + \sum_{i \in N} \lambda_i h_i(x) + \sum_{j \in M} \mu_j^{\bar{G}} (g_j(x) - \bar{G}_j) + \mu_j^{\underline{G}} (-g_j(x) + \underline{G}_j) \quad (2.11)$$

Based on the "Karush–Kuhn–Tucker" theorem, an optimal solution is only reached when the following 4 sets of conditions are met:

1. First-order Optimality conditions

$$\frac{\partial \mathcal{L}}{\partial x} = 0 \quad (2.12)$$

$$\frac{\partial \mathcal{L}}{\partial \lambda_i} = 0, \quad \forall i \in N \quad (2.13)$$

2. Feasibility conditions (-)

3. Complementary slackness conditions

$$\mu_j^{\bar{G}} (g_j(x) - \bar{G}_j) = 0, \quad \forall j \in M \quad (2.14)$$

$$\mu_j^{\underline{G}} (-g_j(x) + \underline{G}_j) = 0, \quad \forall j \in M \quad (2.15)$$

3. Positivity conditions

$$\mu_j^{\bar{G}}, \mu_j^{\underline{G}} \geq 0, \quad \forall j \in M \quad (2.16)$$

Vectors λ and μ are called KKT multipliers, or dual variables of the constraints. The decision variables x are called the primal variables.

2.3.2. CONSENSUS AND INNOVATION: THE FUNDAMENTALS

The Consensus and Innovation method is a decentralized method that can be implemented at both the nodal or the regional level. It utilizes the KKT strategy in order to establish a set of KKT conditions that each local subproblem needs to satisfy, for the optimal solution to be found. Afterwards, a set of variable updates is defined, updates that take place in every iteration of the C+I iterative algorithm. The updates are formed in such a way that, eventually, the variables obtain values that ensure the KKT conditions are satisfied, at which point we claim that convergence has been achieved and the optimal solution has been found. The C+I method, as described, was firstly established in the works of [21] and [10]. It is noteworthy that C+I is inherently different from the ALR decomposition methods, as, in the former, the KKT optimality conditions are directly solved through the accumulated effects of the variable updates, while the latter methods decompose the optimization problem into a smaller subproblem per entity to solve.

As it can be inferred from the name of the method, the C+I method consists of two main parts: the consensus part, and the innovation part. The consensus part refers to the intention for all nodes in the decentralized network to reach consensus on the value of the Locational Marginal Price (LMP), a term representing the increase in cost per unit of power. The innovation part refers to the innovative integration of the KKT optimality conditions into local updates/operations, that take place in order for the former to be satisfied.

Generally speaking, the C+I algorithm we focus on in this Thesis does offer the benefits that decentralized solutions offer, which were presented at the beginning of Section 2.2. Additionally, the algorithm has various advantages and disadvantages of its own, as indicated by [2]: it can be implemented at both the nodal or the regional level, allowing for a shift in the balance and trade-off between communication costs and local processing requirements. Each iteration of the algorithm is faster than that of the Lagrangian-based methods, although more iterations are required due the fact that all of the subproblems are required to reach consensus on the LMP.

Due to the improvements of our adaptation, which is firstly fully established in the Master Thesis of Pedro Parreira[17], more advantages arise. One advantage results from an applied practise where, in each iteration, physical measurements such as voltage and current are locally measured from each device in the network. This effectively eliminates a lot of computational requirements, since the physical network restrictions represented by some constraints of the problem are trivially satisfied when identifying the actual state of the electric network. Another advantage of our implementation is that, while it takes place at the nodal level, which equates greater communication requirements, a modification keeps the number of communicated variables low. This modification involves some changes in the optimality conditions so that the neighbouring voltage levels no longer need to be communicated, reducing the amount of communicated variables by one. Both of the aforementioned practises can be observed in Chapter 3, where the algorithm is described in detail. Overall, the resulting algorithm is relatively simple to implement, as it consists mainly of iterative variable updates, information exchange amongst nodes and collection of information from the physical devices themselves.

3

THE CONSENSUS AND INNOVATION ALGORITHM FOR UNIPOLAR DC MICROGRIDS: THEORY

This section describes the C+I algorithm for the DC-OPF problem, already introduced in Section 2.3.2. The presented version of the algorithm was lastly developed by Pedro Pereira[17], and was modified/improved by us, wherever mentioned as such in the report. Firstly, in section 3.1, an overview of the structure of the algorithm is presented. Then, in section 3.2, we explain the way in which the DC-OPF problem is mathematically modelled under the C+I algorithm. Finally, in section 3.3, the mechanics behind the “droop control” functionality are explained, which plays an essential role in the effectiveness of the algorithm.

3.1. OVERVIEW OF THE STRUCTURE: ENTITIES AND COMMUNICATIONS

THIS section describes the general structure of the C+I algorithm for the DC-OPF problem, in terms of entities and communications. We start off with the fact that there exist two Layers of components: the Physical Layer and the Cyber Layer. The Physical Layer represents the actual physical micro-grid, i.e., the converters (generators, loads) that are connected to each other through electrical lines. Each one of these converters is assumed to have a processing unit attached to it, a unit referred to as a “node”. The Cyber Layer represents these nodes, which are connected to each other in the same way their respective converters are connected to their neighbours. As explained in section 2.3.2, each of these nodes iteratively performs some variable updates until so-called “convergence” is achieved, i.e., until the KKT conditions established by using the KKT strategy are satisfied.

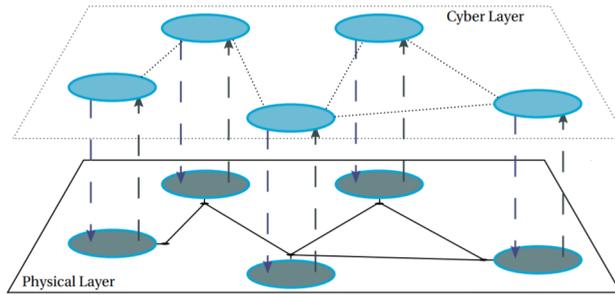


Figure 3.1: The form of interaction occurring between the Physical and Cyber Layers

Figure 3.1 presents the form of interaction occurring between the Physical and Cyber Layers. There are, actually, two types of communication taking place: Inter-nodal communication and inter-layer communication. Regarding inter-nodal communication, the nodes exchange variables that aid them in reaching consensus and achieving convergence. As for inter-layer communication, each converter sends its physical measurements (voltage/current/power) to its node, which then uses those measurements to perform its necessary work. Part of this work is to regulate the relationship between voltage and current, a task referred to as “droop control”. The decided voltage/current is then communicated back to the converter of the node, in a way that the converter is instructed on how to act, specifically, how to adjust its voltage, power etc. Different converters have different capabilities. The results of this behavior are later communicated again back to the node, and the cycle continues until convergence is achieved. The aforementioned procedure is demonstrated in Figure 3.2.

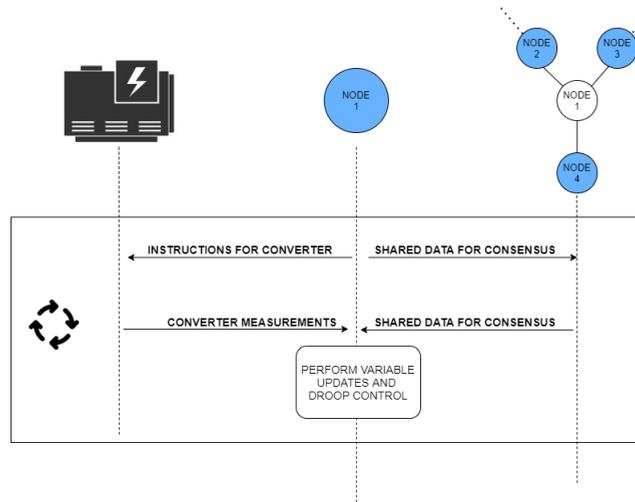


Figure 3.2: The communication occurring during each iteration of the algorithm (enclosing rectangle), along with the data exchanged

3.2. ADAPTING THE C+I ALGORITHM FOR THE DC-OPF PROBLEM

HAVING described the DC-OPF problem and its relevant parameters in section 2.1, and having introduced the fundamentals behind the C+I method in section 2.3, we are now ready to explain the way in which the C+I algorithm is mathematically defined for the DC-OPF problem. We begin by once again presenting the optimization problem, as it was initially formulated in section 2.1.

$$\min_p \sum_{m \in N} A_m p_m^2 + B_m p_m \quad (3.1)$$

$$p_m = u_m \cdot i_m^s = u_m \sum_{n \in \Omega_m} i_{mn} = u_m \sum_{n \in \Omega_m} G_{m,n} (u_m - u_n), \quad \forall m, n \in N \quad (3.2)$$

$$i_{mn} = \sum_{n \in \Omega_m} G_{mn} (u_m - u_n) \leq \bar{I}_{mn}, \quad \forall m, n \in N \quad (3.3)$$

$$\underline{P}_m \leq p_m \leq \bar{P}_m, \quad \forall m \in N \quad (3.4)$$

$$\underline{U}_m \leq u_m \leq \bar{U}_m, \quad \forall m \in N \quad (3.5)$$

Constraint 3.2 represents the “power mismatch” equation, that is, the total power generated at node m is equal to the difference between its total incoming and outgoing power. The fundamental equations from the Appendix section A.1 can be used to represent the right-hand side of the constraint as power. Constraint 3.3 ensures that the total current flowing through a line does not exceed the maximum current limit. Outgoing current is given a positive sign, while incoming current is given a negative sign. Constraints 3.4 and 3.5 ensure that both the power and the voltage generated, respectively, stay within the appropriate limits.

3.2.1. FORMULATING THE KKT OPTIMALITY CONDITIONS

Given the rules surrounding the formulation of the Lagrangian function in section 2.3.1, based off the KKT strategy and the specific optimization problem, the Lagrangian function in this case is formulated as follows.

$$\begin{aligned}
\mathcal{L} = & \sum_{m \in N} A_m p_m^2 + B_m p_m \\
& + \sum_{m \in N} \lambda_m (u_m \sum_{n \in \Omega_m} G_{m,n} (u_m - u_n) - p_m) \\
& + \sum_{m \in N} \sum_{n \in \Omega_m} \mu_{mn} (G_{mn} (u_m - u_n) - \bar{I}_{mn}) \\
& + \sum_{m \in N} \mu_m^{\bar{P}} (p_m - \bar{P}_m) \\
& + \sum_{m \in N} \mu_m^{\underline{P}} (-p_m + \underline{P}_m) \\
& + \sum_{m \in N} \mu_m^{\bar{U}} (u_m - \bar{U}_m) \\
& + \sum_{m \in N} \mu_m^{\underline{U}} (-u_m + \underline{U}_m)
\end{aligned} \tag{3.6}$$

The notations λ and μ are the dual variables of the formulation, with λ being associated with the “power mismatch” constraint (3.2), while the μ variables are associated with the rest of the constraints. Specifically, λ is referred to as the Locational Marginal Price (LMP), measured in monetary units per Watt (*m.u./W*), and it represents the increase in cost per unit of power. The μ variables represent the violation (or not) of certain restrictions and constraints, such as the maximum current limit for a bus.

The KKT Conditions can now be derived from the Lagrangian function. The First Order Optimality conditions are as follows.

$$\frac{\partial \mathcal{L}}{\partial p_m} = 2A_m p_m + B_m - \lambda_m + \mu_m^{\bar{P}} - \mu_m^{\underline{P}} \quad (3.7)$$

$$\frac{\partial \mathcal{L}}{\partial u_m} = \lambda_m \sum_{n \in \Omega_m} G_{mn}(u_m - u_n) \quad (3.8)$$

$$\begin{aligned} & + \lambda_m u_m \sum_{n \in \Omega_m} G_{mn} - \sum_{n \in \Omega_m} \lambda_n u_n G_{mn} \\ & + \sum_{n \in \Omega_m} G_{mn}(\mu_{mn} - \mu_{nm}) + \mu_m^{\bar{U}} - \mu_m^{\underline{U}} = 0 \\ \frac{\partial \mathcal{L}}{\partial \lambda_m} & = -p_m + u_m \sum_{n \in \Omega_m} G_{mn}(u_m - u_n) = 0 \end{aligned} \quad (3.9)$$

$$\frac{\partial \mathcal{L}}{\partial \mu_{mn}} = \sum_{n \in \Omega_m} G_{mn}(u_m - u_n) - \bar{I}_{mn} \leq 0 \quad (3.10)$$

$$\frac{\partial \mathcal{L}}{\partial \mu_m^{\bar{P}}} = p_m - \bar{P}_m \leq 0 \quad (3.11)$$

$$\frac{\partial \mathcal{L}}{\partial \mu_m^{\underline{P}}} = -p_m + \underline{P}_m \leq 0 \quad (3.12)$$

$$\frac{\partial \mathcal{L}}{\partial \mu_m^{\bar{U}}} = u_m - \bar{U}_m \leq 0 \quad (3.13)$$

$$\frac{\partial \mathcal{L}}{\partial \mu_m^{\underline{U}}} = -u_m + \underline{U}_m \leq 0 \quad (3.14)$$

As we can see, the KKT First-Order Optimality conditions are simply the derivatives of the variables of the Lagrangian function. The only exception is the $\frac{\partial \mathcal{L}}{\partial u_m}$ derivative, which has been adapted in order for the consensus goal of the algorithm to be achievable.

Next, the Complementary Slackness conditions are presented.

$$\mu_{mn} (\bar{I}_{mn} - G_{mn}(u_m - u_n)) = 0 \quad (3.15)$$

$$\mu_{nm} (-\bar{I}_{mn} + G_{mn}(u_m - u_n)) = 0 \quad (3.16)$$

$$\mu_m^{\bar{P}} (\bar{P}_m - p_m) = 0 \quad (3.17)$$

$$\mu_m^{\underline{P}} (p_m - \underline{P}_m) = 0 \quad (3.18)$$

$$\mu_m^{\bar{U}} (\bar{U}_m - u_m) = 0 \quad (3.19)$$

$$\mu_m^{\underline{U}} (u_m - \underline{U}_m) = 0 \quad (3.20)$$

Finally, the Positivity conditions require all μ variables to be positive. Any μ variable with a value other than zero signifies a limit that has been violated (e.g. line congestion, or exceeded voltage upper limit).

$$\mu_{mn}, \mu_{nm}, \mu_m^{\bar{P}}, \mu_m^{\underline{P}}, \mu_m^{\bar{U}}, \mu_m^{\underline{U}} \geq 0 \quad (3.21)$$

Although the notation is omitted, the KKT conditions apply distinctively per node m or per line (m, n) in the grid.

Notice that, in the context of the DC-OPF problem, a substitution is possible in the above conditions: namely, fundamental equation 2 from the Appendix can be used to replace parameter u_n with other, local parameters. This substitution is useful, since it relieves the algorithm of the need to know the external, neighboring voltage u_n , thus fewer variables need to be communicated across nodes. However, it is of vital importance to the algorithm that this substitution is done only after the KKT conditions are derived, and not on the primal constraints (3.2-3.3). This is because consensus, i.e. the effectiveness of the algorithm, depends on the existence of the neighboring LMP (λ_n) in $\frac{\partial \mathcal{L}}{\partial u_m}$, however, as presented by Pedro Parreira[17], any substitution in the primal constraints would result in $\frac{\partial \mathcal{L}}{\partial u_m}$ lacking the neighboring LMP values, thus rendering the algorithm ineffective. The resulting conditions are presented below, please refer to [17] for a more detailed presentation of the substitution process.

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial u_m} = & \sum_{n \in \Omega_m} i_{mn}(\lambda_m + \lambda_n) + u_m \left(\sum_{n \in \Omega_m} G_{mn}(\lambda_m + \lambda_n) \right) \\ & + \sum_{n \in \Omega_m} G_{mn}(\mu_{mn} + \mu_{nm}) + \mu_m^{\bar{U}} - \mu_m^{\underline{U}} \end{aligned} \quad (3.22)$$

$$\frac{\partial \mathcal{L}}{\partial \lambda_m} = \hat{p}_m - p_m = 0 \quad (3.23)$$

$$\frac{\partial \mathcal{L}}{\partial \mu_{mn}} = \sum_{n \in \Omega_m} i_{mn} - \bar{I}_{mn} \leq 0 \quad (3.24)$$

Furthermore, the substitution could also be applied on the Complementary Slackness conditions, but that is not relevant, since these conditions are indirectly enforced through the use of the derivatives in the variable updates.

Having performed the above substitution, the only data necessary to be communicated during inter-nodal communication are a subset of the dual variables, specifically, λ_m and μ_{mn} . As mentioned in section 2.3.2, this constitutes a significant advantage in terms of the data exchanges required by the algorithm, and it would have not been possible without the integration of physical measurements in our algorithm.

Since the above conditions will play a significant role in the updates performed at each iteration of the algorithm, we will be providing an overview of what each condition

represents. The $\frac{\partial \mathcal{L}}{\partial u_m}$ derivative represents the weighted average of the local and neighboring LMPs, while accounting for congestion issues and voltage limit violations. Given that, ideally, the LMPs of connected nodes are aimed to be equal, this condition is referred to as the “consensus” condition. The $\frac{\partial \mathcal{L}}{\partial \lambda_m}$ derivative is referred to as the “power mismatch” condition, and it expresses the difference between the power, as measured directly from the converter by the node (\hat{p}_m^s in node-to-grid communication), and the so-called power setpoint (p_m^s), which is the power variable as used and altered through the updates of the algorithm. Finally, the $\frac{\partial \mathcal{L}}{\partial \mu_{mn}}$ derivative represents the line congestion.

3.2.2. FORMULATING THE UPDATES PER ITERATION

Having formulated the KKT conditions for the DC-OPF problem, we proceed to define the variable updates that will be executed at every iteration of the algorithm.

AN UPDATE STRATEGY OVER THE RELATIONSHIP BETWEEN λ_m AND p_m

As mentioned, the λ_m variable represents the Locational Marginal Price (LMP), is measured in monetary units per Watt (m.u./W), and it represents the increase in cost per unit of power. Thus, it is directly related to the power variable. When attempting to establish update rules for the above two variables, the following goals need to be achieved: Intuitively, and based on the objective function, most power should be generated from the cheapest generators. The expensive generators are only supposed to generate power if all the cheaper ones have reached their maximum generation limits, or if there exist other limitations, like line congestion. In order to avoid oscillation between the LMP and power during their updates, Dolaputra[6] proposed an update mechanism that causes an inter-dependence between the two variables and achieves the aforementioned goals:

Figures 3.3 and 3.4 depict the relationship between the LMP and the power. At each iteration, either the LMP or the power will be updated, and the other variable will adapt accordingly. The idea behind the updates is the following: If the LMP is less than the least possible cost for power generation by a generator, as seen by the cost derivative thresholds on the vertical axis, the generator keeps generating the minimum allowed power (usually zero). A generator is only allowed to generate (more) power if its LMP reaches the lower cost threshold. After the LMP equals/exceeds the threshold, the generator will keep increasingly generating power, until it reaches its maximum power limit, or until the whole network stabilizes at that LMP. As the power changes, the LMP will vary linearly if the generator has a quadratic cost function ($A_m \neq 0$), and will remain constant if the generator has a linear cost function ($A_m = 0$). If the maximum power limit is reached but there is still demand (signified by a continuously increasing LMP), the propagation of LMP will involve another, more expensive generator. The above methodology makes intuitive sense since, in any market, any product (power) sold at price lower than the “set” price is not worth selling, while a well-priced product whose demand is higher than its supply will keep having its price increased and more producers (generators) will engage

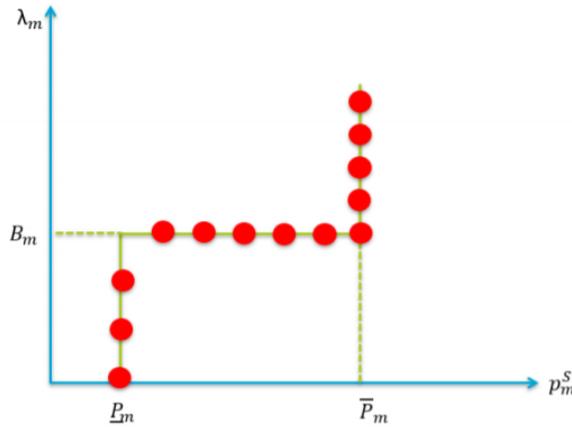


Figure 3.3: The relationship between the LMP and the power, for a generator with a linear cost function

in the trade, producers that sell that product at that higher price.

Having described the relationship between the price and power generation for the generators, it is also important to specify the relationship between the price and power demand for the loads, in case there is a price involved with consuming power. We will refer to that price as revenue, as it is of beneficial nature, contrary to the costs. Perhaps the most important thing to note is that a load will only consume maximum power if the generators that produce it, produce it at a lower cost than the revenue of consumption, i.e. if the LMP is less than the least possible revenue of power consumption. The idea is that a load only requests for more power if producing that amount of power is cheap compared to consuming it, creating revenue. After the LMP equals/exceeds the lower revenue threshold, the load will keep decreasing its request for power, until it turns off, or until the whole network stabilizes at that LMP. Furthermore, similarly to a generator, a load will allow for propagation of the the LMP at maximum/minimum power consumption, as there may be other loads requesting for power.

Looking at Figures 3.3 and 3.4, we can observe two meaningful regions of variable behavior. The first one is referred to as the Constant Power Region, where the power remains constant at its minimum or maximum limit, and the LMP receives a value below the Minimum Marginal Cost ($\lambda_m < B_m + 2A_m P_m$) or above the Maximum Marginal Cost ($\lambda_m > B_m + 2A_m \bar{P}_m$), respectively. The other region is referred to as the Marginal Power Region, where the power has a value between its limits, and the LMP either remains constant and equal to B_m (if the objective function is linear) or receives a value between the Minimum and Maximum Marginal Cost, depending on the value of power.

Going into technicalities, it is worth noting that, at each iteration, the region these variables belong to is identified by the LMP. That is, there are two cases that can unfold:

1. The LMP dictates that the current region is the Constant Power Region. The power

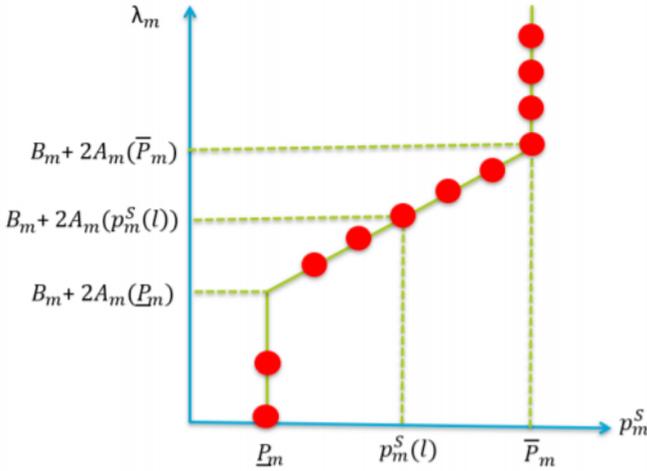


Figure 3.4: The relationship between the LMP and the power, for a generator with a quadratic cost function

is then updated to the appropriate limit, and the LMP is updated using its update equation. If the new LMP is now in the Marginal Power Region, the second case needs to also be executed.

2. If the LMP dictates that the current region is the Marginal Power Region, the power is updated using its update equation, and the LMP is updated with a value in between the Minimum and Maximum Marginal Costs, based on the new value of the power. If, now, the new value of the power is exceeding or equal to any power limit, then for one more time, the power is updated to the appropriate limit, and the LMP is updated using its update equation.

CONSTANT POWER REGION UPDATE EQUATIONS

The update of the LMP takes the below form, where both derivatives $\frac{\partial \mathcal{L}}{\partial u_m}$ and $\frac{\partial \mathcal{L}}{\partial \lambda_m}$ are involved.

$$\lambda_m(l+1) = \lambda_m(l) - a_u^\lambda \frac{\partial \mathcal{L}}{\partial u_m} + a_\lambda^\lambda \frac{\partial \mathcal{L}}{\partial \lambda_m} \tag{3.25}$$

$$p_m(l+1) = \bar{P}_m \tag{3.26}$$

$$a_u^\lambda = \frac{1/(|\Omega_m|+1)}{u_m \sum_{n \in \Omega_m} G_{mn}} \tag{3.27}$$

$$a_\lambda^\lambda = 1 \times 10^{-5} \tag{3.28}$$

As mentioned, $\frac{\partial \mathcal{L}}{\partial \lambda_m}$ represents the “power mismatch”. As presented in the example scenario A.2 from the Appendix, in the event of power demand from a load, the balance between the measured and predicted power will be lost, creating the so-called mismatch and giving the derivative a value, which value will then have an analogous effect on the new LMP, eventually causing power generation/reduction for any demands to be met. For example, a positive derivative translates to a measured power output that is higher than the power setpoint, and so the LMP is increased to encourage power generation. On the other hand, a negative derivative translates to a measured power output that is lower than the power setpoint, and so the LMP is decreased to discourage power generation. Regarding the a_λ^λ tuning parameter, it represents the variation rate of the LMP i.e., how drastically each node is willing to increase/decrease its LMP to match the intended power output. Setting this tuning parameter too high runs the risk of oscillations and very reactive behavior, while setting it too low slows the overall convergence rate of the algorithm. Following past experimentation[6][17], the variation rate was defined as a 1% variation per kW of power mismatch.

Derivative $\frac{\partial \mathcal{L}}{\partial u_m}$ represents, as mentioned, the consensus part of the algorithm, and it includes the neighboring LMPs, hence it is also necessary for the LMP update. The reason the derivative has a disanalogous effect on the new LMP (negative sign) is broken down and explained in detail by both Parreira[17] and Dolaputra[6], and it is hence omitted from this document. However, for the sake of argument, it is worth remembering that this derivative represents the weighted average of the local and neighbouring LMPs: at the absence of congestion and voltage limit violations, this derivative will be positive if the local LMP is greater than the averaged neighboring ones, and negative otherwise. Since the attempt is to equalize the LMPs (reach consensus), a positive derivative should decrease the updated LMP. Through that same explanation, we argue that the tuning parameter a_u^λ , representing the variation rate of the LMP given the difference in LMP between this and the neighboring nodes, needs to have a nominator value of $\frac{1}{|\Omega_n|+1}$, as that evenly distributes the stabilization efforts between neighboring nodes. This new value

overrides the previously established value for a_u^λ in Parreira's work, which failed to take in account the number of neighbours when evaluating this parameter.

Finally, since the LMP needs to remain the same at the absence of congestion and losses, the previous LMP is also part of the update equation.

MARGINAL POWER REGION UPDATE EQUATIONS

The update of the power and the LMP takes the below form, where both derivatives $\frac{\partial \mathcal{L}}{\partial u_m}$ and $\frac{\partial \mathcal{L}}{\partial \lambda_m}$ are involved.

$$p_m(l+1) = p_m(l) - a_u^p \frac{\partial \mathcal{L}}{\partial u_m} + a_\lambda^p \frac{\partial \mathcal{L}}{\partial \lambda_m} \quad (3.29)$$

$$\lambda_m(l+1) = 2A_m p_m(l+1) + B_m \quad (3.30)$$

$$a_u^p = \frac{100}{u_m \sum_{n \in \Omega_m} G_{mn}} \quad (3.31)$$

$$a_\lambda^p = \frac{1}{(|\Omega_n| + 1) * 2} \quad (3.32)$$

Derivative $\frac{\partial \mathcal{L}}{\partial u_m}$ is necessary for the power update, just like it was for the LMP update in the Constant Power Region. This consensus condition is necessary since it describes the way through which the local and neighboring LMPs relate to and affect the power. In the case of the LMP update for the Constant Power Region, we explained how a positive derivative indicates a need for a smaller LMP. Consequently, a smaller LMP indicates a lesser need for power generation, and thus the consensus derivative also has a disanalogous effect on the new value of power (negative sign). The a_u^p parameter represents the variation rate of the power, given the difference in LMP between this and the neighboring nodes, and again, following past experimentation[6][17], the value of its nominator has been defined to be 100, i.e, 100W per “amount of monetary units per power unit” ($W^2/m.u$).

As explained in the Constant Power Region section, the “power mismatch” derivative $\frac{\partial \mathcal{L}}{\partial \lambda_m}$ receives a value upon a detected mismatch, which value is then used to update the LMP. The same logic described in that section, applies here, but in order to update the power setpoint instead. Again, a positive derivative indicates a need for power generation, and vice versa. Regarding the tuning parameter a_λ^p , the logic that applies is a combination of the cases for a_u^λ and a_λ^λ : Just like a_λ^λ , a_λ^p is a variation rate parameter, but for the power, and based on the power mismatch. Furthermore, it is also required to achieve the same goal that a_u^λ is trying to achieve: evenly distributed stabilization efforts. Thus, a value of $\frac{1}{|\Omega_n|+1}$ would seem fitting, overriding the previously established value in Parreira's work. However, because the voltage update, presented in the following section,

is also contributing to the elimination of the power mismatch, the significance of the a_λ^p tuning parameter is further halved to $\frac{1}{(|\Omega_n|+1)*2}$.

Finally, we can observe the LMP update adapting to the new value of power, as it was initially described by the power and LMP update strategy.

THE VOLTAGE UPDATE

The voltage update, under no congestion in a connected line of the node, is as follows, with derivatives $\frac{\partial \mathcal{L}}{\partial u_m}$ and $\frac{\partial \mathcal{L}}{\partial \lambda_m}$ involved once again.

$$u_m(l+1) = u_m(l) - a_u^u \frac{\partial \mathcal{L}}{\partial u_m} - a_\lambda^u \frac{\partial \mathcal{L}}{\partial \lambda_m} \quad (3.33)$$

$$a_u^u = \frac{5}{u_m \sum_{n \in \Omega_m} G_{mn}} \quad (3.34)$$

$$a_\lambda^u = \frac{1/(|\Omega_n|+1)*2}{u_m \sum_{n \in \Omega_m} G_{mn}} \quad (3.35)$$

The voltage setpoint, apart from being used in various other updates, is also used to evaluate the droop curve. As such, and as mentioned in the previous section, it directly contributes to the elimination of any power mismatch. Thus, both derivatives $\frac{\partial \mathcal{L}}{\partial u_m}$ and $\frac{\partial \mathcal{L}}{\partial \lambda_m}$ are involved in its update equation, for the same reasons they are involved in the power update equation. It is worth noting that increasing the voltage helps reduce transmission losses across a line, since a higher voltage requires less current to satisfy the same amount of power demand. Furthermore, at the start of each iteration, the voltage setpoint assumes the value of the measured voltage from the grid, and is only updated after the power and LMP updates are complete.

The a_u^u tuning parameter is a variation rate parameter for the voltage, given the difference in LMP between this and the neighboring nodes. A positive derivative indicates a need for less power generation overall, so there is consequently a need for reducing the voltage. Following past experimentation[17], the value of its nominator has been defined to be 5V per “monetary unit per power unit” ($V * W/m.u.$).

The a_λ^u tuning parameter represents the variation rate of the voltage given the power mismatch. As mentioned, the a_λ^u tuning parameter splits the necessary work with the a_λ^p parameter: that is, it assumes a nominator of $\frac{1}{(|\Omega_n|+1)*2}$ in order to collaborate with the power setpoint in eliminating any power mismatch. However, it performs the opposite operation on the voltage than the one performed on the power setpoint: Rather than increasing the voltage, a positive $\frac{\partial \mathcal{L}}{\partial \lambda_m}$ would cause the voltage setpoint to diminish, since a lower voltage encourages current flow, and the purpose of this parameter is more focused on optimal current flow given the power, rather than alternating the power flow itself.

In a micro-grid, a line cannot allow an unlimited amount of current. This “maximum current” limitation is represented by the variable \bar{I}_{mn} . In the case of line congestion, the voltage update is performed differently than before, in order to contribute to the elimination of this issue:

$$u_m(l+1) = \frac{\bar{I}_{mn}}{G_{mn}} + u_n(l) \quad (3.36)$$

By utilizing the fundamental equation 4 from the Appendix, we can shape the voltage update into 3.36, effectively assigning the proper voltage to node m so that the current in the congested line (m, n) is maximized and kept stable to that \bar{I}_{mn} limit. If multiple lines are congested, the neighbor of minimum voltage is selected for the update operation.

Using fundamental equation 2 from the Appendix, update 3.36 is reshaped into equation 3.37, effectively relieving us of the need to know variable $u_n(l)$.

$$u_m(l+1) = u_m(l) + \frac{\bar{I}_{mn} - i_{mn}}{G_{mn}} \quad (3.37)$$

THE μ_{mn} DUAL VARIABLES UPDATE - CONGESTION MANAGEMENT

The μ_{mn} dual variables update is presented below.

$$\begin{aligned} \mu_{mn}(l+1) &= \mathbb{P} \left[\mu_{mn}(l) + \beta_{mn} \frac{\partial \mathcal{L}}{\partial \mu_{mn}} \right] \\ \mathbb{P} \Rightarrow \mu_{mn}(l+1) &= 0, \text{ if } (\mu_{mn} \cap \frac{\partial \mathcal{L}}{\partial \mu_{mn}}) \leq 0 \end{aligned} \quad (3.38)$$

$$\beta_{mn} = 5 \quad (3.39)$$

As stated for all μ variables at condition 3.21, any non-zero value indicates an issue, in this case, line congestion. This non-zero value can only occur if the $\frac{\partial \mathcal{L}}{\partial \mu_{mn}}$ derivative is positive, i.e., if the current flowing through line (m, n) is greater than the highest allowed. Even in the case of a positive derivative, the update does not allow μ_{mn} to receive a negative value. Tuning parameter β_{mn} represents the growth rate of the μ_{mn} variable, given the excess current, and has been given a value of 5[17].

The effect of the μ_{mn} variable can be demonstrated with a simple example: suppose line congestion occurs and μ_{mn} receives a non-zero value. This change will affect derivative $\frac{\partial \mathcal{L}}{\partial u_m}$, and, subsequently, the LMP update 3.25. This will cause the LMP of node m to decrease and the LMP of node n to increase, limiting power flow in the (m, n) direction and pushing for power consumption in the (n, m) direction, effectively reducing the current flowing through line (m, n) .

THE μ_m^U DUAL VARIABLE UPDATES – VOLTAGE LIMITS

The μ_m^U dual variables update is presented below.

$$\bar{\mu}_m^U(l+1) = \mathbb{P} \left[\bar{\mu}_m^U(l) + \beta^{\bar{U}} \frac{\partial \mathcal{L}}{\partial \bar{\mu}_m^U} \right] \quad (3.40)$$

$$\mu_m^U(l+1) = \mathbb{P} \left[\mu_m^U(l) + \beta^U \frac{\partial \mathcal{L}}{\partial \mu_m^U} \right] \quad (3.41)$$

$$\beta^{\bar{U}} = \beta^U = 10 \quad (3.42)$$

The exact same logic that applied for the μ_{mn} variables update, applies here as well. The only difference is that the issue these variables represent is voltage limit violations, with derivatives $\frac{\partial \mathcal{L}}{\partial \bar{\mu}_m^U}$ and $\frac{\partial \mathcal{L}}{\partial \mu_m^U}$ calculating the amount of voltage exceeding a limit. Tuning parameters $\beta^{\bar{U}}$ and β^U represent the growth rate of the μ_m^U variables, given the excess voltage, and have been given a value of 10[6].

THE μ_m^P DUAL VARIABLE UPDATES – POWER LIMITS

The μ_m^P dual variables update is presented below.

$$\bar{\mu}_m^P(l+1) = \mathbb{P} \left[\bar{\mu}_m^P(l) + \beta^{\bar{P}} \frac{\partial \mathcal{L}}{\partial \bar{\mu}_m^P} \right] \quad (3.43)$$

$$\mu_m^P(l+1) = \mathbb{P} \left[\mu_m^P(l) + \beta^P \frac{\partial \mathcal{L}}{\partial \mu_m^P} \right] \quad (3.44)$$

$$\beta^{\bar{P}} = \beta^P = 0.5 \quad (3.45)$$

The exact same logic that applied for the μ_{mn} variables update, applies here as well. The only difference is that the issue these variables represent is power limit violations, with derivatives $\frac{\partial \mathcal{L}}{\partial \bar{\mu}_m^P}$ and $\frac{\partial \mathcal{L}}{\partial \mu_m^P}$ calculating the amount of power exceeding a limit. Tuning parameters $\beta^{\bar{P}}$ and β^P represent the growth rate of the μ_m^P variables, given the excess power, and have been given a value of 0.5[6].

In contrast to the rest of the variables, these μ_m^P variables are not used in any other update, and thus do not affect the convergence rate of the algorithm. They can, however, be useful in verifying that the algorithm is functioning as intended.

3.3. DROOP CONTROL

As discussed in section 3.1, the droop control functionality takes place as part of the work for a node, and the results of droop control are then communicated to the converter in order to adjust its behavior. Droop control is essentially the act of defining a droop curve, such that the converter can optimally set its own power, current and voltage, and the relationship between them, depending on the specific converter's capabilities. In our case, droop control ensures that, whenever there is a shift in either the voltage or the current, the remaining variable adapts accordingly, and power is maintained stable.

Given that the maximum and minimum current ($\overline{I_m}$, $\underline{I_m}$) that a converter allows to flow through it are known, the form of a droop curve is as presented in Figure 3.5.

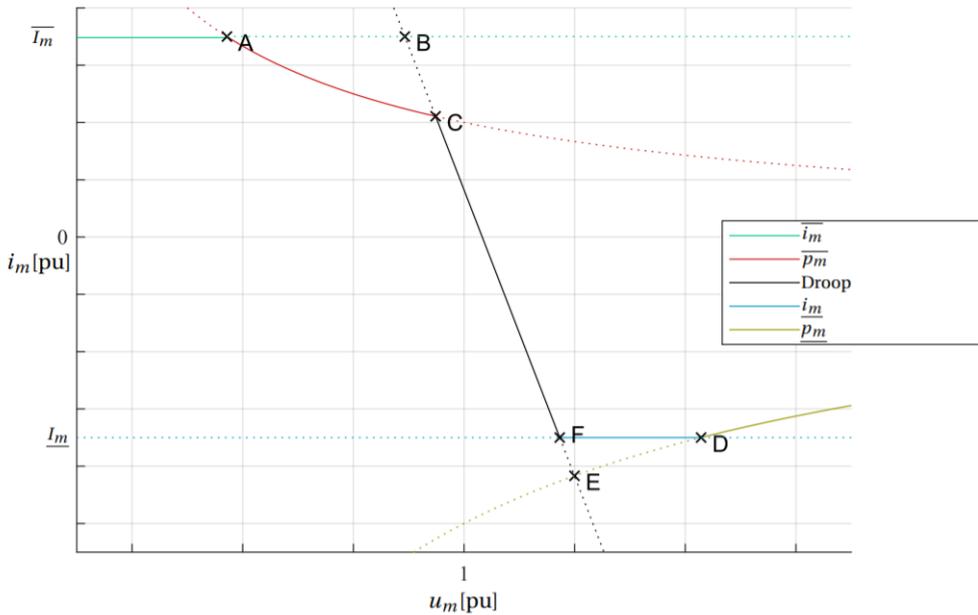


Figure 3.5: The form of a droop curve, with its 5 regions of operation

It is possible to identify 5 regions of operation which describe the behavior of the converter, depending on its own state. When the converter is supplying/demanding maximum or minimum current, its behavior is mandated by the bright green and blue lines, respectively. This means that its voltage may change, but the current may not, as indicated by the horizontal lines. When the converter is at the regions of maximum or minimum power, i.e., the red and olive-green lines, respectively, both current and voltage may change, but their product must remain constant ($u_m \cdot i_m = p_m$). Finally, in the case where the converter is not at any operational limit (points C to F), its behavior is described by a linear function:

$$i_m^s = f(u) = -d * u_m + i_m^0 \tag{3.46}$$

where i_m^0 is the value of the current when the voltage is equal to one ($u_m = 1$), and d is the so-called “droop slope”. The droop slope d is actually a tuning parameter: the smaller the slope, the less reactive a converter is to a voltage oscillation, meaning it will provide less current as compensation for the change, but it will remain more stable against voltage oscillations on the network. On the other hand, a higher droop slope means that a node will compensate more for power imbalances, but it will be more susceptible to oscillation problems. In our implementation of the C+I algorithm, the droop slope is kept constant per device, in all regions of operation, after experimentally identifying a droop slope that provided a good trade-off between reactivity and stability.

3.3.1. THE OBJECTIVE OF DROOP CONTROL

Apart from indirectly aiding towards the convergence of the algorithm, the establishment of a droop curve also helps a converter preserve stability. The objective of droop control is demonstrated in Figure 3.6.

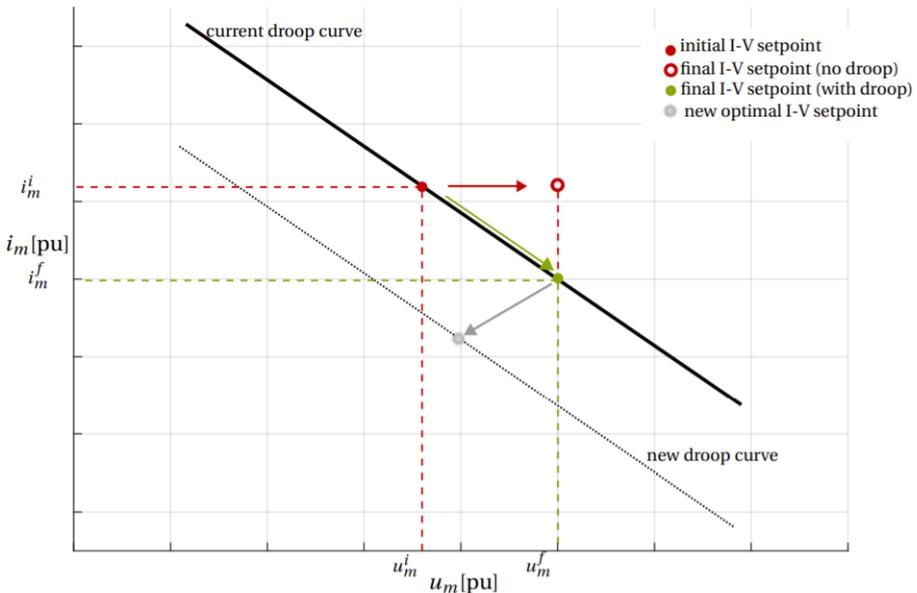


Figure 3.6: A demonstration of the droop curve being redefined

Let us describe the example of Figure 3.6. Suppose a converter is capable of altering its current. If the local voltage automatically increases (transition from red dot to hollow red dot), it is indicated that, somewhere in the network, more power is being generated than consumed. If no droop control was present, the power output of that node would

eventually increase, possibility worsening the network instability. However, with droop control present, the converter is instructed to maintain its current in accordance with the droop curve, and so it will lower its current to match that curve (green dot). Based on the observed events, the droop curve established by the algorithm for the next iteration could be thought of as the gray line, with the gray dot being the new optimal voltage-current setpoint.

3.3.2. DROOP CONTROL PER OPERATION REGION

Let us now describe the behavior of a converter for each of the regions, based on our algorithm. It is important to note that we assume constant power and current limits, that do not change over time. Additionally, for simplification purposes, and since our experimental setup does not suffer from such limitations, our algorithm does not account for the maximum and minimum current regions. The way they can be handled, however, will be briefly explained.

OPERATING WITHIN THE LIMITS – LINEAR REGION

As mentioned, the goal in this region is to define an optimal droop curve, so that the converter can be instructed to properly adjust its power generation. In order to define a droop curve, two variables need to be defined, as seen from equation 3.46: droop slope d and constant i_m^0 . If we look at that linear equation of the droop curve and at Figure 3.6, we can observe that the droop slope d describes the variation rate of the current, with regards to any change in the voltage. Therefore, it can be written as

$$d = -\frac{\Delta i^s}{\Delta u} \quad (3.47)$$

where Δi^s is the range between the maximum and minimum possible current for the converter, given the voltage deviation, and Δu is the maximum voltage deviation that the converter can experience. Based on the above definitions, it holds that

$$\overline{i^s} = \overline{p^s}/\underline{u}, \quad \underline{i^s} = \underline{p^s}/\overline{u} \quad (3.48)$$

where \overline{u} and \underline{u} are the maximum and minimum possible voltages that can occur, given the voltage deviation. \overline{u} and \underline{u} are to be defined by the algorithm in order for an optimal droop slope d to be defined, and, consequently, the droop curve. The droop slope can then be rewritten as

$$d = \frac{\underline{p^s}/\overline{u} - \overline{p^s}/\underline{u}}{\Delta u} \quad (3.49)$$

A proper and realistic range of voltage deviation Δu must also be defined. In accordance with DC standardization discussions, in order to decrease power output from the maximum to the minimum limit, i.e., $\Delta p^s = \overline{P}_m^s - \underline{P}_m^s$, and vice versa, it would require a voltage deviation of 10V, i.e., $\Delta u = 10V$. In other words, if a load is requesting power

from a generator of zero power output, a 10V variation in voltage between the two can cause the generator to switch to producing maximum power for the load. Consequently, it needs to be ensured that variables \bar{u} and \underline{u} have a difference of 10V. The value of Δi^s per device is discussed in Chapter 5, where it is more relevant. Our choice of values creates a steep slope that is both reactive to power imbalances, and resistant enough to oscillation problems.

Based on 3.1, the linear equation of the droop curve is reformulated as follows

$$i_m^0 = i_m^s(l+1) + d * u_m(l+1) = \frac{p_m^s(l+1)}{u_m(l+1)} + d * u_m(l+1) \quad (3.50)$$

and, after the droop slope d has been calculated, the local variable updates $p_m^s(l+1)$ 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 $u_m(l+1) < \underline{U}_m$, variable $u_m(l+1)$ is set to the value of the exceeded limit. A more visual representation of the procedure used to derive i_m^0 from the defined droop slope and power/voltage setpoints can be seen in Figure 3.7.

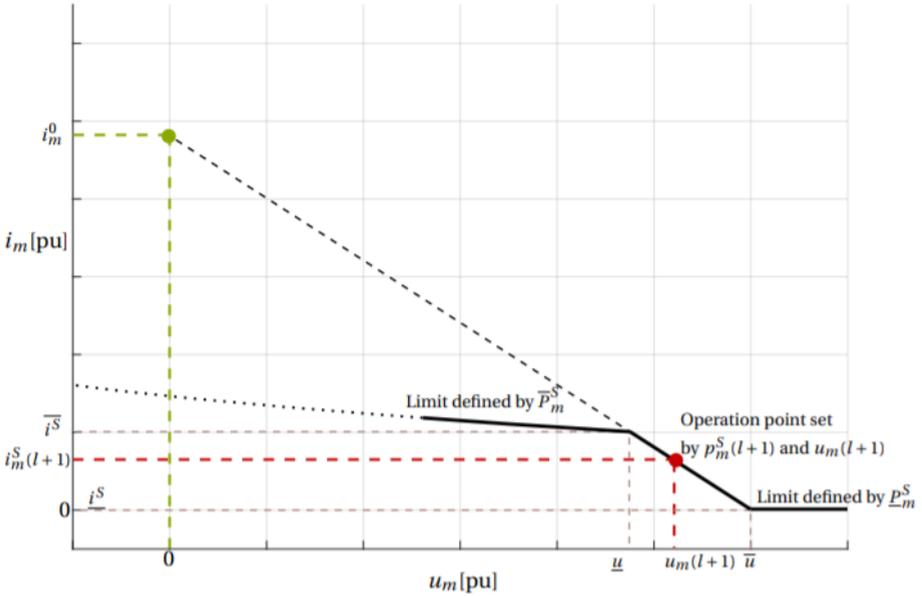


Figure 3.7: A representation of the procedure used to derive i_m^0 from the defined droop slope d and power/voltage setpoints $p_m^s(l+1)$ and $u_m(l+1)$

OPERATING AT THE MAXIMUM OR MINIMUM CURRENT REGION

So far, we have been referring to the node as a computational unit representing a converter. If that computational unit is actually a Power Electronics (PE) interface, then we are given the ability to safeguard the converter against extreme scenarios. In other words, the PE interface can arbitrarily determine the current or power of the converter in order

to protect it. In the case of a converter experiencing maximum or minimum current, the converter is arbitrarily set, by the PE interface, to exert maximum/minimum current, and the voltage is allowed to vary naturally on its own. Once a power limit is reached, we apply the reasoning for the maximum and minimum power regions.

OPERATING AT THE MAXIMUM OR MINIMUM POWER REGION

The main objective here is to safeguard the integrity of the converters, while also keeping the algorithm optimal. In order to achieve both safety and optimality, the variables \underline{u} and \bar{u} can be employed as so-called "soft and hard voltage thresholds", respectively, for both the maximum and minimum power regions. These thresholds can be seen as a subdivision of each operation region into three sub-regions. For example, when in the minimum power region, the converter should not interfere with the provided minimal power, unless the local voltage is under the defined hard threshold. This is necessary because, in the opposite case, small voltage oscillations over the soft threshold would cause the converter to alternate between turning on and off during this "confusion", greatly increasing the overall costs.

Our implementation and setup bypasses the concerns expressed above, disabling a device from completely turning off when at zero power. Thus, the values of \underline{u} , \bar{u} , and, consequently, of the droop slope d , remain the same as they were in the linear region of droop control.

4

THE CONSENSUS AND INNOVATION ALGORITHM FOR UNIPOLAR DC MICROGRIDS: IMPLEMENTATION AND TECHNICAL DETAILS

This chapter is dedicated to describing the design considerations and technical details behind our implementation of the OPF algorithm. Section 4.1 describes the physical components that we used to set up the network and the algorithm. Sections 4.2 and 4.3 focus on the way inter-nodal and inter-layer communication were practically achieved. Finally, Section 4.4 presents a diagram that provides a clearer, visual representation of the implementation and the relevant processes.

4.1. PHYSICAL SETUP: NODES AND GRID COMPONENTS

The OPF algorithm can only be utilized given that the devices of the network can be calibrated, in terms of power, voltage or current, and that those attributes can be effectively measured. As a result, the decision was to use Power Circuit Boards (PCB), designed by DC Opportunities, on devices that were compatible with such equipment and provided those capabilities. Each PCB had an ESP32 micro-processor attached to it, which was acting as the node of the device, as defined in section 3.1.

Regarding the devices themselves, two types of devices were used, one acting as a generator, and another acting as a load. The Mean Well CSP-3000 Power Supplies were used as generators, while custom LED lights were used as loads. These devices were connected to each other through power cables, forming the electric network.

Having described the specific devices and components used for the setup, we may now proceed to describe how the inter-nodal and inter-layer communication was implemented.

4.2. INTER-NODAL COMMUNICATION

Regarding inter-nodal communication, the ESP32s controlling each device use their WIFI capabilities to establish a connection with an Access Point (AP) and then communicate with neighbouring ESP32s using the UDP communication protocol. While UDP communication is faster than TCP communication, it runs the risk of lost data. Hence, in each iteration of the algorithm, if data are not received within a time limit, either because they were lost or because the timeout was reached, then the node expecting that data uses the data received in the last iteration instead. By establishing an appropriate time limit, it can be ensured that the effectiveness of the algorithm is not harmed by the amount of data lost during inter-nodal communication.

4.2.1. NEIGHBOUR DISCOVERY

In order for a node to communicate with its neighbours, it has to know who these neighbours are, a process known as "neighbour discovery". Even though the devices themselves are physically connected through an electric cable, that does not provide a way for the ESP32 controlling one device to somehow know which ESP32 is controlling the neighbouring device. Thus, a method for "neighbour discovery" had to be devised.

One idea we came up with was to infer the distance between ESP32s using signal strength; it is quite reasonable to assume that the neighbouring devices of a node are

closer to it than its non-neighbouring ones. The ESP32 possesses Bluetooth technology, and it is possible to infer distance using Bluetooth signal strength[5][11], however, there are quite a lot of disadvantages that make it ineffective in our case, such as flight path interference, emission power, receiver sensitivity, and noise in the measurements. Bluetooth can be used, however, where accuracy is of low priority, and where low power consumption and low implementation cost is of higher importance. In the opposite case, using WIFI signal strength is a much better alternative. Specifically, a special method often-used is called "WIFI fingerprint positioning", where the Received Signal Strength Indication (RSSI) of each device from the AP is collected, and then a strategy is used to estimate the distance between two devices by using their RSSIs. Some of the available strategies include k-nearest neighbour techniques[16], random forest learning[22], and supervised machine learning[23]. Eventually, we decided that the complexity involved in using WIFI fingerprint positioning for neighbour discovery was unnecessary, compared to the custom solution we devised next.

The neighbour discovery method we eventually adopted involved a central coordinator, i.e. a server, that would be aware of the network layout and the neighbours of each device. Of course, since our OPF algorithm is decentralized, the centralized server is only used during the initialization phase so that we avoid having to configure each ESP32 separately. During the initialization phase of the algorithm, each ESP32 contacts the server and informs it of two things: its MAC address and its IP address. The server, being aware of the MAC address corresponding to each node, informs that node about the IP addresses of its neighbours. The server also provides the necessary initialization data for the algorithm.

4.3. INTER-LAYER COMMUNICATION

Inter-layer communication depends heavily on the specifics of each device in question. The Mean Well CSP3000 Power Supplies used as generators can only have their voltage levels configured, and that is possible through Pulse-Width Modulation (PWM) signals. Hence, the node-to-grid direction of communication occurs through the use of PWM signals and duty cycles. As for the grid-to-node direction of communication, it occurs through Analog-to-Digital Conversion (ADC) measurements, retrieving the device's voltage and current levels and, by extension, its power levels. As for the LED lights used as loads, both directions of communication utilize a microcontroller on their PCB, acting as a mediator between the ESP32 and the LED light. Through the microcontroller, the current levels of the LED light can be configured, while the measurements retrieved include the voltage and current levels and, by extension, its power levels.

4.4. DIAGRAM OF IMPLEMENTATION

The basics behind the technical implementation of the algorithm have been explained, and diagram 4.1 can now provide a clearer, visual representation of the whole process. Since the algorithm itself has already been explained in detail in chapter 3, we will only be making a few points about the diagram that indicate any innovation in its architec-

ture:

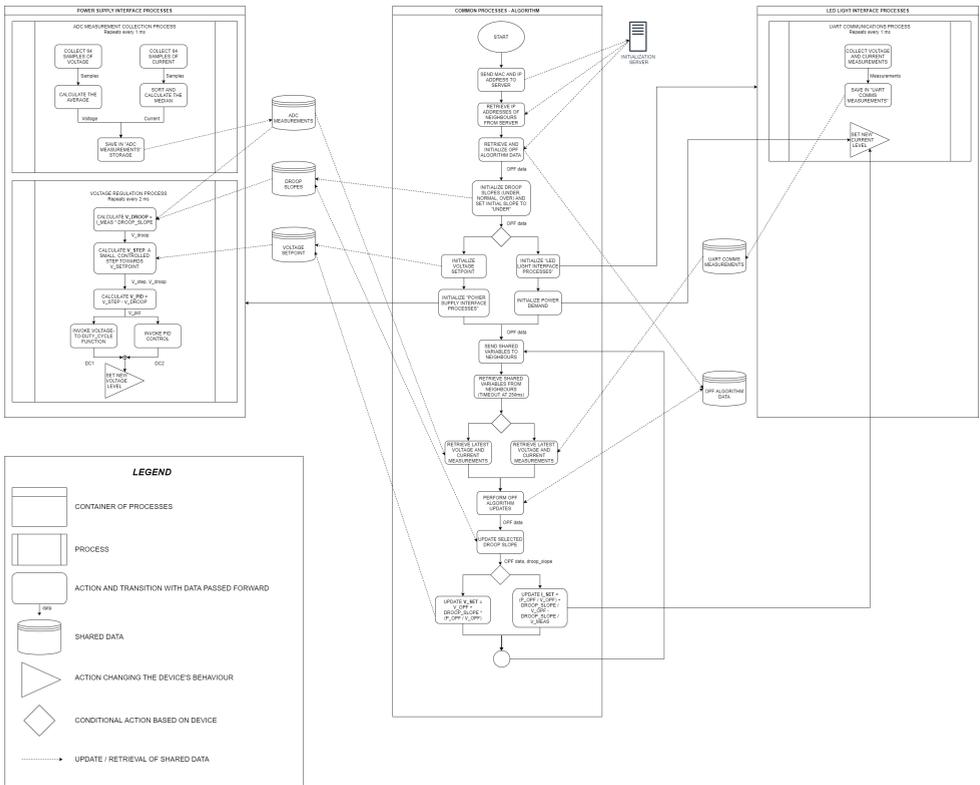


Figure 4.1: The diagram of our implementation of the OPF algorithm

- The implementation consists of 3 parallel sets of processes: the central set is responsible for running the algorithm itself, and it is the common set of processes that all devices run upon initialization of the algorithm. The other 2 sets are dedicated to one type of device each, and are responsible for the inter-layer communication occurring between the node and the device it controls.
- The set of processes responsible for the power supply continuously runs PID control in order to regulate the voltage and keep it stable across varying current levels. The PID control receives its feedback from a parallel process that continuously updates the measured voltage levels. In order for the PID control to avoid the slow transition from one voltage level to another, when the difference between the two is big, we incorporated a voltage-to-duty-cycle function into the process, that we created through interpolation. The function approximates the intended voltage, and then the PID control takes over, performing small, controlled steps in order to smoothly reach the intended voltage level.

- Whenever a new voltage setpoint is decided by the algorithm (V_{OPF}), it is re-evaluated through droop control based on the theoretical current setpoint, producing the theoretical voltage setpoint V_{SET} . The PID control process is then responsible to make sure that the voltage setpoint is met fast and steadily, but while also applying its own droop control to re-evaluate the theoretical voltage setpoint based on the actual current levels. For both applications of droop control, the appropriate droop curve is used, the one that has been established by the algorithm.
- The set of processes responsible for the LED light continuously retrieve the voltage and current measurements through a UART channel, connecting the ESP32 and the microcontroller, the mediator. Similarly to the case of the power supplies, the current setpoint decided by the algorithm (I_{OPF}) is re-evaluated through droop control based on the theoretical voltage setpoint, producing the theoretical current setpoint, which is eventually also re-evaluated based on the retrieved actual current levels, resulting in I_{SET} .
- The algorithm achieves fast execution through the parallel coordination of the aforementioned processes. The "ADC measurements collection" process repeatedly runs with intervals of 1ms, with the "Voltage regulation" process running with 2ms intervals to ensure that at least one new set of measurements has been collected before its next run. The "UART communications" process also runs with a 1ms interval. Furthermore, the fact that all the grid measurements are continuously collected in parallel to the algorithm ensures that there is zero delay between the moment the algorithm requests the latest measurements and the moment they are available for retrieval, i.e., zero delay during grid-to-node communication. This delay has been a concern in the work of Pedro Parreira[17], who last worked on the simulated algorithm, and it has been effectively eliminated in this implementation of it. Another concern in the work of Parreira was the possible loss of data during inter-node communication. Specifically, Parreira attempted to give timeouts of 300ms and 500ms to the inter-node communication process, but none of them were sufficient; each iteration took quite longer to complete, because of the time required to simulate the grid, resulting in a non-converging algorithm. Hence, he decided to have no timeouts and simply artificially dismiss 25% of the communication, which rendered the algorithm functioning and converging again. In our case, we impose a 250ms timeout to the inter-node communication, which has been shown to cause less than 5% of the communication data to be lost, and our algorithm remains effective. Thus, both the timeout and data loss aspects of the inter-node communication have been improved.

5

CASE STUDIES

In this chapter, we present some case studies with the purpose of demonstrating the effectiveness of the algorithm under various different scenarios. The chapter is split into 3 sections, where Section 5.1 is concerned with 2-node networks of one generator and one load, and Section 5.2 focuses on 3-node networks of two generators and one load. In Figure 5.1, we can see the two Mean Well CSP3000 power supplies used as the generators, and the LED light used as the load. Section 5.3 shares some concluding remarks.

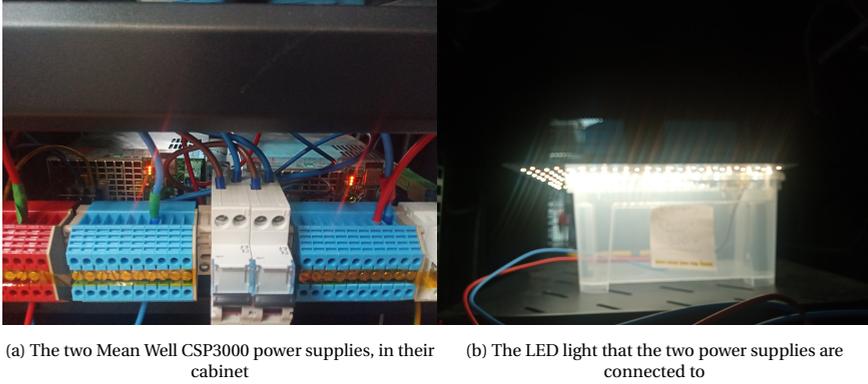


Figure 5.1: The two Mean Well CSP3000 power supplies used as the generators, and the LED light used as the load

5.1. ONE GENERATOR AND ONE LOAD

The setup consists of one Mean Well CSP3000 power supply and one LED light, connected to each other. Since we experienced quite a lot of current fluctuations at very low current levels, we added some extra resistance in the connection between the two devices, having the initial current start at about $1.25A$.

Both the power supply and the LED light initialize their voltage levels at $u_m = 350V$, with a lower voltage limit of $\underline{U}_m = 340V$ and an upper voltage limit of $\overline{U}_m = 360V$. At the start of the algorithm, all generators must be off, i.e., $p_m = 0W$, and have a lower power limit of $\underline{P}_m = 0W$ and an upper power limit of $\overline{P}_m = 3000W$. As for the LED light, we initialize its lower power limit to $\underline{P}_m = -85W$ because, by construction, the maximum current that it can achieve is just shy of $0.250A$, and because the lower voltage limit is $340V$. Note that we care about the lower voltage limit when deciding the maximum power demand, not the upper one, because we want the power demand to be satisfiable in any voltage range, with the algorithm deciding what the optimal voltage is for optimality, not by necessity. The cost coefficients $B_m = 0$ and $A_m = 0$ were assigned to the LED light, meaning there is no cost function related to power consumption, and the load will always request for full power. As a result, the upper power limit is also set to $\overline{P}_m = -85W$. The power demand at initialization is $p_m = -85W$, since all loads request full power at startup. The cost coefficients $B_m = 3.0$ and $A_m = 0$ were assigned to the power supply.

EXPERIMENT 1 - NORMAL

In this first experiment, we aimed to observe how the power supply starts providing power to the LED once it enters its Marginal Power Region. With this first experiment, we also aimed to test the speed of our implementation in terms of iterations per second. The results of the first experiment are presented in Figure 5.2.

Looking at the graph of the current, we can observe that the current of the power supply starts at 1.25A and slowly increases, because of the resistors in the line heating up. The current of the LED light starts at the maximum possible 0.25A. The equivalent behavior occurs with the measured power from both devices. Unfortunately, by construction of the network, the requested current is immediately satisfied by the network, even though the power setpoint p_m of the power supply is still at zero. This unwanted behavior is eliminated in later, larger experiments with different setups. Regardless, once the LMP of the power supply reaches its lower threshold, $B_m = 3$, the power supply starts supplying the power needed to match the currently measured power in the network, and the LMP stabilizes, since the power generated has not reached the upper limit. The reason the LMP does initially increase is mainly attributed to the power mismatch constantly experienced by the power supply due to the added resistance, and not so much on the power request by the LED light, as the latter is immediately satisfied. This is also a reason the LMP grows relatively slowly, reaching the lower threshold within 10 seconds. It is worth noticing that the LMP of the LED light also stabilizes, as a result of the power supply LMP holding it down to its level.

The voltage level of the power supply drops to the lower limit, 340V, and the voltage of the LED light follows accordingly. This occurs because the voltage has a tendency to drop when there is a request for power or increasing current, to allow for that to happen.

Finally, we can observe that the algorithm performs about 500 iterations in a matter of 10 seconds, which translates to about 50 iterations per second. We consider this quite a good speed, given that the best indication of the algorithm's hypothetical optimal speed comes from the simulated algorithm in Pedro Parreira's work[17], where the simplest 2-3 node networks have a runtime of 12 seconds for 500 iterations; with about 8 of those seconds being attributed to the simulation of the grid mechanics, we are left with 4 seconds for the actual algorithm, coming down to 125 iterations per second. If we do not optimistically leave out the time for simulating the grid, we end up with 40 iterations per second, which is less than the speed achieved by our implementation.

EXPERIMENT 2 - LED LIGHT TURNS ON HALFWAY THROUGH

The next experiment with this setup involved having the LED off and only turning it on halfway through the process, to observe the impact this shift in power demand would have on the algorithm. The results of the experiment are presented in Figure 5.3.

We can see that the results are similar to the previous experiment, only this time, the LMP starts growing only once the LED turns on, at about 40 seconds into the process. This happens because, although in the beginning there is still a power mismatch that drives the LMP of the power supply up, the LMP of the LED light remains at zero, and thus pushes the power supply LMP down with it. Furthermore, the moment the LED

light turns on, we see a sudden spike in the measured power, current and voltage, all of which stabilize immediately as the power demand is satisfied.

EXPERIMENT 3 - $\underline{P}_m = -40W$ FOR THE LED LIGHT

This next experiment shown in Figure 5.4 is the same as the last one, only the power demand of the LED has been halved, down to $\underline{P}_m = 40W$. The purpose of this experiment is to verify that the power demand can be adjusted as desired, with the algorithm behaving in the same way.

Indeed, aside from the difference in the power setpoint, measured power and measured current, the rest of the data remains the same.

EXPERIMENT 4 - $\bar{I}_{mn} = 1.5A$

The next experiment, Figure 5.5, is identical to the second experiment, only the upper current limit of the line between the power supply and the LED light has been artificially capped to $\bar{I}_{mn} = 1.5A$. The purpose of this experiment is to observe the different behavior of the algorithm in case there is current congestion in a certain line.

Because the change is artificial, we can observe the current growing past that 1.5A limit, however, the moment the limit is reached (around the 30sec mark), the dual variable for the line current in that direction, μ_{mn} , starts growing. Since the line experiences no relief, the dual variable keeps growing, giving more weight to the LMP of the LED light. As a result, the LMP of the LED light manages to "escape" the other LMP, which can no longer hold it down to its level. This is desirable behavior because, in the case of current overflow in a line, we want the receiving end to consume or pass on that current, and the sending end to lower its current output so that the current in the line is limited back to acceptable levels. By having the LMP of the receiving end increase, and the LMP of the sending end decrease, we are trying to achieve exactly that.

EXPERIMENT 5 - $B_m = 2.0$ AND $A_m = 0$ FOR THE LED LIGHT

In the next experiment, Figure 5.6, we actually gave a cost function to the load, with cost coefficients $B_m = 2.0$ and $A_m = 0$. This also requires us to switch the upper power limit to $\bar{P}_m = 0W$, so that the LED light can turn off if consuming any power is impractical. The experiment is otherwise identical to the first experiment, but now that the load actually has a cost function, we expect it to eventually turn off once its LMP exceeds the upper threshold of its cost derivative, $B_m = 2.0$.

We can observe that, at about the 10sec mark, the LMP of both the devices reaches value $\lambda_m = 2.0$. At that moment, and being a load with a linear cost function, the LED light immediately turns off, as the LMP keeps growing towards the lower threshold of the power supply, where it stabilizes. Counter-intuitively, the power supply does start and keeps providing power even though the LED is now off, because of its own power mismatch that is it experiencing due to the added resistance, that keeps its LMP high.

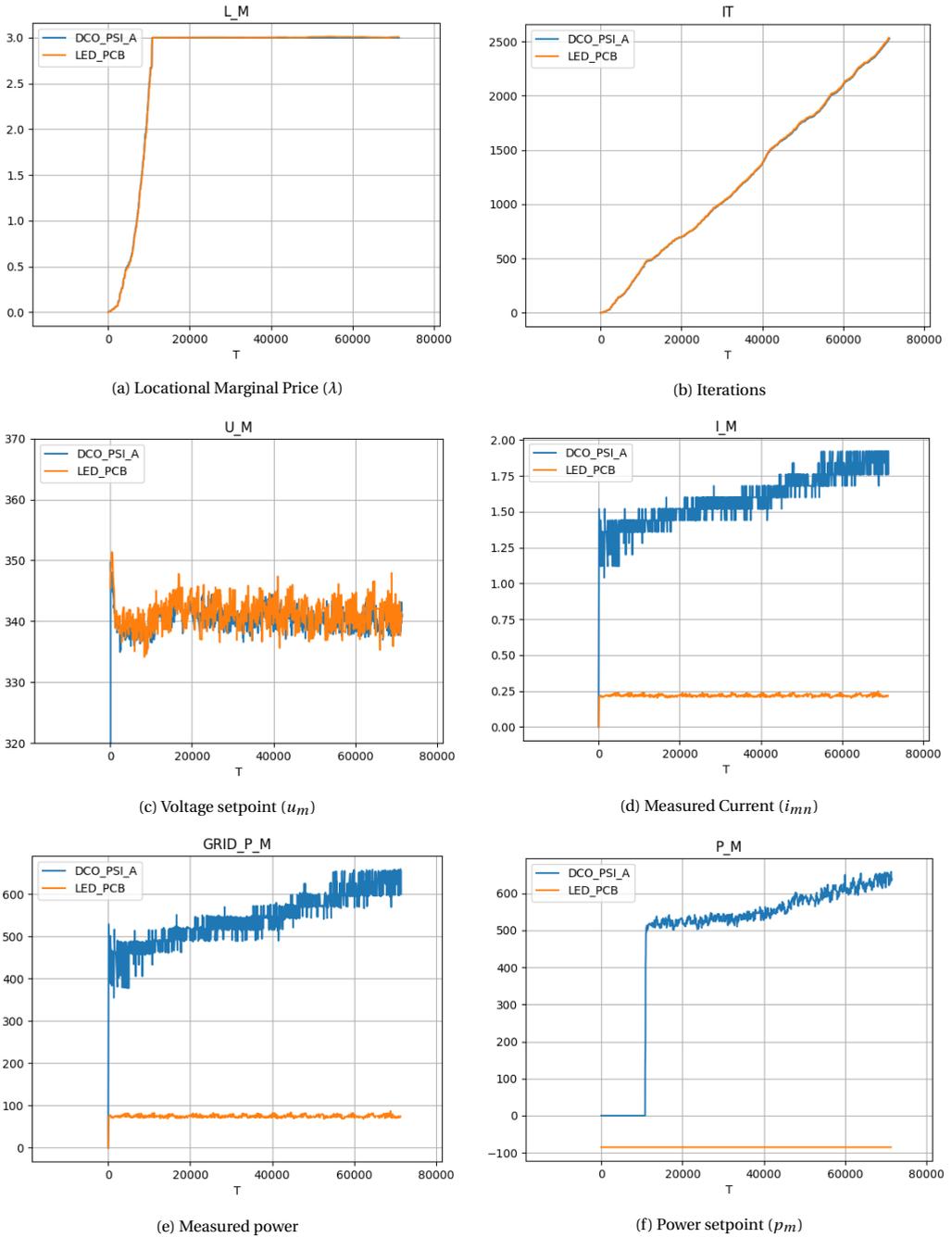
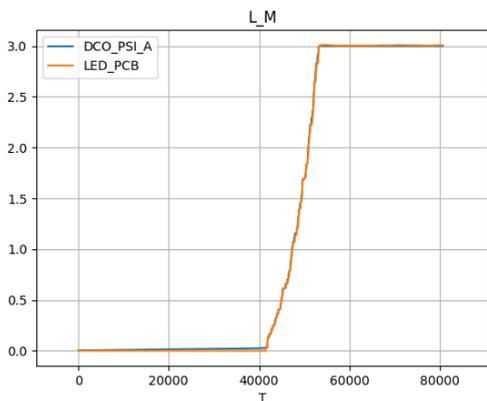
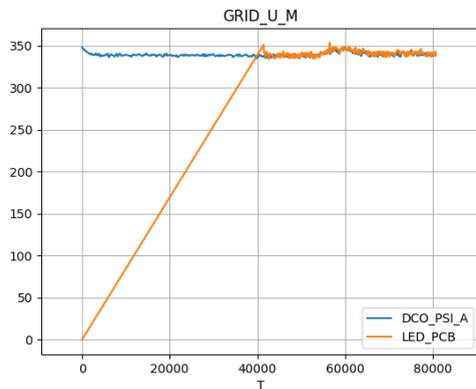


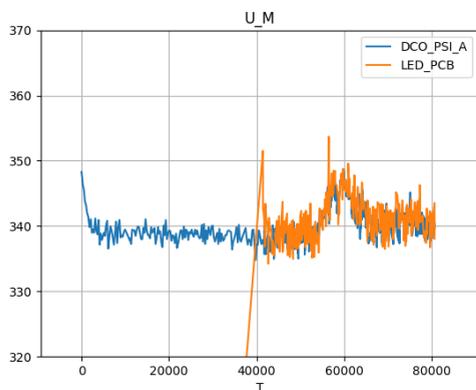
Figure 5.2: The value of various variables over time (ms)
Experiment 1 - Normal



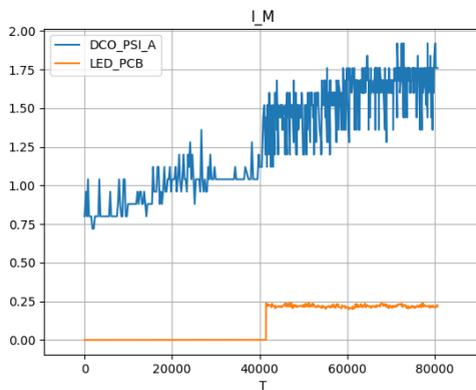
(a) Locational Marginal Price (λ)



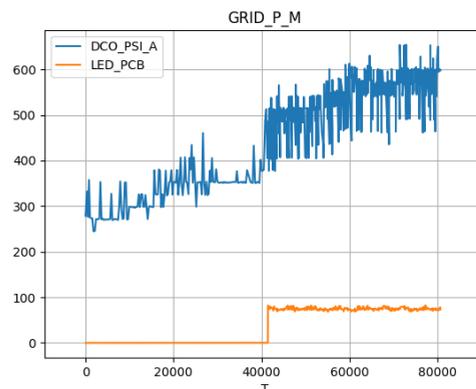
(b) Measured voltage



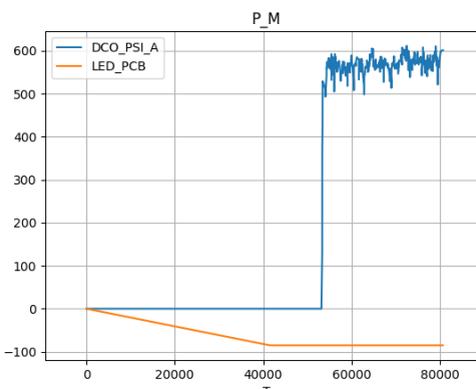
(c) Voltage setpoint (u_m)



(d) Measured Current (i_{mn})

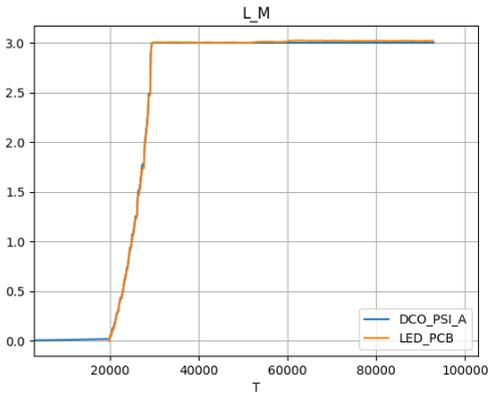


(e) Measured power

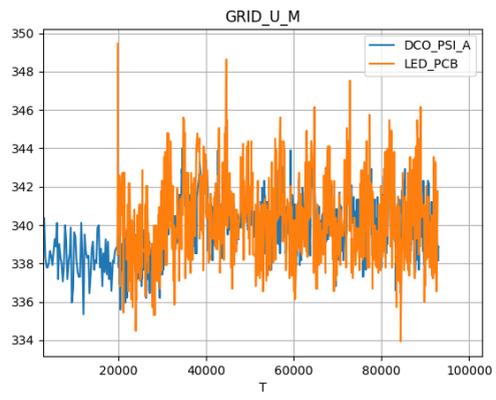


(f) Power setpoint (p_m)

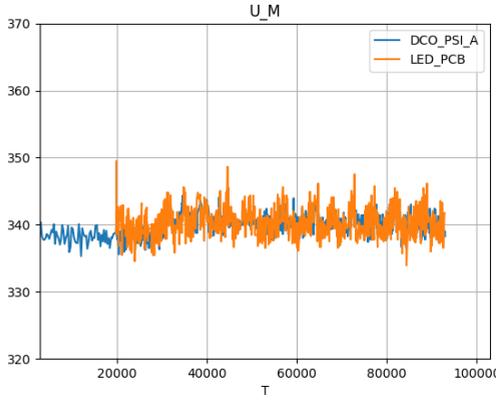
Figure 5.3: The value of various variables over time (ms)
 - Experiment 2 - LED light turns on at the 40sec mark



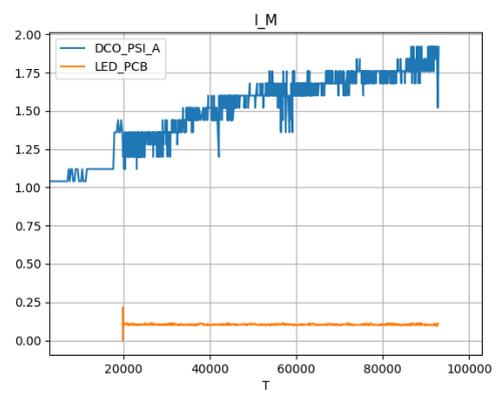
(a) Locational Marginal Price (λ)



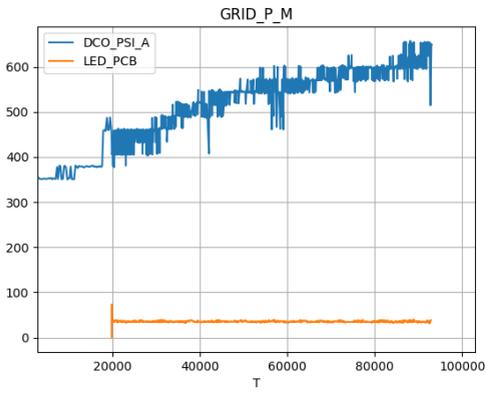
(b) Measured voltage



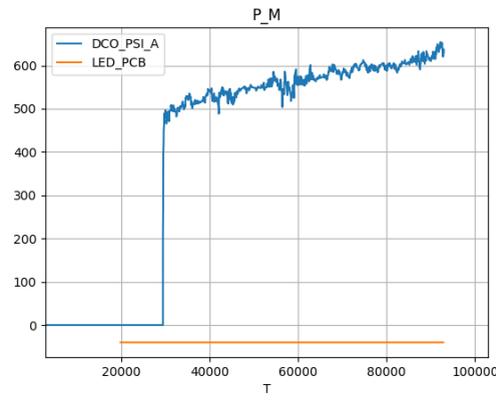
(c) Voltage setpoint (u_m)



(d) Measured Current (i_{mn})

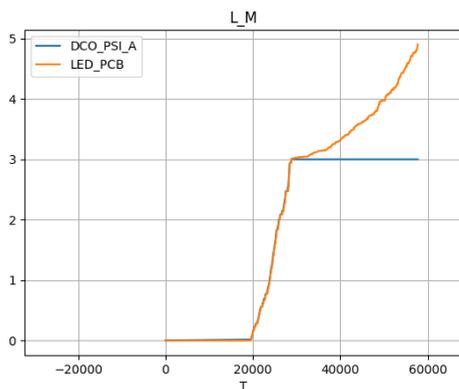


(e) Measured power

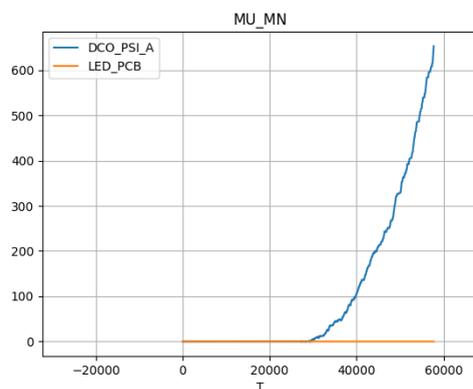


(f) Power setpoint (p_m)

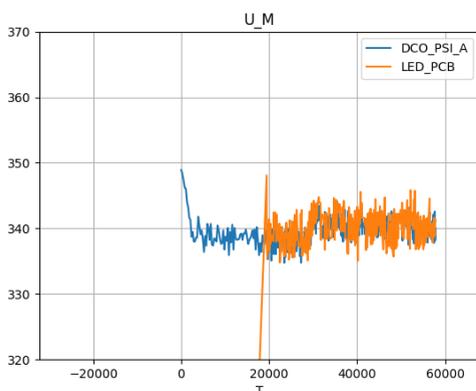
Figure 5.4: The value of various variables over time (ms)
Experiment 3 - $P_m = -40W$ for the LED light



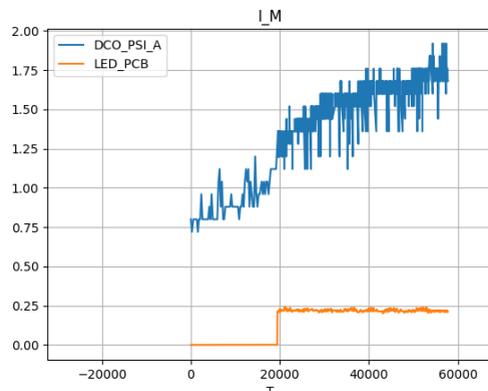
(a) Locational Marginal Price (λ)



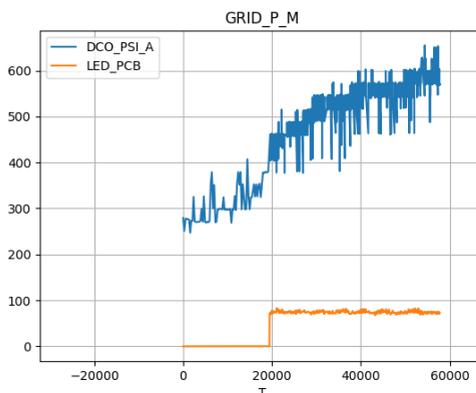
(b) Dual variable for line current (μ_{mn})



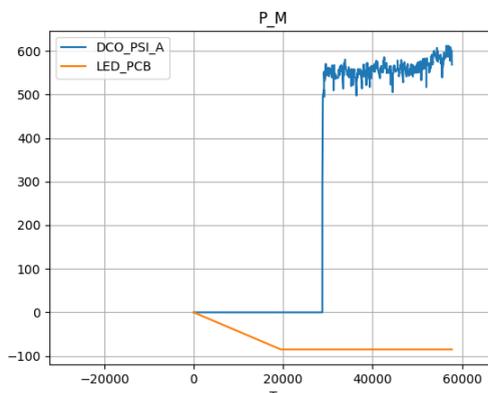
(c) Voltage setpoint (u_m)



(d) Measured Current (i_{mn})

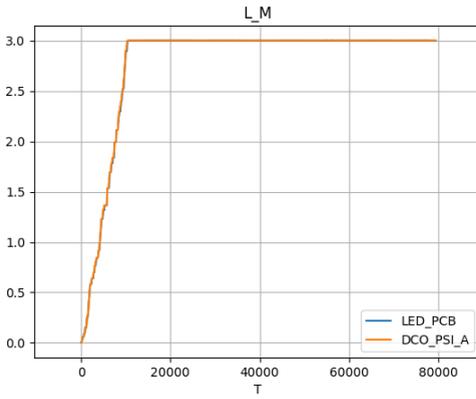


(e) Measured power

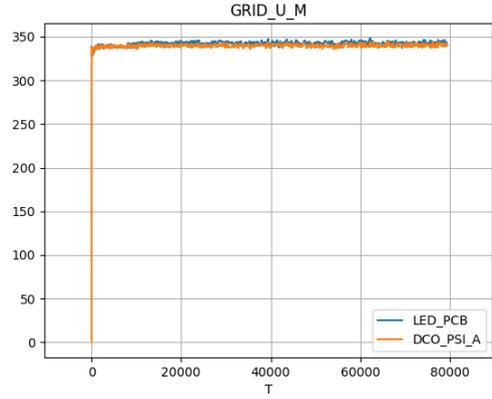


(f) Power setpoint (p_m)

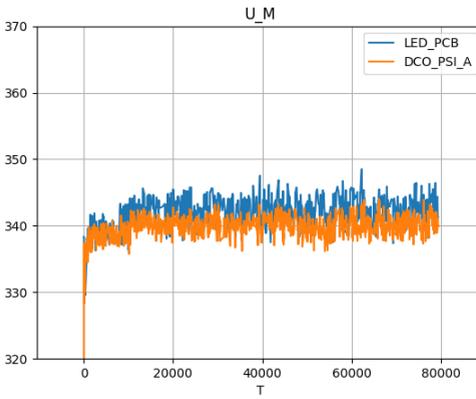
Figure 5.5: The value of various variables over time (ms)
- Experiment 4 - $\bar{I}_{mn} = 1.5A$



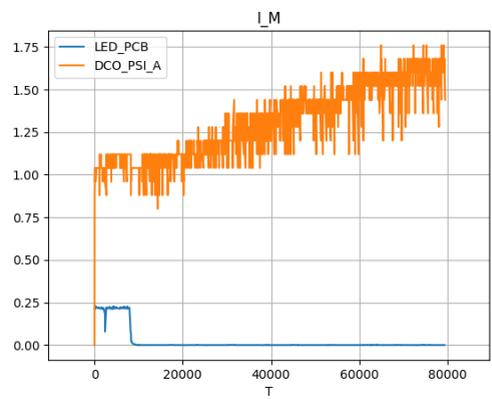
(a) Locational Marginal Price (λ)



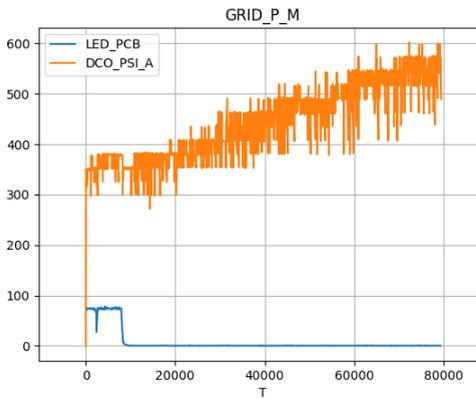
(b) Measured voltage



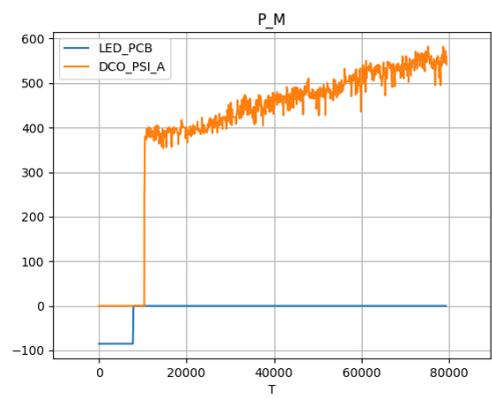
(c) Voltage setpoint (u_m)



(d) Measured Current ($i_{m,n}$)



(e) Measured power



(f) Power setpoint (p_m)

Figure 5.6: The value of various variables over time (ms)
 - Experiment 5 - $B_m = 2.0$ and $A_m = 0$ for the LED light

5.2. TWO GENERATORS AND ONE LOAD

The setup consists of two Mean Well CSP3000 power supplies and one LED light, with each power supply connected only to the LED light. Having performed some hardware upgrades, the current fluctuations experienced in the previous set of experiments no longer existed, and thus there was no need for the extra resistance this time around. This change also affected the growth rate of the LMP, which made us change the cost coefficients of each device from integer values to decimals between 0 and 1. Finally, it is worth noting that the graphs depicted as "Line Current i" present the current measured at the line connecting that device to its "i"th neighbour.

Both the power supplies and the LED light initialize their voltage levels at $u_m = 350V$, with a lower voltage limit of $\underline{U}_m = 330V$ and an upper voltage limit of $\overline{U}_m = 370V$. At the start of the algorithm, all generators must be off, i.e., $p_m = 0W$, and have a lower power limit of $\underline{P}_m = 0W$ and an upper power limit of $\overline{P}_m = 3000W$. As for the LED light, we initialize its lower power limit to $\underline{P}_m = -165W$ because, by construction, the maximum current that it can achieve is just shy of $0.5A$, and because the lower voltage limit is $330V$. The cost coefficients $B_m = 0$ and $A_m = 0$ were assigned to the LED light, meaning there is no cost function related to power consumption, and the load will always request for full power. As a result, the upper power limit is also set to $\overline{P}_m = -165W$. The power demand at initialization is $p_m = -165W$, since all loads request full power at startup. The cost coefficients $B_m = 0.03$ and $A_m = 0$ were assigned to the first power supply, and $B_m = 0.06$ and $A_m = 0$ to the second one.

NORMAL

In this first experiment, we simply aim to observe only the cheap power supply provide the requested power. The results of the first experiment are presented in Figure 5.7.

At startup, similarly to the case of the previous set of experiments, the power demand is immediately satisfied by the joint and balanced efforts of the two power supplies. This can be observed through the graphs of the two line currents, the measured current and the measured power. Once the lower threshold of the cost derivative for the cheap power supply (orange line) is reached, the LMP stabilizes, and the cheap power supply is instructed to provide a constant amount of power to satisfy the power demand. Unfortunately, the actual power and current flow in the network does not shift, because of the voltage equivalence that naturally occurs between the power supplies. This behavior can be altered by artificially manipulating the voltage difference between the two generators, as we will see in a later experiment. As of this point, however, the experiment fails to show the cheap power supply take over the power generation, and only presents that the algorithm instructs it to do so.

LED LIGHT TURNS ON AT THE 27SEC MARK

This experiment, seen in Figure 5.8, is identical to the last one, with the only difference being that the LED light is turned on at about 27 seconds into the experiment, where both power supplies transition from supplying no power to providing a balanced amount each for the LED light.

CHEAP GENERATOR HAS A LOWER VOLTAGE LIMIT OF 350V

For this experiment, Figure 5.9, we set the lower voltage limit of the cheap power supply (orange line) to $\underline{U}_m = 350V$, instead of the previous value of 330V. This twitch has the purpose of showing how a voltage difference between the two power supplies can give priority to one power supply over the other, when it comes to power generation.

We can now observe that, once the LED light turns on at about 17 seconds, the cheap power supply starts providing all the necessary power, with the other power supply staying off, something caused by the voltage difference established. And when the LMP reaches the lower threshold of the cost derivative of the cheap power supply, the cheap power supply is instructed to provide the full power required, after which the system converges.

$\bar{P}_m = 50W$ FOR THE CHEAP GENERATOR

This experiment, Figure 5.10, focuses on showing how the expensive power supply also contributes to power generation, if the cheap one can not satisfy the full power demand. For that purpose, we set the upper power limit for the cheap power supply to $\bar{P}_m = 50W$.

We can observe that, even though the $B_m = 0.03$ threshold of the cheap power supply is reached by the LMP, the algorithm notices that it cannot provide the 75-80W required, and so the LMP keeps growing, reaching the threshold of the second power supply, which contributes along with the first power supply towards the total power generation.

$\bar{I}_{23} = 0.15A$

This experiment, Figure 5.11, focuses on artificially limiting the maximum current that can pass through the line connecting the expensive power supply to the LED light, so that the efforts of eliminating that overflow can be observed. Hence, we set $\bar{I}_{23} = 0.15A$.

Although the LMP of the LED light reaches the threshold of the cheap power supply, the overflow of current drives it so high that the LMP of the cheap power supply cannot keep it to its level anymore. Additionally, the LMP of the expensive power supply keeps being pushed down and remains at zero. This is the expected behavior, since, in any overflow line, the power at the origin-side of the current is limited, while the power consumption at the receiving end is increased, in order to relieve the line of the excess current. Finally, we can observe the voltage growing and reaching the upper limit for the first time, since there is a non-stop push for relief of the overflow line, something that never occurs because it is an artificial change.

DYNAMICALLY CHANGING POWER DEMAND

The experiment shown in Figure 5.12 simply had the purpose of showing that power demand can be manipulated at random times, and that the system appropriately adjusts to these changes.

As we can see from the graphs of the power setpoint and measured power/current, whenever the power demand changes (green line), so does the instructed power by the algorithm, and so does the provided/measured power and current.

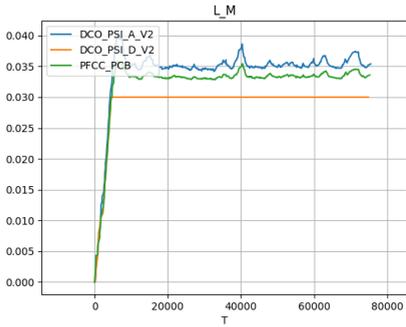
DYNAMICALLY CHANGING COST

For this final experiment, Figure 5.13, we waited until the LMP had stabilized at the lower threshold of the cheap power supply, and then, at the 20th second, we changed the cost coefficient of that power supply from $B_m = 0.03$ to $B_m = 1$. The idea is to watch the cheap power supply, that has now turned into an expensive one, to give up all power generation, and for the LMP to increase, reaching the next-best power supply, which now takes over the power generation.

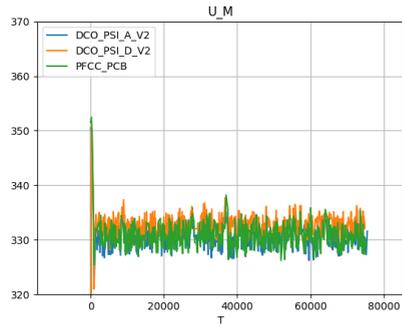
Indeed, our move caused the LMP to be freed from its stable state, the power setpoint of the cheap power supply went down to zero, and the power generation was taken over by the expensive power supply, at the same power setpoint initially set for the cheap one.

5.3. CONCLUDING REMARKS

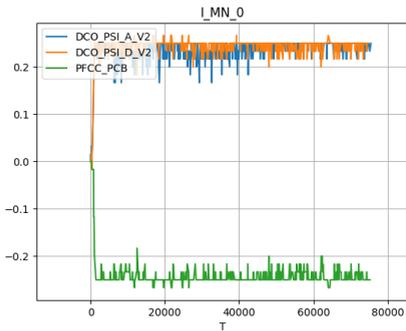
We have tested out various scenarios on the 2 and 3-node setups, verifying that the algorithm is indeed working as expected. We have also achieved fast algorithm execution, with a rate of about 50 iterations per second, and identified that the growth rate of the LMP is the biggest factor affecting convergence time. Due to our setup, we have not managed to demonstrate that the power flowing through the network is directly affected by our algorithm, which was an important goal of ours; that achievement is demonstrated in a future chapter, using the same algorithm on a larger, more appropriate setup.



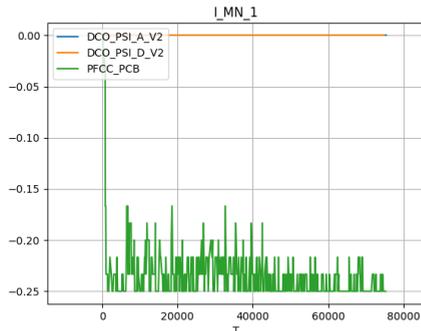
(a) Locational Marginal Price (λ)



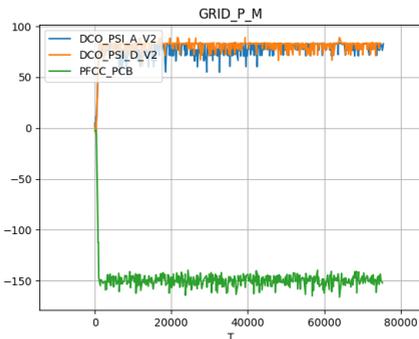
(b) Voltage setpoint (u_m)



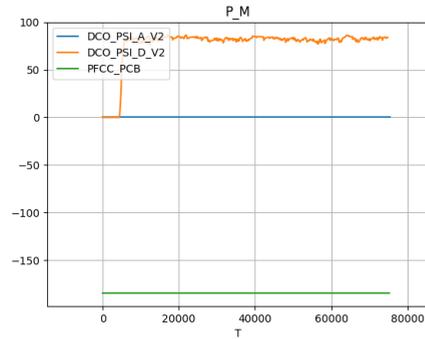
(c) Line Current 1



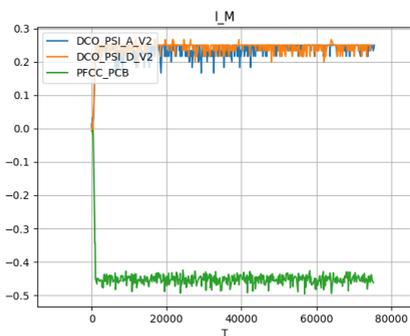
(d) Line Current 2



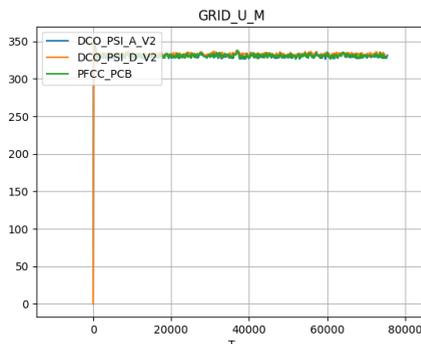
(e) Measured Power



(f) Power setpoint (p_m)



(g) Measured Current



(h) Measured voltage

Figure 5.7: The value of various variables over time (ms)
- Experiment 1 - Normal

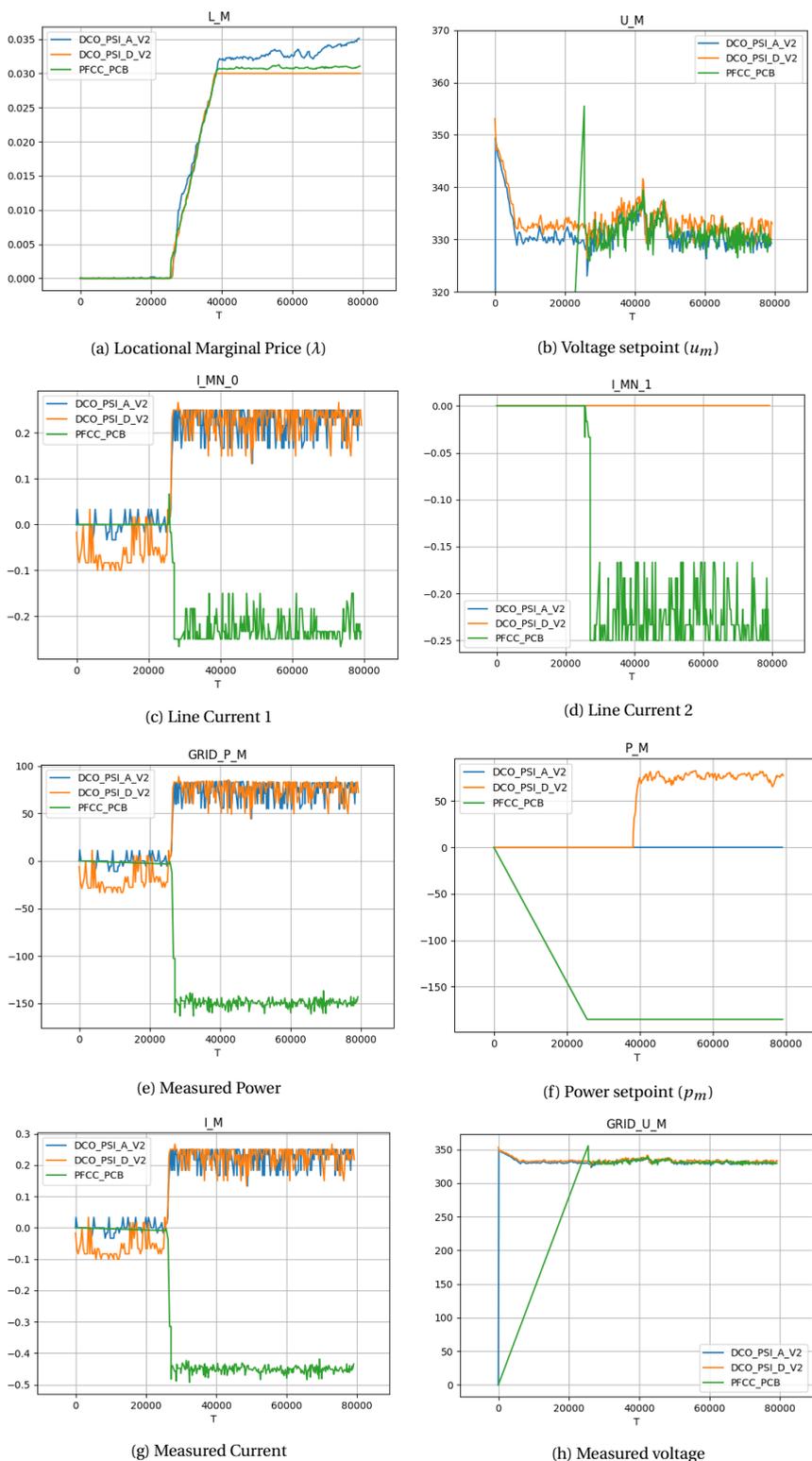
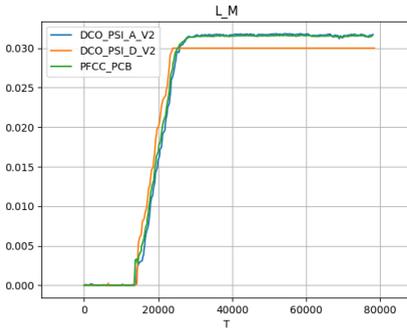
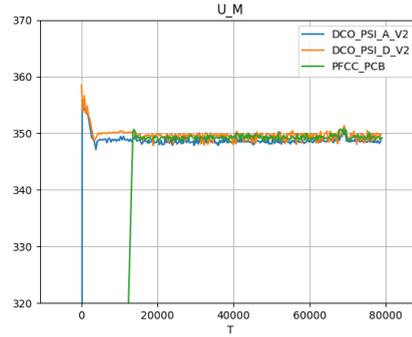


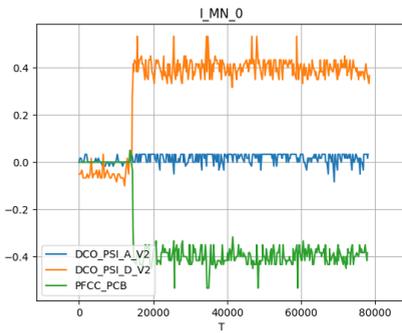
Figure 5.8: The value of various variables over time (ms)
- Experiment 2 - LED light turns on at the 27sec mark



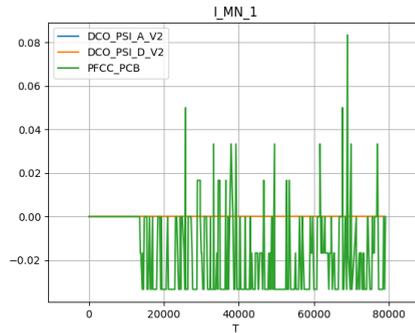
(a) Locational Marginal Price (λ)



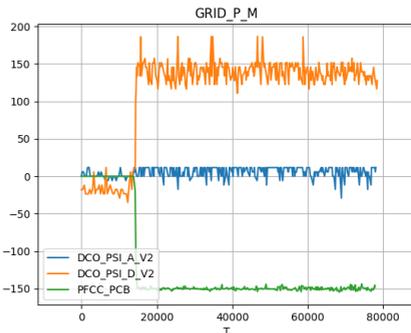
(b) Voltage setpoint (u_m)



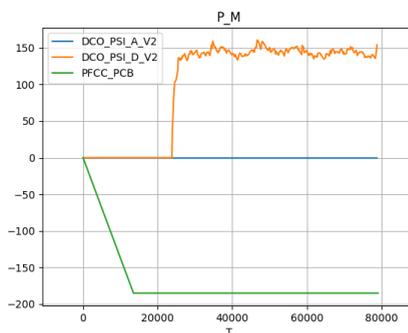
(c) Line Current 1



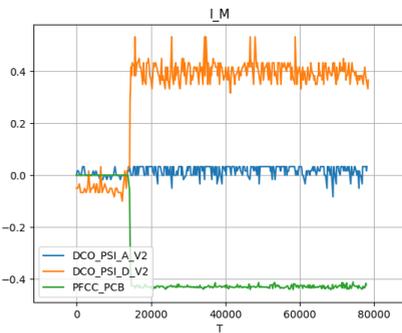
(d) Line Current 2



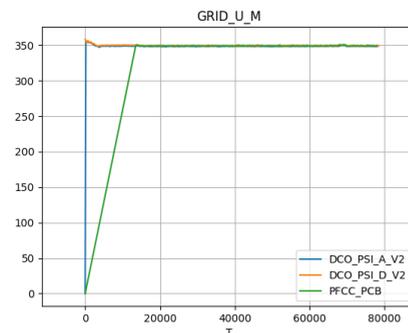
(e) Measured Power



(f) Power setpoint (p_m)



(g) Measured Current



(h) Measured voltage

Figure 5.9: The value of various variables over time (ms)
 - Experiment 3 - Cheap generator has a lower voltage limit of 350V

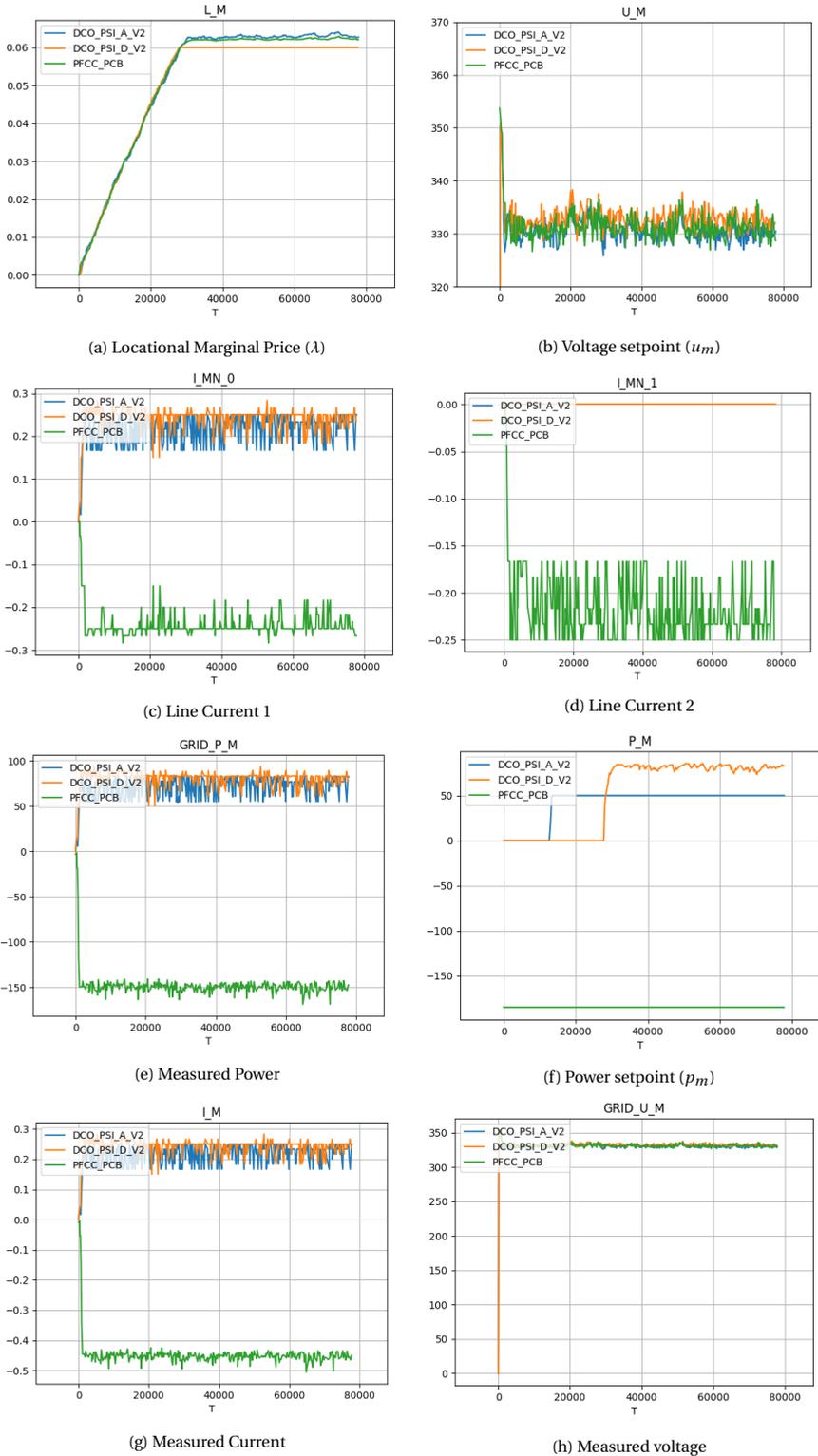
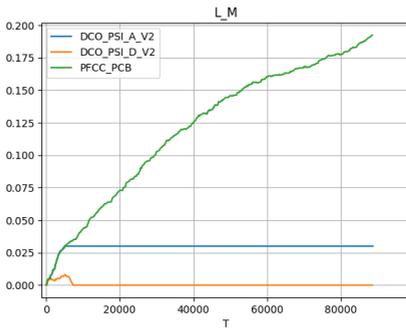
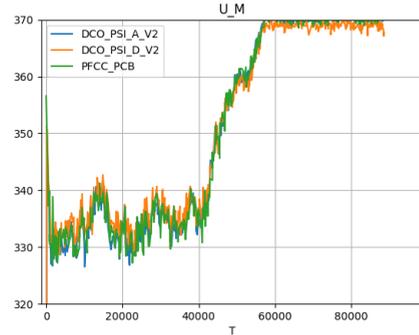


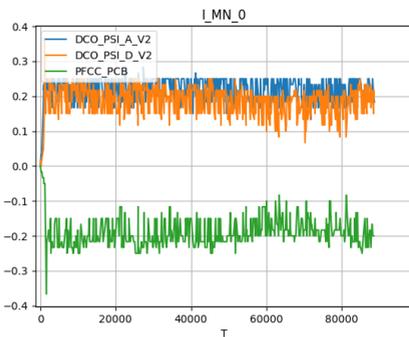
Figure 5.10: The value of various variables over time (ms)
 - Experiment 4 - $\bar{P}_m = 50W$ for the cheap generator



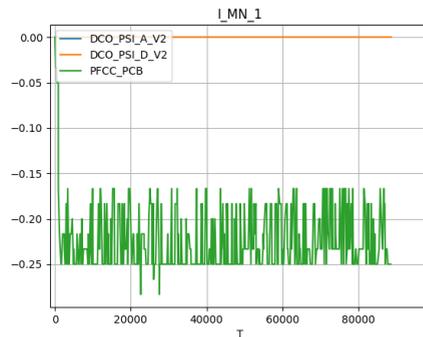
(a) Locational Marginal Price (λ)



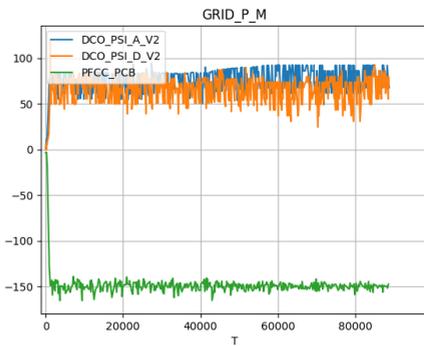
(b) Voltage setpoint (u_m)



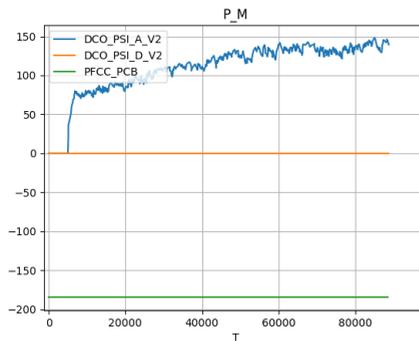
(c) Line Current 1



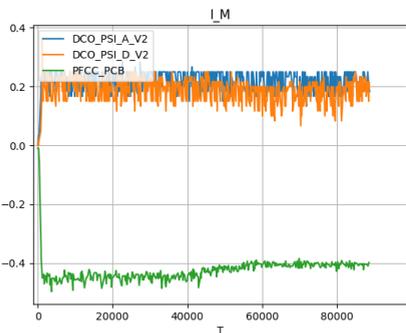
(d) Line Current 2



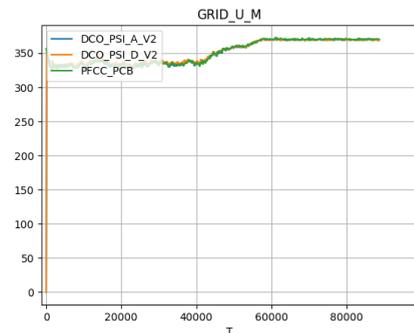
(e) Measured Power



(f) Power setpoint (p_m)



(g) Measured Current



(h) Measured voltage

Figure 5.11: The value of various variables over time (ms)
 - Experiment 5 - $I_{23} = 0.15A$

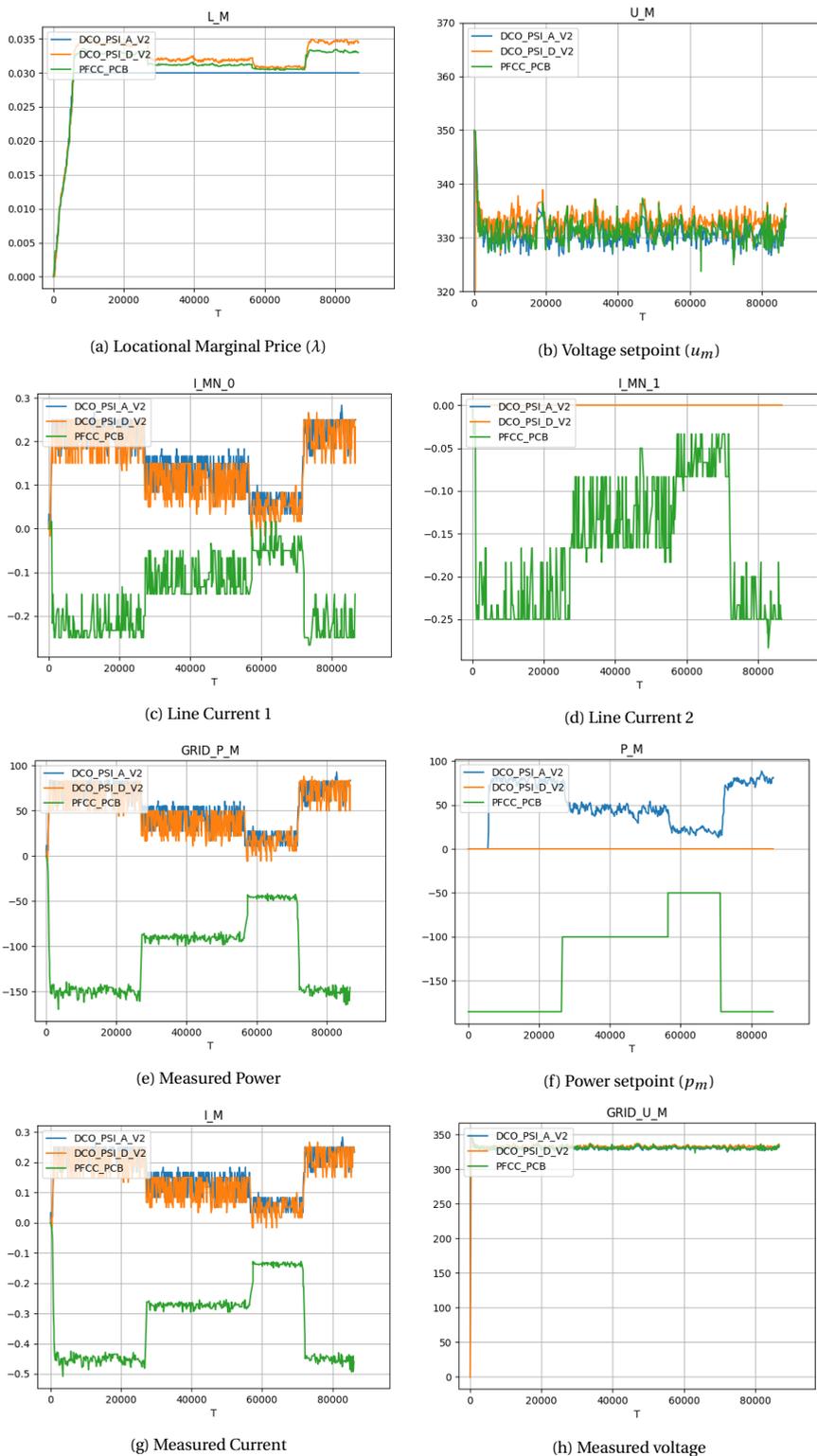


Figure 5.12: The value of various variables over time (ms)
- Experiment 6 - Dynamically changing power demand

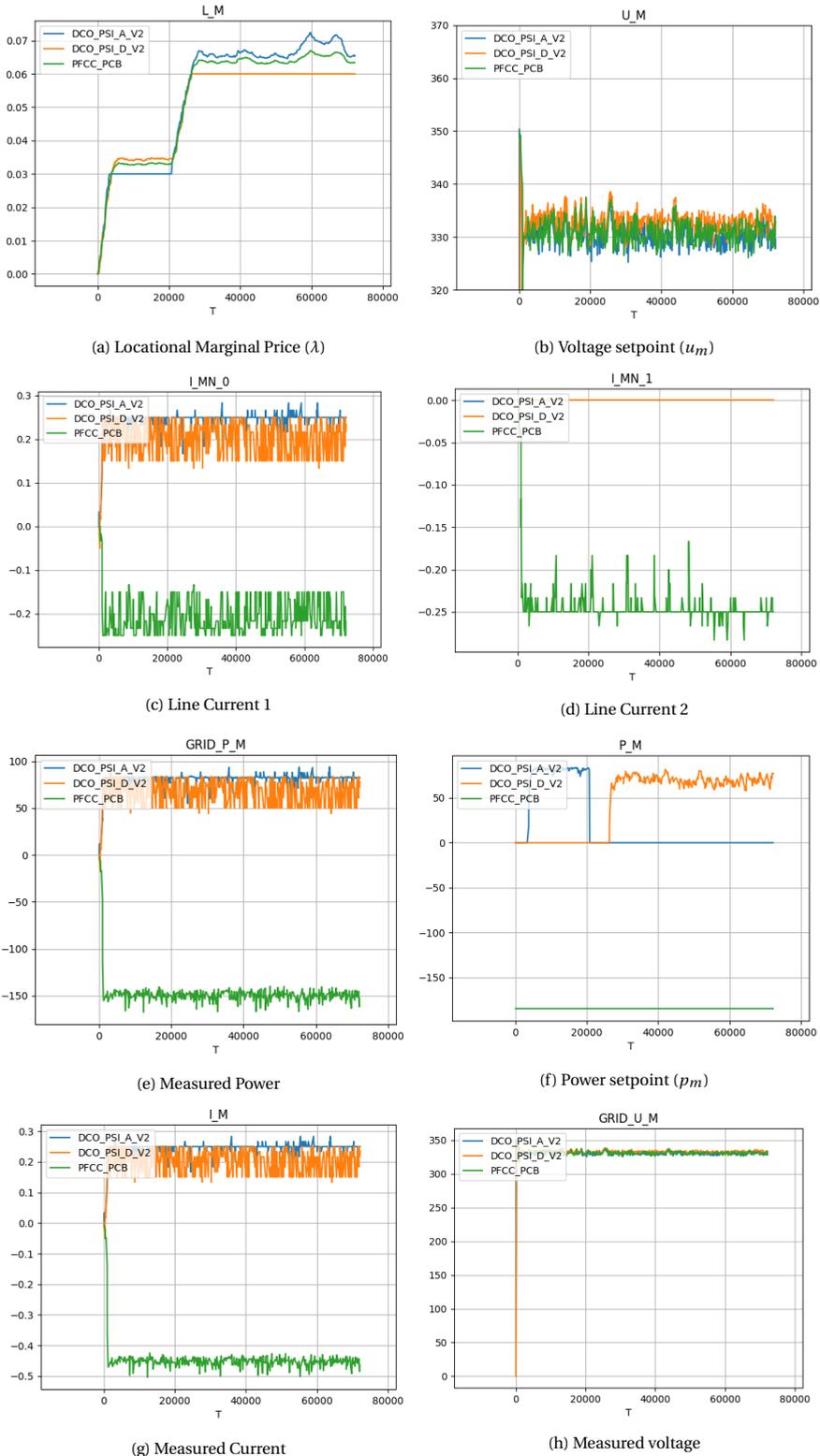


Figure 5.13: The value of various variables over time (ms)
- Experiment 7 - Dynamically changing cost

6

THE CONSENSUS AND INNOVATION ALGORITHM: A MODIFIED ALGORITHM FOR REGION-BASED OPF

This chapter introduces a way of modifying the C+I algorithm, so that a node may now be in control of more than one devices, essentially turning the problem from nodal OPF to region-based OPF. The C+I algorithm cannot be trivially extended to support region OPF, because of some new challenges that arise: If the amount of communication is to remain the same, then the LMP of one region is responsible for dictating the behavior of not one, but multiple devices, which involves questions like "how to update the total power and power per device, based on a single LMP value?", and "based on which device's cost coefficients will the new value of the LMP depend on?".

The incentive for this modification came when we considered a specific device as a load in our network: a metal plate with two types of LED lights installed on it, one with a warm color and another with a bright white color. The two LED lights could be manipulated separately through the same PCB, which lead to the idea of coming up with an appropriate algorithm for this kind of device.

Section 6.1 describes the modified algorithm, while section 6.2 provides case studies that verify its effectiveness. Section 6.3 shares some concluding remarks.

6.1. THE MODIFIED ALGORITHM

We begin by explaining how the original, nodal OPF model of the C+I algorithm is modified when transitioning to a region-based OPF model. We then proceed to describe the new KKT conditions that are introduced to the problem, the way the variable updates are affected, and the additional challenges that arise as a result. Finally, we explain the approach taken in order to solve these issues.

6.1.1. THE NEW OPTIMIZATION PROBLEM

The optimization problem from section 2.1 is now modified as follows:

$$\min_p \sum_{m \in N} \sum_{s \in S_m} A_{ms} p_{ms}^2 + B_{ms} p_{ms} \quad (6.1)$$

$$p_m = \sum_{s \in S_m} p_{ms} = u_m i_m^s = u_m \sum_{n \in \Omega_m} G_{m,n} (u_m - u_n), \quad \forall m, n \in N \quad (6.2)$$

$$i_{mn} = \sum_{n \in \Omega_m} G_{mn} (u_m - u_n) \leq \bar{I}_{mn}, \quad \forall m, n \in N \quad (6.3)$$

$$\underline{P}_m \leq p_m \leq \bar{P}_m, \quad \forall m \in N \quad (6.4)$$

$$\underline{P}_{ms} \leq p_{ms} \leq \bar{P}_{ms}, \quad \forall m \in N \quad (6.5)$$

$$\underline{U}_m \leq u_m \leq \bar{U}_m, \quad \forall m \in N \quad (6.6)$$

Essentially, the only necessary modification was the addition of power variables p_{ms} for each device s within a node m , something that the original power variables were not accounting for. The original power variables p_m could be removed from the problem definition, and be replaced with the sum of the p_{ms} variables, but are not, because we do

end up requiring them defined in the devised algorithm we came up with. Furthermore, we make the convenient assumption that the sum of all the power limits per device of a node are equal to the power limit of the node, formally, $\bar{P}_m = \sum_{s \in S_m} \bar{P}_{ms}$ and $\underline{P}_m = \sum_{s \in S_m} \underline{P}_{ms}$. This assumption helps us avoid extra challenges and complexity.

We can observe that three extra conditions have been added. The first condition is an equality (6.2), rationally requiring that the sum of the power setpoints of each device of a node equals the total power setpoint for that node. The other two conditions (6.5) require that each power setpoint of a device is also within its own power limits, not just the node's. The latter conditions introduce two additional dual variables, $\mu_{ms}^{\bar{P}}$ and $\mu_{ms}^{\underline{P}}$, as we can see in the modified Lagrangian below:

$$\begin{aligned}
\mathcal{L} = & \sum_{m \in N} \sum_{s \in S_m} A_{ms} p_{ms}^2 + B_{ms} p_{ms} \\
& + \sum_{m \in N} \lambda_m (u_m \sum_{n \in \Omega_m} G_{m,n} (u_m - u_n) - p_m) \\
& \quad + \sum_{m \in N} \lambda_m (\sum_{s \in S_m} p_{ms} - p_m) \\
& + \sum_{m \in N} \sum_{n \in \Omega_m} \mu_{mn} (G_{mn} (u_m - u_n) - \bar{I}_{mn}) \\
& \quad + \sum_{m \in N} \mu_m^{\bar{P}} (p_m - \bar{P}_m) \\
& \quad + \sum_{m \in N} \sum_{s \in S_m} \mu_{ms}^{\bar{P}} (p_{ms} - \bar{P}_{ms}) \\
& \quad + \sum_{m \in N} \mu_m^{\underline{P}} (-p_m + \underline{P}_m) \\
& \quad + \sum_{m \in N} \sum_{s \in S_m} \mu_{ms}^{\underline{P}} (-p_{ms} + \underline{P}_{ms}) \\
& \quad + \sum_{m \in N} \mu_m^{\bar{U}} (u_m - \bar{U}_m) \\
& \quad + \sum_{m \in N} \mu_m^{\underline{U}} (-u_m + \underline{U}_m)
\end{aligned}$$

The resulting KKT conditions are presented below:

$$\frac{\partial \mathcal{L}}{\partial p_{ms}} = 2A_{ms}p_{ms} + B_{ms} + \lambda_m + \mu_{ms}^{\bar{P}} - \mu_{ms}^P \quad (6.7)$$

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial u_m} &= \lambda_m \sum_{n \in \Omega_m} G_{mn}(u_m - u_n) \\ &+ \lambda_m u_m \sum_{n \in \Omega_m} G_{mn} - \sum_{n \in \Omega_m} \lambda_n u_n G_{mn} \\ &+ \sum_{n \in \Omega_m} G_{mn}(\mu_{mn} - \mu_{nm}) + \mu_m^{\bar{U}} - \mu_m^U = 0 \end{aligned} \quad (6.8)$$

$$\frac{\partial \mathcal{L}}{\partial \lambda_m} = -p_m + u_m \sum_{n \in \Omega_m} G_{mn}(u_m - u_n) - p_m + \sum_{s \in S_m} p_{ms} = 0 \quad (6.9)$$

$$\frac{\partial \mathcal{L}}{\partial \mu_{mn}} = \sum_{n \in \Omega_m} G_{mn}(u_m - u_n) - \bar{I}_{mn} \leq 0 \quad (6.10)$$

$$\frac{\partial \mathcal{L}}{\partial \mu_m^{\bar{P}}} = p_m - \bar{P}_m \leq 0 \quad (6.11)$$

$$\frac{\partial \mathcal{L}}{\partial \mu_{ms}^{\bar{P}}} = p_{ms} - \bar{P}_{ms} \leq 0 \quad (6.12)$$

$$\frac{\partial \mathcal{L}}{\partial \mu_m^P} = -p_m + \underline{P}_m \leq 0 \quad (6.13)$$

$$\frac{\partial \mathcal{L}}{\partial \mu_{ms}^P} = -p_{ms} + \underline{P}_{ms} \leq 0 \quad (6.14)$$

$$\frac{\partial \mathcal{L}}{\partial \mu_m^{\bar{U}}} = u_m - \bar{U}_m \leq 0 \quad (6.15)$$

$$\frac{\partial \mathcal{L}}{\partial \mu_m^U} = -u_m + \underline{U}_m \leq 0 \quad (6.16)$$

The original power derivative $\frac{\partial \mathcal{L}}{\partial p_m}$ is missing from this list, because, by construction of our algorithm, p_m is no longer a decision variable. Instead, the decision variables are p_{ms} , which, once decided for, consequently define the p_m variable. Additionally, since we make the above relationship between p_{ms} and p_m explicit by construction in the algorithm, it no longer needs to be accounted for, and the lambda derivative $\frac{\partial \mathcal{L}}{\partial \lambda_m}$ can be simplified to what it was originally (3.9). Furthermore, in the original algorithm, the power derivative $\frac{\partial \mathcal{L}}{\partial p_m}$ and power dual variables $\frac{\partial \mathcal{L}}{\partial \mu_m^P}$, $\frac{\partial \mathcal{L}}{\partial \mu_m^{\bar{P}}}$ are not used in the algorithm; the requirements and conditions they represent are implicitly satisfied by the algorithm, specifically, by the updates of the power variable. A similar approach is adopted in this case as well, hence, the new power derivative $\frac{\partial \mathcal{L}}{\partial p_{ms}}$ and new power dual variables

$\frac{\partial \mathcal{L}}{\partial \mu_{ms}^P}$, $\frac{\partial \mathcal{L}}{\partial \mu_{ms}^{\bar{P}}}$ are not directly used, and the conditions they represent are implicitly satisfied.

6.1.2. THE NEW STRATEGY

We have justified a lot of modelling decisions with the excuse of the algorithm and its construction, so now it is time to describe the devised algorithm. The modifications to the algorithm are only related to the power and LMP variable update strategy, please refer to Section 3.2.2 for a reminder of the original strategy. Based on the original strategy, the LMP would dictate the region of operation, Constant or Marginal Power Region, and the updates of the LMP and power would take place accordingly. In this case, however, where we have multiple devices per node, each device could be intercepted at a different point and region on the cost derivative by the LMP, as portrayed by Figure 6.1.

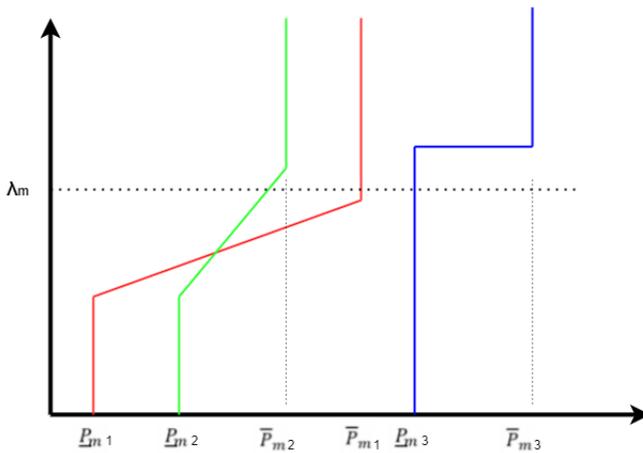


Figure 6.1: A depiction of how one LMP value intercepts the cost derivative function of each device of the node at different regions. Device 1 is intercepted in its upper Constant Power Region, device 2 is intercepted in its Marginal Power Region, and device 3 is intercepted in its lower Constant Power Region

We could have handled the set of updates for each device separately, based on where each device is intercepted, but that was introducing far more challenges than the alternative. The idea is to keep the power variable p_m update the same as in the original algorithm, and then decide how that power would be split amongst the candidate devices of the node. This approach still created a lot of challenges to be solved, but they were quite more manageable. The main challenges could be summed up as follows: which region do we base the LMP and power updates on? How do we split the decided power amongst the devices of the node? In case we are performing the updates corresponding to the Marginal Power Region, which cost coefficients dictate the new value of the LMP?

Our solution takes a scenario-based approach. We break down each of the possible 5 scenarios, in terms of where does the LMP intercept each cost derivative, and explain how the aforementioned challenges are handled for each. It is important to remember

that, in both the original and this new version of the algorithm, if the updates are performed according to the Constant Power Region (CPR), then the LMP is updated based on external factors, and the power is deterministically defined based on local factors, and vice versa for the Marginal Power Region (MPR).

LOWER CONSTANT POWER REGION

If all devices are intercepted in the lower CPR, the updates take place according to the CPR: all devices are set to provide their minimum power \underline{P}_{ms} (minimum power generation for generators, maximum power demand for loads), and the LMP is updated as in the original algorithm. Naturally, the power setpoint p_m is set to \underline{p}_m .

UPPER CONSTANT POWER REGION

If all devices are intercepted in the upper CPR, the updates take place according to the CPR: all devices are set to provide their maximum power \overline{P}_{ms} (maximum power generation for generators, minimum power demand for loads), and the LMP is updated as in the original algorithm. Naturally, the power setpoint p_m is set to \overline{p}_m .

LOWER AND UPPER CONSTANT POWER REGION

If some devices are intercepted in the lower CPR, and some others in the upper CPR, the updates take place according to the CPR: all devices are set to provide their maximum or minimum power, according to where each was intercepted by the LMP, and the LMP is updated as in the original algorithm. Naturally, the power setpoint p_m is set to be the sum of the power setpoints p_{ms} decided per device.

MARGINAL POWER REGION

If all devices are intercepted in the MPR, the updates take place according to the MPR. The power setpoint p_m is updated as in the original algorithm, and each device calculates its own cost for producing the newly decided amount of power. The devices take priority based on that cost, with the cheapest device given the most priority. Each prioritized device sets its own power setpoint p_{ms} to maximum power \overline{P}_{ms} , if a generator, and to minimum power \underline{P}_{ms} , if a load. The process stops once we run out of available power p_m , for instance, if we assume that the remaining power to allocate is x , and $\overline{P}_{ms} > x$ for the generator with priority, then we set $p_{ms} = x$. Finally, the LMP is updated as in the original algorithm, with the cost coefficients B_m and A_m used being those of the device with the highest priority.

This approach is based on the fact that, given that the power setpoint is already decided by external factors when in the MPR, it makes sense to allocate most of that power to devices that utilize it best, i.e., generators that produce it cheaply or loads that are closer to turning off than others. Following the same line of thought, the LMP is best to be dictated by the most prioritized device, as that is the device that makes the most impact, and it is also the device that has a priority in dictating its transition compared to the others, something that is achieved through the LMP.

CONSTANT AND MARGINAL POWER REGION

If some devices are intercepted in the CPR, and some others in the MPR, the updates take place according to the MPR. All devices intercepted in the CPR are set to provide their maximum or minimum power, according to where each was intercepted by the LMP. Once that process is over, the remaining power is essentially the new power setpoint p_m that was calculated, minus the power setpoints p_{ms} of the devices in the CPR. The rest of the process follows exactly as in the previous scenario of all devices in the MPR, but this time, the LMP is dictated by the device with the highest priority in the MPR, as it makes no sense for the LMP to be dictated by a device in the CPR.

6.2. CASE STUDIES

Our case studies take place on a network consisting of one Mean Well CSP3000 Power Supply as a generator, and the metal plate with the two LED lights mentioned at the start of this chapter, as the load. The goal we wanted to achieve by using this setup was to observe the algorithm give the appropriate priority to each of the LED lights, based on their cost functions, while handling both of them through a single node and only receiving a single LMP value. The cost functions are configured in such a way, that each of the 5 aforementioned scenarios is encountered.

Both the power supply and the LED light initialize their voltage levels at $u_m = 350V$, with a lower voltage limit of $\underline{U}_m = 330V$ and an upper voltage limit of $\overline{U}_m = 370V$. At the start of the algorithm, all generators must be off, i.e., $p_m = 0W$, and have a lower power limit of $\underline{P}_m = 0W$ and an upper power limit of $\overline{P}_m = 3000W$. As for the LED light, we initialize its lower power limit to $\underline{P}_m = -165W$ because, by construction, the maximum current that each of the two LED lights can achieve is just shy of $0.250A$, and because the lower voltage limit is $330V$. The upper power limit is set to $\overline{P}_m = 0W$. Each of the two LED lights gets a balanced amount of power demand, formally, $\underline{P}_{ms} = -82.5W$.

ONE LED LIGHT STAYS ON

For the first experiment, the cost coefficients $B_m = 0.2$ and $A_m = 0$ were assigned to the first LED light, and $B_m = 0.4$ and $A_m = 0$ to the second. The cost coefficients $B_m = 0.3$ and $A_m = 0$ were assigned to the power supply. The purpose of this experiment was to observe the less efficient LED light turn off before the power supply starts providing power, since consuming power is not beneficial for that LED in this particular setup. The results can be seen in Figure 6.2.

Indeed, we can observe that, once the LMP reaches the value of 0.2, which is the cost threshold for the less efficient LED light, the light turns off, as indicated by the sudden shift in its power demand (p_source_0) and measured power ($grid_p_source_0$). By the time the LMP stabilizes at the threshold of the power supply, 0.3, only one LED light is left on, and the power supply is instructed to provide the appropriate amount of power, which is less than what the LED light requests, since part of it is automatically satisfied by our setup. Another unfortunate side effect of our setup is the fact that the requested power is automatically satisfied, even before the power supply is instructed to provide it,

which is the same effect we had in the experiments with the original algorithm.

BOTH LED LIGHTS TURN OFF SIMULTANEOUSLY

For the second experiment, the cost coefficients $B_m = 0.2$ and $A_m = 0$ were assigned to both the LED lights. The cost coefficients $B_m = 0.25$ and $A_m = 0$ were assigned to the power supply. The purpose of this experiment was to observe both LED lights turn off before the power supply has a chance to provide power, since that is not efficient for any of the two. The LMP is supposed to stabilize at the cost threshold of the two LED lights, since there is no request for power, and consequently no request for a higher LMP, after the two LEDs turn off. The results can be seen in Figure 6.3.

Once the LMP reaches the value of 0.2, we can see that both LED lights turn off simultaneously, and by the time the LMP stabilizes at 0.25, the power supply doesn't need to provide power anymore because there is no demand from the environment. Notice how the variables p_source and $grid_p_source$ follow the same pattern of behavior for both LED lights. Furthermore, contrary to what we expected, the LMP stabilizes at the cost threshold of the power supply, but that is mostly attributed to the fact that, even after the LED lights turn off, there is still some minor power mismatch measured by the system that slowly pushes the LMP upwards.

BOTH LED LIGHTS TURN OFF SEQUENTIALLY

For the next experiment, the cost coefficients $B_m = 0.2$ and $A_m = 0$ were assigned to the first LED light, and $B_m = 0.4$ and $A_m = 0$ to the second. The cost coefficients $B_m = 0.5$ and $A_m = 0$ were assigned to the power supply. This time, the idea was to again observe both LED lights turn off, but one after the other. Additionally, we again expected the LMP to stabilize at the cost threshold of the most effective LED light, since there is no request for power, and consequently no request for a higher LMP, after both the LEDs turn off. The results can be seen in Figure 6.4.

We can observe that the less efficient LED turns off when the LMP hits its threshold, 0.2, and then the second LED follows the same behavior when the LMP reaches its threshold next, 0.4. However, this time the LED takes much more time to completely turn off. The reason behind this behavior is the fact that the system, at this point, is very low on both generated and demanded power, and so there is no increase in the LMP to quickly push the LED to turn off; instead, the LMP stabilizes at the threshold of the LED, which simply keeps the power at the current levels, or rather, allows it to slowly fade away.

We then slightly modified the experiment by unbalancing the power demands per LED light, setting them to $\underline{P}_{m1} = -45W$, $\underline{P}_{m2} = -82.5W$ and, consequently, $\underline{P}_m = -127.5W$. We also changed the cost coefficient of the second LED light from $B_m = 0.4$ to $B_m = 0.3$, for the purposes of faster convergence and demonstration. The purpose of this experiment is to simply observe a different demand for power by each LED light. The results can be seen in Figure 6.5.

Indeed, before the first LED turns off, the system requests a total amount of power of about $127W$, and, after the low-demand LED turns off, the total request for power

drops by about $45W$, at $82.5W$, and finally the system once again takes its time turning completely off.

Finally, we modified the original experiment of this subsection in yet another way: we added a quadratic cost coefficient to both LED lights, $A_{m1} = A_{m2} = 0.001$. This was expected to cause a very rapid transition for each LED light from full demand to zero demand, because, once each LED light enters its Marginal Power Region, the effects of the quadratic cost coefficient would make the LMP jump from the lower cost threshold to the higher cost threshold almost instantly. The results can be seen in Figure 6.6.

We can see that the LMP grows rapidly and immediately causes both LED lights to turn off, but stops its rapid growth after escaping the Marginal Power Region of the efficient LED light, i.e., after exceeding its upper cost threshold, and slowly grows towards 0.5 , the lower cost threshold of the power supply, because of any remaining power mismatch in the system. Thus, the behavior is indeed as expected.

6.3. CONCLUDING REMARKS

Through our experiments, we have verified that the algorithm is indeed effective in all 5 possible scenarios of LMP-to-cost-function relationships. A setup of greater scale would be more convincing of this conclusion, but physical restrictions prevented us from implementing one. The results also indicate that convergence is achieved almost, if not as fast, as in the original algorithm, because the growth rate of the LMP remains the same, since all the operations within a region are like a blackbox to the rest of the network, which only gets to know the newly decided value of the LMP for that region. The main weakness of the algorithm is its non-optimality in certain scenarios where many devices of a region are in their Marginal Power Region: while the optimal solution may be for those devices to each provide/request a fraction of their maximum power, our algorithm takes a sequential approach, only allowing a device to provide/request power if all devices of higher priority are already providing/requesting maximum power. Of course, such scenarios are very rare, because of the preconditions required for them to happen, namely non-zero quadratic coefficients, overlapping Marginal Power Regions and belonging to the same region.

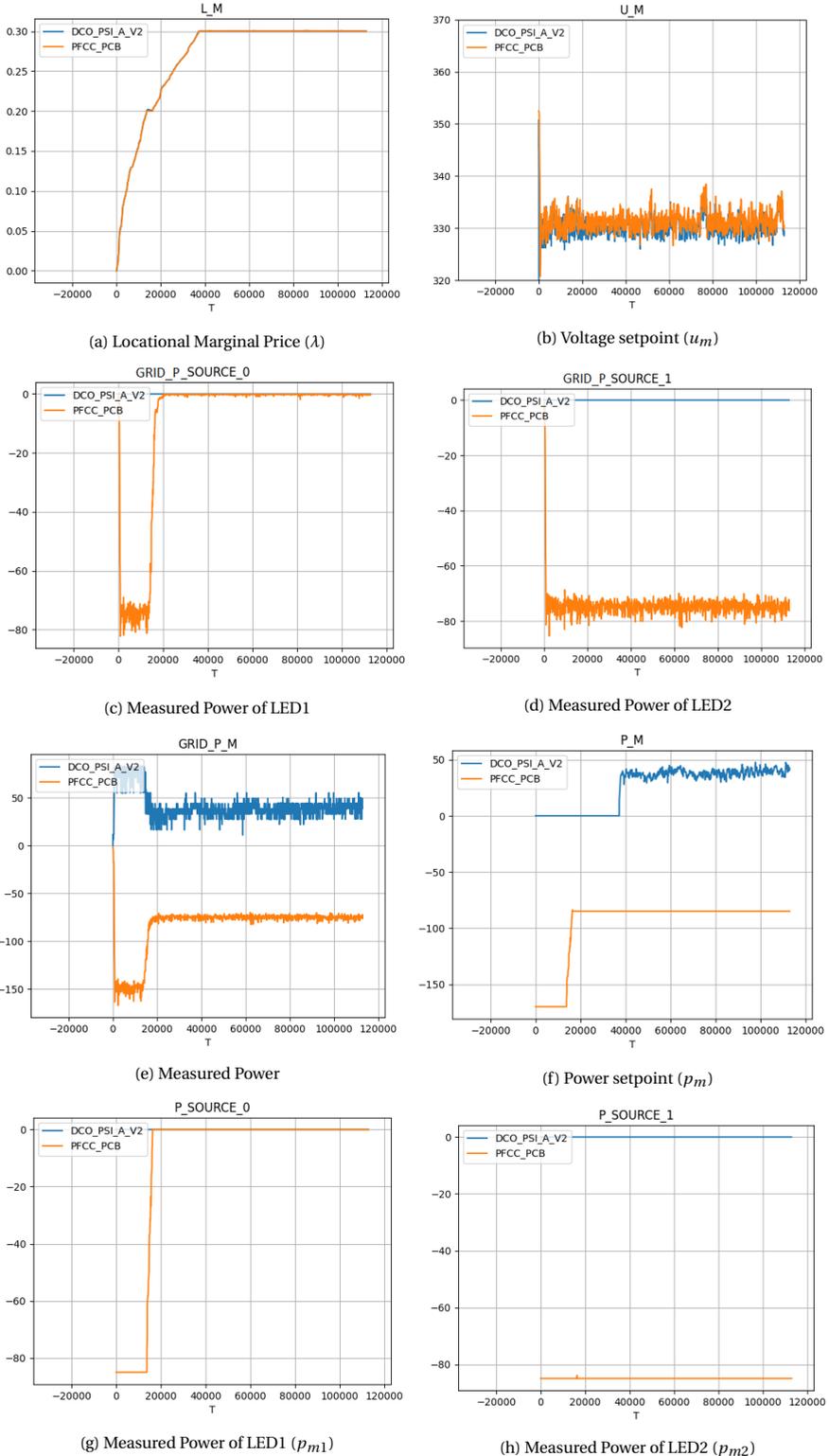
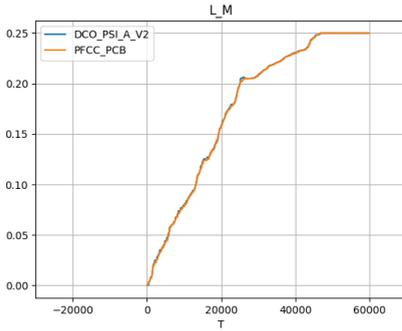
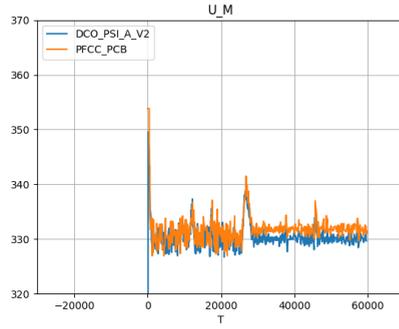


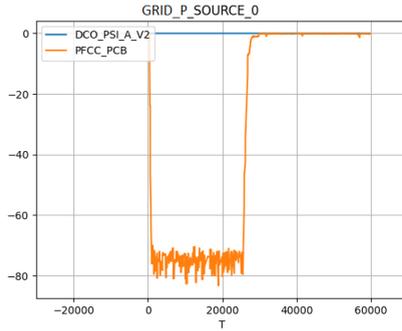
Figure 6.2: The value of various variables over time (ms)
- Experiment 1 - One LED light stays on



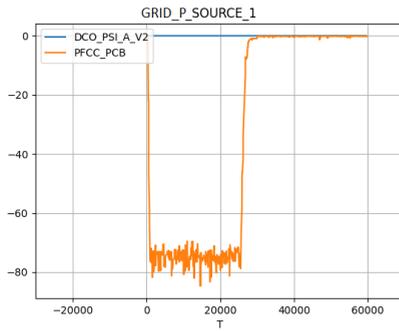
(a) Locational Marginal Price (λ)



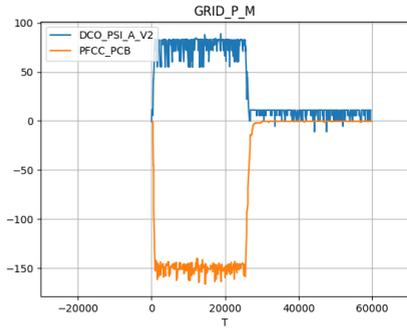
(b) Voltage setpoint (u_m)



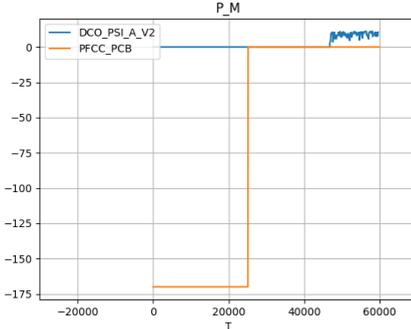
(c) Measured Power of LED1



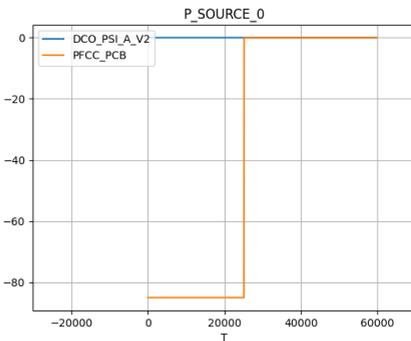
(d) Measured Power of LED2



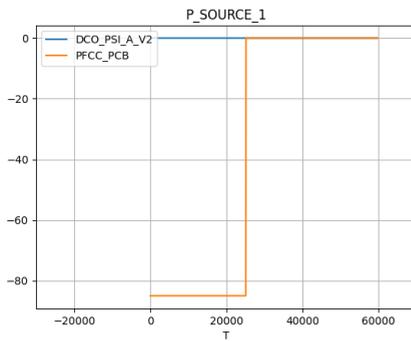
(e) Measured Power



(f) Power setpoint (p_m)



(g) Measured Power of LED1 (p_{m1})



(h) Measured Power of LED2 (p_{m2})

Figure 6.3: The value of various variables over time (ms)
- Experiment 2 - Both LED lights turn off simultaneously

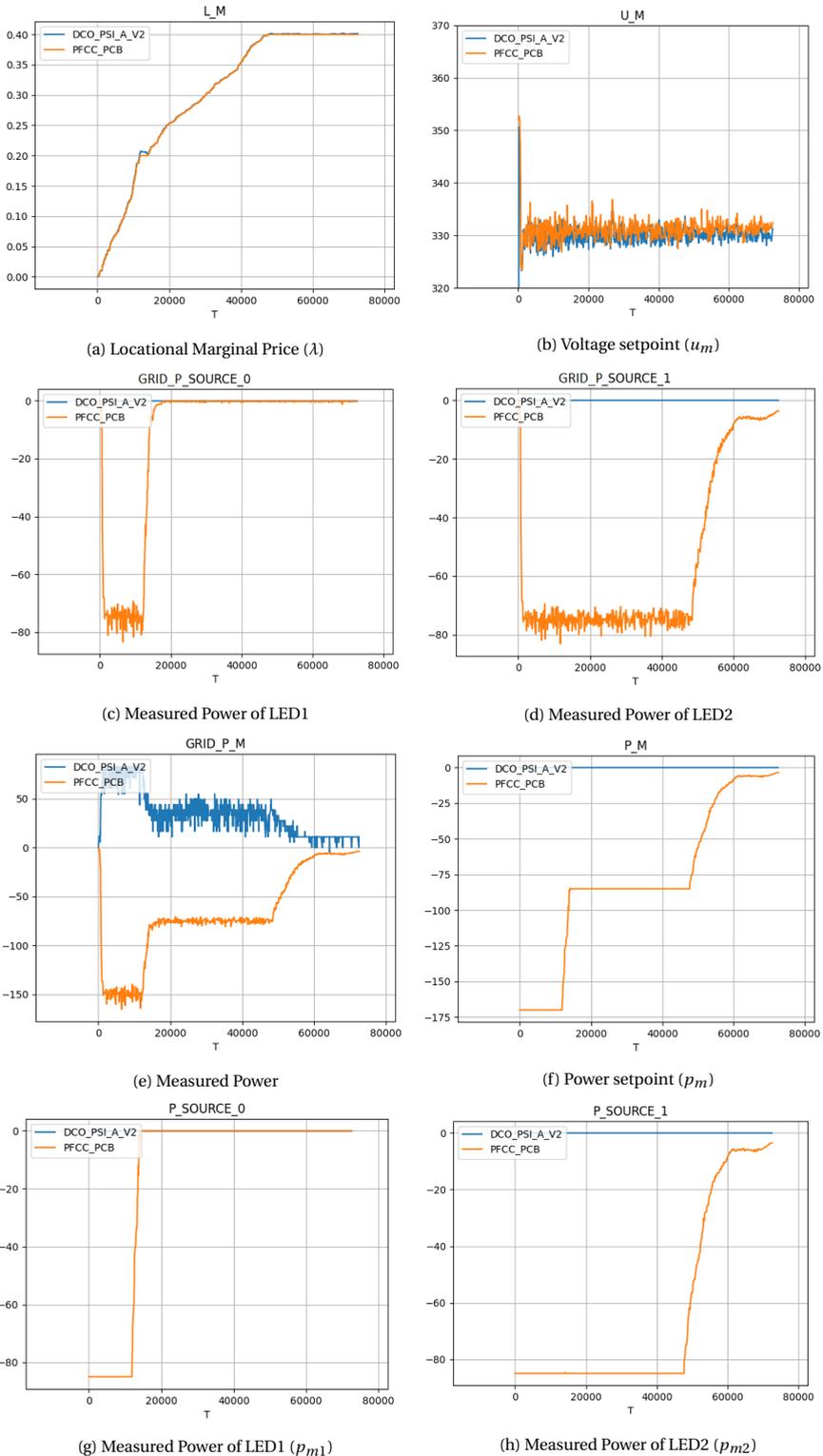
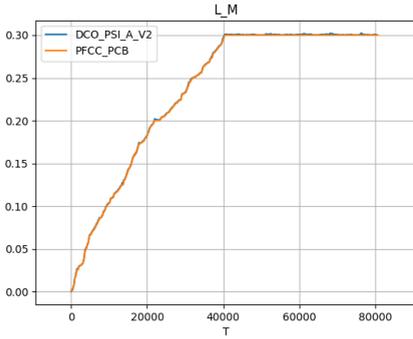
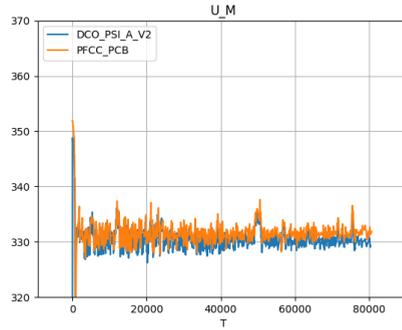


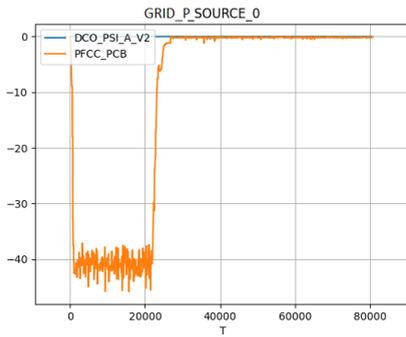
Figure 6.4: The value of various variables over time (ms)
 - Experiment 3 - Both LED lights turn off sequentially



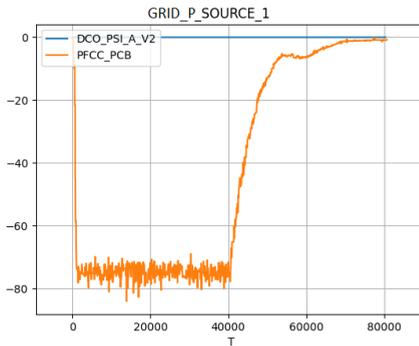
(a) Locational Marginal Price (λ)



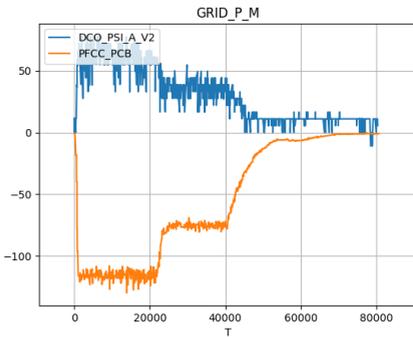
(b) Voltage setpoint (u_m)



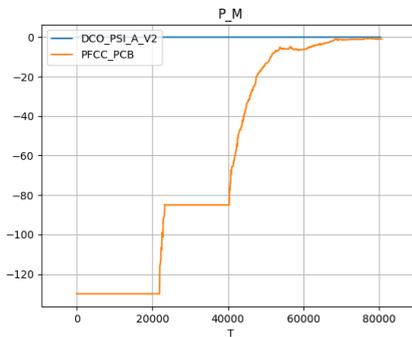
(c) Measured Power of LED1



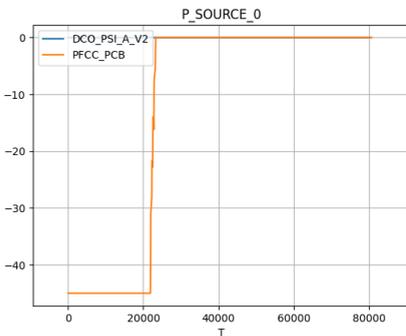
(d) Measured Power of LED2



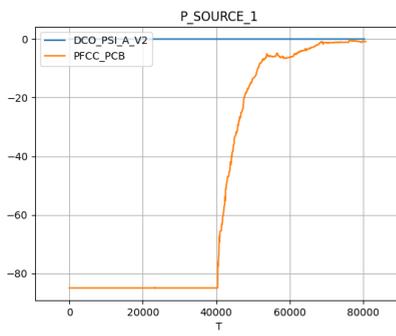
(e) Measured Power



(f) Power setpoint (p_m)



(g) Measured Power of LED1 (p_{m1})



(h) Measured Power of LED2 (p_{m2})

Figure 6.5: The value of various variables over time (ms)

- Experiment 4 - Both LED lights turn off sequentially - $p_{m1} = -45W, p_{m2} = -82.5W$

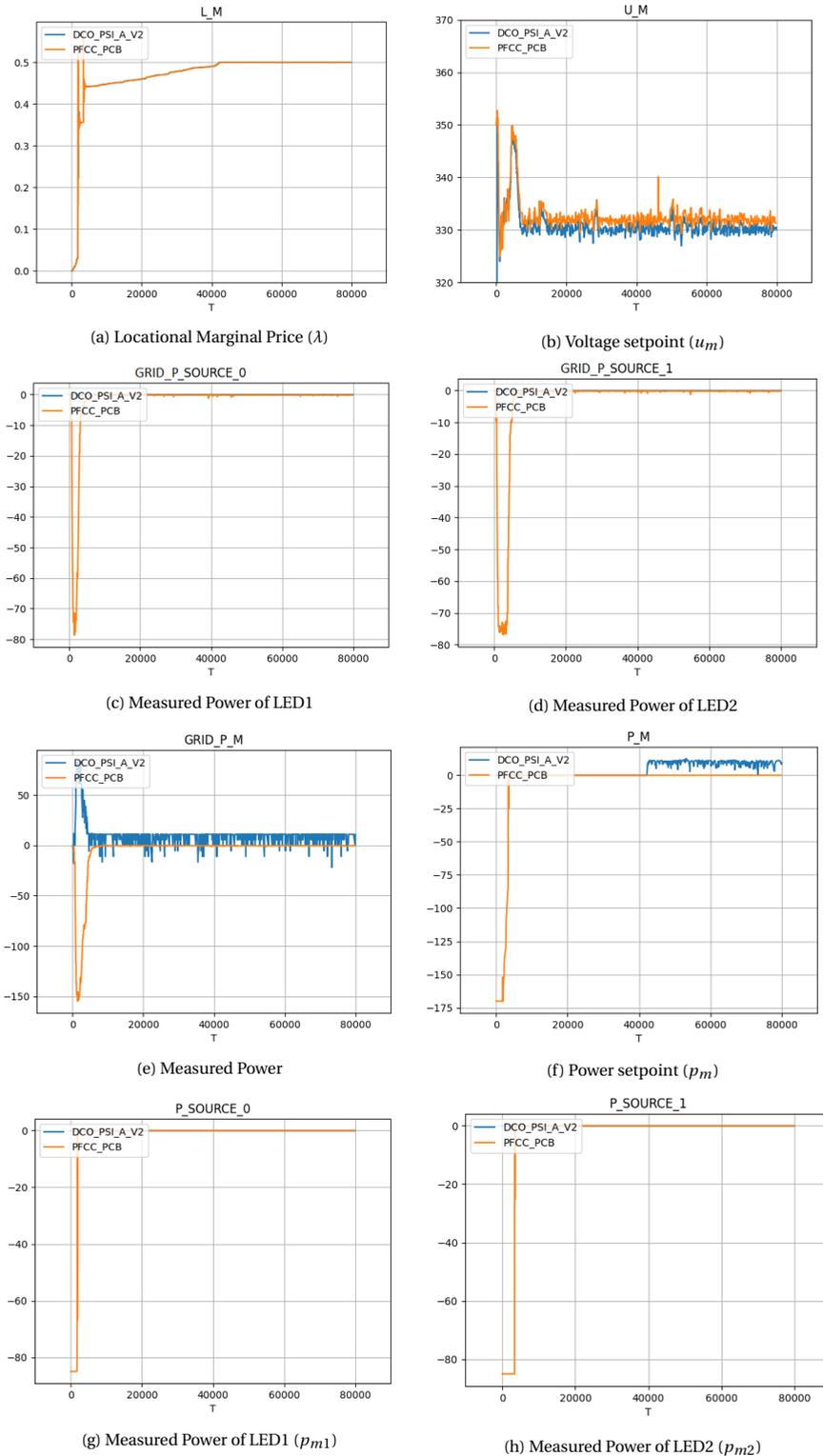


Figure 6.6: The value of various variables over time (ms)
 - Experiment 5 - Both LED lights turn off sequentially - $A_{m1} = A_{m2} = 0.001$

7

THE CONSENSUS AND INNOVATION ALGORITHM: ISLANDING AND DE-ISLANDING

As discussed in Section 1.4, the capability for an OPF algorithm to be applicable to scenarios of islanding and de-islanding can prove to be of great utility. This chapter focuses on proving that our C+I algorithm does have that capability, by demonstrating its effectiveness using experiments, one of which takes place at the Green Village, an innovation site where technologies in the field of sustainable energy provision are tested and applied in a real-life environment.

In order to properly demonstrate the effectiveness of our algorithm in a scenario of islanding or de-islanding, a 5-node network was devised, which is visually represented by Figure 7.1. An actual photograph of the setup is also presented in Figure 7.2. In this case, two of the PCBs are Solid State Protection (SSP) modules, containing 5 slots for connections to neighbours that can be enabled and disabled, hence why they prove useful for this kind of experiments. Each SSP module is considered a node in the network, though, in contrast to the other nodes, they do not control any physical converter. We can observe that module "SSP 2" has 2 neighbours, connected at its first two slots, while module "SSP" has 3 neighbours, connected at slots 0, 2 and 3.

The idea is to have the LED request for power that should be provided by either the expensive power supply, Power Supply D, if the connection between the SSPs is inactive (islanding), or should be provided by the cheap power supply, Power Supply A, if the connection between the SSPs is active (de-islanding). With that purpose in mind, we set the cost of Power Supplies A and D to $B_m = 0.5, A_m = 0$ and $B_m = 1, A_m = 0$, respectively. All devices initialize their voltage levels at $u_m = 350V$, with a lower voltage limit of $\underline{U}_m = 330V$ and an upper voltage limit of $\overline{U}_m = 370V$. Both power supplies set $\underline{P}_m = 0W$ and $\overline{P}_m = 3000W$. As for the LED light, we initialize its lower power limit to $\underline{P}_m = -165W$

because, by construction, the maximum current that it can achieve is just shy of $0.500A$, and because the lower voltage limit is $330V$. The cost coefficients $B_m = 0$ and $A_m = 0$ were assigned to the LED light, meaning there is no cost function related to power consumption, and the load will always request for full power. As a result, the upper power limit is also set to $\bar{P}_m = -165W$. Finally, all power and cost variables for the SSP modules are set to 0 (zero) because, as far as variables are of concerned, the SSPs just propagate information, and we essentially treat them as loads with no request for power.

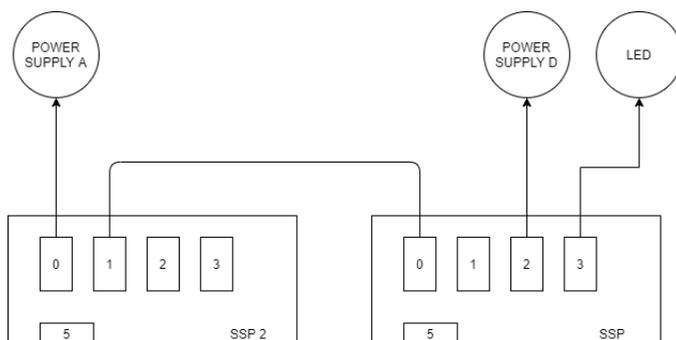


Figure 7.1: A sketch of the 5-node testing network

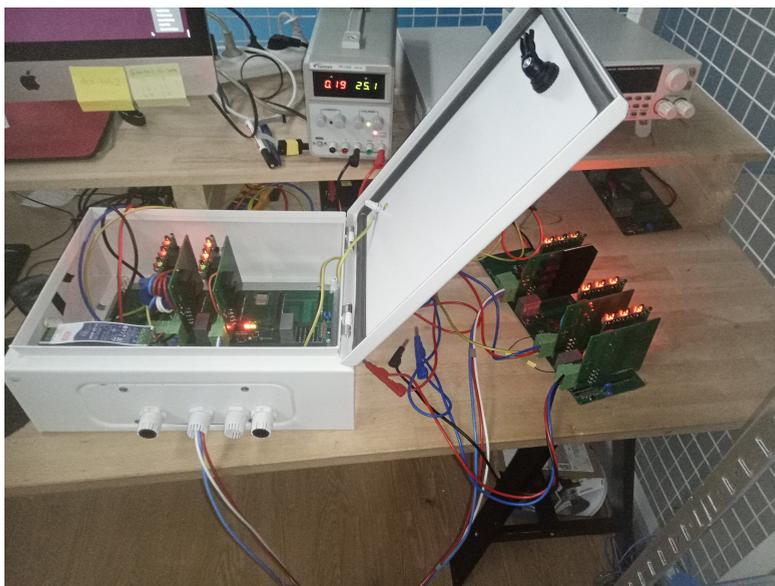


Figure 7.2: The 5-node laboratory testing network

5 NODES - SHIFT FROM EXPENSIVE TO CHEAP

At the beginning of this experiment, the link connecting the two SSPs is inactive, leaving the system in islanding mode. The purpose of this first experiment is to observe the expensive power supply step away from its role as the provider of power, allowing for the cheap power supply to take over, once the link is re-enabled. The results of the experiment can be seen in Figure 7.3.

We can observe that, at first, the LMP grows to 0.5 in the left section of the network, and to 1 in the right section, because of each power supply trying to maintain its lower-threshold LMP. Consequently, both power supplies are instructed to provide power, with the expensive one providing about 165W, which is what the LED requests. The right SSP experiences an incoming current of 0.5A (i_{m2}), because of the expensive power supply, and an outgoing current of again 0.5A (i_{m3}), provided to the LED.

A little bit before the 17:13:00 mark, we re-enable the link between the two SSPs. We can then immediately observe that the LMP of the right section stops being maintained at a value of 1, and gradually drops down to 0.5, the lower cost threshold of the cheap power supply. The initial cause for this behaviour is the sudden drop of power at the time of re-enabling the link, and afterwards, the connected system stabilizes at its new LMP value, 0.5. During this transition, we can also observe the shift in measured power, with the cheap power supply taking over the power generation, and leaving only a fraction of it to the expensive one, a behavior attributed to the setup rather than the algorithm; after all, we can see that there is no request for power to the expensive power supply by the power setpoint p_m . Indeed, we shall see, at a later experiment, that the setup is the cause for this involvement of the expensive power supply.

Finally, at the 17:14:13 mark, we disable the link between the two SSPs again, and the system returns back to its initial behavior, with the expensive power supply taking over the power generation. It is also interesting to notice the shift in the "iterations per second" rate whenever we change the state of the connection: The slope for the right section of the network decreases when the whole system is connected, because there are more neighbours to communicate with and this takes up more time, while the slope increases for that same right section when the system is disconnected, which equates more iterations per second. Hence, it is obvious that even one additional neighbour makes a significant change in the "iterations per second" rate. In Section 5.1, our estimation of this rate was 50 iterations per second, with the simulated algorithm of Pedro having a rate of about 40 iterations per second for the same amount of neighbours. In this experiment, we can observe that, for the right section of the network, 500 iterations take up about 15 seconds, resulting in a rate of 33 iterations per second. This is a positive result, because we can see that the rate remains at satisfying levels, even with the addition of extra operations for the whole (de-)islanding scenario to work.

Finally, we can observe that the algorithm performs about 500 iterations in a matter of 10 seconds, which translates to about 50 iterations per second. We consider this quite a good speed, given that the best indication of the algorithm's hypothetical optimal speed comes from the simulated algorithm in Pedro Parreira's work[15], where the simplest 2-3 node networks have a runtime of 12 seconds for 500 iterations; with about 8 of those

seconds being attributed to the simulation of the grid mechanics, we are left with 4 seconds for the actual algorithm, coming down to 125 iterations per second. If we do not optimistically leave out the time for simulating the grid, we end up with 40 iterations per second, which is less than the speed achieved by our implementation

Another interesting observation is that the voltage is quite unstable, where most of the devices seem to follow around the pattern of the oscillating, expensive power supply. We suspect this is largely caused by the low current levels overall, a suspicion that we back up in a later experiment.

5 NODES - SHIFT FROM CHEAP TO EXPENSIVE

For this experiment, we left the link connecting the two SSPs active, with the system in de-islanding mode, and also initially disabled the expensive power supply. The purpose of this experiment is to verify that, indeed, nothing about the power generation changes if a power supply joins the network, that is more expensive than the current provider of power.

We can observe that, at first, the LMP grows to 0.5 in the whole network, because of the cheap power supply trying to maintain its lower-threshold LMP. Consequently, that power supply is instructed to provide power, about 165W, which is what the LED requests. The right SSP experiences an incoming current of 0.5A (i_{m0}), because of the cheap power supply, and an outgoing current of again 0.5A (i_{m3}), provided to the LED.

At the 17:26:50 mark, we enable the expensive power supply. While the LMP does not retain its global value, we unexpectedly observe the cheap power supply lowering its power generation and allowing room for the expensive power supply to contribute, which is most likely attributed to the fact that the expensive power supply automatically provides some power as long as it is part of the network, just like the case in our previous experiment. Nonetheless, we can see that our algorithm still only instructs the cheap power supply to provide power, just less than before.

Finally, at the 17:27:55 mark, we disable the link between the two SSPs, and the system shifts so that the expensive power supply takes over the power generation, with the right section LMP growing to a value of 1. Once again, there is also an increase in the "iterations per second" rate of the right section, now that it is isolated.

5 NODES - GREEN VILLAGE - SHIFT FROM EXPENSIVE TO CHEAP

Our final and grander experiment takes place at the Green Village, an out-of-the-lab innovation site where technologies in the field of sustainable energy provision, such as ours, are tested and applied in a real-life environment. A sketch of our setup is presented by Figure 7.5, and a picture of the operation environment can be seen in Figure 7.6. Notice that the new setup is almost identical to the laboratory 5-node setup, with the only difference being that the power supplies are connected at different slots to the SSPs, and that the two SSPs are connected to each other through a long cable that crosses through a couple of street lights, consuming current of up to 2A. Our goal with this experiment is the same as before: to verify that power is provided by the expensive power supply,

if the system is in islanding mode, and by the cheap power supply, if the system is in de-islanding mode.

Just like in the laboratory experiment, at first, the LMP grows to 0.5 in the left section of the network, and to 1 in the right section, because of each power supply trying to maintain its lower-threshold LMP. Consequently, both power supplies are instructed to provide power, with the expensive one providing about 165W, which is what the LED requests. The right SSP experiences an incoming current of 0.5A (i_{m4}), because of the expensive power supply, and an outgoing current of again 0.5A (i_{m3}), provided to the LED. We can also notice a couple of huge spikes in the power setpoint, that are, interestingly, not recorded in the measured power itself, so we can go ahead and discard them as noise.

At the 29:16:30 mark, we re-enable the link between the two SSPs. We can then immediately observe that the LMP of the right section stops being maintained at a value of 1, and gradually drops down to 0.5, the lower cost threshold of the cheap power supply, and the now-connected system stabilizes at its new LMP value, 0.5. During this transition, we can also observe the shift in measured power, with the cheap power supply taking over the whole power generation. There are two interesting things to be noted here regarding the power: First, and contrary to the laboratory experiment, the expensive power supply completely stops generating any power, an indication that the reason it did not do so in the lab experiment was the low current levels. Secondly, the measured power of the cheap power supply is now much higher, due to the high demand and consumption of the street lights across the line connecting the two SSPs. Of course, that demand is not part of the problem or the algorithm, it is an external factor, that does, however, affect our measurements, without negatively affecting the result: the LED still is provided its requested 0.5A (i_{m3}), yet there is an incoming (i_{m4}) and outgoing (i_{m1}) current of more than 2A on the left section SSP. Furthermore, the voltage levels have been stabilized a lot more, which backs up our aforementioned argument in the laboratory experiment that voltage stability can be obtained through higher current levels.

Finally, at the 29:16:32 mark, we disable the link between the two SSPs again, and the system returns back to its initial behavior, with the expensive power supply taking over the power generation.

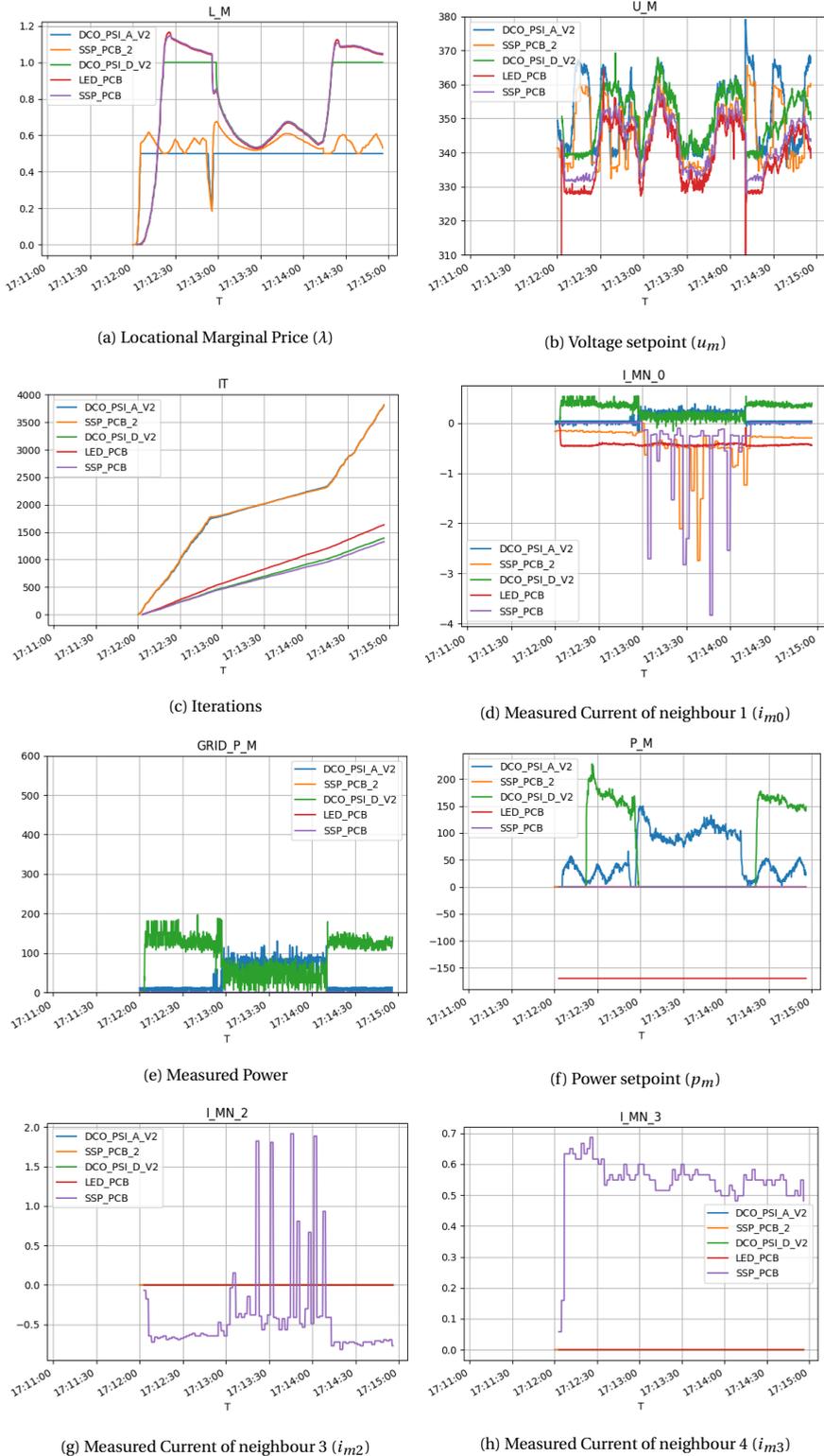


Figure 7.3: The value of various variables over time (hh:mm:ss)
 - Experiment 1 - 5 nodes - Shift from expensive to cheap

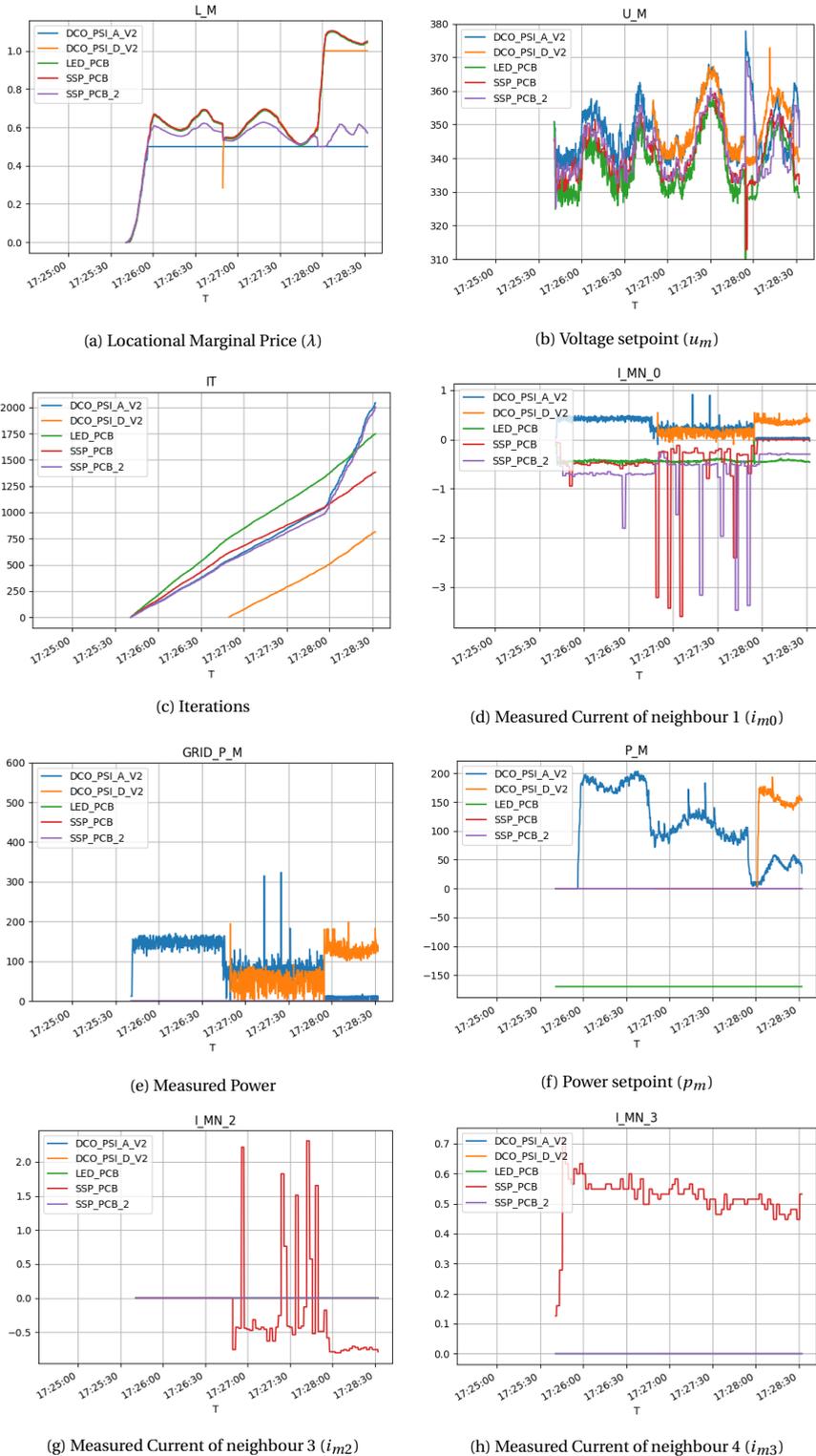


Figure 7.4: The value of various variables over time (hh:mm:ss)
 - Experiment 1 - 5 nodes - Shift from cheap to expensive

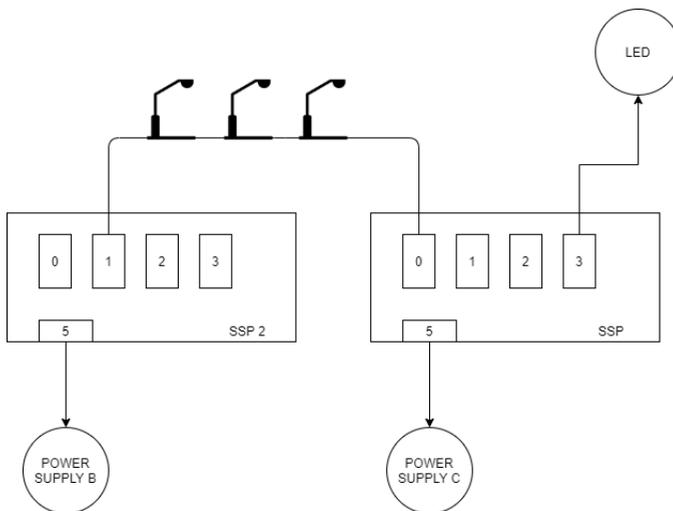


Figure 7.5: A sketch of the 5-node testing network at the Green Village

7

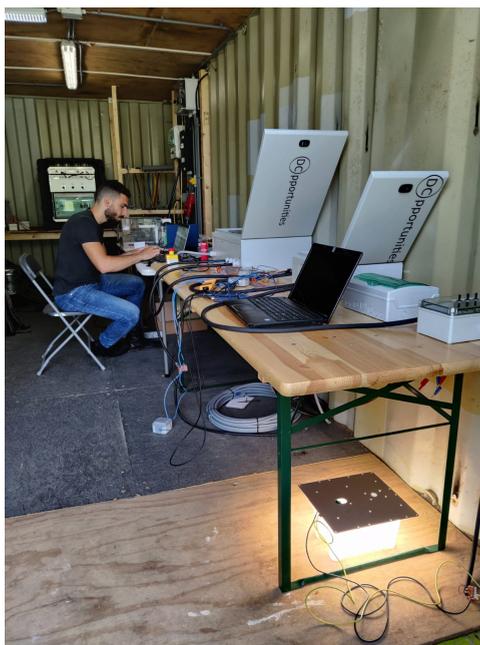
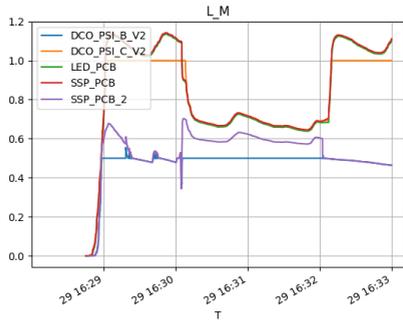
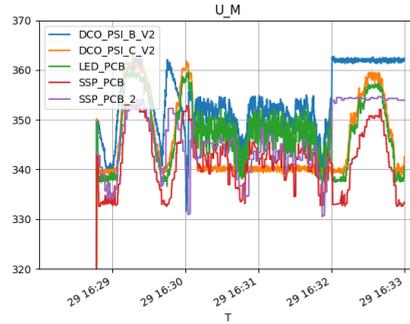


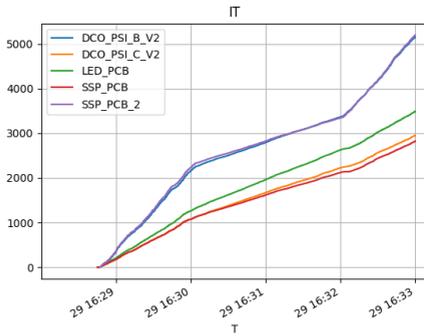
Figure 7.6: The 5-node setup and testing network at the Green Village



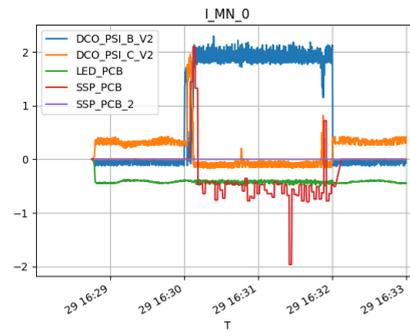
(a) Locational Marginal Price (λ)



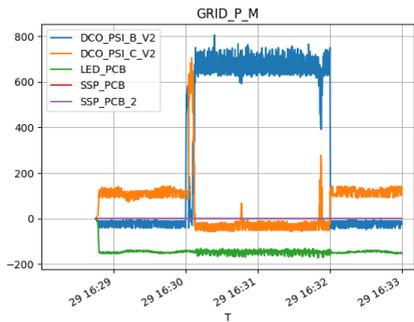
(b) Voltage setpoint (u_m)



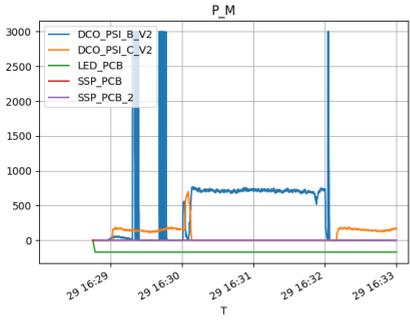
(c) Iterations



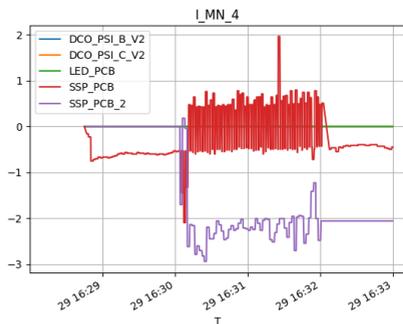
(d) Measured Current of neighbour 1 (i_{m0})



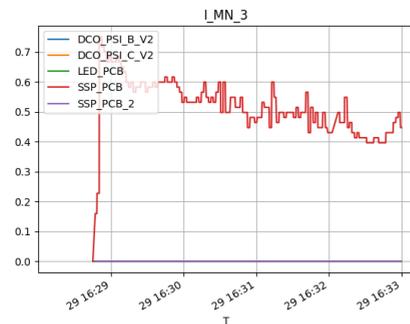
(e) Measured Power



(f) Power setpoint (p_m)



(g) Measured Current of neighbour 5 (i_{m4})



(h) Measured Current of neighbour 4 (i_{m3})

Figure 7.7: The value of various variables over time (dd:hh:mm)
 - Experiment 1 - 5 nodes - Green Village - Shift from expensive to cheap

8

CONCLUSION

In order to present our conclusions and work, we will be focusing on answering each of the research questions, along with providing all the relevant achievements and newly-found knowledge:

How is the C+I algorithm to be implemented for a physical environment where, compared to the simulated one, various additional factors like noise, delays, and voltage/current regulation come into play? What design choices and improvements can be performed in order to eliminate the negative effects of these additional factors?

As described in Chapter 4, our physical implementation consists of ESP32 processors, Power Circuit Boards, and the devices they are connected to, an implementation that allows a wide variety of coordinated operations to take place in parallel. Specifically, our algorithm achieves fast execution through the parallel coordination of processes like measurements collection, voltage regulation and neighbour communication. The fact that all the grid measurements are continuously collected in parallel to the algorithm ensures that there is zero delay during grid-to-node communication, eliminating the related concerns expressed in Parreira's work[17]. Another concern in the work of Parreira was the possible loss of data during inter-node communication, which was effectively taken care of in our work, resulting in less than 5% of the communication data to be lost and still maintaining an effective algorithm. Thus, through the aforementioned practices and the overall structure of our implementation, we achieved not only the creation of a functional C+I algorithm, but also an effective and efficient one.

How does the algorithm perform, given these extra factors, in terms of convergence time and overall execution speed, which are the traditional metrics the algorithm is evaluated on?

Our implementation of the DC-OPF algorithm has been verified as applicable and functional on a physical network, specifically, on small-scale networks of 2 and 3 nodes. Through testing out various different scenarios (Section 5), we ensured that the above conclusion holds for all of them. Examples of such scenarios include current overflow,

prioritizing generators, power limit handling and dynamic reconfiguration of costs and power demands. We also verified that the C+I algorithm can work on a physical network using the same exact parameter values as used in the simulations, although we modified a few of them to properly adapt to the number of neighbours of a device, as they are supposed to by definition (Section 3.2.2). In terms of the number of iterations, we have shown that our algorithm is better or on-par with the number of iterations of the simulated algorithm, at least considering small networks, which we were experimenting on. In terms of the time for convergence, we realized that the most significant factor towards convergence time was the value of the cost coefficients of each device and the growth rate of the LMP, since most of the other transitions were instantaneous.

How can the original, nodal-OPF C+I algorithm be extended for region-OPF, while maintaining its performance regarding metrics such as optimality, convergence time and execution speed?

Our work has shown that it is possible to implement a modified version of the C+I algorithm for region-based OPF, i.e., multiple devices being controlled by a single node, that is still effective. The modified C+I algorithm has allowed us to prove and demonstrate the effectiveness of the algorithm on new scenarios, different to those that the original algorithm had been tested on, such as the prioritization of loads and giving them specific cost functions. The algorithm lacks optimality in scenarios where multiple devices controlled by the same node have to operate in their Marginal Power Region, a very rare scenario. Nonetheless, convergence is still guaranteed with speeds similar to that of the original algorithm.

How does the C+I algorithm perform on scenarios of islanding and de-islanding?

The C+I algorithm, after having undergone minor adjustments to support (de-)islanding scenarios, has proven to be very effective on at least a 5-node network. The network was set up at the Green Village, an innovation site where technologies in the field of sustainable energy provision are tested and applied in a real-life environment, which further proves the effectiveness of the algorithm on large-scale, real and physical environments.

FUTURE WORK

While the algorithm has been verified to be functional and effective on small-scale networks, it would be interesting to test it on larger-scale networks, whether that means a greater number of nodes, or a bigger existing infrastructure, with longer cable lines and generators/loads of greater magnitude.

It is also interesting to note that, while most of our case studies focused on proving the functionality of the algorithm in varying scenarios, it would be worthwhile to test scenarios that specifically make convergence time-costly, such as current congestion scenarios, which were not easily replicable in our experiments. Another example of a scenario that makes convergence difficult is a network with a great amount of nodes between a load and a cheap generator, which would make the propagation of the Locational Marginal Price slow, since the algorithm does not consist of individually-solved sub-problems but of collaborative efforts towards solving one big problem. Additionally, any such stress-test results could be compared with implementations of other OPF al-

gorithms, to compare the performance between the two algorithms, and practically find where the C+I algorithm falls short, and where it prevails.

Finally, we know that the C+I algorithm has been extended, from the unipolar version of Parreira[17], to the bipolar version of Najjar[15]: it would be interesting to implement the bipolar version of the algorithm for a physical grid, to test its effectiveness.

Regarding our modified C+I algorithm for region-OPF, a solution could be devised for the current, non-optimal handling of multiple same-region devices in the Marginal Power Region, or a new approach altogether could be proposed for adjusting the C+I algorithm for region-OPF.

EPILOGUE

"A cool breeze rustled the trees, dried the sweat from my skin, and soothed my aching bones. It whispered in my ear and shared a secret, which echoed in my brain, like a drumbeat that wouldn't stop: "There is no finish line, Goggins. There is no finish line..."

APPENDICES

A. FUNDAMENTALS OF ELECTRICITY

A.1. FUNDAMENTAL LAWS OF ELECTRICITY

The equations as shown below hold, given the characteristics of an electric network as described in section 2.1. For power converters m , n , and a transmission line (m, n) , the following hold:

$$p_m = u_m * i_m \quad (1)$$

Ohm's Law

$$i_{mn} = G_{mn} * (u_m - u_n) \quad (2)$$

Kirchhoff's Law

$$i_m = \sum_{n \in \Omega_m} i_{mn} \quad (3)$$

$$u_m - u_n = R_{mn} * i_{mn} = \frac{i_{mn}}{G_{mn}} \quad (4)$$

A.2. THE INTERACTION BETWEEN A LOAD AND A GENERATOR

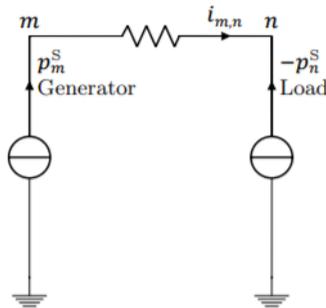


Figure 1: The interaction between a load and a generator

In the event of power demand from a load, the load will automatically experience a voltage drop because of power withdrawal, resulting in a neighboring generator having a higher voltage. This voltage difference triggers power flowing from that generator to the load, and then the generator is required to produce enough power to balance-out the load's demand.

BIBLIOGRAPHY

- [1] Tarek Alskaf and Gijs van Leeuwen. “Decentralized Optimal Power Flow in Distribution Networks Using Blockchain”. In: *International Conference on Smart Energy Systems and Technologies (SEST)* (2019). DOI: [10.1109/SEST.2019.8849153](https://doi.org/10.1109/SEST.2019.8849153).
- [2] Javad Mohammadi Amin Kargarian and Junyao Guo. “Toward Distributed/Decentralized DC Optimal Power Flow Implementation in Future Electric Power Systems”. In: *IEEE Transactions on Smart Grid* PP99 (2016). DOI: [10.1109/TSG.2016.2614904](https://doi.org/10.1109/TSG.2016.2614904).
- [3] Niveen Badra Amr Khaled and Almoataz Y. Abdelaziz. “Optimal Power Flow Methods: A Comprehensive Survey”. In: *IEEJ* 7 (2016).
- [4] G. Cohen. “Auxiliary problem principle and decomposition of optimization problems”. In: *Journal of Optimization Theory and Applications* 32 (1980), pp. 277–305. DOI: [10.1007/BF00934554](https://doi.org/10.1007/BF00934554).
- [5] Paul-Olivier Dehaye. *Inferring distance from Bluetooth signal strength: a deep dive*. URL: <https://medium.com/personaldata-io/inferring-distance-from-bluetooth-signal-strength-a-deep-dive-fe7badc2bb6d>.
- [6] Diwangkoro M. Dolaputra. “Fully Distributed Optimal Power Flow”. MA thesis. Delft, The Netherlands: TU Delft, 2018. URL: <https://repository.tudelft.nl/islandora/object/uuid%3A8dc35dc4-0a6f-4560-a503-e0bbb4a6b1a9>.
- [7] Aouss Gabash Erfan Mohagheghi Mansour Alramlawi and Pu Li. “A Survey of Real-Time Optimal Power Flow”. In: *Energies* 11 (2018). DOI: [10.3390/en1113142](https://doi.org/10.3390/en1113142).
- [8] Tomaso Erseghe. “Distributed Optimal Power Flow Using ADMM”. In: *IEEE Transactions on Power Systems* 29 (5 2014). DOI: [10.1109/TPWRS.2014.2306495](https://doi.org/10.1109/TPWRS.2014.2306495).
- [9] Cedric Jozs Fariba Zohrizadeh and Ming Jin. “A survey on conic relaxations of optimal power flow problem”. In: *European Journal of Operational Research* 287 (2020), pp. 391–409. DOI: [10.1016/j.ejor.2020.01.034](https://doi.org/10.1016/j.ejor.2020.01.034).
- [10] Soumya Kar Javad Mohammadi and Gabriela Hug. “Distributed Approach for DC Optimal Power Flow Calculations”. In: *arXiv: Optimization and Control* (2014). URL: <https://arxiv.org/abs/1410.4236>.
- [11] Johan Larsson. “Distance Estimation and Positioning Based on Bluetooth Low Energy Technology”. MA thesis. Stockholm, Sweden: KTH Royal Institute of Technology, 2015. URL: <http://www.diva-portal.org/smash/get/diva2:859549/FULLTEXT01.pdf>.
- [12] Laura Ramirez-Elizondo Laurens Mackay Tsegay Hailu and Pavol Bauer. “Decentralized current limiting in meshed DC distribution grids”. In: *IEEE International Conference on DC Microgrids (ICDCM)* (2015). DOI: [10.1109/ICDCM.2015.7152045](https://doi.org/10.1109/ICDCM.2015.7152045).

- [13] Sudipta Chakraborty M. Godoy Simoes Caroline S. Uriarte and Felix. A. Farret. "Cost Considerations on Fuel Cell Renewable Energy Systems". In: *IEEE* (2006). DOI: [10.1109/IAS.2006.256843](https://doi.org/10.1109/IAS.2006.256843).
- [14] Nestor Michelena and Panos Papalambros. "Convergence Properties of Analytical Target Cascading". In: *AIAA Journal* 41 (5 2003). DOI: [10.2514/2.2025](https://doi.org/10.2514/2.2025).
- [15] Mohamed-Youssef Najjar. "Decentralized Optimal Power Flow for Bipolar DC Grids". MA thesis. Delft, The Netherlands: TU Delft, 2020. URL: <http://resolver.tudelft.nl/uuid:681cac5f-1a39-4490-86f7-04f1eadda127>.
- [16] Jongtaek Oh and Jisu Kim. "Adaptive -nearest neighbour algorithm for WiFi fingerprint positioning". In: *ICT Express* 4 (2 2018). DOI: [10.1016/j.ictex.2018.04.004](https://doi.org/10.1016/j.ictex.2018.04.004).
- [17] Pedro G. L. Parreira. "Fully Distributed Optimal Power Flow for Low Voltage DC Grids - An optimisation solution using physical measurements". MA thesis. Delft, The Netherlands: TU Delft, 2019. URL: <https://repository.tudelft.nl/islandora/object/uuid:1baf1c80-9415-4f0d-810f-40777f456aed?collection=education>.
- [18] Laurens Mackay Sahil Karambelkar and Shantanu Chakraborty. "Distributed Optimal Power Flow for DC Distribution Grids". In: *IEEE Power and Energy Society General Meeting* (2017). DOI: [10.1109/PESGM.2018.8586629](https://doi.org/10.1109/PESGM.2018.8586629).
- [19] Fraunhofer ISE Simon Philips and Werner Warmuth. *Photovoltaics Report*. 2019. URL: <https://www.ise.fraunhofer.de/en/publications/studies/photovoltaics-report.html>.
- [20] Gabriela Hug Soumya Kar and Javad Mohammadi. "Asynchronous distributed approach for DC Optimal Power Flow". In: *IEEE Eindhoven PowerTech* (2015). DOI: [10.1109/PTC.2015.7232606](https://doi.org/10.1109/PTC.2015.7232606).
- [21] Javad Mohammadi Soumya Kar Gabriela Hug and José M. F. Moura. "Distributed State Estimation and Energy Management in Smart Grids: A Consensus + Innovations Approach". In: *IEEE Journal of Selected Topics in Signal Processing* 8 (6 2014), pp. 1022–1038. DOI: [10.1109/JSTSP.2014.2364545](https://doi.org/10.1109/JSTSP.2014.2364545).
- [22] Jinah Kim Sunmin Lee and Nammee Moon. "Random forest and WiFi fingerprint-based indoor location recognition system using smart watch". In: *Human-centric Computing and Information Sciences* 9 (2019). DOI: [10.1186/s13673-019-0168-7](https://doi.org/10.1186/s13673-019-0168-7).
- [23] Masumi Shirakawa Tomoya Nakatani Takuya Maekawa and Takahiro Hara. "Estimating the Physical Distance between Two Locations with Wi-Fi Received Signal Strength Information Using Obstacle-aware Approach". In: *Proceedings of the ACM on interactive, mobile, wearable and ubiquitous technologies* 2 (3 2018), pp. 1–26. DOI: [10.1145/3264940](https://doi.org/10.1145/3264940).
- [24] Kai-Chao Yao Wei-Tzer Huang and Chun-Ching Wu. "Using the Direct Search Method for Optimal Dispatch of Distributed Generation in a Medium-Voltage Microgrid". In: *Energies* 7 (12 2014), pp. 8355–8373. DOI: [10.3390/en7128355](https://doi.org/10.3390/en7128355).

- [25] Houde Liu Yinliang Xu Hongbin Sun and Qing Fu. “Distributed solution to DC optimal power flow with congestion management”. In: *International Journal of Electrical Power and Energy Systems* 95 (2018), pp. 73–82. DOI: [10.1016/j.ijepes.2017.08.009](https://doi.org/10.1016/j.ijepes.2017.08.009).