UNIVERSITÀ DI BOLOGNA

School of Engineering

Master Degree in Automation Engineering

Distributed Control Systems

Distributed Dual Gradient Tracking for Microgrid Control

Professor: **Giuseppe Notarstefano**

Students:
Alessandro Sitta
Leonardo Fiocchi

Academic year 2019/2020

Abstract

Energy efficiency and renewable energy are pushing towards rethinking the idea of grids. Not only conventional power supplies are present nowadays, but also other sources of energy e.g. solar panel, biomass generator and wind turbines. The notion of microgrid enters here, as a localized group of electricity sources and sinks that operates with macrogrid. The main advantageous point is that the microgrid can operate autonomously from the conventional power generators. In other words, it represents a distributed energy resources and loads that can operate in a coordinate way. Such challenging distributed optimization set-up consider each node of the network as an agent whose overall goal is to minimize the sum of the local cost functions. As in a real-world application, each function depends on a local variable that will be subjected to local and coupling constraints. Duality theory helps to handle the distributed algorithm in a simple and intuitive manner.
In the paper a network of N agents, arranged in generator, storage units, controllable loads and a trade node, that must solve a microgrid control problem is considered. The solution is carried out introducing the Dual Method and then applying the Distributed Gradient Tracking Algorithm on the cost-coupled problem.

Contents

Introduction

Motivations

Traditional fossil fuels are being seen with skepticism in recent years for their inevitable environmental pollution and scarcity for the future years. This has led to a shift towards renewable energy sources (RES) that are now being inserted in the general grid. The present situation suggests that the level of penetration of RES can only increase in future years and this creates new challenges for the network architecture and control of the grids. In fact, despite all the benefits, some issues arise such as the intermittent power generation they can provide. During the day the energy required by the users can be compensated by the RES most of the time, while during night solar power cannot be exploited anymore, for example [1]. Power system operations and planning are increasingly being subject of research and studies to integrate the uncertainty that RES introduce into the system.
The concept of microgrid integrated with the general power supply instead of a centralized grid is advantageous to this problem. A microgrid represents distributed energy resources and loads that can operate in a coordinate way. The system can be represented by N agents that cooperatively will solve an optimization problem, subjected to local and coupling constrains due to the single nodes power and the overall power balance of the microgrid. In order to reach as much energy efficiency as possible a distributed optimization is computed. In fact, this is a Model Predictive Approach, but it should be handled in a distributed manner.
The goal is to design an optimization-based feedback control law for a (spatially distributed) network of dynamical systems. The leading idea is the principle of receding horizon control, which informally speaking consists of solving at each time step an optimization problem (usually termed optimal control problem), in which the system model is used to predict the system trajectory over a fixed time window. After an optimal solution of the optimal control problem is found, the input associated to the current time instant is applied and the process is repeated.

Chapter 1

Problem description

The project deals with the design and implementation of a distributed optimization algorithm to solve an optimization problem arising in microgrid control. The project is divided in two tasks. The first requires implementing the Distributed Gradient Tracking algorithm, while in the second the practical example of a microgrid is described. The dual problem of the microgrid primal system should be solved using the solution found in the first task.
This project is a detailed and interesting practical problem of optimization amenable to distributed computation. It requires the minimization of the sum of local cost functions, each one depending on a local variable, subject to a local constraint for each variable and a coupling constraint involving all the decision variables. In this problem, the global optimal solution is obtained by stacking all the local variables. This feature leads easily to so-called big-data problems having a very highly dimensional decision variable that grows with the network size. Similar example setups include resource allocation, Communication, Signal Processing, Controls and Robotics or network flow optimization (e.g., in smart grid energy management).

Task 1

Suppose we have N agents that want to cooperatively solve the concave program

$$max \sum_{i=1}^{N} q_i(\lambda) \tag{1.1}$$

where $\lambda \in \mathbb{R}^S$, and $q_i : \mathbb{R}^S \to \mathbb{R}$ are strictly concave, quadratic functions, i.e.,

$$q_i(\lambda) = -(\lambda^T Q_i \lambda + r_i^T \lambda), i \in \{1, .., N\} \tag{1.2}$$

where $Q_i \in \mathbb{R}^{SxS}$ is a positive definite matrix. Design a software, written in MATLAB, implementing the Distributed Gradient Tracking algorithm presented, e.g., in [2]. Use the software to solve a random instance of problem (1.1).

Task 2

Consider a network of N agents, arranged in generator, storage units, controllable loads and a trade node, that must solve a microgrid control problem i.e.

$$\begin{aligned}
&\min_{p_1,..,p_N} \sum_{i=1}^{N} f_i(p_i) \\
&subj.to \sum_{i \in GEN} p_{gen,i}^{\tau} + \sum_{i \in STOR} p_{stor,i}^{\tau} + \sum_{i \in CONL} p_{conl,i}^{\tau} + p_{tr}^{\tau} - D^{\tau} = 0 \\
&\qquad p_i \in X_i, \\
&\qquad \forall i \in \{1, .., N\}, \\
&\qquad \forall \tau \in \{1, .., T+1\}
\end{aligned} \tag{1.3}$$

where cost functions f_i and local constraints X_i are summarized in Table 1. Derive the dual problem of (1.3), when dualizing only the coupling constraints $\sum_{i \in GEN} p_{gen,i} + \sum_{i \in STOR} p_{stor,i} + \sum_{i \in CONL} p_{conl,i} + p_{tr} - D = 0$. The dual problem should have the same structure of problem (1.1).
Notice that in the coupling constraint we have explicitly written the type of each p_i, e.g., $p_{gen,i}$,in order to better highlight the nature of each contribution.

Type	Cost function $f_i(p_i)$	Local constraints X_i
GEN	$\sum_{\tau=1}^{T+1}[\alpha_1 p_{gen,i}^{\tau} + \alpha_2 (p_{gen,i}^{\tau})^2]$	$\underline{p} \leqslant p_{gen,i}^{\tau} \leqslant \overline{p}, \tau \in [1, T+1]$ $\underline{r} \leqslant p_{gen,i}^{\tau+1} - p_{gen,i}^{\tau} \leqslant \overline{r}, \tau \in [1, T+1]$
STOR	$\sum_{\tau=1}^{T+1}[\epsilon (p_{gen,i}^{\tau})^2]$	-$d_{stor}^{\tau} \leqslant p_{stor,i}^{\tau} \leqslant c_{stor}^{\tau}, \tau \in [1, T+1]$ $q_{stor,i}^{\tau+1} = p_{stor,i}^{\tau} + p_{stor,i}^{\tau}, \tau \in [1, T]$ $0 \leqslant q_{stor,i}^{\tau} \leqslant q_{max}^{\tau}, \tau \in [1, T+1]$
CONL	$\sum_{\tau=1}^{T+1}[\beta \cdot max\{0, p_{des,i}^{\tau} - p_{conl,i}^{\tau}\}$ $+ \sum_{\tau=1}^{T+1}[\epsilon (p_{conl,i}^{\tau})^2]$	-P $\leqslant p_{conl,i}^{\tau} \leqslant P, \tau \in [1, T+1]$
{N}	$\sum_{\tau=1}^{T+1}[-c_1 p_{tr}^{\tau} + c_2 \lvert p_{tr}^{\tau} \rvert] + \sum_{\tau=1}^{T+1}[\epsilon (p_{tr}^{\tau})^2]$	-E $\leqslant p_{tr}^{\tau} \leqslant E, \tau \in [1, T+1]$

Table 1.1: Cost functions and local constraints for the microgrid control problem.

Finally, the necessary modifications to the software developed in Task 1 are made to solve an instance of the derived dual problem. The software should also return the optimal solution of the primal problem (1.3). The problem data must be generated randomly. Generate a set of simulations and show the results.

Chapter 2

Algorithm

From this chapter the Algorithms and set-ups used are presented and explained. First, the set-up created for the overall problem is presented, followed by the Centralized solution. Then the Dual Method is introduced in order to move to a cost coupled problem, which can be solved in a Distributed way using the Gradient Tracking Algorithm.

2.1 Set Up

The problem consists in minimizing the sum of different cost functions, respectively of Generators, Storage elements, Controllable Loads and Power Trading cost. Each of them is constrained by local constraints, and finally a coupling constraint is present. In this chapter the overall set up of the problem is explained.
The workflow of presentation is this: for each kind of node the relative cost function and constraints will be reported. After that the coupling constrained is introduced and its structure adapted for the MATLAB implementation. Finally, they will be grouped together in a structure necessary to find the optimal solution using MATLAB.
The decision variable for the minimization problem is going to be called p, which will be a vector composed of $(T+1)N$ components, where $[0, T+1]$ is the time horizon and N is the number of agents. At some point during the process p will be augmented to create z, in order to deal with some particular formulations of the cost functions.

2.1.1 Generators

The *cost function* for the generator i is:

$$f_i(p_i) = \sum_{\tau=1}^{T+1} [\alpha_1 p^{\tau}_{gen,i} + \alpha_2 (p^{\tau}_{gen,i})^2] \tag{2.1}$$

which will be rewritten in matrix form as:

$$f_i(p_i) = R_i^T \begin{bmatrix} p^1_{gen,i} \\ \vdots \\ p^{T+1}_{gen,i} \end{bmatrix} + \begin{bmatrix} p^1_{gen,i} & \dots & p^{T+1}_{gen,i} \end{bmatrix} Q_i \begin{bmatrix} p^1_{gen,i} \\ \vdots \\ p^{T+1}_{gen,i} \end{bmatrix} \tag{2.2}$$

where matrices R_i^T and Q_i are as follows:

$$R_i^T = \begin{bmatrix} \alpha_1^1 & \dots & \alpha_1^1 \end{bmatrix} \qquad Q_i = \begin{bmatrix} \alpha_2 & 0 & \dots \\ \vdots & \ddots & \\ 0 & & \alpha_2 \end{bmatrix} \tag{2.3}$$

The dimensions of vector R_i^T and matrix Q_i are according to the dimension of the decision variable (and therefore of the time horizon).
The *local constraints* for the generators are of two types: a simple upper-lower bound and an upper-lower bound on the difference between the power generated at time $\tau + 1$ and τ. Given that for the local constraint the form to which the aim is for is of the kind:

$$Ax \leqslant b$$

Then the first bounds are written in this way:

$$\begin{bmatrix} -I \end{bmatrix}_{T+1} \cdot \begin{bmatrix} p^1_{gen,i} \\ \vdots \\ p^{T+1}_{gen,i} \end{bmatrix} \leqslant \begin{bmatrix} -\underline{p} \\ \vdots \\ -\underline{p} \end{bmatrix}_{(T+1)\times 1} \tag{2.4}$$

$$\begin{bmatrix} I \end{bmatrix}_{T+1} \cdot \begin{bmatrix} p^1_{gen,i} \\ \vdots \\ p^{T+1}_{gen,i} \end{bmatrix} \leqslant \begin{bmatrix} \overline{p} \\ \vdots \\ \overline{p} \end{bmatrix}_{(T+1)\times 1} \tag{2.5}$$

While the second local constraint has been rewritten in this way:

$$\begin{bmatrix} -1 & 1 & 0 & 0 & \dots & 0 \\ 0 & -1 & 1 & 0 & \dots & 0 \\ \vdots & 0 & \ddots & \ddots & 0 & \vdots \\ 0 & \dots & 0 & -1 & 1 & 0 \\ 0 & \dots & 0 & 0 & -1 & 1 \end{bmatrix}_{T\times(T+1)} \cdot \begin{bmatrix} p^1_{gen,i} \\ \vdots \\ p^{T+1}_{gen,i} \end{bmatrix}_{(T+1)\times 1} \leqslant \begin{bmatrix} \overline{r} \\ \vdots \\ \overline{r} \end{bmatrix}_{T\times 1} \tag{2.6}$$

$$\begin{bmatrix} 1 & -1 & 0 & 0 & \dots & 0 \\ 0 & 1 & -1 & 0 & \dots & 0 \\ \vdots & 0 & \ddots & \ddots & 0 & \vdots \\ 0 & \dots & 0 & 1 & -1 & 0 \\ 0 & \dots & 0 & 0 & 1 & -1 \end{bmatrix}_{T\times(T+1)} \cdot \begin{bmatrix} p^1_{gen,i} \\ \vdots \\ p^{T+1}_{gen,i} \end{bmatrix}_{(T+1)\times 1} \leqslant \begin{bmatrix} -\underline{r} \\ \vdots \\ -\underline{r} \end{bmatrix}_{T\times 1} \tag{2.7}$$

In order to keep the simplex notation possible, the A-matrices of the local constraints in equations (2.4), (2.5), (2.6), (2.7) are named respectively, $A_{1,g}$, $A_{2,g}$, $A_{3,g}$, $A_{4,g}$. While the b-vector of the same local constraints are named, respectively, $b_{1,g}$,$b_{2,g}$,$b_{3,g}$,$b_{4,g}$. In this way the matrix for the first generator's constraints will become:

$$[A_{gen1}] = \begin{bmatrix} A_{1,g,1} \\ A_{2,g,1} \\ A_{3,g,1} \\ A_{4,g,1} \end{bmatrix}_{2(2T+1)\times(T+1)} \qquad [b_{gen1}] = \begin{bmatrix} b_{1,g,1} \\ b_{2,g,1} \\ b_{3,g,1} \\ b_{4,g,1} \end{bmatrix}_{2(2T+1)\times 1} \tag{2.8}$$

In this way all the local constraints for the generators can be written compactly in this form:

$$[A_{tot}]_{2G(2T+1)\times G(T+1)} \cdot [p]_{G(T+1)\times 1} \leqslant [b_{tot}]_{2G(2T+1)\times 1} \tag{2.9}$$

$$\begin{bmatrix} [A_{gen1}] & 0 & \dots & 0 \\ 0 & [A_{gen2}] & 0 & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & [A_{genG}] \end{bmatrix} \cdot \begin{bmatrix} [p_{gen,1}] \\ \vdots \\ [p_{gen,G}] \end{bmatrix} \leqslant \begin{bmatrix} [b_{gen1}] \\ [b_{gen2}] \\ \vdots \\ [b_{genG}] \end{bmatrix} \tag{2.10}$$

where $G \leqslant N$ is the number of Generators in the Network.

2.1.2 Storages

The *cost function* for the storage i is:

$$f_i(p_i) = \sum_{\tau=1}^{T+1} [\epsilon(p^{\tau}_{gen,i})^2] \tag{2.11}$$

which will be rewritten in matrix form as:

$$f_i(p_i) = \begin{bmatrix} p^1_{stor,i} & \dots & p^{T+1}_{stor,i} \end{bmatrix} \cdot Q_i \cdot \begin{bmatrix} p^1_{stor,i} \\ \vdots \\ p^{T+1}_{stor,i} \end{bmatrix} \tag{2.12}$$

where matrix Q_i is as follows:

$$Q_i = \begin{bmatrix} \epsilon & 0 & \dots \\ \vdots & \ddots & \\ 0 & & \epsilon \end{bmatrix}$$

The dimensions of Q_i are according to the dimension of the decision variable (and therefore of the time horizon).
The *local constraints* for the storage are more complex than the previous case. The first kind is a simple upper-lower bound, while the second one has an upper-lower bound through the variable q^{τ}_{stor}. Following the same reasoning done for the generators, the first result is found.
The first bounds are written in this way:

$$[I]_{T+1} \begin{bmatrix} p^1_{stor,i} \\ \vdots \\ p^{T+1}_{stor,i} \end{bmatrix} \leqslant \begin{bmatrix} -c_{stor} \\ \vdots \\ -c_{stor} \end{bmatrix} \tag{2.13}$$

$$[-I]_{T+1} \begin{bmatrix} p^1_{stor,i} \\ \vdots \\ p^{T+1}_{stor,i} \end{bmatrix} \leqslant \begin{bmatrix} d_{stor} \\ \vdots \\ d_{stor} \end{bmatrix} \tag{2.14}$$

While for the second local constraint, in order for the dimensions to be coherent, a procedure like this has been followed:

$$\begin{bmatrix} -1 & 1 & 0 & 0 & \dots & 0 \\ 0 & -1 & 1 & 0 & \dots & 0 \\ \vdots & 0 & \ddots & \ddots & 0 & \vdots \\ 0 & \dots & 0 & -1 & 1 & 0 \\ 0 & \dots & 0 & 0 & -1 & 1 \end{bmatrix}_{T\times(T+1)} \cdot \begin{bmatrix} q^1_{stor,i} \\ \vdots \\ q^{T+1}_{stor,i} \end{bmatrix} = [I]_{T\times(T+1)} \cdot \begin{bmatrix} p^1_{stor,i} \\ \vdots \\ p^{T+1}_{stor,i} \end{bmatrix}$$

The formulation of $p_{stor,i}$ with respect to $q_{stor,i}$ has been computed, exploiting the pseudo-inverse of the matrix with dimensions $T\times(T+1)$. Therefore, denoting as K the matrix that must be inverted we get the remaining two local constraints:

$$[K^+]_{(T+1)\times T} \cdot \begin{bmatrix} [I] & \vdots & 0 \end{bmatrix}_{T\times(T+1)} \cdot \begin{bmatrix} p^1_{stor,i} \\ \vdots \\ p^{T+1}_{stor,i} \end{bmatrix}_{(T+1)\times 1} \leqslant \begin{bmatrix} q_{max} \\ \vdots \\ q_{max} \end{bmatrix}_{(T+1)\times 1} \tag{2.15}$$

$$[-K^{+}]_{(T+1)\times T}\cdot\Big[[I]\ \vdots\ 0\Big]_{T\times(T+1)}\cdot\begin{bmatrix}p^{1}_{stor,i}\\ \vdots\\ p^{T+1}_{stor,i}\end{bmatrix}_{(T+1)\times 1}\leqslant\begin{bmatrix}0\\ \vdots\\ 0\end{bmatrix}_{(T+1)\times 1}\tag{2.16}$$

In order to keep the simplex notation possible, the A-matrices of the local constraints in equations (2.13), (2.14), (2.15), (2.16) are named respectively, $A_{1,s}$, $A_{2,s}$, $A_{3,s}$, $A_{4,s}$. While the b-vector of the same local constraints are named, respectively, $b_{1,s}$,$b_{2,s}$,$b_{3,s}$,$b_{4,s}$. In this way the matrix for the first storage's constraints will become:

$$[A_{stor1}]=\begin{bmatrix}A_{1,s,1}\\ A_{2,s,1}\\ A_{3,s,1}\\ A_{4,s,1}\end{bmatrix}_{4(T+1)\times(T+1)}\qquad [b_{stor1}]=\begin{bmatrix}b_{1,s,1}\\ b_{2,s,1}\\ b_{3,s,1}\\ b_{4,s,1}\end{bmatrix}_{4(T+1)\times 1}\tag{2.17}$$

In this way all the local constraints for the storage nodes can be written compactly in this way:

$$[A_{tot}]_{4S(T+1)\times S(T+1)}\cdot[p]_{S(T+1)\times 1}\leqslant[b_{tot}]_{4S(T+1)\times 1}\tag{2.18}$$

$$\begin{bmatrix}[A_{stor1}] & 0 & \dots & 0\\ 0 & [A_{stor2}] & 0 & \vdots\\ \vdots & \ddots & \ddots & 0\\ 0 & \dots & 0 & [A_{storS}]\end{bmatrix}\cdot\begin{bmatrix}[p_{stor,1}]\\ \vdots\\ [p_{stor,S}]\end{bmatrix}\leqslant\begin{bmatrix}[b_{stor1}]\\ [b_{stor2}]\\ \vdots\\ [b_{storS}]\end{bmatrix}\tag{2.19}$$

where $S\leqslant N$ is the number of Storage elements in the Network.

2.1.3 Controllable Loads

The *cost function* for the Controllable Load i is:

$$f_i(p_i)=\sum_{\tau=1}^{T+1}[\beta\cdot max\{0,p^{\tau}_{des,i}-p^{\tau}_{conl,i}\}+\sum_{\tau=1}^{T+1}[\epsilon(p^{\tau}_{conl,i})^2]\tag{2.20}$$

with local constraints of the type:

$$-P\leqslant p^{\tau}_{conl,i}\leqslant P$$

where the term

$$\min_{x}(\max(0,x-x_0))$$

is handled through the Epigraph Formulation, becoming:

$$\min_{x,y} y\tag{2.21}$$

$$\text{subj. to } y \geqslant \max(0, x - x_0)$$

Therefore, the final formulation for the Controllable loads problem is:

$$\begin{aligned}
&\min_{p_{conl,1},y} \sum_{\tau=1}^{T+1} [\beta \cdot y] + \sum_{\tau=1}^{T+1} [\epsilon (p^{\tau}_{conl,i})^2] \\
&\text{subj. to } y \geqslant -p^{\tau}_{conl,i} - p_{des,i} \\
&\qquad\quad y \geqslant 0 \\
&\qquad\quad p^{\tau}_{conl,i} \leqslant P \\
&\qquad\quad -p^{\tau}_{conl,i} \leqslant P
\end{aligned} \tag{2.22}$$

The matrix form will be:

$$\min_{p_{conl,1},y} \begin{bmatrix}\beta \dots \beta\end{bmatrix} \begin{bmatrix} y_1 \\ \vdots \\ y_{T+1} \end{bmatrix} + \begin{bmatrix} p^1_{conl,i} & \dots & p^{T+1}_{conl,i} \end{bmatrix} \begin{bmatrix} \epsilon & & \\ & \ddots & \\ & & \epsilon \end{bmatrix} \begin{bmatrix} p^1_{conl,i} \\ \vdots \\ p^{T+1}_{conl,i} \end{bmatrix}$$

To handle also this problem in the same way as the others, the decision variable p has been augmented with y, and in this way, finding z:

$$z = \begin{bmatrix} p^1_{conl,i} & \dots & p^{T+1}_{conl,i} & y_1 & \dots & y_{T+1} \end{bmatrix}^T \tag{2.23}$$

Finding the quadratic form:

$$\min_{z_i} z^T \cdot Q_c \cdot z + R_c \cdot z \tag{2.24}$$

where the matrices R_c and Q_c are as follows:

$$R_c = \begin{bmatrix} 0 & \dots & 0 & \beta & \dots & \beta \end{bmatrix} \qquad Q_c = \begin{bmatrix} \epsilon & & & & & \\ & \ddots & & & & \\ & & \epsilon & & & \\ & & & 0 & & \\ & & & & \ddots & \\ & & & & & 0 \end{bmatrix}$$

The local constraints are written accordingly. Note that $p_{des,i}$ has been modelled as a sine function with a random component superimposed.

$$-\begin{bmatrix} [I]_{T+1} & \vdots & [I]_{T+1} \end{bmatrix}_{(T+1)\times 2(T+1)} \cdot \begin{bmatrix} z \end{bmatrix}_{2(T+1)\times 1} \leqslant \begin{bmatrix} p^1_{des,i} \\ \vdots \\ p^{T+1}_{des,i} \end{bmatrix}_{(T+1)\times 1} \tag{2.25}$$

$$-\Big[[0]_{T+1} \ \vdots \ [I]_{T+1}\Big]_{(T+1)\times 2(T+1)} \cdot \big[z\big]_{2(T+1)\times 1} \leqslant \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix}_{(T+1)\times 1} \tag{2.26}$$

$$\Big[[I]_{T+1} \ \vdots \ [0]_{T+1}\Big]_{(T+1)\times 2(T+1)} \cdot \big[z\big]_{2(T+1)\times 1} \leqslant \begin{bmatrix} P \\ \vdots \\ P \end{bmatrix}_{(T+1)\times 1} \tag{2.27}$$

$$-\Big[[I]_{T+1} \ \vdots \ [0]_{T+1}\Big]_{(T+1)\times 2(T+1)} \cdot \big[z\big]_{2(T+1)\times 1} \leqslant \begin{bmatrix} P \\ \vdots \\ P \end{bmatrix}_{(T+1)\times 1} \tag{2.28}$$

In this way, following the same reasoning as the previous cases, the matrices $A_{conl,i}$ and $b_{conl,i}$ are created for the controllable load i (using (2.25), (2.26), (2.27), (2.28)), and subsequently the total matrices for the local constraint of all the controllable loads are created:

$$[A_{tot}]_{4C(T+1)\times 2C(T+1)} \cdot [z]_{2C(T+1)\times 1} \leqslant [b_{tot}]_{4C(T+1)\times 1} \tag{2.29}$$

$$\begin{bmatrix} [A_{conl1}] & 0 & \dots & 0 \\ 0 & [A_{conl2}] & 0 & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & [A_{conlC}] \end{bmatrix} \cdot \begin{bmatrix} [z_1] \\ \vdots \\ [z_C] \end{bmatrix} \leqslant \begin{bmatrix} [b_{conl1}] \\ [b_{conl2}] \\ \vdots \\ [b_{conlC}] \end{bmatrix} \tag{2.30}$$

with $C \leqslant N$ number of Controllable Loads in the Network.

2.1.4 Power Trading Cost

The device $i = N$ is the connection node with the main grid and its *power cost* is modelled as the function:

$$f_{tr} = \sum_{\tau=1}^{T+1} [-c_1 p_{tr}^{\tau} + c_2 |p_{tr}^{\tau}|] + \sum_{\tau=1}^{T+1} [\epsilon (p_{tr}^{\tau})^2] \tag{2.31}$$

with local constraints of the type:

$$-E \leqslant p_{tr}^{\tau} \leqslant E \qquad \forall i \in 1, \dots, N$$

which is equal to:

$$\min_{p_{tr}} -\begin{bmatrix} c_1 & \dots & c_1 \end{bmatrix} \begin{bmatrix} p_{tr}^1 \\ \vdots \\ p_{tr}^{T+1} \end{bmatrix} + \begin{bmatrix} p_{tr}^1 & \dots & p_{tr}^{T+1} \end{bmatrix} \begin{bmatrix} \epsilon & & \\ & \ddots & \\ & & \epsilon \end{bmatrix} \begin{bmatrix} p_{tr}^1 \\ \vdots \\ p_{tr}^{T+1} \end{bmatrix} + \begin{bmatrix} c_2 & \dots & c_2 \end{bmatrix} \begin{bmatrix} |p_{tr}^1| \\ \vdots \\ |p_{tr}^{T+1}| \end{bmatrix}$$

In general

$$\min_{p_{tr}} \begin{bmatrix} |p_{tr}^1| \\ \vdots \\ |p_{tr}^{T+1}| \end{bmatrix}$$

is equivalent to:

$$\min_{x,y} y \tag{2.32}$$

$$\text{subj. to} \begin{bmatrix} |p_{tr}^1| \\ \vdots \\ |p_{tr}^{T+1}| \end{bmatrix} \leqslant \begin{bmatrix} y^1 \\ \vdots \\ y^{T+1} \end{bmatrix}$$

As already shown, the decision variable p_{tr} has been augmented with y, and in this way finding z:

$$z = \begin{bmatrix} p_{tr}^1 & \dots & p_{tr}^{T+1} & y_1 & \dots & y_{T+1} \end{bmatrix}^T \tag{2.33}$$

Finding the quadratic form:

$$\min_z z^T \cdot Q_{tr} \cdot z + R_{tr} \cdot z \tag{2.34}$$

where the matrices R_{tr} and Q_{tr} are as follows:

$$R_{tr} = \begin{bmatrix} -c_1 & \dots & -c_1 & c_2 & \dots & c_2 \end{bmatrix} \qquad Q_{tr} = \begin{bmatrix} \epsilon & & & & & \\ & \ddots & & & & \\ & & \epsilon & & & \\ & & & 0 & & \\ & & & & \ddots & \\ & & & & & 0 \end{bmatrix}$$

For the local constraints some adjustments are necessary. In fact, for example in the first one, there is:

$$\begin{bmatrix} p_{tr}^1 \\ \vdots \\ p_{tr}^{T+1} \\ y^1 \\ \vdots \\ y^{T+1} \end{bmatrix}_{2(T+1)\times 1} - \begin{bmatrix} y^1 \\ \vdots \\ y^{T+1} \end{bmatrix}_{(T+1)\times 1} \leqslant \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix}_{(T+1)\times 1}$$

That is transformed using the augmented decision variable in this local constraint:

$$\Big[[I]_{T+1} \;\; \vdots \;\; [-I]_{T+1}\Big]_{(T+1)\times 2(T+1)} \cdot \begin{bmatrix} p_{tr}^1 \\ \vdots \\ p_{tr}^{T+1} \\ y^1 \\ \vdots \\ y^{T+1} \end{bmatrix}_{2(T+1)\times 1} \leqslant \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix}_{(T+1)\times 1} \tag{2.35}$$

On the same footprint, the second constraint is:

$$\Big[[-I]_{T+1} \;\; \vdots \;\; [-I]_{T+1}\Big]_{(T+1)\times 2(T+1)} \cdot \begin{bmatrix} p_{tr}^1 \\ \vdots \\ p_{tr}^{T+1} \\ y^1 \\ \vdots \\ y^{T+1} \end{bmatrix}_{2(T+1)\times 1} \leqslant \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix}_{(T+1)\times 1} \tag{2.36}$$

While for the upper and lower bounds E it will be:

$$\Big[[I]_{T+1} \;\; \vdots \;\; [0]_{T+1}\Big]_{(T+1)\times 2(T+1)} \cdot \begin{bmatrix} p_{tr}^1 \\ \vdots \\ p_{tr}^{T+1} \\ y^1 \\ \vdots \\ y^{T+1} \end{bmatrix}_{2(T+1)\times 1} \leqslant \begin{bmatrix} E \\ \vdots \\ E \end{bmatrix}_{(T+1)\times 1} \tag{2.37}$$

$$\begin{bmatrix}[-I]_{T+1} & \vdots & [0]_{T+1}\end{bmatrix}_{(T+1)\times 2(T+1)} \cdot \begin{bmatrix} p_{tr}^{1} \\ \vdots \\ p_{tr}^{T+1} \\ y^{1} \\ \vdots \\ y^{T+1} \end{bmatrix}_{2(T+1)\times 1} \leqslant \begin{bmatrix} E \\ \vdots \\ E \end{bmatrix}_{(T+1)\times 1} \tag{2.38}$$

Thus, the optimization problem (2.31) is subjected to conditions (2.35), (2.36), (2.37), (2.38). Given that the Trading cost is modelled as if only on the last node, then the final A_{tot} and b_{tot} matrices are block matrices created stacking one on top of the others the matrices at equations (2.35), (2.36), (2.37), (2.38).

2.1.5 Coupling Constraint

The microgrid problem presented is composed of an optimization problem over a sum of functions, subjected to the local constraints presented until now. Moreover, a *coupling constraint* that connects all the nodes is introduced.

$$\sum_{i\in GEN} p_{gen,i}^{\tau} + \sum_{i\in STOR} p_{stor,i}^{\tau} + \sum_{i\in CONL} p_{conl,i}^{\tau} + p_{tr}^{\tau} - D^{\tau} = 0 \tag{2.39}$$

$$\forall \tau \in \{1,..,T+1\}$$

In order to solve in a centralized way the optimal problem, the equality constraints are structured in order to be in the form:

$$[A_{eq}] \cdot [z] = [b_{eq}] \tag{2.40}$$

where A_{eq} is created by stacking side by side the $A_{eq,i}$ of each node. As explanatory case, the generators A_{eq} matrix is shown, since all the others are created similarly.

$$\begin{bmatrix} 1 & 0 & \dots & 0 & 0 & \dots & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 & \dots & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & 0 & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & 1 & 0 & 0 & \dots & 0 \end{bmatrix}_{T\times 2(T+1)} \tag{2.41}$$

Matrix (2.41) is copied side by side, as many times as the number of agents in the Network (in this case only generators). Finally, for the b_{eq} vector, it will simply be the vector of the D^{τ}, therefore of dimension $(T+1)\times 1$.

2.1.6 Centralized Optimal Solution

The Centralized Optimal Solution is found using the MATLAB command *quadprog*. All the matrices presented so far have been constructed in order to be fed to *quadprog*. Note that the decision variable has been augmented with y for all the components of the network, in order to take into account the Epigraph Formulation discussed above (see (2.21), (2.32)). For this reason the matrices that in previous discussion were not augmented are modified to take into account this factor.
The optimal solution returned by *quadprog* consists of the optimal value function and the optimal vector z. Therefore, the components of p are extracted from z to be further exploited for the analysis on the Distributed Section.

2.2 Dual problem framework

The presented problem is a *constraint-coupled* problem. It may be of interest to derive the dual problem of (1.3), dualizing only the coupling constraints. The first thing to consider is $f_i(p_i)$, recalling that the decision vector $p_i \in \mathbb{R}^{T+1}$. In the presented framework this function can be of 4 different kinds, which however are all quadratic functions and therefore convex, implying that our problem is convex. Then in order to have $g_i(p_i)$ convex we can consider the coupling constraints and write them in the linear form:

$$g_i(p_i) = H_i p_i + h_i$$

Therefore, the overall problem can be rewritten as:

$$\begin{aligned} &min \sum_{i=1}^{N} f_i(p_i) \\ &subj.to \sum_{i=1}^{N} g_i(p_i) = 0, \\ &p_i \in X_i, \end{aligned} \tag{2.42}$$

with $H_i \in \mathbb{R}^{(T+1)\times(T+1)}$, $h_i \in \mathbb{R}^{(T+1)}$, $\lambda \in \mathbb{R}^{(T+1)}$ with the $H_i p_i + h_i$ of the kind:

$$H_i p_i + h_i = \begin{bmatrix} [H_i p_i + h_i]_1 \\ \vdots \\ [H_i p_i + h_i]_{T+1} \end{bmatrix}$$

In particular, in the framework presented in (1.3) the resulting matrices will be:

$$H_i = I_{(T+1)\times(T+1)}, \forall i \in 1, \dots, N$$

$$h_i = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix}_{(T+1)\times 1} \qquad \forall i \in 1, \dots, N-1$$

$$h_N = \begin{bmatrix} -D^1 \\ \vdots \\ -D^{T+1} \end{bmatrix}$$

The next step is to write the Lagrangian:

$$L((p_1, \dots, p_N), \lambda) = \sum_{i=1}^{N} (f_i(p_i) + \lambda^T (H_i p_i + h_i)) \tag{2.43}$$

And from there the Dual Function is defined as follows:

$$\begin{aligned} q(\lambda) &= \inf_{p_1, \dots p_N} L((p_1, \dots, p_N), \lambda) \\ &= \inf_{p_1, \dots p_N} \sum_{i=1}^{N} (f_i(p_i) + \lambda^T (H_i p_i + h_i)) \\ &= \sum_{i=1}^{N} \inf_{p_i} (f_i(p_i) + \lambda^T (H_i p_i + h_i)) \end{aligned}$$

So by writing:

$$q_i(\lambda) = \inf_{p_i \in X_i} (f_i(p_i) + \lambda^T (H_i p_i + h_i)) \tag{2.44}$$

The Dual Problem results in:

$$\max_{\lambda \in \mathbb{R}^{T+1}} \sum_{i=1}^{N} q_i(\lambda) \tag{2.45}$$

In this way the original problem can be rewritten as a *cost-coupled* problem, with structure as (1.1). Moreover, since the starting problem has only equality constraints, the resulting dual problem is *unconstrained.* Instead, for what concerns the local constraints, they are also considered in this new framework, in particular, restricting the infimum of the i_{th} component of the Lagrangian to the *feasible set.* The dual problem can then be solved with centralized or decentralized methods as a cost-coupled problem. In any case the solution of the dual problem represents a lower bound for the primal problem, if moreover strong duality conditions hold its solution will be equivalent to the primal problem's solution.

2.3 Quadratic function Set Up

It has been shown that the cost functions for generators, storage, controllable loads and power trading costs can be all written as quadratic functions (see (2.2), (2.12), (2.24), (2.34)). Therefore, the resulting problem will be in the form of (1.2). For this problem the aim is to maximize the scalar objective function, where λ is the *optimization variable*. Some considerations can be made: at first this is an *unconstrained* optimization problem (e.g. $\lambda \in \mathbb{R}^{T+1}$), which means that an optimal solution can always be found; then, since the cost function is strictly convex, the optimal solution is *unique*. This comes from the property that: $-f$ is concave only if f is convex. Thus the original problem (1.1) is immediately cast as:

$$\min_{\lambda \in \mathbb{R}^{T+1}} \sum_{i=1}^{N} q_i(\lambda) \tag{2.46}$$

and $q_i : \mathbb{R}^{T+1} \to \mathbb{R}$ are strictly convex, quadratic functions, i.e.,

$$q_i(\lambda) = +(\lambda^T Q_i \lambda + r_i^T \lambda), \qquad i \in \{1, .., N\},$$

where $Q_i \in \mathbb{R}^{(T+1)\times(T+1)}$ is a positive definite matrix. The Q_i matrix takes different values for generators, storages, etc. The r_i vector depends not only on the cost function of the specific agent, but it encodes also the $\lambda^T(H_i p_i + h_i)$ contribution, therefore considering for example the i^{th} agent being a generator:

$$R_i = \begin{bmatrix} \alpha_1^1 \\ \vdots \\ \alpha_1^1 \end{bmatrix} + \begin{bmatrix} (\lambda^1)^T p_i^1 \\ \vdots \\ (\lambda^{T+1})^T p_i^{T+1} \end{bmatrix} \qquad Q_i = \begin{bmatrix} \alpha_2 & 0 & \dots \\ \vdots & \ddots & \\ 0 & & \alpha_2 \end{bmatrix} \tag{2.47}$$

Similarly each of the different component can be written in this way. Recalling that the i^{th} component of the dual cost is:

$$q_i(\lambda) = \inf_{p_i \in X_i} (f_i(p_i) + \lambda^T(H_i p_i + h_i)) \tag{2.48}$$

It is now of interest to solve this *cost-coupled* problem in a distributed way. In order to do so the Gradient Tracking Distributed Algorithm is presented.

2.4 Gradient Tracking Distributed Algorithm

In order to fully exploit the potential of the network, to reduce computational burden for a single *master* node and to preserve privacy, the Gradient Tracking Distributed Algorithm [3] can be used to solve the dual problem, which is just a cost-coupled problem.
The advantage of this method is that it exhibits a fast convergence rate because it allows for the use of a *constant* step-size. In order to have convergence we need some assumptions to be fulfilled. In particular, we should have each $q_i(\lambda)$ *sufficiently* regular, a network representable as a *fixed undirected graph* and with a weighted adjacency matrix A *doubly stochastic* [3].
Then, by denoting λ_i^t as solution estimate of a minimum λ^* at agent i, the algorithm for the *maximization* problem is:

$$\lambda_i^{t+1} = \sum_{j=1}^{N} a_{ij}\lambda_j^t + \gamma d_i^t \tag{2.49}$$

$$d_i^{t+1} = \sum_{j=1}^{N} a_{ij}y_j^t + (\nabla q_i(\lambda_i^{t+1}) - \nabla q_i(\lambda_i^t)) \tag{2.50}$$

where the consensus iteration $\lambda_i^{t+1} = \sum_{j=1}^{N} a_{ij}\lambda_j^t$ is meant to enforce an agreement among the agents, and the descent direction $-\sum_{h=1}^{N} \nabla q_h(\lambda_h^t)$ still requires a global knowledge that is not locally available. To overcome this issue, the exact (centralized) descent direction is replaced by a local one. This local estimate of the descent direction is called here d_i^t, and is updated through a dynamic *average consensus* iteration to eventually track the true gradient direction $\sum_{h=1}^{N} \nabla q_h(\lambda_h^t)$. The problem is in the computation of $\nabla q_i(\lambda_i^{t+1})$ as, in general, an explicit expression for $q_i(\lambda_i)$ is not available. The problem is solved by noticing that:

$$p_i^{t+1} = arg \min_{p_i \in X_i} (f_i(p_i) + (\lambda_i^{t+1})^T (H_i p_i + h_i)) \tag{2.51}$$

$$\nabla q_i(\lambda_i^{t+1}) = H_i p_i^{t+1} + h_i \tag{2.52}$$

In a similar way it is possible to derive $\nabla q_i(\lambda_i^{t+1})$ and to write the overall algorithm as:

$$\lambda_i^{t+1} = \sum_{j=1}^{N} a_{ij}\lambda_j^t + \gamma d_i^t \tag{2.53}$$

$$p_i^{t+1} = arg \min_{p_i \in X_i} (f_i(p_i) + (\lambda_i^{t+1})^T (H_i p_i + h_i)) \tag{2.54}$$

$$d_i^{t+1} = \sum_{j=1}^{N} a_{ij} y_j^t + H_i(p_i^{t+1} - p_i^t) \tag{2.55}$$

Notice that in the presented framework (1.2), the first assumption is respected as q_i is differentiable, i.e. $\exists \nabla q_i(\lambda)$ for each $\lambda \in \mathbb{R}^{T+1}$. Additional information on this assumption are given in [2]. Then, if the assumptions on the network are also fulfilled, the algorithm converges to the global minimum. Note that, assuming strong convexity, as the $p_i^{t+1} = arg \min_{p_i \in X_i}(f_i(p_i) + (\lambda_i^{t+1})^T (H_i p_i + h_i))$ results to be unique at each iteration, p_i^t will converge to the unique global minimum.

Chapter 3

Results

For the simulations, it is considered an heterogeneous network of N=10 units with 4 generators, 3 storage devices, 2 controllable loads and 1 connection to the main grid. We consider the network to have a corresponding G fixed and undirected and a weighted adjacency matrix A doubly stochastic. We assume that each unit predicts its power generation strategy over a horizon of T = 12 slots. In order to fit the microgrid control problem in our set-up, we let each p_i be the whole trajectory over the prediction horizon $[0, T]$, examples and details in (2.2), (2.12), (2.23), (2.33), for all the device types. As for the cost functions we consider $f_i(p_i) = \sum_{\tau=1}^{T+1} f_i^\tau(p_i^\tau)$. Each local constraint X_i encodes the heterogeneous dynamics and bounds on the state of the units. We set a constant step size $\gamma = 6e^{-3}$. We considered random values for the unit parameters and random sine functions for D and p_{des}.

In Figure 3.1 we show the primal cost error, $|\sum_{i=1}^{N} f_i(p_i) - f^*|$, and the dual cost error, $|\sum_{i=1}^{N}(f_i(p_i) + \lambda_i^T p_i) - f^*|$, where f^* is the optimal cost computed with a centralized algorithm. It can be seen that both cost errors converge to the optimal cost f^* in a fast way, since we are using a constant step size. This is expected, in fact agents do not only cooperate on looking for the minimum of the problem, but also on tracking the gradient for descending in the cost itself. Tracking the true gradient direction translates in a faster cost error convergence.

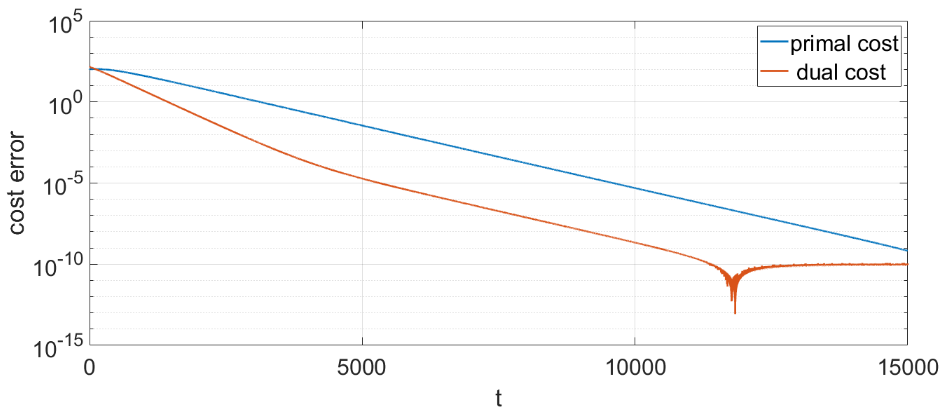

Figure 3.1: Evolution of the primal and dual cost error that shows convergence to the optimal cost.

In Figure 3.2 we show the state error, where p^* is the optimal state computed with a centralized algorithm. It can be seen that the state converges to the optimal one in a liner way, in a semi-logarithmic scale.

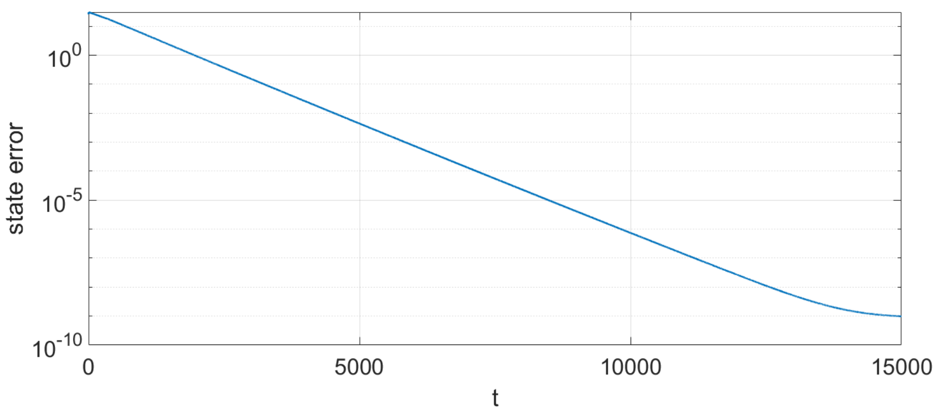

Figure 3.2: Evolution of the state error $||p - p^*||$ that shows convergence to the optimal state.

In Figure 3.3 we show the violation of the constraints. It can be seen that at the beginning the violation is greater than 0 as only the last agent has knowledge on D, but as the iterations continue this knowledge is spread to the neighbors and the algorithm is then able to track the coupling constraints.

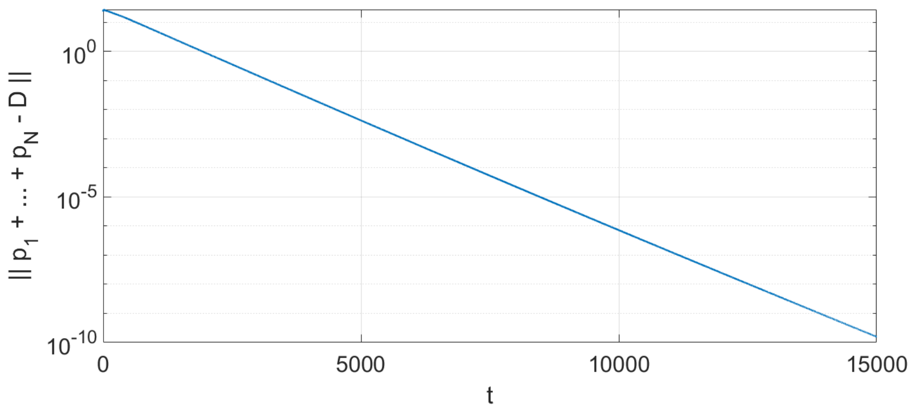

Figure 3.3: Evolution of the violation of coupling constraints $||\sum_{i=1}^{N} p_i - D||$ that shows asymptotic vanishing violation.

In Figure 3.4 we show the tracking error, with d_{avg} that is the mean of all the d_i computed at each iteration. The d_i is an estimation of the gradient for each agent, and what it can be seen is that all the d_i converge to the same value of the real gradient.

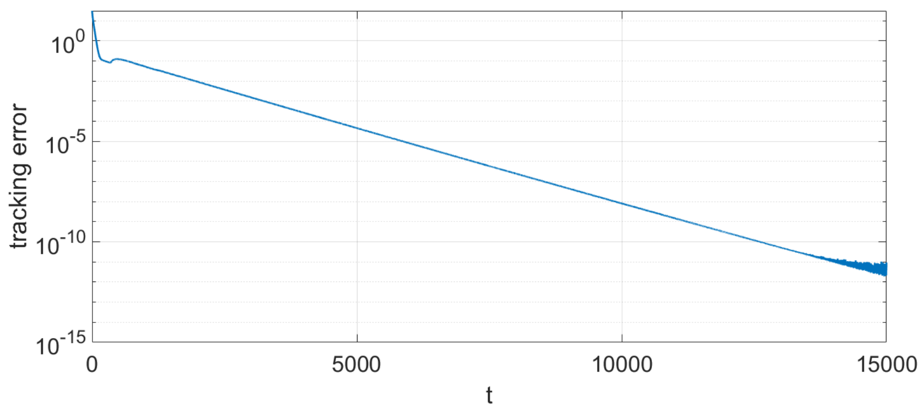

Figure 3.4: Evolution of the tracking error $||\sum_{i=1}^{N} d_i - d_{avg}||$ that shows convergence of the estimation of the gradient.

In Figure 3.5 we show the consensus error, with λ_{avg} that is the mean of all the λ_i computed at each iteration. The λ_i is the variable for the dual problem $q_i(\lambda)$, and what it can be seen is that all the λ_i converge to the same value of the dual variable, the optimal one.

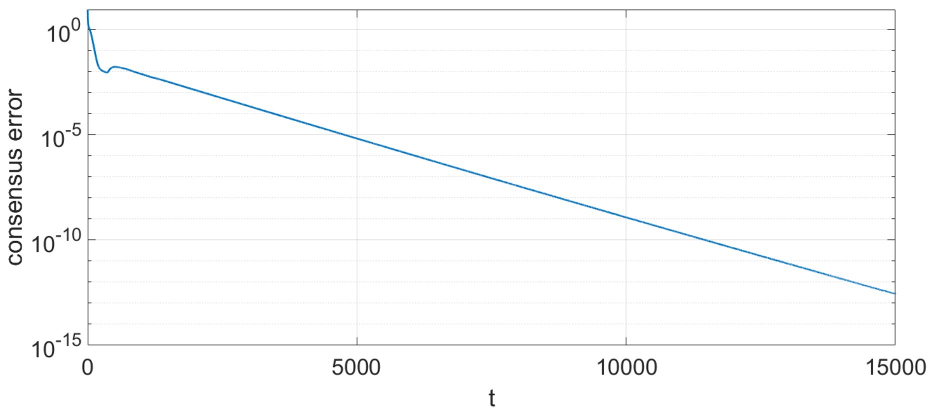

Figure 3.5: Evolution of the consensus error $||\sum_{i=1}^{N} \lambda_i - \lambda_{avg}||$ that shows convergence of the dual variable.

Conclusions

The problem has been solved using a constructive approach. Starting from the base problem and building on it the various component of the microgrid. At first the set-up has been developed and modelled in order to reduce each cost function and constraint to the initial simpler problem. On this bases a Centralized Optimal Solution has been found using MATLAB in order to evaluate the Distributed part and also to use as a tool for solving the simpler optimization problem of each agent after having applied duality. In fact, to the constraint-coupled primal problem set-up we applied duality in order to cast it in a cost-coupled problem solved using the Gradient Tracking Distributed Algorithm. The results presented show how the dual problem solves the same problem as the primal one. Moreover the error between the dual cost (distributed approach) and the optimal cost (centralized approach) is decreasing much faster, this is due to the fact that the Gradient Tracking algorithm uses a constant step size. This, as all the other results found during simulation, confirm that the gradient tracking algorithm is a very efficient and fast way to solve optimization problems structured in the proper way.

References

[1] Alireza Majzoobi and Amin Khodaei. “Application of Microgrids in Addressing Distribution Network Net-Load Ramping”. In: Sept. 2016. DOI: 10.1109/ISGT.2016.7781276.

[2] I. Notarnicola G. Notarstefano and A. Camisa. “Distributed optimization for smart cyber-physical networks”. In: *Foundations and Trends in Systems and Control (to appear)* (2019). [Online]. URL: https://arxiv.org/pdf/1906.10760.

[3] I. Notarnicola and G. Notarstefano. “Constraint-coupled distributed optimization: Relaxation and duality approach”. In: *IEEE Transactions on Control of Network Systems* (2019).