Distributed Model Predictive Operation Control of Interconnected Microgrids

ALEXANDRE FOREL
The upward trends in renewable energy deployment in recent years bring new challenges to the development of electrical networks. Interconnected microgrids appear as a novel bottom-up approach to the production and integration of renewable energy. Using model predictive control (MPC), the energy management of several interconnected microgrids is investigated. An optimisation problem is formulated and distributed onto the individual units using the alternating direction method of multipliers (ADMM). The microgrids cooperate to reach a global optimum using neighbour-to-neighbour communications.

The benefits of using distributed operation control for microgrids are analysed and a control architecture is proposed. Two algorithms are implemented to solve the optimisation problem and their advantages or differences are confronted.

**Keywords:** microgrid, energy management, cooperative control, model predictive control, ADMM
Abstract

Förnybara energikällor har ökat under senaste åren. Det innebär nya utmaningar för evolutionen av elektriska nät. Microgrids är en bottom-up ansats för produktion och integrering av förnybar energi.
Energiförsörjning av flera sammankopplade Microgrids studeras in detta arbete genom modellbaserad prediktiv kontroll (MPC). Ett optimeringsproblem formuleras på de enskilda enheterna med Alternating Direction Method of Multipliers (ADMM) och parallell beräkningar härledas. Microgrids samarbetar för att nå en global lösning av neighbour-to-neighbour kommunikation.
Distribuerad energiförsörjning av microgrids analyseras och två kontroll algoritmer utformas.

Keywords: Microgrid, energiförsörjning, distribuerad kontroll, MPC, ADMM
1 Introduction 1
1.1 Integration of renewable energy sources . . . . . . . . . . . . . . . . . . . . 1
1.1.1 Energy storage systems . . . . . . . . . . . . . . . . . . . . . . . . . . . 1
1.2 Microgrids: a bottom-up approach to future energy systems . . . . . . . . 2
1.2.1 Existing microgrids . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 2
1.2.2 Control layers of microgrids . . . . . . . . . . . . . . . . . . . . . . . . 3
1.2.3 Energy management of microgrids . . . . . . . . . . . . . . . . . . . . 3
   1.2.3.1 Optimal control . . . . . . . . . . . . . . . . . . . . . . . . . 3
   1.2.3.2 Model predictive control . . . . . . . . . . . . . . . . . . . 3
   1.2.3.3 Distributed cooperative control . . . . . . . . . . . . . . . 4
1.3 Problem statement . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 4
1.4 Related work . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 4
1.5 Contributions . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 5
1.6 Outline . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 6

2 Theory and methods 7
2.1 Notation . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 7
2.2 System definition and optimal control . . . . . . . . . . . . . . . . . . . . . 7
2.3 Model predictive control . . . . . . . . . . . . . . . . . . . . . . . . . . . . 8
   2.3.0.1 Discount factor . . . . . . . . . . . . . . . . . . . . . . . . . . . 9
   2.3.1 Application to microgrids . . . . . . . . . . . . . . . . . . . . . . . . 10
2.4 Control architectures for energy management . . . . . . . . . . . . . . . . 10
   2.4.1 Centralised model predictive control . . . . . . . . . . . . . . . . . 11
   2.4.2 Distributed hierarchical control . . . . . . . . . . . . . . . . . . . 11
   2.4.3 Distributed neighbour-to-neighbour MPC . . . . . . . . . . . . . . 12
      2.4.3.1 Cooperation and global optimality . . . . . . . . . . . . . . 12
   2.4.4 Distributed control characteristics . . . . . . . . . . . . . . . . . . 12
2.5 Summary . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 13
3 Cooperative distributed optimisation
3.1 Introduction from optimisation theory: convexity
3.1.1 Convex and strictly convex set
3.1.2 Convex and strictly convex functions
3.1.3 Importance of convexity in optimisation theory
3.2 ADMM precursors and optimisation methods
3.2.1 Constrained problem and dual ascent
3.2.1.1 Lagrangian relaxation and Lagrangian dual problem
3.2.2 Separable optimisation problem
3.2.3 Dual decomposition
3.2.4 Augmented Lagrangian method
3.3 Alternating direction method of multipliers
3.3.1 General form of ADMM
3.3.2 ADMM in practice
3.3.2.1 Residuals: a criterion for convergence
3.3.2.2 Penalty parameter
3.3.3 Separable ADMM algorithm and parallelisation
3.3.3.1 Update step of separable ADMM
3.4 Application to consensus problem
3.5 Summary

4 Problem Formulation
4.1 Presentation of the case study
4.2 Single microgrid model
4.2.1 Model presentation
4.2.1.1 State of charge of the electrical storage system
4.2.1.2 Forecasts: load demand and renewable energy in-feed
4.2.1.3 Disturbances on the system
4.2.2 Optimisation problem formulation
4.2.3 Constraints
4.2.3.1 Load balancing constraint
4.2.3.2 Limits on the generating units
4.2.3.3 Limits on renewable energy
4.2.3.4 Limits on stored energy
4.2.4 Cost function
4.2.4.1 Cost of producing energy
4.2.4.2 Cost function for local problem on prediction horizon
4.2.5 Convexity
4.3 A small grid of interconnected microgrids
4.3.1 Connecting microgrids and power flow
4.3.2 Distributed optimisation problem formulation
4.3.3 Node power and edge power flow
4.3.3.1 New coupling constraint
4.3.3.2 Cost for transmitting power
# List of figures

2.1 Receding horizon control with a finite prediction horizon [48] .......................... 9  
4.1 Exemplary microgrid ................................................................. 26  
4.2 Naive forecast with time delay ..................................................... 28  
4.3 Cost associated with the stored energy state ..................................... 31  
4.4 Four interconnected microgrids ..................................................... 33  
6.1 Power generation and transmission over a one week simulation ............... 51  
6.2 Simulation results for Microgrid 1 over a one week simulation ................. 52  
6.3 Power transmission on the electric network over a simulation of one week ... 53  
6.4 Objective cost difference between central and distributed ADMM algorithms 54  
6.5 Local voltage phase angle evolution of central and distributed ADMM algorithms 55  
6.6 ADMM algorithms with varying the step-size $\rho$ ................................. 57  
6.7 Number of iterations for convergence with varying step-size ................. 58  
6.8 Number of iterations for convergence with varying absolute tolerance ....... 58  
6.9 Computing time for central controller and ADMM with substitution with varying precision ................................................................. 59
# List of tables

5.1 Classification of decision variables for microgrid $i$ .......................... 41  
5.2 Decision variables and parameters used in Sequential ADMM for microgrid $i$ 42  
5.3 Decision variables and parameters used in ADMM with substitution for microgrid $i$ ................................................................. 44  
6.1 Parameters of model predictive control .............................................. 48  
6.2 Parameters for microgrid models ......................................................... 48  
6.3 Parameters for transmission network ................................................... 48  
6.4 Parameters for ADMM ................................................................. 49
List of Algorithms

1. Separable ADMM algorithm with parallel computation .................................. 23
2. Distributed sequential ADMM ................................................................. 43
3. Distributed simultaneous ADMM with substitution .............................. 45
## List of acronyms

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADMM</td>
<td>alternating direction method of multipliers</td>
</tr>
<tr>
<td>ESS</td>
<td>energy storage system</td>
</tr>
<tr>
<td>LQR</td>
<td>linear-quadratic regulator</td>
</tr>
<tr>
<td>MIL</td>
<td>mixed integer linear</td>
</tr>
<tr>
<td>MPC</td>
<td>model predictive control</td>
</tr>
<tr>
<td>PID</td>
<td>proportional-integral-derivative</td>
</tr>
<tr>
<td>RES</td>
<td>renewable energy sources</td>
</tr>
</tbody>
</table>

xv
Chapter 1

Introduction

1.1 Integration of renewable energy sources

Due to more and more concern around the use of fossil fuels and the consequential demand for more sustainable energy, the share of renewable electricity generation has greatly increased in the last decades [35]. Energy production is transitioning from a centralised grid of highly capable conventional generators, operating on fossil or nuclear, to small scale renewable-based production sites. This paradigm shift leads the electrical power network to face key problems in order to increase the global share of renewable-based electricity.

One of the main problems encountered in grids with a high share of renewable electricity lies in renewable energies intermittent nature and the possible asynchronism of supply and demand [38]. For instance, when solar energy is highly available may not be the same moment it is most needed or used, and the same can apply for wind energy. Renewable detractors often argue that these electricity sources may not be sufficient to ensure the base load, which is a constant power demand that has to be generated through the day.

1.1.1 Energy storage systems

In this context, energy storage system (ESS) are getting more and more attention and are sometimes considered as the keystone of tomorrow’s electrical grid [44]. These devices are based on different processes: electrochemical batteries such as lithium-ion or sodium-sulphur batteries for instance, or mechanical processes such as pumped-hydro storage, compressed-air energy storage or flywheels.

In any case, the use of electrical energy storage can benefit an electrical network in diverse ways and on different time scales:

- On a fast time scale, storage units can provide power for primary frequency control to compensate for deviations in energy production and consumption [31].
• On a longer time scale, using ESS can help flattening the production curve with regards to the peak load demand, and used for economical dispatch [8, 5]. From an energy management point of view, the batteries also allow storing and shifting the use of renewable-based electricity in place of a fossil or nuclear generator.

By storing renewable infeed when it is most available, and redistributing it when it is most needed, ESS can provide a solution to increase the global share of renewable generation. This could also help to reduce greenhouse-gases emissions which aligns with recent political decisions. Reducing greenhouse-gases and addressing climate change are goals that have been targeted during the 21st conference of the parties [36] and have also been endorsed by the European Union. An objective is for instance to aim at reducing greenhouse emission by 20% by 2020 compared to the levels of 1990, and by 80% by 2050 [43].

1.2 Microgrids: a bottom-up approach to future energy systems

The term microgrid (MG) generally designates a local group of electric units and loads. They can either be connected to the main electrical grid or operate autonomously in so called islanded mode. Even though the term is not fixed and it is common to find different type of microgrids in literature, a microgrid is commonly characterised by having a relatively high share of renewable but intermittent energy sources and can use storage to optimise the use of the former. Controllable loads or electric vehicles can also be part of a microgrid [40]. As such, a typical example of microgrid is a residential area composed of households equipped with generation sources such as solar panels, wind turbines or combined heat and power together with electrical batteries. Microgrids can be independently installed, and later connected together to form a network of microgrids. This can be seen as a bottom-up strategy: gradually building a decentralised electrical grid.

Microgrids can have a prominent place in the energy system of tomorrow especially in developing countries or remote areas as it may allow more proximity between loads and supply, and therefore lead to fewer transmission losses.

1.2.1 Existing microgrids

Microgrids in islanded mode are particularly interesting for the electrification of remote areas as deserts cut the electrical grid and leave villages without connection to the global grid, see e.g. [18]. Another application can be areas that are electrically isolated due to geographical reasons such as islands. On Kodiak Island for instance mechanicals flywheels are operated in order to decrease the use of diesel generators and focus on using wind and hydro power [25]. Other examples of existing microgrids can be seen in West African states. Although it varies for each country, it is estimated that less than 10 percent of the rural population has access to electricity [16], and less than 20 percent of
1.2. MICROGRIDS: A BOTTOM-UP APPROACH TO FUTURE ENERGY SYSTEMS

Communities are connected to power infrastructure. The development of relatively cheap microgrids with small battery units could be a solution for electrifying in a bottom-up approach: electrifying the villages in an islanded fashion first and slowly connecting them together thanks to the plug-and-play dimension of microgrids [42].

1.2.2 Control layers of microgrids

Motivated by conventional power systems, a control hierarchy of three layers with three time scales has been advocated for microgrids as well [26]. The layers are as follows.

1. Primary control ensures the voltage and frequency stability of the grid by balancing supply and demand and ensuring power sharing by implementing a proportional control called droop control. It operates in a decentralised manner within a typical time scale of some milliseconds to seconds.

2. Secondary control operates slower than primary control and aims at compensating the steady-state deviation in both voltage and frequency. The time scale is some seconds to minutes.

3. Tertiary or operation control (also called energy management) has the goal of economically optimising the operation of the grid. It focuses on the dispatch of power between the different units of the system in order to reach an optimal behaviour. It operates on the time scale of minutes to hours.

1.2.3 Energy management of microgrids

1.2.3.1 Optimal control

Optimal control characterises the technique of designing a control input by solving an optimisation problem. A common way of formulating this problem revolves on using an objective function that is minimised over a set of inputs restricted by a set of constraints. This approach is especially suitable for the operation control of microgrids as both the objective function and the constraints are highly customisable and adaptable to the problem. Moreover, there is a strong parallel between optimisation theory in control and in economics that is applicable to the energy management of microgrids. In [4] for instance a method of deducing shadow costs from an optimisation problem is presented.

1.2.3.2 Model predictive control

Model predictive control (MPC) is an optimal control method that aims at solving an optimisation problem over a prediction horizon. The length of the prediction horizon is chosen considering the precision of a forecast model or system model. Model predictive control offers a lot of advantages because of its ability to include supply and/or demand forecasts. It is also possible to develop MPC with plug-and-play aspects which makes it suitable for distributed control [49].
1.2.3.3 Distributed cooperative control

One key aspect of microgrids control is to coordinate the individual agents to reach a global optimum and, as such, having a network of cooperating and not competing agents. Even though a central controller is often the easiest way of reaching global optimality, it is not often easy to implement, and in particular the complexity of calculation and the communication burdens increase with the size of the network.

Distributing the problem onto cooperative agents that reach the same global solution can have many advantages as in [28]. Distributing the computation may lead to a better scalability as the computational and communication burdens are expected to increase slower than with a centralised controller. Communications required for designing the control law are exchanged with the local neighbours only. Because there is no central entity that gathers the data, this process can also lead to more privacy. With an appropriate distributed control architecture, a microgrid would not be able to use the communicated data from its neighbours to deduce their decisions or strategies. Moreover, distributed control schemes may be more adaptable as units can be connected/disconnected without having to recalculate the controller completely. This behaviour can be described as plug-and-play and allows for more flexibility and robustness. In case of an accidental line disconnection that cut a grid in two parts, for instance, a network could continue to operate thanks to distributed control.

1.3 Problem statement

This thesis focuses on the energy management of microgrids. Our target is to maximise the use of the available renewable infeed by storing energy in the ESS and reusing it when needed. By employing forecasts for the future renewable infeed and load demand, an optimal control law is designed that schedules the use of the battery in charge or discharge. The microgrids should be controlled locally but also be able to cooperate by transmitting power on the network. By designing a fully distributed and cooperative algorithm, we expect to reach a control scheme doted of the aforementioned traits of local controllers: scalability, privacy and adaptability.

The consequences of solving the economic dispatch problem without any central entity should be investigated and the implementation of distributed neighbour-to-neighbour control should be detailed. In particular, the energy management problem is expected to be solved without too much increase in the computation time compared to the central control case. The solution algorithms should be applicable to different network architectures in order to be considered adaptable.

1.4 Related work

Model predictive control has been recently very popular in the field of microgrid research and especially when applied to the tertiary control layer. In [33, 34], a mixed integer linear (MIL) MPC-based operation controller is presented on a single microgrid. In [14] the aspect of uncertainties on the forecasts is analysed and solved with a minimax model.
The theoretical background of decomposing an optimisation problem and solving it in a distributed fashion has also recently received some serious attention. Different approaches exist and several problem structures have been investigated. Dual decomposition methods have been investigated in [19] and [10] for instance. In [2], a methodology to derive and apply the alternating direction method of multipliers (ADMM) is presented. This method is expected to yield better performances than dual decomposition when applied to certain optimisation problems. To further investigate the speed of convergence of ADMM-based optimisation, other methods have been proposed: [50, 51, 52] for instance introduce a promising ADMM-based algorithm over graphs with good performances. Combining this theoretical background on distributed optimisation and the research on microgrid models, distributed control algorithms have been investigated for interconnected microgrids. In particular, the ADMM method has been recently applied to solve energy management problems in a distributed way. In [17] a distributed ADMM method is presented for solving dynamic network energy management based on message passing. [47] extends the previous work using predictions models and including curtailable loads and electric vehicles in the grid model. In [54] a fully distributed ADMM-based method for cooperating microgrids is proposed for a small network of interconnected microgrid. It is however based on fully controllable power flows.

## 1.5 Contributions

The main focus of this thesis is to develop a fully distributed cooperative ADMM-based algorithm in order to reach global optimality. Although distributed energy management of interconnected microgrids has already been investigated in literature, this thesis aims at deriving power flows over the network with an approach based on the local voltage phase angle of microgrids. By assuming that the power flows are not directly controllable, this method provides a more general overview. Fully controllable power flow are attainable using power electronics but these equipments may not always be available in practice. Instead, DC power flow approximations are applied over inductive lines.

Therefore, the main contribution of this thesis is to extend the works presented in Section 1.4 by solving the energy management problem of several interconnected microgrids using a fully distributed neighbour-to-neighbour algorithm with non fully controllable power flows. To solve the problem, two distributed algorithms based on ADMM are derived. Because the efficiency of such algorithms is problem dependant, an analysis of the tuning of these controllers will be presented. A comparison of their characteristics, benefits and disadvantages is also provided. Eventually, the results of the two solution algorithms are compared to the central controller and confronted in terms of computation speed.
1.6 Outline

The remainder of this thesis is organised as follows. Chapter 2 introduces theoretical notions on optimisation problems and model predictive control and presents different control architectures, from centralised to distributed. Chapter 3 focuses on the theory of cooperative control and presents ADMM. In Chapter 4 the microgrid model is presented, as well as the case study of four interconnected microgrids, and their formulation as an optimisation problem is described. Chapter 5 presents distributed controllers based on ADMM in order to solve the problem of energy management of interconnected microgrids. Chapter 6 shows some numerical results and especially aims at comparing the distributed controllers to the central one.
Chapter 2

Theory and methods

This chapter introduces theoretical notions on model predictive control (MPC) and presents different architecture for the control of multi-agent systems. In particular, the transition from centralised control to fully distributed control is investigated.

2.1 Notation

Let discrete time $t$ be denoted as an argument by brackets $[t]$. A sequence from $a$ to $b$ is denoted $[a; b]$ and if $k$ is a discrete time variable, $k \in [a; b]$ denotes all natural numbers from $a$ to $b$ included. Let $f : \mathbb{R}^n \rightarrow \mathbb{R}$ be the objective function of an optimisation problem over $x$ where $n \in \mathbb{N}$, $f^*$ is an optimal solution value of the optimisation problem, and $x^*$ denotes an antecedent of this optimal solution, i.e. $f(x^*) = f^*$. The transpose of a vector $x \in \mathbb{R}^n$ is denoted $x^T$ and its Euclidean norm is denoted $\|x\|_2$.

2.2 System definition and optimal control

The state-space model of a system in discrete-time is a mathematical description of the evolution of the physical system. It is expressed in (2.1) where $(n, m) \in \mathbb{N}^2$, $x \in \mathbb{R}^n$ is the vector of measurable states, $u \in \mathbb{R}^m$ is the vector of control input, $A \in \mathbb{R}^{n \times n}$ is the state matrix and $B \in \mathbb{R}^{n \times m}$ is the input matrix. It is expressed as

$$x[k + 1] = Ax[k] + Bu[k].$$

(2.1)

The system can be controlled by choosing an appropriate control sequence for $u$ for all discrete time steps $k \in \mathbb{N}$. In order to respect constraints, the state and input vectors can be restricted to an acceptable set of states $X \subset \mathbb{R}^n$, and an acceptable set of inputs $U \subset \mathbb{R}^m$ respectively. This bounds can model safe operation behaviour, saturation or desired limits on the variables.

Optimal control is a mathematical method for deriving control policies by solving an optimisation problem. The cost or objective function $f$ is a function of state variables $x$ and control variables $u$, where the state variable evolves according to the state-space
model in (2.1). Expressing the optimisation problem on the acceptable sets of states and inputs has the form

\[
\begin{align*}
\text{minimise} \quad & f(x, u) \\
\text{subject to} \quad & \begin{cases} 
    x[k + 1] = Ax[k] + Bu[k] \\
    x \in \mathcal{X} \\
    u \in \mathcal{U}
\end{cases}
\end{align*}
\]\n
(2.2)

By solving this problem, an optimal input trajectory \( u^* [k] \) can be derived.

### 2.3 Model predictive control

Model predictive control is a form of optimal control that is based on the iterative optimisation of a system model over a finite prediction horizon. It has been developed in the 1980s to be applied in the process industries such as chemical plants or oil refineries. Since then, model predictive control has been widely adopted in industry. Its main benefits compared to proportional-integral-derivative (PID) or linear-quadratic regulator (LQR) controllers is that it can take into account predictions of future events in the control design. MPC might however be computationally more expensive, as it consists of solving multiple optimisation problems on the prediction horizon. The development of relatively cheap and powerful microprocessors that have made computational power more easily available might be one of the reason for the recent adoption rates in MPC. A deeper look into the theory of MPC and stability analysis can be found in [24].

At time \( k \), the current plant state is sampled and a cost minimising control strategy is computed via a numerical minimisation algorithm over a finite or infinite horizon in the future: \( [k, k + T] \) where \( T \in \mathbb{N} \) is called the prediction horizon. In practice, the choice of the prediction horizon has to be found heuristically. According to [7], a long prediction horizon can usually ensure stability, but on the other hand, large horizons can make predictors lose their accuracy as a consequence of modelling errors, on the forecasts or the system models for instance. The mathematical formulation of model predictive control can be derived from (2.2) as

\[
\begin{align*}
\text{minimise} \quad & \sum_{t=k}^{k+T} f(x[t], u[t]) \\
\text{subject to} \quad & \begin{cases} 
    x[t + 1] = Ax[t] + Bu[t], \quad \forall t \in [k; k + T] \\
    x \in \mathcal{X} \\
    u \in \mathcal{U}
\end{cases}
\end{align*}
\]\n
(2.3)

where \( x \in \mathbb{R}^{n \times T} \) and \( u \in \mathbb{R}^{m \times T} \). In this case, the optimal solution \( u^* \in \mathbb{R}^{m \times T} \) is a matrix on the whole prediction horizon.
2.3. MODEL PREDICTIVE CONTROL

Implementing model predictive control means that only the first step of the control input sequence $u^*[k+1]$ is actually implemented, the following inputs are discarded. After the update and at the next time instant, the state is sampled again and the calculations are repeated starting from the new current state, yielding a new control and new predicted state paths. The prediction horizon keeps being shifted forward and for this reason this implementation of model predictive control is also called receding horizon control. It is important to remember that in the state variable sequence $x$, only the first data $x[k]$ is a measurement whereas the sequence $x[k+1; k+T]$ is deduced from the system model and the derived input $u^*$. Figure 2.1 illustrates the behaviour of MPC and shows the evolution of the control input through the prediction period.

The main advantages of model predictive control, compared to more classic controllers are that

- it can handle very well multi-input/multi-output problems,
- it can easily include constraints on the state and control variables, and
- it is possible to include forecasts in the minimisation over the prediction horizon.

![Figure 2.1. Receding horizon control with a finite prediction horizon [48]](image)

2.3.0.1 Discount factor

As the system or the prediction model disturbances get higher as the horizon period increases, and in order to give more importance to early prediction, a discount factor $\gamma$
can be used in the objective function. The minimisation problem is then of the form

\[
\begin{align*}
\text{minimise} \quad & \sum_{t=k}^{k+T} \gamma^{t-k} f(x[t], u[t]) \\
\text{subject to} \quad & x[t + 1] = Ax[t] + Bu[t], \quad \forall t \in [k; k + T] \\
& x \in \mathcal{X} \\
& u \in \mathcal{U}
\end{align*}
\]

(2.4)

where the discount factor \( \gamma \) is a real value in the interval \((0; 1)\) typically chosen close to 1.

### 2.3.1 Application to microgrids

Model predictive control is especially suitable for the energy management of microgrids because of its inherent capability to include forecasts in the optimisation problem. In the case of microgrids the forecast can be

- prevision on the load demand, that varies through the day but can be estimated thanks to historical data, or
- forecasts on the amount of solar or wind power available that can be deduced from meteorological models.

In both cases, the more accurate the forecast model, the better the performances. An example of forecast model calculation and its application to energy management of microgrids can be found in e.g. [47].

### 2.4 Control architectures for energy management

In the case of interconnected microgrids, operation control is a multi-agent problem. In most cases, the easiest controller to implement is a central controller that has full knowledge of the individual agents states and can control them. In this case, and if the problem exhibits specific properties (convexity for instance, see Section 3.1), the solution of the optimisation problem is said globally optimal which means that the cost function can not be further minimised and no other control sequence could yield a better result from the optimisation point of view. One of the main focuses of this thesis is to provide a fully distributed control architecture where each agent can locally compute and implement its control law.

In this section different control architectures from central control to distributed are presented. Because the use of the terms hierarchical, distributed, decentralised may have different meaning according to the context, one important aspect of this section is to explain clearly what is the meaning of the terms in the context of this thesis.
2.4. CONTROL ARCHITECTURES FOR ENERGY MANAGEMENT

2.4.1 Centralised model predictive control

Centralised MPC consists of one unique central controller that solves a global optimisation problem with regards to a cost function regrouping the inputs and outputs of all the units involved. It is generally assumed that the centralised controller disposes of all the data of all agents.

In such conditions, the centralised controller yields the optimal solution of the optimisation problem. The drawbacks can be

- its communicational burden and its weak scalability as adding/removing a unit requires a reformulation of the problem and recalculation of the controller, and
- the computing time that gets higher with the number of units.

Centralised control is especially suitable for model predictive control considering its intrinsic ability for handling several states and variables [37].

2.4.2 Distributed hierarchical control

This control structure can be considered as an intermediate step between central control and fully distributed control. It designates a multi-agent system communicating with a central entity, at the higher level that coordinates their actions. The local regulators are placed at a lower level.

Hierarchical control with MPC is sometimes called Coordinated Distributed MPC [19], as the term hierarchical often describes a control architecture organised in different layers of different time-scales, such as the three layers of microgrid control mentioned in Section 1.2.2.

The main idea is to separate and distribute the computation in parallel to all the individual agents. The central entity receives all the predicted control solutions and coordinate the agents. It can for instance calculate a price that will impact the local cost function: by varying the price, the central entity can influence the local optimal solution of the individual agents, and as such lead the global system to the optimal global solution.

The main benefit of this method is to implement some parallel computation that can have a big impact on the global computing time, and especially for huge networks. However, since the coordinator needs to communicate with all agents, it does not solve the communicational burden of the central entity gathering and broadcasting some data to all agents. The central entity may for instance gather some information on the state of the units, and broadcast them a corresponding price. Hierarchical control can use iterative algorithm to converge. An iterative hierarchical control with broadcast/gather communications can be found in e.g. [5].

As in central control, the calculations of the central entity are expected to increase with the size of the network. However, as some of the computation is done in parallel, hierarchical control is expected to yield better performances in term of computation time.
2.4.3 Distributed neighbour-to-neighbour MPC

Contrary to hierarchical control, a distributed control scheme does not use any central entity. Instead, the units locally calculate the control law by exchanging data with their connected neighbours. This communication scheme is called neighbour-to-neighbour communications, see e.g. [19]. The data exchanged can be some informations on the state of the local subsystems, or some informations on the desired control sequence. In distributed control, the local agent do not necessarily have global knowledge of the system. This architecture differs from decentralised control which is characterised by using neither a central entity nor communications between the units.

In distributed energy management, each local microgrid has some knowledge on the behaviour of the others. The individual agents do not have global knowledge but are able to communicate with their local neighbours and as such influence their control sequence. In order to converge to a global optimum, the local optimisation problems are often solved iteratively and use a combination of older predicted states and newly computed control sequences. As a result of these iterated calculations, distributed control schemes have to bear a computational and communicational burden in order to converge to a global optimum [21]. The benefits in terms of computation time are therefore fully dependant on the type of problem, the size of the system and the network topology. In general, and although it depends strongly on the structure of the problem: the bigger the network size, the more advantageous distributed control is expected to be. In [29] for instance, the convergence speed of centralised and distributed algorithms with MPC are confronted with varying network size.

2.4.3.1 Cooperation and global optimality

The case of multi-agent optimisation has a strong parallel with game theory. An analysis of this topic can be found in e.g. [22]. As local agents tend to optimise their local objective, it is probable that a behaviour that minimise the local cost may not be beneficial to the global network. This behaviour could be qualified as competitive. On the contrary, endorsing a locally sub-optimal behaviour in order to reach a global optimal could be qualified as a cooperative behaviour. According to [46], independent algorithms show competitive behaviours where each local agent tends to move towards a Nash equilibrium, whereas iterative and cooperating methods tend to output a Pareto optimal solution, as provided by an ideal centralised controller.

2.4.4 Distributed control characteristics

To sum up, we can highlight three choices that characterise a certain type of distributed control architecture.

- The units communicate to all the other units or only to some local neighbours.
• The units communicate once every sampling interval (non-iterative process) or they can communicate several times within a sampling interval (iterative process).

• Each agent minimises an independent local cost function or locally minimises a global cost function.

2.5 Summary

In this chapter the mathematical formulation of a discrete-time system and its inclusion in optimisation problems was presented. Model predictive control was introduced and different approaches for its implementation were described. The benefits of applying distributed MPC to the energy management of microgrids appear promising in terms of local control and actuation of the microgrids, reducing the communicational burden and decreasing the computation needed with big networks.

The following section will investigate the decomposability and separability of optimisation problems. In the case of a non fully-separable problem, optimisation methods will be proposed to apply distributed control with fast convergence. In particular the theory behind an iterative mathematical method to locally update a global cost function will be presented by introducing ADMM.
Chapter 3

Cooperative distributed optimisation

This chapter aims at introducing the background and theory behind distributed ADMM. First some basic notions on convexity are presented in order to identify the characteristics and problematics of solving an optimisation problem in a distributed way. Then, optimisation methods relying on the dual problem formulation are presented. Their separability aspects are investigated for problems with specific structures. Finally, the method to derive ADMM is deduced by introducing anterior optimisation methods such as dual ascent, dual decomposition and the method of multipliers. A more detailed presentation of the alternating direction method of multiplier can be found in [2].

3.1 Introduction from optimisation theory: convexity

Convexity in optimisation is a central property necessary to ensure that, if a globally optimal solution that respects the constraints exists, it is attainable and unique [30]. In this section are introduced notions on convexity for both a set and a function.

3.1.1 Convex and strictly convex set

A set $S$ is convex when for all $(x_1, x_2) \in S^2$ there is a path linking $x_1$ to $x_2$ that remains in the set $S$. This condition is expressed by

$$\forall \lambda \in [0, 1], \lambda x_1 + (1 - \lambda) x_2 \in S.$$  \hspace{1cm} (3.1)

A set $S$ is strictly convex if for all $(x_1, x_2) \in S$ there is a path linking $x_1$ to $x_2$ that strictly stays in the interior of the set $S$ denoted $\text{int}(S)$, i.e.

$$\forall \lambda \in (0, 1), \lambda x_1 + (1 - \lambda) x_2 \in \text{int}(S).$$  \hspace{1cm} (3.2)
CHAPTER 3. COOPERATIVE DISTRIBUTED OPTIMISATION

3.1.2 Convex and strictly convex functions

In a similar way to convex sets, the notion of convexity can be applied to functions. First, the notion of domain of a function $f$ has to be defined. The domain of a function $f: \mathbb{R}^n \rightarrow \mathbb{R} \cup \{+\infty\}$ is defined by

$$\text{dom}(f) = \{x | f(x) \neq +\infty\}. \quad (3.3)$$

The concept of domain (3.3) allows to deduce the following definitions for the function $f$.

A function $f$ is said to be proper when its domain is non-empty

$$\text{dom}(f) \neq \emptyset. \quad (3.4)$$

A function $f$ is convex if for every pair of points $(x_1, x_2) \in \text{dom}(f)$, the chord between the two points lies above the function, i.e.

$$\forall \lambda \in [0, 1], \forall x_1, x_2 \in \text{dom}(f), f(\lambda x_1 + (1 - \lambda) x_2) \leq \lambda f(x_1) + (1 - \lambda) f(x_2). \quad (3.5)$$

A function $f$ is strictly convex if for every pair of points $(x_1, x_2) \in \text{dom}(f)$, the chord between the two points lies strictly above the function, i.e.

$$\forall \lambda \in (0, 1), \forall x_1 \neq x_2 \in \text{dom}(f), f(\lambda x_1 + (1 - \lambda) x_2) < \lambda f(x_1) + (1 - \lambda) f(x_2). \quad (3.6)$$

3.1.3 Importance of convexity in optimisation theory

The concept of convexity is crucial in optimisation theory [30], as a

- convex function $f$ has a unique minimum value $f^*$, which means that a local minimum of a convex function is also a global minimum, and a
- strictly convex function $f$ has a unique minimum value $f^*$ and a unique antecedent $x^*$ such that $f(x^*) = f^*$.

As such, an optimisation problem is said to be convex, if the cost function $f$ is convex, and if the acceptable set of input $S$ formed by the constraints is convex [30].
3.2 ADMM precursors and optimisation methods

This section presents the path taken in [2] to deduce ADMM. This is done by progressing from existing distributed optimisation methods and combining them. In this part, all optimisation problems are supposed convex.

3.2.1 Constrained problem and dual ascent

Let us consider a convex optimisation problem with equality and inequality constraints of the form

\[
\begin{align*}
\text{minimise} & \quad f(x) \\
\text{subject to} & \quad Hx = h \\
& \quad Gx \leq g
\end{align*}
\]  

(3.7a)  

(3.7b)  

(3.7c)

where \( x \in \mathbb{R}^n \), \( H \in \mathbb{R}^{p_e \times n} \), \( h \in \mathbb{R}^{p_e} \), \( G \in \mathbb{R}^{p_i \times n} \), \( g \in \mathbb{R}^{p_i} \). Furthermore; \( f : \mathbb{R}^n \rightarrow \mathbb{R} \) is supposed convex and proper. In this problem, (3.7a) shows the minimisation of the objective function, (3.7b) ensures the \( p_e \) equality constraints and (3.7c) ensures the \( p_i \) inequality constraints.

The optimisation problem (3.7) is said to be the **primal** problem. In the following subsection 3.2.1.1, the dual problem associated to the primal problem (3.7) will be introduced. Formulating the dual problem provides a way of solving the optimisation problem without projecting onto the feasible set, which might cause extensive calculations. If the optimisation problem has a specific structure, it may also allow for distributed calculation methods and parallelisation.

3.2.1.1 Lagrangian relaxation and Lagrangian dual problem

The Lagrangian relaxation is an optimisation method that consists in relaxing a constraint by assigning it a cost in the objective function if it is not respected [11]. In this case, the Lagrangian relaxation of the optimisation problem (3.7) becomes

\[
L(x, \lambda, \mu) = f(x) + \lambda^T(Hx - h) + \mu^T(Gx - g)
\]

(3.8)

where \( \lambda \in \mathbb{R}^{p_e} \) and \( \mu \in \mathbb{R}^{p_i} \) are called the **Lagrange multipliers**, or **dual vectors**. The Lagrange multipliers are sometimes called the price or shadow cost when used in economical optimisation problems [4].

This allows to rewrite the optimisation as a dual problem over the dual function \( d \). This is called the Lagrangian dual problem as seen in (3.9).

\[
\begin{align*}
\text{maximise} & \quad d(\lambda, \mu) = \inf_x L(x, \lambda, \mu) \\
\text{subject to} & \quad \mu \succeq 0 \\
& \quad \text{element-wise positive}
\end{align*}
\]  

(3.9)
CHAPTER 3. COOPERATIVE DISTRIBUTED OPTIMISATION

The dual problem is a concave optimisation problem as it maximises a concave objective function and the constraint set is convex. Furthermore, the convexity of the dual problem does not rely on the convexity of the primal problem \[30\]. Let \(f^* \in \mathbb{R}\) be the optimal solution of the primal problem (3.7) and \(d^* \in \mathbb{R}\) be the optimal solution of the dual problem (3.9). In general, the dual problem solution provides a lower bound for the primal one, and \(d^* \leq f^*\).

Let us assume that there exists a feasible solution. Then, the condition of strong duality holds, i.e. the dual and primal optimal value are equal: \(d^* = f^*\). Therefore, minimising the primal problem (3.12) is equivalent to maximising the dual problem (3.9) \[4\].

Dual ascent \[2, 4\] is an optimisation method that takes advantage of this fact. It solves the optimisation problem (3.7) by alternatively minimising the primal problem and maximising the dual problem (3.9). The update step of dual ascent at iteration \(\nu\) can be expressed by

\[
\begin{align*}
    x^{\nu+1} & := \arg\min_x L(x, \lambda^\nu, \mu^\nu) \quad (3.10a) \\
    \lambda^{\nu+1} & := \lambda^\nu + \alpha (Hx^{\nu+1} - h) \quad (3.10b) \\
    \mu^{\nu+1} & := \mu^\nu + \alpha (Gx^{\nu+1} - g) \quad (3.10c)
\end{align*}
\]

where the dual problems (3.10b), (3.10c) are solved using gradient ascent \[2\]. In this update sequence, \(\alpha \in \mathbb{R}^+\) is a step-size parameter that has to be chosen appropriately. One of the major benefits of the dual ascent method is that it can allow for distributed computation if the problem structure is suitable \[2\]. In the following sections we will investigate these problems with particular structures.

### 3.2.2 Separable optimisation problem

An optimisation problem on a cost function \(f : \mathbb{R}^n \to \mathbb{R}\) is said separable if the cost function is decomposable into \(N \in \mathbb{R}\) subset as

\[
f(x) = \sum_{i=1}^{N} f_i(x_i)
\]

where \(x = (x_1, x_2, ..., x_N)\) is such that \(x_i \in \mathbb{R}^{n_i}\) are subvectors of \(x\).

The minimisation of the objective function \(f\) can be separated as the sum of the subfunctions \(f_i\) as

\[
\min_x f(x) = \min_x \sum_{i=1}^{N} f_i(x_i) . \quad (3.12)
\]

An optimisation is said fully-separable when there is no coupling variable or constraints linking the different subfunctions. If an optimisation problem is fully-separable, it is almost immediate to distribute onto the \(N\) subsystems using

\[
x^* = \arg\min_x f(x) = [\arg\min_{x_1} f(x_1) ... \arg\min_{x_N} f(x_N)] . \quad (3.13)
\]
3.2. ADMM PRECURSORS AND OPTIMISATION METHODS

3.2.3 Dual decomposition

Dual decomposition is an extension of the dual ascent method if the cost function $f$ is separable in $N$ subsets. This method is suitable when the problem has coupling constraints. By relaxing these coupling constraints, the problem can be decomposed in several subproblems [32].

Let us consider an optimisation problem with only equality constraints

\begin{align}
\text{minimise} \ f(x) \\
\text{subject to} \ Hx = h.
\end{align}

The matrix $H$ has a specific structure. It can be partitioned onto $N$ subsets as

\[ H = [H_1, H_2, \ldots, H_N]. \]

The constraints of the optimisation problem (3.14b) can be subsequently expressed as

\[ Hx = \sum_{i=1}^{N} H_i x_i = h. \]

This specific form for the constraint allows the Lagrangian function to be partitioned accordingly as $L(x, y) = \sum_{i=1}^{N} L_i(x, y)$ where

\[ L_i(x, \lambda) = f_i(x_i) + \lambda^T H_i x_i - \frac{1}{N} \lambda^T q, \quad \forall \ i \in [1, N]. \]

This operation decomposes the primal problem into distributively solvable subproblems, combined with a dual problem that can be considered as a master problem, see e.g. [32]. In the same way as in Section(3.2.1), the minimisation algorithm updates successively the primal and the dual problems. However, in this case, the primal problem can be separated on all partitioned variables $x_i$ which leads to

\[
\begin{cases}
    x_i^{v+1} := \arg\min_{x_i} L_i(x_i, \lambda^v), & i \in [1, N] \\
    \lambda^{v+1} := \lambda^v + \alpha (H x^{v+1} - q)
\end{cases}
\]

where $\alpha$ is the step-size. The value of the step-size can have a strong impact on the convergence speed. In [10] for instance, a method is proposed that divides $\alpha$ by a factor of 10 if the algorithm has not converged after 100 iterations.

The algorithm can be described as a broadcast and gather updating scheme. The individual $x_i$ are updated in parallel using a shared value for $\lambda$ and they are then gathered for the dual update step. Dual decomposition works in the same way as dual ascent but allows for parallel computation and distributed implementation. One of the drawbacks of dual decomposition is that it requires a relatively high number of iterations to converge [2]. This is what motivates the research for a faster algorithm which leads to the following methods.
3.2.4 Augmented Lagrangian method

Augmented Lagrangian methods were developed in order to add robustness to the dual ascent method, but also to obtain a convergent distributed algorithm that does not require strict convexity or finiteness of $f$, as stated in [2]. In practice it consists in adding another quadratic penalty term for diverging from the equality constraint. Compared to the previous dual ascent Lagrangian, the quadratic term converts the convex problem into a strongly convex one.

The augmented Lagrangian of problem (3.14) can be formulated as

$$L_p(x, \lambda) = f(x) + \lambda^T (H x - q) + \frac{\rho}{2} \|H x - q\|^2_2$$

(3.19)

where $\rho > 0$ is the penalty parameter. When the optimal solution $x^*$ is found in the feasible set, the additional quadratic term is zero, and the value of the augmented Lagrangian is unchanged. Similarly, choosing $\rho = 0$ transforms the problem back to a standard non-augmented Lagrangian.

In the same way as in dual ascent, the method of multiplier solves the optimisation problem (3.14) in an alternating way between the primal and dual problems. The dual problem is now a maximisation of the dual augmented Lagrangian. The method of multipliers updates as

$$\begin{cases} x^{v+1} := \arg\min_x L_p(x, \lambda^v), \\ \lambda^{v+1} := \lambda^v + \rho(H x^{v+1} - q). \end{cases}$$

(3.20)

The method of multiplier has been discussed in [1] where it is stated that this method provides good performances when the problem has a high dimensionality or a lot of constraints. In the following section, an attempt to adapt the method of multiplier to distributed computation is presented.

3.3 Alternating direction method of multipliers

3.3.1 General form of ADMM

The alternating direction method of multipliers is a decomposition-coordination procedure that aims at solving a global optimisation problem by decomposing it into local subproblems and coordinating them into finding the optimal solution. It can be considered as an attempt to mix the benefits of the above presented methods: dual decomposition (distributed computation) and augmented Lagrangian methods for constrained optimisation (non-strict convexity and convergence speed).

Let us consider the convex optimisation problem of the general form

$$\begin{align*}
\text{minimise} & \quad f(x) + g(z) \\
\text{subject to} & \quad H x + G z = q
\end{align*}$$

(3.21)

(3.22)
where the functions $f$ and $g$ are convex, $x \in \mathbb{R}^n$, $z \in \mathbb{R}^m$, $H \in \mathbb{R}^{p \times n}$, $G \in \mathbb{R}^{p \times m}$ and $q \in \mathbb{R}^p$. The essential difference from the previous convex problem (3.7) is that the variable $x$ has been split into two parts: $x$ and $z$, and the objective function has been adapted onto these two variables accordingly.

The augmented Lagrangian can be formulated in a similar way as in the previous section, and the algorithm is a mix between the augmented Lagrangian as seen in the the method of multipliers and the distributed computing of the dual decomposition method. It has the form

$$L_p(x, z, \lambda) = f(x) + g(z) + \lambda^T (H x + G z - q) + \frac{\rho}{2} \|H x + G z - q\|_2^2 \quad (3.23)$$

where $\rho \in \mathbb{R}^+$ is the penalty parameter and is strictly positive.

The ADMM algorithm consists of three steps: an $x$ minimisation step, a $y$ minimisation step and a dual update over $y$. It is formulated as

$$\begin{aligned}
    x^{v+1} &:= \arg\min_x L_p(x, z^v, \lambda^v) \\
    z^{v+1} &:= \arg\min_z L_p(x^{v+1}, z, \lambda^v) \\
    \lambda^{v+1} &:= \lambda^v + \rho (H x^{v+1} + G z^{v+1} - q).
\end{aligned} \quad (3.24)$$

### 3.3.2 ADMM in practice

#### 3.3.2.1 Residuals: a criterion for convergence

ADMM being an iterated algorithm, a criterion for establishing convergence has to be chosen in practice. In order to quantify the convergence, residuals on the iterated results can be defined. In this thesis a single residual on the dual problem is chosen as

$$\|s^{v+1}\|_2 = \|\rho (\lambda^{v+1} - \lambda^v)\|_2 \leq \epsilon^{\text{dual}}. \quad (3.25)$$

This dual residual can be considered as a measure of primal feasibility. The difference between iterated Lagrangian multipliers in (3.25) relates directly to the respect of the feasibility constraints, see (3.23). The residual represents how much does the solution at the $v$-th iteration respects the relaxed equality constraint: the bigger the residual, the further the solution is from respecting the equality constraint. Therefore it is also a measurement of the convergence of the algorithm. As it gets closer to the optimal solution, the dual residual should decrease. A termination criterion can be defined when, for instance, the dual residual becomes less or equal than an absolute tolerance parameter $\epsilon^{\text{dual}}$.

In literature, it is common to find two residual criteria: a primal one and a dual one. Definitions can be found, in e.g. [2] where residuals are calculated from the Karush-Kuhn-Tucker conditions. Another method is presented in [47] for example.

#### 3.3.2.2 Penalty parameter

The choice of the penalty parameter $\rho$ is important to decrease the number of iterations needed for convergence although it is particularly problem dependant [47]. As it can
be deduced from (3.24), a big value of the penalty parameter will increase the influence
of the relaxed constraint penalty term in the objective function. It is important to take
into account this behaviour when working in a distributed fashion. There are however
methods to determine the value of $\rho$, as in [12].
In practice, and depending on the optimisation problem, it might be advantageous to
vary the value of the step-size according to the relative importance of the primal or dual
residuals. This is called *adaptive* tuning, and a method can be found in e.g. [2].
Usually, if the primal residual is greater than the dual residual by a certain factor then the
penalty parameter can be increased. Similarly, if it is smaller than the dual residual by a
certain factor, the penalty parameter can be decreased. Another method for varying the
parameter according to the primal or dual residual can be found in [17].

### 3.3.3 Separable ADMM algorithm and parallelisation

If the optimisation problem is separable in $N$ subsets $X_i, i \in [1, N]$ that are closed and
convex, then the problem can be distributed onto $N$ agents. Each local agent $i \in [1; N]
minimises its local objective function (3.26) as

$$x_i^{v+1} := \arg\min_{x_i} L_p(x_1^v, \ldots, x_i^v, \ldots, x_N^v, z^v, \lambda^v).$$ (3.26)

In this local problem, all variables $x_j^v$ where $j \neq i$ are fixed parameter and only $x_i$ is an
optimisation decision variable. Here it is important to note that in the ADMM update step
(3.24), the variables are updated in a Gauss-Seidel fashion, meaning that the calculation of
variable $z^{v+1}$ uses the newly available results for $x^{v+1}$. In (3.26), the update do not follow
the Gauss-Seidel update as it allows for some parallel calculation. This parallelisation is
done on the variables $x_i$ only.

#### 3.3.3.1 Update step of separable ADMM

If the optimisation problem presented in 3.22 has some separability properties on vari-
ables $X_i, i \in [1, N]$, the corresponding ADMM algorithm can be expressed as in Algorithm
1, where all calculations on $x_i$ can be performed in parallel. An example of this implemen-
tation can be found in e.g. [27]. In this algorithm, the optimisation architecture can be
described as hierarchically distributed, as the Lagrangian update is unique and computed
on a unique node that gathers all information from the distributed calculations.
Algorithm 1: Separable ADMM algorithm with parallel computation

Init: $\nu = 0$

while ($r^\nu > \epsilon^r$ and $s^\nu > \epsilon^s$) and $\nu < \nu_{\text{max}}$ do

for all $i$ (in parallel) do

\[ x_i^{\nu+1} = \operatorname{argmin}_{x_i} L_p(x_1^\nu, ..., x_{i-1}^\nu, x_i, x_{i+1}^\nu, ..., x_N^\nu, \lambda^\nu); \]

Gather all $x_i^{\nu+1}$;

$z$ update: $z^{\nu+1} = \operatorname{argmin}_z L_p(x^{\nu+1}, z, \lambda^\nu)$;

Lagrangian update: $\lambda^{\nu+1} = \lambda^\nu + \rho (H x^{\nu+1} + G z^{\nu+1} - q)$;

Calculate residuals $r^\nu$ and $s^\nu$;

$\nu = \nu + 1$

3.4 Application to consensus problem

Consensus problem are widely known in literature on distributed computation. A simple form can be found in [2] but more complex formulations of this problem exist, such as [53] dealing with asynchronous distributed optimisation or [6] that presents inexact consensus. In general, consensus problems were first applications of ADMM on distributed computation.

Applied on a graph of $N$ agents, the goal is to minimise a separable cost function $f$ as described by (3.12) and reach a consensus so that all subfunctions $f_i$ use the same value for the variables $x_i$. The constraint set can be expressed as

\[ X = \{ (x_1, ..., x_N) \mid x_1 = x_2 = ... = x_i = ... = x_N \} \]  \hspace{1cm} (3.27)

where $x_i \in \mathbb{R}^n$ and $f_i : \mathbb{R}^n \to \mathbb{R}$.

The problem can be reformulated by adding a common global variable $z \in \mathbb{R}^n$ that represents the value to which all variables $x_i$ should converge. The optimisation problem can be formulated as

\[ \min_{x,z} \sum_{i=1}^{N} f_i(x_i) \]  \hspace{1cm} (3.28)

subject to $x_i - z = 0, \quad i = 1, ..., N$. \hspace{1cm} (3.29)

The distance from $x_i$ to $z$ is then penalised and introduced in the cost function using augmented Lagrangian relaxation. The update step of the algorithm for solving the consensus problem with ADMM as in [2] is

\[
\begin{align*}
    x_i^{\nu+1} &= \operatorname{argmin}_{x_i} \left( f_i(x_i) + \lambda_i^{\nu} (x_i - z^{\nu}) + \frac{\rho}{2} \| x_i - z^{\nu} \|_2^2 \right) \\
    z^{\nu+1} &= \frac{1}{N} \sum_{i=1}^{N} (x_i^{\nu+1} + \frac{1}{\rho} \lambda_i^{\nu}) \\
    \lambda^{\nu+1} &= \lambda^\nu + \rho (x^{\nu+1} - z^{\nu+1})
\end{align*}
\]  \hspace{1cm} (3.30)
CHAPTER 3. COOPERATIVE DISTRIBUTED OPTIMISATION

One first important aspect of this problem is that the coupling appears only in the constraint set: the local objective function $f_i$ depends only on the local variable $x_i$. This is not always the case, and in particular it will not be the case in the following sections of this thesis. As updating $z^{\nu+1}$ requires knowledge of all the variables $x_i$, this problem would correspond to distributed hierarchical control. The updates on $x_i^{\nu+1}$ can be computed in parallel, and the update step of $z^{\nu+1}$ can be considered as a Gauss-Seidel pass.

3.5 Summary

In this chapter, the theoretical background of the alternating direction method of multipliers was given. The optimisation notions presented now allow us to identify the difficulties that may arise when developing distributed and parallel control. Both the convexity of the acceptable sets, and the objective functions of the study case will be investigated in the following chapter. Moreover, the non fully-separability of the optimisation problem, due to coupling constraints on the subsystems, will be analysed.
Chapter 4

Problem Formulation

4.1 Presentation of the case study

This thesis will apply model predictive energy management to a relatively small network of interconnected microgrids. In general and according to the so called curse of dimensionality, the complexity of solving an optimisation problem increases greatly with the size of a network [5]. In particular, adding extra dimensions to a mathematical problem may lead to great increase in the computing time.

The problem considered is an energy management problem with safety constraints on the different units. The aim is to design a control law for all the unit in an electrical network in order to satisfy the demand with optimal cost of generation and dispatch. The renewable energy infeed and the load profile are given by forecast models. This is an inherent aspect of this type of study and will be decisive when using predictive control. The objective remains to maximise the use of renewable energies (wind or solar) by storing the energy in the battery when available in order to reuse it when necessary instead of conventional generation.

In this thesis, the optimisation problem is expressed strictly in linear and quadratic terms. Therefore there is no MIL dimension to it, contrary to [33, 34] for instance. Although this may be a strong assumption and not closely linked to the real behaviour, it is important for us to maintain convexity in the optimisation problem in order to solve the problem distributively later on.

In this chapter the model of a single connected microgrid will be introduced and the exemplary case of four interconnected microgrids will be presented allowing power transmission between the individual microgrids. Here, $k$ is the discrete time instant variable, expressed in hours. Since the third control layer of microgrids is the one with the slowest dynamic, a sampling period of 30 min is considered.
4.2 Single microgrid model

4.2.1 Model presentation

The microgrids are assumed to be controlled by local operators. The model for a single microgrid is composed of both power generating units, storage and loads. The control law aims at satisfying the demand by adapting the generated power. The microgrid model considered here consists of

- a **renewable energy sources (RES)**, e.g. a wind turbine or a solar unit,
- a **conventional generator** using fossil fuels,
- an **energy storage unit**, e.g. a battery,
- a **load**, and
- a **point of common coupling**.

This microgrid model is kept simple but still holds onto the specifications of microgrids, which are

- a combination of consumption and production sites, geographically close, and
- an electricity storage unit used to maximise the use of a renewable energy generation.

Each unit has bounds due to both safety and physical limits. They will be expressed in the optimisation problem formulated as constraints later in this chapter. Furthermore, the units are all interconnected. Figure 4.1 presents the model considered for a microgrid \( i \in \Omega \) where \( \Omega \) is the set of all microgrids in the network. All microgrids are assumed to have a similar structure. Thus this chapter will present the model for microgrid \( i \). It is equivalently applicable to all microgrids in the set \( \Omega \).
The variables \((u_G,i(k), u_{R,i}(k), u_{S,i}(k))\) denote the setpoints for the conventional generator, the RES and the storage unit respectively at discrete time instant \(k\). The vector \((p_G,i(k), p_{R,i}(k), p_{S,i}(k))\) is the power generated at time instant \(k\) by the conventional generator, the RES and the storage unit respectively. The transmitted power at time instant \(k\), \(p_{N,i}(k)\), can be deduced as the sum of local production and consumption, as
\[
p_{N,i}[k] = p_{G,i}[k] + p_{R,i}[k] + p_{S,i}[k] + w_{L,i}[k] \tag{4.1}
\]
where \(w_{L,i}[k]\) is the load power at time instant \(k\). In the case of an islanded microgrid, there is no power transmitted and therefore the local balance of power would be
\[
p_{G,i}[k] + p_{R,i}[k] + p_{S,i}[k] + w_{L,i}[k] = 0. \tag{4.2}
\]

### 4.2.1.1 State of charge of the electrical storage system

The state of the considered system is denoted by \(x_i[k]\) and represents the state of charge of the local ESS. Its evolution can be described by
\[
x[k+1] = x[k] - T_S p_{S,i}[k] \tag{4.3}
\]
where \(T_S\) is the sampling time and \(p_{S,i}[k]\) the power fed into or taken out of the ESS at time instant \(k\). As a convention, it is decided that a positive value for \(p_S\) means that the ESS acts as a generator and, therefore, the stored energy decreases. On the contrary, a negative value for \(p_{S,i}\) means that the storage unit is charged and consumes power from the grid.

### 4.2.1.2 Forecasts: load demand and renewable energy in-feed

The forecast model considered is a naive one: the predictions for the demand and the renewable energy infeed on the prediction horizon are equal to the last measured values. This may be the simplest forecast model and as such may not lead to the most efficient behaviour. The MPC is implemented using a certainty equivalent approach, i.e., the controller assumes the forecasts to be certain. This method is not robust and can lead to infeasibility. If the error on a forecast is relatively high, the control law may lead to unsafe behaviour.

Moreover, a delay of one sampling time interval is introduced between the measurement of the load and renewable energy values and the calculation of the control law. At time \(k\), the value used for the forecasts \((\tilde{w}_{R,i}[k], \tilde{w}_{R,i}[k])\) is derived from the last measured value \((w_{R,i}[k-1], w_{R,i}[k-1])\) using the naive model. Because it is a pessimistic approach, the robust behaviour of the microgrid is ensured with soft constraints on the ESS. An illustration of naive forecast with time delay can be seen in Figure 4.2.1.2. In blue is plotted in full line the actual evolution of \(w\) from time \(k-1\) to \(k\), and in dashed line is represented the actual figure evolution of \(\tilde{w}\). In red is plotted the forecast \(\tilde{w}\) used over the prediction horizon. In this example, an error on the forecast is present at time \(k\), and increases over the prediction horizon.
The mathematical formulation of the certainty equivalent naive forecasts with time delay is described by (4.4) where \((w_{R,i}, w_{L,i}) \in \mathbb{R}^2\) is the vector of measurement of the renewable energy available and of the load power, and \((\tilde{w}_{R,i}, \tilde{w}_{L,i}) \in \mathbb{R}^{T \times 2}\) is their forecast values on the prediction horizon. At time \(k\), it has the form

\[
\begin{align*}
\hat{w}_{R,i}[t] &= w_{R,i}[k-1], \\
\hat{w}_{L,i}[t] &= w_{L,i}[k-1],
\end{align*}
\]

\forall t \in [k; k+T]. \tag{4.4}

### 4.2.1.3 Disturbances on the System

Because there are errors on the forecast values, the control sequence for the microgrids units \((u_{G,i}[k], u_{R,i}[k], u_{S,i}[k])\) may not be equal to the actual power \((p_{G,i}[k], p_{R,i}[k], p_{S,i}[k])\) at time \(k\). The units are impacted in different ways as described in (4.5). The renewable generated power cannot be bigger than the actual available power. The storage unit acts as a slack bus and ensures the local balance of power by compensating for all disturbances.

\[
\begin{align*}
p_{G,i}[k] &= u_{G,i}[k] \\
p_{R,i}[k] &= \min(u_{R,i}(k), w_{R,i}[k]) \\
p_{S,i}[k] &= u_{S,i}[k]
\end{align*}
\]

\tag{4.5}

### 4.2.2 Optimisation problem formulation

There are two main parts in the optimisation problem formulation. The constraints describe limits for the variables. They can take the form of inequality constraints that express an upper or lower bound on a variable, or they can be equality constraints on multiple variables. The objective function, also called cost function or utility function, is a weighted non-linear convex function expressing the cost of usage or production of the different
units. The formulation of the cost function can be tuned so as to put more importance on specific objectives. Let us introduce the local optimisation problem formulation that aims at solving the energy management problem by maximising the use of the renewable energy, while respecting safety and physical constraints on both the decision variables \( u_i = (u_G, i[k], u_R, i[k], u_S, i[k], u_N, i[k])^T \), the power \( p_i = (p_G, i[k], p_R, i[k], p_S, i[k], p_N, i[k])^T \) and the system state \( x_i[k] \).

### 4.2.3 Constraints

In order to comply with the physical model and safety limits of the system, the optimisation problem restricts the variables with equality and inequality constraints. The constraints must be satisfied at all time instants, which means that when calculating the control sequence at time instant \( k \), the constraints on the predicted state and control inputs have to be satisfied for all time instants \( t \in [k; k + T] \).

#### 4.2.3.1 Load balancing constraint

One important equality constraint is ensuring the local balance of production and consumption, i.e., satisfying the load demand at all time. It is expressed by

\[
\begin{align*}
u_{G, i}[t] + u_{R, i}[t] + u_{S, i}[t] + u_{N, i}[t] + \tilde{w}_{L, i}[t] &= 0, \quad \forall t \in [k; k + T].
\end{align*}
\]  

As mentioned in (4.2.1.2) \( \tilde{w}_{L, i}[t] \) is a forecast. This means that the production/consumption balance described by (4.6) would not be respected in practice. In order to stabilise the system and ensure the local power balance, the storage unit acts as a slack bus: compensating for the gaps between predictions and reality. This is included in the optimisation problem formulation later on in 4.2.3.4.

#### 4.2.3.2 Limits on the generating units

All generating units, i.e. the conventional generator, the renewable energy source and the battery device have a bounded power output. The bounds can be derived from the physical limits, or by safety recommendations. In the optimisation problem, these limits take the form

\[
\begin{align*}
\frac{p_{G, i}}{p_{G, i}} &\leq \left( \frac{p_{G, i}[t]}{p_{G, i}} \right) \leq \left( \frac{p_{G, i}}{p_{G, i}} \right), \quad \forall t \in [k; k + T],
\end{align*}
\]  

where \( (p_{G, i}, p_{R, i}, p_{S, i}) \in \mathbb{R}^3 \) are the lower limits on the power output of the conventional generator and the RES and the storage unit respectively, and \( (\bar{p}_{G, i}, \bar{p}_{R, i}, \bar{p}_{S, i}) \in \mathbb{R}^3 \) are the higher limits.

It is important in this thesis to keep the problem linear and convex as this will allow to reach a globally optimal solution in a distributed fashion. Therefore, to avoid mixed-integer linear programming, the lower limits of the conventional generator are relaxed so that \( p_{G, i} = 0 \). In a real application, this value is usually \( p_{G, i} > 0 \). However, the storage unit
has either a positive or a negative power output, according to if it is feeding the system, or charging itself respectively.

4.2.3.3 Limits on renewable energy

The setpoint of the renewable energy source must always be inferior to the real power infeed

\[
\begin{align*}
    p_{R,i}[t] & \leq \bar{w}_{R,i}[t], & \forall t \in [k; k + T]. \\
    p_{R,i}[t] & \leq u_{R,i}[t], & \forall t \in [k; k + T].
\end{align*}
\] (4.8)

Once again, as the MPC relies on forecasts on the amount of renewable energy available, the value of \(\bar{w}_{R,i}[t]\) is only a prediction.

4.2.3.4 Limits on stored energy

The state of charge of the battery unit is also bounded by its lower limit \(E_{S,i} \in \mathbb{R}\) and higher limit \(\overline{E_{S,i}} \in \mathbb{R}\) as stated in the following hard constraint

\[
E_{S,i} \leq x_i[t] \leq \overline{E_{S,i}}, \quad \forall t \in [k, k + T].
\] (4.9)

Because there are uncertainties in both load and renewable infeed forecasts, the storage unit is chosen to behave as the slack bus, compensating the prediction errors. In order to incorporate this unpredictable behaviour in the system constraints, the storage bounds are tightened and expressed as a soft constraint, as in [14]. The worst case prediction error is assumed to be

\[
\begin{align*}
    \bar{e}_{p,i}^+ &= T_S \bar{P}_{S,i}, \text{ worst case positive forecast error} \\
    \bar{e}_{p,i}^- &= T_S \bar{P}_{S,i}, \text{ worst case negative forecast error}
\end{align*}
\] (4.10)

Soft bounds are then expressed as the bounds that would ensure safe operation of the system, even in the case of small constraint violation of the storage units. They are expressed as follow

\[
\begin{align*}
    E_{S,i}^{\text{soft}} &= E_{S,i} - \bar{e}_{p,i}^- \\
    \overline{E_{S,i}}^{\text{soft}} &= E_{S,i} - \bar{e}_{p,i}^+
\end{align*}
\] (4.11)

To include the soft constraints in the optimisation problem, a slack variable representing the cost of overpassing the soft limits \(z_{S,i}[t]\) is introduced in both the cost function and a constraint equation. The constraint equation on the storage slack variable is

\[
E_{S,i}^{\text{soft}} - z_{S,i}[t] \leq x_i(t) \leq \overline{E_{S,i}}^{\text{soft}} + z_{S,i}[t], \quad \forall t \in [k; k + T].
\] (4.12)

The penalty added in the main optimisation objective function could be a linear or quadratic cost, e.g.

\[
J_{S,i}[t] = c_{S,i}(z_{S,i}[t])^2, \quad \forall t \in [k; k + T]
\] (4.13)

where \(c_{S,i} \in \mathbb{R}\) is an arbitrary weight. The soft constraint is designed this way so that there is no cost when the state of charge is within the bounds, and with a quadratically
increasing cost when it is outside of the bound. It should be noted that soft constraints allow some constraint violation which means that the optimal solution might be outside of the soft bounds. The amount of constraint violations can be tuned with the weight $c_{S,i}$ and is as such problem dependant.

Both hard and soft constraints can be applied to the same variable. Figure 4.3 shows an example of both hard and soft constraint calculation with the parameters $(E_{S,i}, E_{S, i}^{\text{soft}}) = (0, 6), (E_{S,i}^{\text{soft}}, E_{S,i}^{\text{soft}}) = (0.5, 5.5)$ and $c_{S,i} = 5$.

![Figure 4.3. Cost associated with the stored energy state](image)

### 4.2.4 Cost function

The objective function of an optimisation problem is designed to achieve a desired behaviour of a system. For the energy management of microgrids, this can be for instance

- to flatten the supply curve of the conventional generator, as peaks in power generation are often costly, see e.g. [5], or
- to model the economical cost of running a grid.

In both cases, the objective function is designed by assigning weighted costs to different control variables. These costs can be linear or quadratic for instance, and their design should conserve the convexity of the optimisation problem.

#### 4.2.4.1 Cost of producing energy

In order to maximise the use of the RES generators and minimise the use of the conventional generator, different weights are defined. The weighted costs are chosen linearly and summed to form the usage cost $J_{U,i}$ as

$$J_{U,i}(t) = \beta_{G,i} u_{G,i}(t) + \beta_{R,i} u_{R,i}(t) + \beta_{S,i} u_{S,i}(t)$$

(4.14)
(\beta_G, i, \beta_R, i, \beta_S, i) \in \mathbb{R}^3$ are real weights. Choosing a positive weight for $\beta_G, i$ and a negative one for $\beta_R, i$ for instance would induce the input solution to use more the RES unit than the conventional generator.

### 4.2.4.2 Cost function for local problem on prediction horizon

As mentioned in Section 2.3.0.1, a discount factor $\gamma$ is added in the local cost function to give more importance to earlier predictions. The local objective function $f_i$ of microgrid $i$, when considering both the cost associated to energy production, and the cost associated to soft constraint of the stored energy, both expressed in a model predictive fashion is

\[
\begin{align*}
\text{minimise} & \quad \sum_{t=k}^{k+T} \gamma^t J_i(x_i[t], u_i[t]) \\
\text{subject to} & \quad x_i[t+1] = x_i[t] - Tsp_S, i[t], \forall t \in [k; k+T] \\
& \quad x_i \in X_i \\
& \quad u_i \in U_i
\end{align*}
\]

where $X_i$ and $U_i$ are the acceptable sets for the state $x_i$ and the input $u_i$ respectively, and where

\[
J_i(x_i[t], u_i[t]) = J_{U,i}(u_i[t]) + J_{S,i}(x_i[t]) = (\beta_{G,i} u_{G,i}[t] + \beta_{R,i} u_{R,i}[t] + \beta_{S,i} u_{S,i}[t]) + c_{S,i}(z_{S,i}[t])^2.
\]

As such, $J_i : (\mathbb{R} \times \mathbb{R}^3) \rightarrow \mathbb{R}$ and $(\beta_{G,i}, \beta_{R,i}, \beta_{S,i}, c_{S,i})$ are tunable weights. It is clear that a compromise has to be taken whether to give more importance to the soft constraint (4.13) respect with a relatively high $c_{S,i}$, or focus on the economical usage cost $J_{U,i}$ (4.14).

### 4.2.5 Convexity

The convexity of an optimisation problem depends on the convexity of the objective function and of the acceptable set of inputs. A common way of proving convexity is to show that the problem is composed of subfunctions that preserve convexity. One can for instance refer to [3] for an exhaustive description of operations that preserve convexity. Since the cost function is only composed of linear and quadratic terms, it is directly deductible that the objective function $f$ is convex. Although the weighted cost for the RES power $\beta_{R,i}$ can be negative, it does not impede convexity as $u_{R,i}[t]$ is bounded. Moreover, as all the inputs are bounded, and no mixed integer programming is introduced in the problem, the acceptable set, and therefore the optimisation problem, are convex.

### 4.3 A small grid of interconnected microgrids

The second part of the case study presentation introduces a combination of four microgrids as modelled in Section 4.2 and represented in Figure 4.1. There are several lines that connect them. In practice, transmitting power over a line results in resistive power losses.
4.3. A SMALL GRID OF INTERCONNECTED MICROGRIDS

One of the objectives of the optimisation is to obtain a trade off between the power transmission between microgrids as it results in losses, and the use of conventional generation. The model of the network considered in the case study is presented on Figure 4.4. In this section, a transmission line connecting two microgrids can be equivalently described as a line or an edge.

![Diagram of interconnected microgrids](image)

**Figure 4.4.** Four interconnected microgrids

### 4.3.1 Connecting microgrids and power flow

Connecting microgrids allows individual units to cooperate with each other by transmitting power through the transmission lines. In order to reach a global optimum, the operation strategy of microgrids may aim at transmitting from MGs that have a high share of renewable in-feed to MGs that would otherwise use their local conventional generator.

### 4.3.2 Distributed optimisation problem formulation

The network considered here is composed of four identical interconnected microgrids. Therefore, the optimisation problem can be decomposed as

- four local optimisation problem of a single microgrid as described in Section 4.2, and
- a global coupling on the transmitted power that appears both in the problem constraints and in the objective function.
4.3.3 Node power and edge power flow

There are two optimisation variables due to interconnecting microgrids:

- \( P_{N,i} \) is the microgrid infeed at the point of common coupling at node \( i \). By convention, a negative net transmission power is seen as a load, which means that node \( i \) feeds power into the grid. A positive net transmission power is seen as a generator, which means that node \( i \) receives power from the grid.

- \( P_{e,i\rightarrow j} \) is the edge power flow, the power flowing from node \( i \) to node \( j \)

The net transmitted power \( P_{N,i} \) at node \( i \) is the sum of all the edge power flow \( P_{e,i\rightarrow j} \) that node \( i \) sends to its neighbours denoted by the index \( j \) in the set of neighbours \( \Omega_i \), i.e.

\[
\sum_{j \in \Omega_i} P_{e,j\rightarrow i}[t] = P_{N,i}[t], \quad \forall t \in [k; k+T].
\] (4.17)

**Bounds on transmitted power:** In the case of interconnected microgrids, we introduce limits on the transmitted power over a line, but not on the net transmitted power at a node. This is an arbitrary choice as the network considered is relatively small, but it could be adapted according to the network structure. For an edge \( e \) in the network, it is of the form

\[
P_{e,e}[t] \leq P_{e,e}[t] \leq P_{e,e}[t], \quad \forall t \in [k; k+T].
\] (4.18)

This is to model the possible saturation of a line and can represent either a physical limit or a safety bound. For instance, high voltage transmission lines can usually transmit more power than lines used in distribution networks.

4.3.3.1 New coupling constraint

To ensure the global balance of the network with power transmission on the network, the sum of all transmitted power is assumed to be zero. This means that all the power that is sent by microgrids is also received somewhere (lossless lines). It is expressed as

\[
\sum_{i=1}^{I} p_{N,i}[t] = 0, \quad \forall t \in [k; k+T]
\] (4.19)

where \( I \) is the number of agents in the network and \( p_{N,i}(t) \) is the net transmitted power at node \( i \) at time instant \( t \).

However, it is not possible to express this equality constraint in a distributed optimisation problem because an agent has only knowledge of its local variables, and some of its neighbours data. Therefore, it is only possible to formulate a local version of this balance constraint for a microgrid \( i \) in the network

\[
u_{G,i}[t] + u_{R,i}[t] + u_{S,i}[t] + u_{N,i}[t] + \tilde{w}_{L,i}[t] = 0, \quad \forall t \in [k; k+T].
\] (4.20)
4.3. A SMALL GRID OF INTERCONNECTED MICROGRIDS

4.3.3.2 Cost for transmitting power

**Power loss:** It is assumed that no power loss occurs in the transmission lines, and as it appears clearly in (4.19), no power is dissipated. However, this is an assumption that can be quite far from reality. In literature, and although it varies according to the type of the line, one can find the value of resistive power losses to be usually less than 5% of the transmitted active power [39].

In order to account for power losses, we instead add a quadratic term to the objective function. Thus, the power losses are not taken into account directly but are emulated by penalising the overall power transmission. This also emphasises the local use of power. For a line $e$ in the network, it is express as

$$J_{T,e}[t] = \zeta(p_{\epsilon,e}[t])^2, \quad \forall t \in [k; k + T].$$ (4.21)

where the value of $\zeta \in [0, 1]$. We choose $\zeta$ close to 0 in order to reflect the real magnitude of power losses, and to allow for relatively high power flows in the network. It should be noted that the cost arbitrarily assigned to transmitting power, see (4.21), is set lower than the power of producing the same amount of energy locally, see (4.14), and therefore enforcing the agents to cooperate instead of acting individually.

4.3.3.3 Power flow and DC approximation

**Local voltage phase angle:** In practice, it is possible to fully control the power flow on the network thanks to power electronics [9]. However, because most grids may not be equipped with these devices, we choose to derive the power flow from the DC power flow assumptions. DC power flow is a linearised model of an AC power flow that is based on three assumptions: the line resistances are negligible, the voltage profile is flat and the voltage phase angle differences between neighbouring nodes are small. These approximations are often used in literature to analyse the economical dispatch in electrical networks, and a more developed analysis can be found for instance in [45].

The choice of using DC power flow assumptions over fully controllable power flows adds complexity to the distributed cooperative behaviours. It is also worth mentioning that the DC power flow approximations are based on the assumption that no power is dissipated on the lines, which is coherent with our model of no resistive power loss.

To deduce the edge power flow with its neighbours, each unit has only one local controllable variable $\theta_i$ which is the local voltage phase angle at node $i$. In the following we show how it is possible for a local microgrid, MG $i$, to deduce its net transmitted power $p_{N,i}$ and the edge power flow with its neighbours $j \in \Omega_i$, $p_{e,i \rightarrow j}$.

To express the topology of the electrical network, we introduce $\Psi$ the connection matrix, $\Psi \in \{0, 1\}^{N \times N}$ where $N$ is the total number of agents. $\Psi(i, j) = 1$ if there is a connection line between agents $i$ and $j$, $\Psi(i, j) = 0$ otherwise.
CHAPTER 4. PROBLEM FORMULATION

The connexion matrix of the grid introduced in Figure 4.4 is then

\[ \Psi = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix}. \quad (4.22) \]

The edge-node incidence matrix \( F \in \{-1, 0, 1\}^{N \times E} \), where \( E \) is the number of line in the network, is derived by choosing an arbitrary direction for every edge \( e \in \{1, \ldots, E\} \) and defined element wise as

\[ F_{i,e} = \begin{cases} -1 & \text{if node } i \text{ is the sink of edge } e, \\ 1 & \text{if node } i \text{ is the source of edge } e, \\ 0 & \text{otherwise.} \end{cases} \quad (4.23) \]

The edge-node incidence matrix of the four microgrid network is then

\[ F = \begin{pmatrix} -1 & -1 & -1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 1 & 1 \end{pmatrix}. \quad (4.24) \]

The connexion matrix and the edge-node incidence matrix are linked by \( \Psi = FF^T \). \quad (4.25) \]

In a similar way, we can introduce \( Y \in \mathbb{R}^{N \times N} \) the admittance matrix, also called Laplacian matrix as

\[ Y = F \text{diag}(y) F^T. \quad (4.26) \]

where \( y = (y_1, \ldots, y_E) \in \mathbb{R}^E \) is vector of admittances on all edges. Finally, \( \theta = (\theta_1, \ldots, \theta_N) \in \mathbb{R}^N \) is defined as the vector of all phase angles, and the net transmitted power matrix and the edge power flow matrix are respectively given by

\[ \begin{align*} \{ & P_N = Y \theta \\ & P_e = \text{diag}(y) F^T \theta. \} \quad (4.27) \end{align*} \]

The edge power flow matrix is composed of all \( P_{e,ij} \) such that

\[ \forall (i, j), \quad P_{e,ij} = \text{sign}(j - i)P_{e,i\rightarrow j}. \quad (4.28) \]

4.3.3.4 Applying power flow calculation to the case study

In the end, there are two relations between the phase angle and the transmitted power over the electrical network, deduced from the DC power flow assumptions as

\[ P_N = \begin{pmatrix} y_{12} + y_{13} + y_{14} & -y_{12} & -y_{13} & -y_{14} \\ -y_{12} & y_{12} & 0 & 0 \\ -y_{13} & 0 & y_{13} + y_{14} & -y_{34} \\ -y_{14} & 0 & -y_{34} & y_{34} + y_{14} \end{pmatrix} \theta, \quad \text{and} \quad (4.29) \]
4.4 Global optimisation problem

Let us formulate the optimisation problem of the network of $I$ interconnected microgrids. The local optimisation cost functions described in Section 4.2.4.2 can now be extended to a global cost function $f$ by summing all local cost functions $f_i$ and introducing the transmission costs as

$$f = \sum_{i=1}^{I} \sum_{t=1}^{k+T} \gamma^{t-k} \left( J_{U,i}(u_i[t]) + J_{S,i}(x_i[t]) + J_{T,i}(\theta(t)) \right).$$

Finally, the mathematical formulation of the MPC optimisation problem at a simulation time step $k$ is given by

$$\text{minimise } \quad f = \sum_{i=1}^{I} \sum_{t=1}^{k+T} \gamma^{t-k} \left( J_{U,i}(u_i[t]) + J_{S,i}(x_i[t]) + J_{T,i}(\theta(t)) \right)$$

subject to

$$x_i[t+1] = x_i[t] - T_S p_{S,i}[t], \quad \forall t \in [k; k+T]$$

$$x_i \in \mathcal{X}_i$$

$$u_i \in \mathcal{U}_i$$

$$\theta \in \mathcal{T}$$

where $\mathcal{X}_i$, $\mathcal{U}_i$ and $\mathcal{T}$ are the acceptable sets of $x_i$, $u_i = (u_{G,i}[k], u_{R,i}[k], u_{S,i}[k], u_{N,i}[k])^T$ and $\theta$ respectively. The acceptable sets can be formed by reassembling all the equality and inequality constraints stated in this chapter. The constraint sets are then defined as

$$\mathcal{X}_i = \left\{ x_i \mid \forall t \in [k; k+T], \begin{array}{l} x_i[t+1] = x_i[t] - T_S p_{S,i}[t] \\ \frac{E_{S,i}}{E_{S,i} \text{soft}} \leq x_i[t] \leq \frac{E_{S,i} \text{soft}}{E_{S,i}} - z_{S,i}[t] \\ \frac{E_{S,i} \text{soft}}{E_{S,i}} - z_{S,i}[t] \leq x_i[t] \leq \frac{E_{S,i} \text{soft}}{E_{S,i}} + z_{S,i}[t] \end{array} \right\},$$

$$\mathcal{U}_i = \left\{ u_i \mid \begin{array}{l} x_i[t+1] = x_i[t] - T_S p_{S,i}[t] \\ \frac{E_{S,i}}{E_{S,i} \text{soft}} \leq x_i[t] \leq \frac{E_{S,i} \text{soft}}{E_{S,i}} - z_{S,i}[t] \\ \frac{E_{S,i} \text{soft}}{E_{S,i}} - z_{S,i}[t] \leq x_i[t] \leq \frac{E_{S,i} \text{soft}}{E_{S,i}} + z_{S,i}[t] \end{array} \right\},$$

$$\mathcal{T} = \left\{ \theta \mid \begin{array}{l} x_i[t+1] = x_i[t] - T_S p_{S,i}[t] \\ \frac{E_{S,i}}{E_{S,i} \text{soft}} \leq x_i[t] \leq \frac{E_{S,i} \text{soft}}{E_{S,i}} - z_{S,i}[t] \\ \frac{E_{S,i} \text{soft}}{E_{S,i}} - z_{S,i}[t] \leq x_i[t] \leq \frac{E_{S,i} \text{soft}}{E_{S,i}} + z_{S,i}[t] \end{array} \right\}.$$
U_i = \begin{cases} 
\begin{align*}
& u_{G,i}[t] + u_{R,i}[t] + u_{S,i}[t] + u_{N,i}[t] + \tilde{w}_{L,i}[t] = 0 \\
& p_{G,i} \leq u_{G,i}[t] \leq \bar{p}_{G,i} \\
& p_{R,i} \leq u_{R,i}[t] \leq \bar{p}_{R,i} \\
& p_{S,i} \leq u_{S,i}[t] \leq \bar{p}_{S,i} \\
& u_{R,i}[t] \leq \tilde{w}_{R,i}[t] 
\end{align*}
\end{cases} 
\right. 
(4.35a)
(4.35b)
(4.35c)
(4.35d)
(4.35e)

\mathbb{T} = \left\{ \theta \mid \forall t \in [k; k + T], \quad P_k = \text{diag}(\gamma) F^T \theta \right\} 
(4.36a)
(4.36b)

By regrouping the cost function of individual microgrids in 4.33 and describing the acceptable sets over the decision variables, the central optimisation problem with ideal knowledge is formulated.

### 4.4.1 Subsystem coupling

In this case, all the generating units can be controlled through set-points whereas the load is not controllable. Without power transmission, the system would be decoupled, i.e. a dynamical system composed of distinct dynamical subsystems that can be independently actuated.

However, the global balance of power and consumption and the deduction of edge power flow from the local voltage phase angles introduce cross-coupling between individual microgrids. The subsystems actuators are decoupled but the microgrids have common objectives and constraints which make them interact with each other [41].

Because of this coupling, the global optimisation problem (4.33) is not fully-separable. In the next chapter, the ADMM method is applied in order to distribute the problem on the different subsystems.

### 4.5 Summary

This chapter presented the models of both a single microgrid and a network composed of four interconnected microgrids with similar structures. The optimisation problem was formulated as the objective function has been described, and all the constraints on the state and input sets have been stated.

The optimisation problem has been built to best reflect our expectations in term of control law. The main objective being to maximise the use of the available renewable energy infeed, the safe behaviour of the grid has also been included. The ability of individual microgrids to cooperate by exchanging power has also been integrated.

The main challenge of the following chapter is then to approach this problem in a distributed fashion. By solving optimisation problems and exchanging informations in an iterative way, the overall optimisation problem is solved in a distributed fashion.
Chapter 5

Design of distributed controllers

In this chapter, we will derive fully distributed neighbour-to-neighbour algorithms converging to the global optimum. The system considered is the four node grid presented in Section 4.3. In the following each microgrid is equivalently called a node or an agent. This chapter focuses on the notion of local variables, which describes intrinsic, private variables that can be either measured, controlled or deduced at a specific node. The opposite of local variables would be global variables, that require information on several agents of the system to be deduced.

First, we analyse the central controller calculation and then try to break it down into individual agents. It is important to clearly explain which data are known locally and which data are not available, requiring the individual agents to communicate and cooperate.

5.1 Central control analysis and global variables

The central controller can be derived by regrouping the constraints from the local optimisation problems from Section 4.2 and from the global constraints linked to power transmission, as in Section 4.3. Once the central problem is formulated, it is relatively direct to deduce the solution, as the central controller is assumed to be able to get knowledge from all the agents in the network.

The central controller outputs the solution for both the generated powers and the transmitted powers on the whole network. From an optimisation point of view, using the local voltage angles as a decision variable does not change much for the central controller. However, from an energy management point of view, the system has fewer degrees of freedom than with fully controllable power flows. In particular, the saturation of a line has strong consequences when handling non-fully controllable power flows.
### 5.1.1 Cost function analysis

The cost function of the central optimisation problem can be expressed as in (4.32). As the central controller has access to every variable, it is expressed as

\[
 f(u[k], x[k]) = \sum_{i=1}^{I} \sum_{t=k}^{k+T} \left( \beta_{G,i} u_{G,i}[t] + \beta_{R,i} u_{R,i}[t] + \beta_{S,i} u_{S,i}[t] + \xi \sum_{j \in \Omega_i} (u_{i,-j}[t])^2 + c_{S,i}(z_{S,i}[t])^2 \right) J_i(u_{i}[t], x_{i}[t]) .
\]

In this equation, it appears that all variables are private local variables, and as such directly controllable, except one: \( u_{N,i} \) which is the net transmitted power at the transmission node of MG \( i \). \( u_{N,i} \) is then a global variable, and it is why the central problem is not fully separable.

### 5.1.2 Local knowledge of microgrids

The controllable variable used to calculate power flow is the phase angle of a microgrid \( \theta_i \). The edge power flow and the net transmitted power flow \( u_{N,i} \) are deduced using (4.27). Therefore it appears that to calculate the variable \( u_{N,i} \), microgrid \( i \) needs to have knowledge of its own local phase angle \( \theta_i \) and of its neighbours phase angle \( \theta_j \) where \( j \in \Omega_i \), and \( \Omega_i \) is the neighbouring set of microgrid \( i \). As such, the net transmitted power \( u_{N,i} \) is not a locally controllable variable. Instead, microgrid \( i \) must reach a consensus with all its neighbours.

To model the fact that neighbouring microgrids may have a different knowledge or prediction for the same variable, the convention \( X_{i,j} \) is used on a certain shared variable \( X \) where \( i \) denotes the node that the variable refers to, and \( j \) indicates the node at which the value is derived.

For example, \( \theta_{j,i} \) describes the value of the voltage phase angle at node \( j \), as seen at node \( i \). If microgrid \( i \) has \( \omega_i = |\Omega_i| \) neighbours, then microgrid \( i \) locally has a total of \( \omega_i \) private variables \( \theta_{j,i} \) reflecting its own estimate of its neighbour angles.

### 5.1.3 Constraints on neighbouring knowledges

The central controller is assumed to have global knowledge of the network, which means that all variables can be communicated to all agents. Therefore inter subsystems coupling can be expressed as constraints in the central controller optimisation problem formulation

\[
 \forall j \in \Omega_i, \quad \begin{cases} \theta_{i,j} = \theta_{i,i} \\ \theta_{j,i} = \theta_{j,j} \end{cases} .
\]

### 5.2 Working toward a distributed solution

The previous section analysed the global variables in the central optimisation problem resulting in coupling between the subsystems. In this section, the goal is to distribute the
central problem on all the subsystems while still ensuring the respect of inter-systems constraints. Table 5.1 lists which variables are private and which ones are global variables subject to inter-system coupling.

<table>
<thead>
<tr>
<th>Fully controllable local variables</th>
<th>Local variables subject to inter-system coupling: global variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>$u_G,i, u_R,i, u_S,i, \theta_i,i, \theta_j,i$</td>
<td>$u_{N,i}, p_{e,i\rightarrow j}$</td>
</tr>
</tbody>
</table>

### 5.2.1 Relaxing constraints on neighbouring phase angles

In order to distribute the global problem to local microgrids, the constraint in (5.2) is relaxed and the difference between the local and neighbouring phase angle estimates are penalised in the local objective function. The local optimisation problem is then of the form

$$
\min \sum_{t=k}^{k+T} J_i(u_i[t], x_i[t])
$$

where $J_O(\theta_i,i[t], \theta_i,j[t], \theta_j,j[t])$ is a real function that add cost to the optimisation when (5.2) is not respected. The calculation of the relaxed equality constraints $J_O$ depends on the chosen method for distributed optimisation. In this thesis, two methods are proposed both based on an augmented Lagrangian relaxation.

### 5.2.2 Problem equivalent to a double consensus

The power flow equation over a line contains three free variables ($\theta_i,i, \theta_j,j, p_{e,i\rightarrow j}$), linked by

$$
p_{e,i\rightarrow j} = y_{ij}(\theta_j,j - \theta_i,i)
$$

In this thesis, it has been decided that the distributed consensus on transmitted power with local phase angle knowledge would be solved by having on each line a double consensus on both $\theta_i,i$ and $\theta_j,j$, see (5.2). Therefore, agents do not negotiate directly the power flowing between them. Instead they negotiate on their voltage phase angles, and locally deduce the edge power flow.

In term of ADMM optimisation, each agent will include two augmented Lagrangian penalty terms per neighbour in its cost function. For each neighbouring agent $j$, MG $i$ needs to agree on both its local phase angle $\theta_i$, and the desired phase angle for its neighbour $\theta_j$. In the end, the algorithm should have reached the equalities in (5.2) and as such ensure these hard constraints as in the central problem.
5.3 Sequential ADMM

The first solution algorithm proposed is an almost direct application of the ADMM method explained in [2]. The two consensus optimisation problem described in 5.2.2 are split into two successive minimisations. In this manner, a microgrid locally and sequentially updates:

1. its knowledge about its own phase angle $\theta_{i,i}$, and
2. its knowledge about the neighbouring phase angles $\theta_{j,i}$.

Intuitively, one can think that this method of splitting the optimisation problem in two may not be the fastest, as each iteration of the ADMM algorithm solves two optimisation problems. However, the point is to provide flexibility in the type of ADMM algorithm used. One may think of using faster converging algorithm based on ADMM, see e.g. fast ADMM method [13], proximal jacobian or primal dual [52]. This method may be closely linked to the ADMM algorithm in [2] as there is a single Gauss-Seidel pass.

To present the algorithm, we first want to decompose and class the variables in two types: the local optimisation variables of microgrid $i$, and the variables communicated to microgrid $i$ by its neighbours. They decision variables and parameters are regrouped in Table 5.2.

<table>
<thead>
<tr>
<th>Table 5.2. Decision variables and parameters used in Sequential ADMM for microgrid $i$</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Local variables at microgrid $i$</strong></td>
</tr>
<tr>
<td>$\theta_{i,i}$</td>
</tr>
<tr>
<td>$\theta_{j,i}$</td>
</tr>
<tr>
<td>$\lambda_{\theta_{i,i}}$</td>
</tr>
<tr>
<td>$\lambda_{\theta_{j,i}}$</td>
</tr>
<tr>
<td><strong>Communicated variables from neighbour $j$ where $j \in \Omega_i$</strong></td>
</tr>
<tr>
<td>$\theta_{i,j}$</td>
</tr>
<tr>
<td>$\theta_{j,j}$</td>
</tr>
<tr>
<td><strong>Parameters</strong></td>
</tr>
<tr>
<td>$\rho$</td>
</tr>
<tr>
<td>$\nu_{\text{max}}$</td>
</tr>
<tr>
<td>$\varepsilon_{\text{dual}}$</td>
</tr>
</tbody>
</table>
Algorithm 2: Distributed sequential ADMM

**Input:** $\bar{u}$ vector of load and wind forecast on the prediction horizon, $x[k]$ stored energy at time $k$, $\Omega_i$ the set of known neighbours

**Output:** $u$ optimal set-points for generators and net transmitted power, $\theta$ local angle knowledge on local MG and its neighbours

**Initialisation:** $\theta_{i,j} = 0$, $\theta_{j,i} = 0$, $\lambda_{\theta_i,i}^0 = 0$, $\lambda_{\theta_j,j}^0 = 0$;

**while** $v \leq v_{\text{max}}$ and $(s^v_{\text{loc}} > \varepsilon_{\text{dual}}$ or $s^v_{\text{neigh}} > \varepsilon_{\text{dual}})$ **do**

1. **store:** $\theta_{i,i}^v$, $\theta_{j,j}^v$, $\theta_{i,j}^v$ and $\theta_{j,i}^v$

2. **Phase 1:** update individual augmented Lagrangian on local angle
   In this step, $\theta_{i,i}$ is a fixed variable.
   
   $$ (u_{i}^{v+1}, \theta_{i,i}^{v+1}) = \arg\min_{u_i, \theta_{i,i}} \left\{ f_i(u_i, \theta_{i,i}, \theta_{j,j}^v) + \frac{\rho}{2} \sum_{j \in \Omega_i} \| \theta_{i,i} - \theta_{j,j}^v \|_2^2 + \sum_{j \in \Omega_i} \lambda_{\theta_{i,i}(i-j)}^v ((\theta_{i,i} - \theta_{j,j}^v)) \right\} $$
   (5.5)

3. **Phase 2:** Communication step of local angle knowledge $\theta_i$ only
   **for all neighbours $j \in \Omega_i$ do**
   
   - **Send** $\theta_{i,i}^{v+1}$ to neighbour $j$;
   - **Receive** $\theta_{j,j}^v$ from neighbour $j$;

4. **Phase 3:** update individual augmented Lagrangian on neighbour angle
   In this step, $\theta_{i,i}$ is a fixed variable.
   
   $$ (u_{i}^{v+1}, \theta_{i,i}^{v+1}) = \arg\min_{u_i, \theta_{i,i}} \left\{ f_i(u_i, \theta_{i,i}^{v+1}, \theta_{j,j}^v) + \frac{\rho}{2} \sum_{j \in \Omega_i} \| \theta_{i,i} - \theta_{j,j}^{v+1} \|_2^2 + \sum_{j \in \Omega_i} \lambda_{\theta_{i,i}(i-j)}^v ((\theta_{i,i} - \theta_{j,j}^{v+1})) \right\} $$
   (5.6)

5. **Phase 4:** Communication step of local angle knowledge $\theta_{j,j}$ only
   **for all neighbours $j \in \Omega_i$ do**
   
   - **Send** $\theta_{j,j}^{v+1}$ to neighbour $j$;
   - **Receive** $\theta_{j,j}^{v+1}$ from neighbour $j$;

6. **Phase 5:** Local update of $\lambda_{i,\theta_i}$ and $\lambda_{j,\theta_j}$
   **for all neighbours $j \in \Omega_i$ do**
   
   $$ \lambda_{\theta_{i,i}(i-j)}^{v+1} = \lambda_{\theta_{i,i}(i-j)}^v + \rho (\theta_{i,i}^v - \theta_{j,j}^v) $$
   (5.7)

   $$ \lambda_{\theta_{j,j}(i-j)}^{v+1} = \lambda_{\theta_{j,j}(i-j)}^v + \rho (\theta_{j,j}^v - \theta_{j,j}^v) $$

7. **Phase 6:** Calculate residuals
   
   $$ s_{\text{loc}}^{v+1} = \rho \sum_{j \in \Omega_i} \| \lambda_{\theta_{i,i}(i-j)}^{v+1} - \lambda_{\theta_{j,j}(i-j)}^v \|_2 $$
   (5.8)

   $$ s_{\text{neigh}}^{v+1} = \rho \sum_{j \in \Omega_i} \| \lambda_{\theta_{j,j}(i-j)}^{v+1} - \lambda_{\theta_{j,j}(i-j)}^v \|_2 $$

5.3. SEQUENTIAL ADMM


5.4 Simultaneous ADMM with substitution

The second algorithm is inspired from the method presented in [54]. In this work, the authors use a substitution method in the ADMM algorithm to solve the economic dispatch of power in a microgrid. As this paper uses fully controllable power flows, we adapt the substitution method to the optimisation problem with voltage phase angles.

The main idea for the substitution process is that for all ADMM iteration \( \nu \), and for all pairs of microgrids \((i, j) \in \Omega^2\), the following equality holds

\[
\lambda^\nu_{i\rightarrow j} + \lambda^\nu_{j\rightarrow i} = 0.
\] (5.9)

Based on that, one can eliminate the Lagrangian multiplier \( \lambda \) from the update equations, by introducing a new variable, in our case \( \hat{\theta} \) defined as

\[
\hat{\theta}^\nu_{i\rightarrow j} = \theta^\nu_i - \frac{\lambda^\nu_{i\rightarrow j}}{\rho}.
\] (5.10)

It is important to note that the step-size parameter \( \rho \) is now included in the update variable \( \hat{\theta} \). Therefore, if one wants to use a varying step-size as explained in section 3.3.2.2, it is needed to adapt the calculation of \( \hat{\theta} \) at each ADMM iteration. The parameters and the local and global decision variables used in the ADMM algorithm with substitution are presented in Table 5.3.

This second algorithm updates the two consensus problem simultaneously in the optimisation problem. In the next chapter will be investigated the compromise between solving the two consensus problem sequentially or simultaneously regarding to both

<table>
<thead>
<tr>
<th>Table 5.3. Decision variables and parameters used in ADMM with substitution for microgrid ( i )</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Local Variables at microgrid ( i )</strong></td>
</tr>
<tr>
<td>( \theta_{i,i} )</td>
</tr>
<tr>
<td>( \hat{\theta}_{i,i} )</td>
</tr>
<tr>
<td>( \hat{\theta}_{i,i\rightarrow j} )</td>
</tr>
<tr>
<td>( \hat{\theta}_{j,i\rightarrow j} )</td>
</tr>
<tr>
<td><strong>Communicated Variables from neighbour ( j ) where ( j \in \Omega_i )</strong></td>
</tr>
<tr>
<td>( \theta_{i,j} )</td>
</tr>
<tr>
<td>( \theta_{j,j} )</td>
</tr>
<tr>
<td><strong>Parameters</strong></td>
</tr>
<tr>
<td>( \rho )</td>
</tr>
<tr>
<td>( \nu_{\text{max}} )</td>
</tr>
<tr>
<td>( \epsilon_{\text{dual}} )</td>
</tr>
</tbody>
</table>
5.4. SIMULTANEOUS ADMM WITH SUBSTITUTION

the number of ADMM iterations needed for convergence and the real computing time associated, in seconds.

**Algorithm 3:** Distributed simultaneous ADMM with substitution

**Input:** \( \bar{u} \) vector of load and wind forecast on the prediction horizon, \( x[k] \) stored energy at time \( k \), \( \Omega_i \) the set of known neighbours

**Output:** \( u \) optimal set-points for generators and net transmitted power, \( \theta \) local angle knowledge on local MG and its neighbours

**Initialisation:** \( \theta_{i,j} = 0, \theta_{j,i} = 0, \hat{\theta}_{i,(i-j)} = 0, \hat{\theta}_{j,(i-j)} = 0 \);

**while** \( \nu \leq \nu_{\text{max}} \) and \((s^{\nu}_{\text{loc}} > \varepsilon_{\text{dual}} \text{ or } s^{\nu}_{\text{neigh}} > \varepsilon_{\text{dual}})\) **do**

**store:** \( \theta_{i,l}^{\nu}, \theta_{j,l}^{\nu}, \hat{\theta}_{i,(i-j)}^{\nu}, \hat{\theta}_{j,(i-j)}^{\nu} \) and \( \theta_{i,j}^{\nu} \)

**Phase 1:** update individual augmented Lagrangian

\[
(u_{i}^{\nu+1}, \theta_{i,j}^{\nu+1}, \theta_{j,i}^{\nu+1}) = \underset{u_i, \theta_{i,j}, \theta_{j,i}}{\text{argmin}} \left( J(l(u_i, \theta_{i,j}, \theta_{j,i})) + \sum_{j \in \Omega_i} \frac{p}{2} \| \theta_{i,j} - \phi_{i,(i-j)}^{\nu} \|^2 + \sum_{j \in \Omega_i} \frac{p}{2} \| \theta_{j,i} - \phi_{j,(i-j)}^{\nu} \|^2 \right)_{j \neq \theta_{i,j}, \theta_{j,i}, \theta_{j,i}^{\nu}}
\] (5.11)

**Phase 2:** Communication step

**for all neighbours** \( j \in \Omega_i \) **do**

[Send] \( \theta_{i,j}^{\nu+1} \) and \( \theta_{j,i}^{\nu+1} \) to neighbour \( j \);

[Receive] \( \theta_{j,j}^{\nu+1} \) and \( \theta_{i,j}^{\nu+1} \) from neighbour \( j \);

**Phase 3:** Local update of \( \hat{\theta}_{i} \) and \( \hat{\theta}_{j} \)

**for all neighbours** \( j \in \Omega_i \) **do**

\[
\hat{\theta}_{i,(i-j)}^{\nu+1} = \hat{\theta}_{i,(i-j)}^{\nu} + \frac{1}{2} (\theta_{i,j}^{\nu+1} - \theta_{i,j}^{\nu})
\] (5.12)

\[
\hat{\theta}_{j,(i-j)}^{\nu+1} = \hat{\theta}_{j,(i-j)}^{\nu} + \frac{1}{2} (\theta_{j,i}^{\nu+1} - \theta_{j,i}^{\nu})
\]

**Phase 4:** Calculate residuals

\[
s^{\nu+1}_{\text{loc}} = \rho \sum_{j \in \Omega_i} \| \hat{\theta}_{i,(i-j)}^{\nu+1} - \hat{\theta}_{i,(i-j)}^{\nu} \|_2
\] (5.13)

\[
s^{\nu+1}_{\text{neigh}} = \rho \sum_{j \in \Omega_i} \| \hat{\theta}_{j,(i-j)}^{\nu+1} - \hat{\theta}_{j,(i-j)}^{\nu} \|_2
\]
5.4.1 Comparison of the two distributed approaches

Comparing Tables 5.2 and 5.3 shows that both algorithms use and communicate the same number of variables. Moreover the dimensions of \( \hat{\theta}_{i,(i\rightarrow j)} \) and \( \lambda_{\theta_{i,(i\rightarrow j)}} \) are identical which mean that there is no difference in the number of communicated variables. The main difference is that Sequential ADMM requires two communication steps, but in the end both method communicate exactly the same amount of data per iteration.

Sequential ADMM exhibits one Gauss-Seidel pass as it updates variables \( \theta_{i,i} \) and \( \theta_{j,i} \) sequentially, using the last available set of data. In this manner, this algorithm is closer to the ADMM algorithm presented in [2]. One could even argue, that the simultaneous ADMM algorithm is closer to a decomposed method of multiplier, than to ADMM as there is no Gauss-Seidel pass in the update step of this method.

5.5 Summary

This chapter consisted in analysing the global optimisation problem described in the previous chapter and identify the inter-system coupling in order to allow for distributed control. The coupling constraints have been identified and two algorithms based on the alternating direction method of multipliers have been derived in order to control the four microgrid network in a distributed way.

The following chapter will illustrate these two algorithms on a case study. The three controllers (central, simultaneous ADMM with substitution and sequential ADMM) will be compared, both in terms of control designs and in terms of convergence speed. The tuning of the distributed controllers will also be investigated experimentally, as a lot of the parameters presented can vary depending on the problem.
Chapter 6

Results

6.1 Simulation environment

The simulations were run on a server with an Intel™ Xeon™ Processor E5-1620 (10M Cache, 3.60 GHz) using 32GB of RAM. The optimisation problem was formulated using Yalmip [20] and Matlab and is solved using Gurobi [15]. In order to emulate the decentralised behaviour of the agents, the parallel computing toolbox [23] was used in combination with Matlab. This toolbox allows parallel computation on a single processor with several cores or threads. The parallelisation can be tuned and coded within Matlab. Moreover, as this toolbox relies on functions to transmit variables between agents, it helps in developing a more practical code for implementation, and in having a first idea of the communication delays due to distributing computation. One of the objective of this thesis is also to show that using parallel computation allows for better calculation times.

6.2 Simulation parameters

The parameters used for tuning the MPC are summed up in Table 6.1. A sampling time of half an hour is chosen, which remains within the time scale of the third layer of control of microgrids, as stated in Section 1.2.2. The discount factor is chosen as $\gamma = 0.95$ which means that at the last time step, the relative importance is $0.95^5 \approx 77\%$ of the first time step.

The parameter for the local microgrid model are listed in Table 6.2. They are mostly arbitrary and chosen to give more importance in using the RES and the ESS. In this formulation, the usage cost of the ESS is simplified. It could be improved by adding ageing cost for instance.
CHAPTER 6. RESULTS

Table 6.1. Parameters of model predictive control

<table>
<thead>
<tr>
<th>Description</th>
<th>Variable name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sampling time [h]</td>
<td>$T_S$</td>
<td>0.5</td>
</tr>
<tr>
<td>Prediction horizon [sampled time-steps]</td>
<td>$T$</td>
<td>6</td>
</tr>
<tr>
<td>Discount factor</td>
<td>$\gamma$</td>
<td>0.95</td>
</tr>
</tbody>
</table>

Table 6.2. Parameters for microgrid models

<table>
<thead>
<tr>
<th>Description</th>
<th>Variable name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Bounds on power</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Conventional generator [pu]</td>
<td>$[P_G; \bar{P}_G]$</td>
<td>[0;1]</td>
</tr>
<tr>
<td>Storage unit [pu]</td>
<td>$[P_S; \bar{P}_S]$</td>
<td>[-1;1]</td>
</tr>
<tr>
<td>Renewable unit [pu]</td>
<td>$[P_R; \bar{P}_R]$</td>
<td>[0;1]</td>
</tr>
<tr>
<td><strong>Power Generation cost</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Conventional generator cost</td>
<td>$\beta_G$</td>
<td>1</td>
</tr>
<tr>
<td>Storage unit cost</td>
<td>$\beta_S$</td>
<td>0</td>
</tr>
<tr>
<td>Renewable unit cost</td>
<td>$\beta_R$</td>
<td>-1</td>
</tr>
<tr>
<td><strong>Stored energy</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Stored energy limits [pu.h]</td>
<td>$[E_S; \bar{E}_S]$</td>
<td>[0;6]</td>
</tr>
<tr>
<td>Stored energy soft limits [pu.h]</td>
<td>$[E_S^{soft}; \bar{E}_S^{soft}]$</td>
<td>[0.5;5.5]</td>
</tr>
<tr>
<td>Initial stored energy [pu.h]</td>
<td>$x_0$</td>
<td>1</td>
</tr>
<tr>
<td>Weight for soft constraint on storage cost</td>
<td>$c_S$</td>
<td>50</td>
</tr>
</tbody>
</table>

The parameters linked to the global optimisation problem are listed in Table 6.3. The transmission cost weight is chosen relatively low in order to enforce the transmission of power between the microgrids.

Table 6.3. Parameters for transmission network

<table>
<thead>
<tr>
<th>Description</th>
<th>Variable name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Line admittances [pu]</td>
<td>$y_{ij}$</td>
<td>20</td>
</tr>
<tr>
<td>Transmission cost</td>
<td>$\zeta$</td>
<td>0.05</td>
</tr>
<tr>
<td>Bounds on edge power flow [pu]</td>
<td>$[P_e; \bar{P}_e]$</td>
<td>[-1;1]</td>
</tr>
</tbody>
</table>

In the following figures and when it is not mentioned otherwise, the numerical value for the distributed controllers parameter are chosen as listed in Table 6.4. The optimal value for these parameters have been found heuristically. An analysis of the influence of the step-size and the absolute tolerance on residual can be found later on in Section 6.5.1.1. The choice of the maximum number of iteration is arbitrary. Here it is chosen so that the
maximum ADMM computation time at one sampled instant remains between 1 and 2 minutes.

<table>
<thead>
<tr>
<th>Table 6.4. Parameters for ADMM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
</tr>
<tr>
<td>Simultaneous ADMM with substitution</td>
</tr>
<tr>
<td>Step-size</td>
</tr>
<tr>
<td>Absolute tolerance on residuals</td>
</tr>
<tr>
<td>Sequential ADMM</td>
</tr>
<tr>
<td>Step-size</td>
</tr>
<tr>
<td>Maximum iteration</td>
</tr>
</tbody>
</table>

### 6.3 Simulation over one week

Both distributed and central controllers are applied to the network presented in Figure 4.4 simulated over a period of one week with the parameters stated in Section 6.2. The distributed algorithms converge, thus the results for power generation and power transfers along the lines are identical for all controllers. Figure 6.1 shows all the power generated, consumed and transmitted by the microgrids in the network.

As a reminder, the transmission cost weight used here is \( \zeta = 0.05 \). Because of the design of the cost function and as the price for generating renewable power is negative, the agents of the network tend to generate as much renewable as possible and store it in the ESS.

Since the wind or solar in-feed vary according to the location, Figure 6.1 also shows how units tend to cooperate: for instance, microgrid 1 transfers power to microgrid 4 that has lower renewable power available.

Figure 6.2 presents the results for the first microgrid only. It can be seen that the use of the wind energy available is maximised through the simulations, which is precisely the aim of the control design. The figure also shows that the conventional generator produces comparatively very few power through the one week simulation. The few instants it is used are linked to a low renewable energy availability, not only at microgrid 1, but through all the agents.

At time \( t = 1.6 \text{d} \), however, the RES generator uses less than half the wind power available, which could be considered as a waste of energy. This is due to the fact that at this time in the simulation, all the ESS in the different microgrids of the network are fully charged.

One can also note the small irregularities between the RES output and the wind power available: they are mainly due to the forecast model as in Section 4.2.1.2. These uncertainties are corrected by the storage unit that acts as a slack bus: applying the positive or
negative power to compensate for the imbalance in the local consumption/production. At time \( t = 3.3 \), the storage power exceeds the bounds as the uncertainties were especially unfavourable.

The distributed algorithms are based on neighbour-to-neighbour communication only, which means that microgrid 2 can not communicate with microgrid 3 or 4 and, in the same vein, can only transmit power to microgrid 1. However, Figure 6.3 shows that microgrid 2, 3 and 4 can cooperate by exchanging power through the microgrid 1. At \( d = 0.3 \) it can be seen that microgrid 2 sends power to microgrid 1 which is redistributed to microgrids 3 and 4. At time \( d = 3.5 \), it is the opposite behaviour that takes place as microgrids 3 and 4 feed power into the grid whereas microgrids 1 and 2 consume power. It is the only period in the simulation in which microgrid 3 and 4 both feed power into the grid, thanks to a high in-feed of renewable energy. In this problem, the power flow is not fully controllable. Instead it creates a coupling appearing mainly within microgrids 1, 3 and 4 as they are all three connected to one another. In particular, one can see in Figure 6.3 how similar the powers \( p_{e,1\rightarrow3} \) and \( p_{e,1\rightarrow4} \) are. This can be explained by the inter-system coupling on the voltage phase angles. In order to have different powers over lines 2 and 3, the angles \( \theta_3 \) and \( \theta_4 \) would have to be different, which would lead to another power flow between nodes 3 and 4.
Figure 6.1. Power generation and transmission over a one week simulation
Figure 6.2. Simulation results for Microgrid 1 over a one week simulation
6.3. SIMULATION OVER ONE WEEK

Figure 6.3. Power transmission on the electric network over a simulation of one week.
CHAPTER 6. RESULTS

6.4 Comparison between central and distributed controllers

6.4.1 Convergence analysis

The convergence of the distributed algorithms solutions to the central optimal solution can be illustrated by the difference between their minimum cost functions. In Figure 6.4, this is represented by the absolute difference between the minimal costs of the central solution and a distributed algorithm over one day of simulation. This error will be referred to as the residual error. In this simulation, the absolute tolerance on the primal and dual residual is $\varepsilon = 1e - 4$ for both algorithms and is plotted on the second graph as a black dash-dotted line. The simulations were run with a maximum number of $\nu_{\text{max}} = 1000$ ADMM iterations per time-step. The number of iterations needed for convergence is also visible on Figure 6.4. The convergence of the distributed algorithms appears satisfying, as the residual error is less than $1e - 2$ for the sequential ADMM in the worst case, and it is especially low through the simulation for the ADMM with substitution. One can remark that the residual error of the central and distributed controllers is not a direct consequence of the absolute tolerance on the residuals $\varepsilon$. The sequential ADMM algorithm performs especially poorly as it does not guarantee the objective difference to be under the criterion of absolute tolerance on the residuals. Furthermore, this algorithm often requires more than the maximum number of iterations to converge. The ADMM algorithm with substitution yields overall better performances. It has a relatively steady number of iterations needed for each time-step through the whole simulation.

Figure 6.4. Objective cost difference between central and distributed ADMM algorithms
6.4. COMPARISON BETWEEN CENTRAL AND DISTRIBUTED CONTROLLERS

Moreover, the residual error with the central solution remains under the absolute tolerance on the residuals.

### 6.4.2 Convexity analysis

![Graph showing local voltage phase angle evolution of central and distributed ADMM algorithms](image)

**Figure 6.5.** Local voltage phase angle evolution of central and distributed ADMM algorithms

If the cost for power transmission $\zeta$ is strictly positive, and if the power flows are fully controllable, then the optimisation problem is strictly convex. However, the optimisation problem is not strictly convex when controlling the local voltage angle of a microgrid. This means that different control inputs can lead to the same objective cost. This is especially visible in Figure 6.5 where the angles of the microgrids can be seen for all three controllers on the same one week simulation than in the previous section. Although the solution angles are different for all controllers, the result in power flow 6.3 are identical which illustrates the non-strict convexity of the problem. According to [2], a distributed
ADMM algorithm converges also in case of non strict convexity. It is however not the case for dual decomposition for instance. Fixing a reference angle at a particular node would lead the problem to be strictly convex. However, one can argue that using this representation without reference angle is a more general approach. In case of a line disconnection for instance, if the reference node is isolated, the other microgrids could still continue to negotiate on their respective phase angles and exchange power. Not having a reference node for the phase angle also ensures total equivalence of the microgrids.

6.5 Analysing distributed ADMM controllers

6.5.1 Tuning the ADMM-based controllers

6.5.1.1 Step-size

A first step of tuning the distributed algorithms is to find an optimal value for the step-size $\rho$ in terms of number of iterations for convergence. Figure 6.6 presents the variations of convergence speed with varying values for the step-size of both algorithms. The simulations are performed on the first time-step of the previous simulation of one week, see 6.1. The first algorithm (ADMM with substitution) has the typical shape of ADMM algorithms with one local minimum for the required number of iterations for convergence. Although the number of iterations required for convergence can vary greatly for this algorithm, the computing time is relatively unchanged. The second algorithm (sequential ADMM) has globally a longer computing time per iteration. The fact that this algorithm solves two optimisation problem per iteration could be a reason for that.

As stated in Section 3.3.2.2, the value of the penalty parameter $\rho$ is problem dependent. In our case, as the simulation ran on 336 sampled time instants, it is possible that the optimal value for $\rho$ deduced at the first iteration (Figure 6.6) might not be optimal for the rest of the simulation. This behaviour and the influence of the step-size parameter can be observed on Figure 6.7 that shows the evolution of required number of iterations for both algorithms. This figure was calculated on the 7 first sampled time instants and with an absolute tolerance of $\varepsilon = 1e^{-2}$ on the residuals. The sequential ADMM algorithm is especially subject to variations in terms of convergence speed and seem to be slower than ADMM with substitution in general. However, the influence of the step-size seem to have relatively less impact on sequential ADMM which gives it more flexibility. ADMM with substitution shows less variation in terms of iterations through the 7 sampled time instants. It could be more useful in practice as it allows to tune the algorithm on one iteration only, and then consider that it will be sufficient for all the other iterations of the simulation.

6.5.1.2 Absolute tolerance

Another important criterion for convergence is the absolute tolerance choice for both the primal and dual residual as presented in Section 3.3.2.1. Figure 6.8 presents different
convergence speed in terms of iterations with decreasing absolute tolerance for ADMM with substitution algorithm. There are two main remarks that one can make on choosing the parameters for ADMM.

- Varying the absolute tolerance from $\epsilon = 1e^{-2}$ to $\epsilon = 1e^{-8}$ increases the precision by a factor of $10^{-6}$ but only doubles the number of iterations required. This is a very good point for the algorithm as it seems that high precision can be attained without too much computational burden.

- As can be seen on Figure 6.8, the number of iterations before convergence remains relatively stable through the 48 sampled time instants. There are however some peaks where the algorithm requires an abnormally high amount of iterations. However, as these peaks are present for all value of $\epsilon$, one may conclude that reducing the precision does not lead to a high decrease in computing time.
CHAPTER 6. RESULTS

Figure 6.7. Number of iterations for convergence with varying step-size

Figure 6.8. Number of iterations for convergence with varying absolute tolerance
6.6 Computing time of distributed and central controllers

The quickest distributed algorithm presented in this thesis is the ADMM with substitution, tuned with a step-size of $\rho = 100$. In figure 6.8, the computation speed of this distributed algorithm with varying absolute tolerance to the central controller case is confronted.

![Graph showing computing time for central controller and ADMM with substitution with varying precision](image)

**Figure 6.9.** Computing time for central controller and ADMM with substitution with varying precision

In the end, the computing time of the best distributed controller is slower than the centralised controller by an order of magnitudes bigger than 10 to 100. The expected improvements in term of computing speed have not be obtained. These results are calculated with a relatively small network of $I = 4$ microgrids. Testing the distributed controllers on bigger network would give more informations on the benefits in terms of computing time. Moreover, the distributed controller already integrates some kind of communication procedures, which are included in the computing time, whereas the central controller is simulated and computed with an ideal omniscient structure (i.e. global knowledge of all the nodes state without communication burden).

In the end, using distributed computation does not seem to hinder the precision of the calculation. Even though the distributed methods are iterative and require more calculations, relatively high precision are attainable without excessive computing burden.

6.7 Evaluation of the Results

The simulation results provided in this chapter illustrated the convergence of the distributed controllers. The design of the optimal control law is attainable in a distributed
 CHAPTER 6. RESULTS

fashion with a good compromise in terms of precision and additional computation. Because the distributed algorithms rely on iterative methods, they did not yield a faster time of calculation. Yet, the network considered was relatively small with only four agents in the system. The advantages in term of computation speed are mostly expected to occur when largely increasing the size of the network.
Chapter 7
Discussion and Conclusions

In this last chapter, the main results will be discussed and the effectiveness of the distributed methods will be evaluated. The important contributions of this thesis will be summarised and a number of recommendations for future work will be provided.

7.1 Conclusions

Two distributed methods based on the alternating direction method of multipliers were proposed in this thesis. The benefits of using distributed control (scalability, privacy and adaptability, see Section 1.2.3.3) have been investigated. In particular, the adaptability of the proposed control architecture have been discussed. Two consensus on the local and neighbours voltage phase angles has been implemented, leading to a more general applicability of the control.

The alternating direction method of multipliers has been proved especially suitable in the case study, as the problem exhibited some non strict convexity aspects. Although they retain a lot of similarities, the controller structures proposed are relatively broad and let some place for tuning or parametrisation. Most of all, as the choice of parameters is particularly problem dependent, the goal of this thesis was not only to obtain satisfying numerical results, but to provide an understandable and reusable base for distributed control when applied to the energy management of microgrid. As the convergence of both algorithm has been attained through simulation, the customisability of the optimisation methods can be developed. The controller architectures could also be furthered by implementing faster converging methods.

7.2 Future Work

The work presented in this thesis can be extended in the following directions. The provided algorithms could be tested on more complex networks. The variation of the computing time with increasing network sizes could be analysed. The microgrid model considered was relatively simple, and more developed models have already been investigated in literature. In particular, and for an energy management problem, the impact of
disturbances, or the addition of line losses could be interesting directions. The simplifications made on the forecasts model or on the measurement delay could be extended to obtain better performances. The optimisation problem could be extended to include MIL properties. The analysis of such distributed MIL solutions could be for instance focused on how to reach the central optimal solution, or approach it with reasonable precision. The algorithms could be further tuned or it could be extended to be less problem dependent. An adaptive penalty parameter could be calculated at each time-step or ADMM iteration for instance. The distributed control structures could be enhanced with faster converging algorithms, such as e.g. the primal-dual method of multipliers [50, 51, 52], or fast ADMM [13]. Adding warm-start feature or implementing a proximal penalty term in the optimisation problem are other possible improvements.
Bibliography


