Università degli Studi di Padova

Dipartimento di Ingegneria dell'Informazione

Robust, Asynchronous and Distributed Algorithms for Control and Estimation in Smart Grids

Ph.D. candidate
Marco Todescato

Advisor
prof. Ruggero Carli

Co-Advisor
prof. Luca Schenato

Director & Coordinator
prof. Matteo Bertocco

Ph.D. School in
Information Engineering

Department of Information Engineering
University of Padova
2016

"Impossible is just a big word thrown around by small men who find it easier to live in the world they've been given than to explore the power they have to change it. Impossible is not a fact. It's an opinion. Impossible is not a declaration. It's a dare. Impossible is potential. Impossible is temporary. Impossible is nothing."

Muhammad Ali

Abstract

Starting from the second half of the XX^{th} century, desire of energy independence, great climate changes and increasing pollution rate have driven a lot of governments to invest on technologies devoted to energetic development. As a consequence, the electric infrastructure is undergoing a deep renovation both from a production, a transmission as well as a distribution point of view.

To face at the same time environmental and economical issues, many politics have decided to invest on the development and diffusion of technologies aimed at the energy production from renewable resources, e.g., photovoltaic and wind farms, hydroelectric and biomass. Particularly successful has been the development of small size generators, μ-generators, in comparison with those of the classical production plants. These small devices can be easily deployed next to the consumers and interfaced with the network through electronic *inverters*, giving rise to a grid of Distributed Energy Resources (DERs). Thanks to the widespread of these devices, the consumer may now be able to produce, manage and sell his own energy becoming himself a producer, alternatively a so called *prosumer*. It can be easily understood how the new economic figure of the prosumer could affect the deregulation of the energy market which would no more be dominated by huge producers and sellers. Conversely, it would be based on the synergy and equilibrium of a multitude of small producers/consumers.

Such an energetic scenario clashes with the actual physic of the electric grid. Indeed, this had been thought and built according to a *top-down* energy flow, in which the energy, starting from the producers, reaches the consumers in an unidirectional way. Conversely, an electric grid scattered of prosumers would foreseen a *bottom-up* energy flow where the energy, starting from the lowest layer, consisting of the prosumers, would climb up the grid to be redirected as needed. To let the development of such a deregulated market, a cautious design and control of the power inverters is necessary.

This thesis focuses on some essential aspects that must be taken care of during the engineering design and the optimal manage of this new type of power grid. In particular, the work consists of four main parts organized as follows.

1. **Introduction and modeling:** initial chapters are devoted to the introduction of the standard models describing an electric grid.

2. **Devices synchronization:** subsequently, since synchronization among devices is required in order to perform data analysis, several distributed algorithms aimed to face this task are presented.

3. **State estimation:** historically the state of the system is represented by magnitude and phase of the isofrequential sinusoidal voltage signals at each node of the grid. Having knowledge of the state is a first and fundamental task that has to be performed in order to effectively control the grid. The aim of this section is to present distributed, robust and effective algorithms to perform state estimation.

4. **Reactive power injection optimization for voltage support:** once the devices are synchronized and the state is known, the grid can be controlled in order to be stabilized and efficiently optimized. In particular, one crucial task is the support of the voltage profile via reactive power injection in order to avoid possible collapses and consequently power blackouts. This section analyzes the problem of voltage support, addressing the issues related to the stress induced on the grid by the load profile. The problem is presented in an optimization framework and a centralized as well a distributed algorithm to solve it are presented.

Particular attention has been posed on the development of distributed strategies that need only a local exchange of information among the agents constituting the network. This class of techniques is particularly interesting for its scalability property and the possibility to be effectively applied to large scale systems such as power grids.

Sommario

A partire dalla seconda metà del XX secolo, i continui cambiamenti climatici, il crescente tasso di inquinamento ed il connesso desiderio di una maggior indipendenza energetica, hanno spinto molti governi ad intensificare gli investimenti in tecnologie dedicate allo sviluppo energetico. Conseguentemente, si è potuto assistere e si sta tuttora assistendo ad un profondo rinnovamento della struttura fisica della rete elettrica a livello di produzione, trasmissione e distribuzione dell'energia.

L'investimento nello sviluppo di tecnologie dedicate alla generazione di energia da fonti rinnovabili, come il fotovoltaico, l'eolico, l'idroelettrico e le biomasse, si è rivelata una strategia comune e vincente al fine di affrontare contemporaneamente le problematiche di natura ambientale e economica. Di particolare successo si è dimostrato lo sviluppo dei cosiddetti microgeneratori, ovvero generatori di dimensioni ridotte se confrontati con i classici impianti di produzione. Questi possono essere facilmente dislocati nei pressi dei diretti utilizzatori dell'energia elettrica (consumers) ed interfacciati alla rete elettrica di distribuzione attraverso particolari dispositivi elettronici, chiamati *inverters*, dando così vita a quella che è comunemente chiamata rete di fonti distribuite di energia (Distributed Energy Resources DERs). La diffusione di generatori di questo tipo oltre a permettere una distribuzione capillare dell'energia, comporta la nascita di una nuova figura economica nota come *prosumer* (prosumer = producer + consumer): l'utilizzatore può decidere di diventare parzialmente responsabile della produzione di energia tramite i micro-generatori, diventando a sua volta produttore (producer). E' facilmente intuibile l'impatto che una figura di questo genere avrebbe in un processo di liberalizzazione del mercato economico dell'energia, il quale non sarebbe più regolato da pochi grandi produttori ma, piuttosto, basato sulla sinergia e l'equilibrio di una moltitutdine di piccoli produttori/consumatori (prosumers).

Un tale scenario energetico risulta in contrasto con l'attuale struttura fisica della rete elettrica. Quest'ultima è stata infatti pensata e progettata per permettere un flusso di energia di tipo *top-down* in cui, a partire dal produttore, l'energia giunge al consumatore in maniera unidirezionale. Una rete elettrica dominata da prosumers prevederebbe

invece un flusso energetico di tipo *bottom-up*, il quale partendo dal basso, risale per essere eventualemente redistribuito. Al fine di permettere lo sviluppo di un mercato liberalizzato e popolato di prosumers, è necessaria un'oculata progettazione e controllo degli inverters attraverso i quali i micro-generatori si interfacciano alla rete.

Questo lavoro di tesi si occupa di alcuni degli aspetti fondamentali da considerare durante la gestione ottimale di una rete elettrica. In particolare, il lavoro si articola in quattro parti principali così organizzate:

1. **Introduzione e modellazione:** i primi capitoli sono dedicati all'introduzione dei modelli standard di una rete elettrica.

2. **Sincronizzazione dei dispositivi:** successivamente, dal momento che, al fine di effettuare analisi dei dati è richiesta sincronizzazione tra i dispositivi all'interno della rete, vengono presentate diverse tecniche distribuite per affrontare e risolvere questo problema.

3. **Stima dello stato della rete:** storicamente lo stato è rappresentato dai valori assoluti e dalle fasi dei segnali sinusoidali isofrequenziali di tensione ai nodi della rete. La sua conoscenza è necessaria al fine del controllo ed ottimizzazione del funzionamento della rete. In questa parte della tesi, vendono dunque presentati una serie di algoritmi distribuiti e asincroni per la stima dello stato.

4. **Ottimizzazione dell'iniziezione di potenza reattiva per il supporto delle tensioni:** una volta sincronizzati i dispositivi e noto lo stato, la rete può essere controllata al fine di ottimizzarne determinate prestazioni. In questa parte della tesi, ci si occupa della minimizzazione dello stress della rete indotto dal carico di potenza col fine ultimo di supportare i livelli di tensione ed evitare possibili collassi e disastrosi black-out. In particolare vengono presentate due possibili soluzioni, una centralizzata ed una distribuita, per risolvere il problema del supporto delle tensioni.

Durante il lavoro, si è posta particolare attenzione allo sviluppo di tecniche di controllo ed algoritmi di natura distribuita ovvero che necessitano solo uno scambio di informazione locale tra gli agenti costituenti le rete. Tale classe di algoritmi di ottimizzazione e controllo risultano intrinsecamente adatti ad essere implementati e sfruttati in sistemi complessi e di larghe dimensioni come le reti elettriche.

Acknowledgments

First, I would like to thank my advisor Prof. Ruggero Carli for his supervision which started back for years ago, during my master thesis and continued during the awesome period of my PhD. I'd like to thank him for the discussions and for steering my learning along the right path.

I would like to deeply thank all the advisors I had the great fortune to have during these three years of research: Prof. Luca Schenato which together with Prof. Carli, had been a mentor in Padova; Prof. Francesco Bullo which advised me during my period at UCSB and Prof. Florian Dörfler (twice) during my stays at UCLA and at ETH.

A thank goes to Prof. Antonio Franchi which I had the pleasure to meet and know at the MPI in Tübingen.

Not to forget my almost two advisors: Dr. Saverio Bolognani and (now Prof.) John Simpson. I think we had some real good researching discussion as well as some good time during my stays in Boston, Santa Barbara and Zurich.

One thanks goes to Prof. Farina and all the committee that patiently read and reviewed this thesis.

To conclude the research-side and begin the personal-side thanks, I must mention my PhD fellow, (Ing.) Andrea Carron. We had the pleasure to spend together more than one abroad experience as well as the two remaining years in Padova. I really wish him the best for his life and career and, hopefully, to meet him to work again together in the future.

All my PhD colleagues in Italy and abroad (in particular, Andrea, Chiara, Diego, Giacomo, Giulia, Giulia, Irene, Michele, Nicoletta, Sina, Sinan, Yutao) for the good time.

All my buddies in Vicenza. Life without them would be boring and partly meaningless.

A special mention goes to my parents, Andrea and Lucia, and my family, Giulio, Carlotta and Enrico, which always supported and, I am sure, will support me in the future. They always let me live my life as I wanted to, teaching me good and bad, right and wrong to grow up and to eventually become what I am now.

The last, most special and unique thank goes to the person who taught and teaches me love, love for life and for what counts in life. From her I learn to relax, to be less rational, less engineer when possible and to focus when necessary. She supports me every day and during these last three years, had the strength to patiently wait, bear and support me while I was abroad. My “emotional conscience”, Cinzia.

Contents

1 Introduction

"Men die but their works endure."
A.L.Cauchy

1.1 The electric grid

From the XIX century, with the beginning of the public illumination and the systematic use of electricity in everyday life, the electrical power is playing a fundamental and unique role in modern society. One thing our lives often rely on is the capability to have continuous and uninterrupted electric power to supply all the devices we need and the infrastructure that surround us. Nowadays, with the massive spread of electronic devices, this need is even more rooted in our contemporary society. However, even if assumed, this possibility is given to us by the careful design and engineering of the electric infrastructure, which probably represents the biggest engineered system by humankind.

The classical electric infrastructure may be hierarchically divided into three main layers in a *top-down* scheme: the generation, i.e., the power plants; the High Voltage (HV) transmission grid and the Medium/Low Voltage (MV/LV) distribution grid which, together, represent the backbone of the electric infrastructure (Harris Williams, Summer 2014) since they are responsible for delivering the energy to the final customers. According

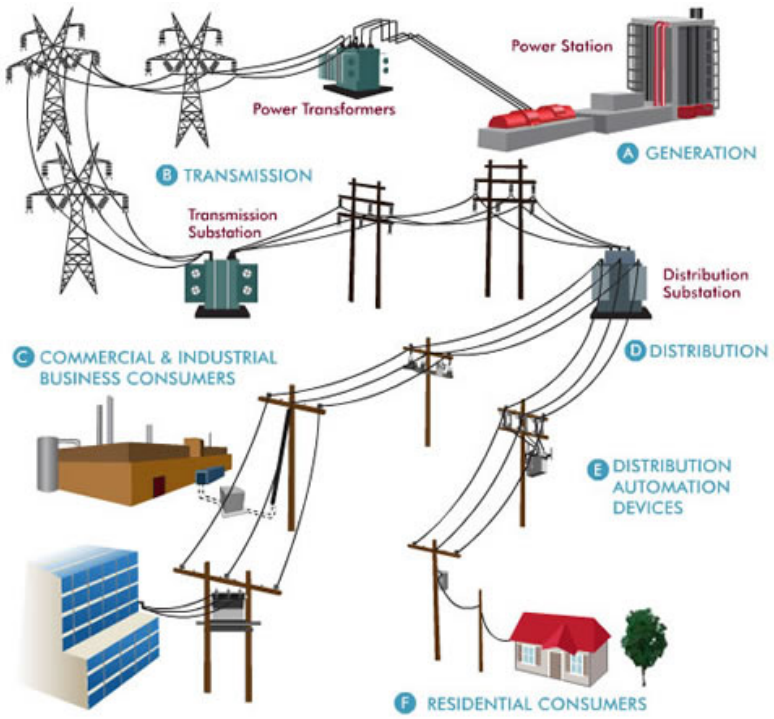

Figure 1.1: Classical electric grid infrastructure.

to this framework, the energy is produced at the power plants. Its voltage level is then increased up to HV by transformers in order to reduce energy losses during transportation. By passing through the transmission grid, energy is carried to substations, where the voltage level is lowered from HV to MV in order to be dispatch to end users, e.g., residential, commercial and industrial customers. Finally, distribution transformers decrease the voltage, from MV to LV, in order to safely distribute electricity to end users.

As can be seen from the Figure 1.1, which depicts a schematic view of the electric structure, we have:

A. **Power station**, where the power is produced.

B. **Transmission grid** consisting of:

   (a) power transformers, i.e., high-voltage transformers, that rise up the voltage level;

   (b) transmission lines used to transport the power which are typically mostly inductive;

   (c) transmission substations connecting different transmission lines.

C. **Commercial & Industrial consumers** which might make use of the power energy directly at the MV level.

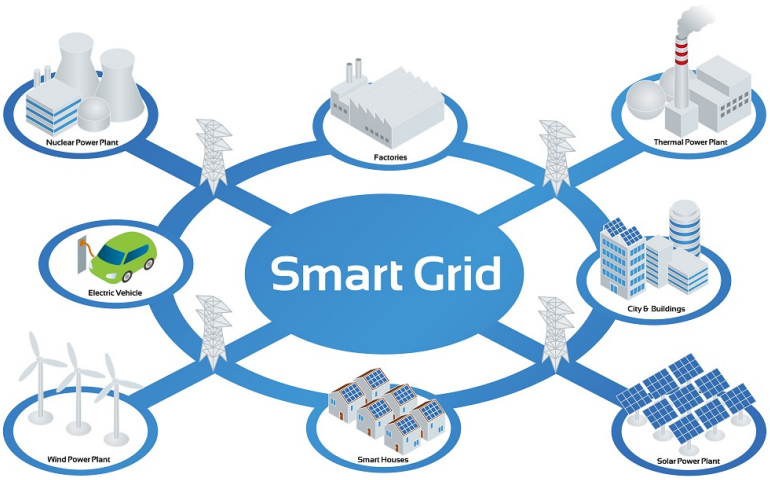

Figure 1.2: Smart grid infrastructure.

D. **Distribution grid** consisting of:

   (a) distribution substations aimed at stepping down the voltage and reroute the power to different branches of the distribution grid;

   (b) distribution lines, e.g., overground and underground lines, which are typically mostly resistive;

E. **Automation devices** used to monitor and step further down the voltage.

F. **Residential customer** which are the LV end user of the grid.

1.2 A new scenario: the Smart Grid

In the last decades, pushed by governments' increasing desire of energy independence, we are witnessing a deep renovation of the electric infrastructure. Major research efforts have focused on the development of *Distributed Energy Resources* (DERs). In particular, as a consequence of the more and more relevant and delicate topic of sustainable environmental development (Twidell, Weir, et al., 2015), renewable and *green* energy sources, e.g., solar, wind, water, biomass, are becoming the main actors (Jiayi, Chuanwen, and Rong, 2008) of this change. Clearly, the possibility to widespread small size renewable energy resources with power ratings less than a few tens of kilowatts would have a major impact from both an energetic, an economic as well as an environmental point of view. First of all, this could increase the reliability of the energy dispatch to final customers. Secondly, the

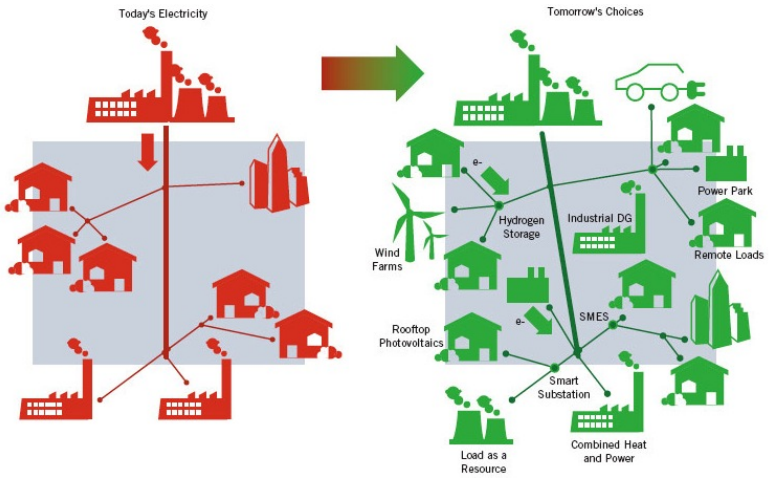

Figure 1.3: Electric infrastructure transition.

outsourcing of the energy production could potentially break down the monopoly of the energy market which is now owned by few big producers. Finally, the massive use of green energy could eventually lead to a complete disuse of traditional and unsustainable energy production sources. The widespread of DERs would completely decentralize the energy production which would now be physically localized next to the consumers rather than far away from them. Thanks to this, the consumers could now be able to produce and dispatch their own energy becoming themselves producers. The ultimate combination of producers and consumers would give rise to a completely new economic figure, the *prosumers*, which would unavoidably contribute to the deregulation of the energy market, which would now be driven by the synergy and the equilibrium of a multitude of producers/consumers rather then dominated by huge producers/sellers.

This new energetic/economic scenario foreseen the birth of the so called *Smart Grid* (Liserre, Sauter, and Hung, 2010) where the production and consumption are no more placed respectively at the top and at the bottom of the infrastructure rather, they occupy different positions at the same level of the network, see Figure 1.2. However, this paradigm clashes with the actual physics of the classical electric grid which has been built according to a top-down criterion (Section 1.1) where the energy physically flows from the power plants, down to the consumers passing through the transmission and the distribution grid. Conversely, this new perspective requires a *bottom-up* energy flow where the energy produced by the prosumers is dispatched and sent all over the grid, where needed. To this end the grid must change its structure as illustratively depicted in Figure 1.3. However, to undertake such an abrupt change of the structure and of the way the grid is operated comes not without side effects (Ipakchi and Albuyeh, 2009).

Since the design of a suitable electric infrastructure from scratch is not possible, the development of new technologies and new paradigms must be carefully designed in order to be effectively and seamlessly integrated with what already exists.

1.3 Manuscript outline

This work focuses on some aspects that may be taken into account during the design, monitoring, control and optimization of a modern smart grid. These regard i) the synchronization of the devices spread over the grid; ii) the state estimation of the grid from the collection of noisy electric measurements; iii) the optimization and control of the reactive injection for the support of the voltage profile. Novel algorithms to solve these problems are presented. They relies on different assumptions regarding, e.g., the sensing and communication capabilities over the network. However, all the solutions proposed share, as least common denominator, a distributed implementation which makes them naturally scalable, intrinsically more robust and then amenable for implementation in large scale systems.

In particular, the first part of the manuscript (the remaining of this chapter and Chapter 2) recall the used notation and presents the required models. The second part of the thesis, contained in Chapter 3 is devoted to the analysis of the synchronization problem among devices. The problem is tackled as a particular type of estimation problem in the framework of localization where only relative measurements among smart devices are available. Chapter 4 contains the third part of the thesis which is devoted to the state estimation problem in a distribution grid. Two different strategies, suitable to the case of synchronous and asynchronous communications, respectively, are presented. The final part, regarding optimization and control, is discussed in Chapter 5. This part is devoted to present a particular safety aspect which is related to voltage support and stress minimization in power systems. In particular, it will be shown a particular measure for the stress induced by the reactive loads on the grid and a possible countermeasure, through reactive power injection, to support the voltage while minimizing the stress in order to improve the network safety. Chapter 6 offers some concluding remarks.

1.4 Mathematical preliminaries & notation

In this section we present the notation used hereafter along the manuscript.

Symbols

The symbols $\mathbb{R}^n$ ($\mathbb{R}^{m\times n}$), $\mathbb{C}^n$ ($\mathbb{C}^{m\times n}$), $\mathbb{Z}^n$ ($\mathbb{Z}^{m\times n}$) and $\mathbb{N}^n$ ($\mathbb{N}^{m\times n}$) denote the vector spaces of n dimensional column vectors ($m \times n$ matrices), where all the entries are real, complex, integer and natural, respectively. Lowercase italic letters, e.g. x, v, z, denote scalar values. Lowercase roman bold letters, e.g., $\mathbf{x}, \mathbf{v}, \mathbf{z}$, denote vectors which we assume by default to be column vectors. Uppercase italic letters denote, e.g. A, B, X, denote matrices. In general, uppercase calligraphic symbols, e.g. $\mathcal{A}, \mathcal{C}, \mathcal{X}$, denote sets. The symbols $\mathbb{1}$ and $\mathbb{I}$ denote the vector of all ones and the identity matrix of suitable dimension, respectively. With the symbol $\mathbb{1}_h$ we denote the canonical vector with all entries equal to zero except the h-th one which is equal to 1. The symbol $\mathbb{0}$ represents both the vector and matrix with all zero entries of suitable dimension. Finally, the symbol $\emptyset$ denotes the empty set.

Scalar, vector and matrix operators

Given a scalar, $|a|$ denotes its absolute value while $\angle a$ denotes its argument. Moreover, with the symbols $\Re(\cdot)$ and $\Im(\cdot)$ we denote its real and imaginary parts, respectively. In the context of complex valued quantities, the symbol $\mathtt{j}$ denotes the imaginary unit. Then, a complex scalar a can be represented both in polar coordinates as $a = |a|\exp(\mathtt{j}\angle a)$, or in rectangular coordinate as $a = \Re(a) + \mathtt{j}\Im(a)$. Given a, with $\overline{a}$ we denote its conjugate. Similar notation is used for vectors. That is, given a vector $\mathbf{a}$, with $|\mathbf{a}|$, $\angle\mathbf{a}$, $\Re(\mathbf{a})$, $\Im(\mathbf{a})$, $\overline{\mathbf{a}}$, we denote the vector of component wise absolute value, argument, real part, imaginary part and conjugate, respectively. In addition, given $\mathbf{a}$, with $\mathbf{a}^T$ we denote its transpose; while with $\mathbf{a}^*$ its conjugate transpose. With a_i we denote the i-th entry of $\mathbf{a}$, i.e., $a_i = [\mathbf{a}]_i$. Given two vectors, the symbol $\odot$ denotes their Hadamard, i.e., component-wise, product, while $\otimes$ denotes their Kronecker product. Given $\mathbf{a}$, with $\|\mathbf{a}\|_p$ we denote its p-norm. When it will not create confusion we will drop the subscript $_p$. We will write $\mathbf{a} \leq \mathbf{b}$ ($\mathbf{a} \geq \mathbf{b}$) if $\mathbf{a}$ is a vector whose entries are component-wise smaller (greater) than the entries of $\mathbf{b}$. Given a vector $\mathbf{a}$, with $[\mathbf{a}]$ or equivalently $\mathrm{diag}(\mathbf{a})$ we denote the diagonal matrix whose main diagonal consists of the element of $\mathbf{a}$.
Similar notation is used for matrices. Then, given A, with $|A|$, $\angle A$, $\Re(A)$, $\Im(A)$, $\overline{A}$, A^T and A^* we denote the same operators as before. Differently, given A, either with a_{ij} or A_{ij} we denote its ij-th scalar entry of, i.e., $a_{ij} = A_{ij} = [A]_{ij}$. With A_{i*} we denote its entire i-th row; similarly, with A_{*j} its entire j-th column. The symbols $\odot$ and $\otimes$ apply to matrices as well. With $\|A\|_p$ we denote the induced p-norm and again, if it will not create confusion, we will drop the subscript $_p$. We will write $A \geq 0$ ($A > 0$) if the matrix A is positive semidefinite (definite). Similarly, we will use the symbol $\leq$ ($<$) for negative

semidefinite (definte) matrices. Given a square matrix A, A^{-1} denotes its inverse; while given a generic matrix A, $A^{\dagger}$ denotes its pseudo-inverse. Given a matrix A, with $\mathrm{tr}(A)$, $\mathrm{im}(A)$ and $\ker(A)$ we denote its trace, image space and null space, respectively. Given a square matrix A, with $\Lambda(A)$ we denote the set containing its eigenvalues; while with $\mathrm{sr}(A)$ and $\mathrm{esr}(A)$ we denote its spectral radius and essential spectral radius, respectively, i.e., its largest and second largest eigenvalue in absolute value. A matrix M is said to be a positive non singular M-matrix if is a matrix with negative off-diagonal elements and positive diagonal ones which can be expressed in the form $M = s\mathbb{I} - B$, with $B_{ij} \geq 0, s > \mathrm{sr}(B)$. A matrix P is said to be stochastic if its elements are nonnegative and satisfy $P\mathbb{1} = \mathbb{1}$. Moreover, it is said doubly stochastic if it is stochastic and $\mathbb{1}^T P = \mathbb{1}^T$. A stochastic matrix P is primitive if it has only one eigenvalue equal to 1 and all the other eigenvalues are strictly inside the unitary circle.

Set operators

Given a set $\mathcal{A}$, the symbol $|\mathcal{A}|$ denotes its cardinality, i.e., the number elements contained in $\mathcal{A}$. Given two sets $\mathcal{A}$ and $\mathcal{B}$, with $\mathcal{B} \subset \mathcal{A}$ ($\mathcal{B} \subseteq \mathcal{A}$) we denote the fact that $\mathcal{B}$ is a proper (non necessarily proper) subset of $\mathcal{A}$. The symbols $\cap$ and $\cup$ represents intersection and union of sets, respectively. The symbol $\times$ denotes the Cartesian product. Given a set $\mathcal{A}$, the symbols $\mathrm{int}(\mathcal{A})$, $\mathrm{bd}(\mathcal{A})$ and $\mathrm{cl}(\mathcal{A})$ denote its interior, boundary and closure, respectively.

Statistical operators

Given a random variable a, this is said to be normally distributed and denoted by $a \sim \mathcal{N}(\mu, \sigma^2)$, if its realizations are drawn from a Gaussian probability density function of mean μ and variance σ^2. Similarly, we will write $\mathbf{a} \sim \mathcal{N}(\mu, \mathbf{\Sigma})$ for normal random vectors. The symbol $\mathbb{E}[\cdot]$ denotes the expectation operator. Given either a random variable or a random vector, the symbol $\mathrm{Var}(\cdot)$ denotes its variance.

Graph theory

A *directed graph* $\mathcal{G}$ is denoted as a pair $(\mathcal{V}, \mathcal{E})$ where $\mathcal{V} = \{1, \ldots, n\}$ is the set of vertices and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the set of directed edges. An edge $e = (i, j) \in \mathcal{E}$ is an arch which is incident on nodes $i, j \in \mathcal{V}$ and is assumed to be directed away from i and directed toward j. The graph $\mathcal{G}$ is said to be *bidirected* or *undirected* if $(i, j) \in \mathcal{E}$ implies $(j, i) \in \mathcal{E}$. Given a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, a *directed path* in $\mathcal{G}$ consists of a sequence of vertices $(i_1, i_2, \ldots, i_r)$ such that $(i_j, i_{j+1}) \in \mathcal{E}$ for every $j \in \{1, \ldots, r-1\}$. Similarly, an *undirected*

path in $\mathcal{G}$ consists of a sequence of vertices $(i_1, i_2, \ldots, i_r)$ such that either $(i_j, i_{j+1}) \in \mathcal{E}$ or $(i_{j+1}, i_j) \in \mathcal{E}$ for every $j \in \{1, \ldots, r-1\}$. Roughly speaking, an undirected path is a path from a node to another node that does not respect the orientation of the edges. The directed graph $\mathcal{G}$ is said to be *strongly connected* (resp. *weakly connected*) if for any pair of vertices $i, j \in \mathcal{V}$ there exists a directed path (resp. undirected path) connecting i to j. Given the directed graph $\mathcal{G}$, the set of neighbors of node i, denoted by $\mathcal{N}_i$, is defined as the set $\mathcal{N}_i = \{j \in \mathcal{V} \,|\, (i, j) \in \mathcal{E}\}$. By convention $i \notin \mathcal{N}_i$, so to denote the set of neighbors plus node i itself we use the symbol $\mathcal{N}_i^+ = \mathcal{N}_i \cup \{i\}$. A directed graph is said to be *regular* if all the nodes have the same number of neighbors. Given a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with $|\mathcal{V}| = n$ and $|\mathcal{E}| = m$, let the *incidence matrix* $A \in \mathbb{R}^{m \times n}$ of $\mathcal{G}$ be defined as $A_{ei} = [a_{ei}]$, where $a_{ei} = 1, -1, 0$, if edge $e \in \mathcal{E}$ is incident on node i and directed away from it, is incident on node i and directed toward it, or is not incident on node i, respectively. Let L be the Laplacian matrix associated with $\mathcal{G}$ defined as $L = A^T A$. Finally, if to each edge $e \in \mathcal{E}$ is associated a weighting factor w_e then, by collecting all the weights in the vector $\mathbf{w}$ and by defining the matrix $W = \mathrm{diag}(\mathbf{w})$, the *weighted* Laplacian matrix associated to $\mathcal{G}$ is defined as $L = A^T W A$.

2

Model of a Smart Grid

"All models are wrong, but some are useful"
George E.P. Box

In this chapter we introduce and present the model of a smart grid as used later on along the manuscript. First of all we recall some basic notions and definitions regarding AC circuits. Most of the concepts here introduced can be found in Andersson (2008) and Von Meier (2006). Secondly, we outline our idea of smart grid, envisioned as a *cyber-physical* system consisting of a physical (electric) layer and a cyber layer equipped with sensing, computation and communication capabilities.

2.1 Basics of AC circuits

Phasorial representation

Consider the generic sinusoidal waveform,

$$a(t) = a_m \sin(2\pi f t + \varphi)\,, \tag{2.1}$$

where a_m represents its amplitude, having the same physical dimension of the waveform $a(t)$, $\omega := 2\pi f$ is the pulsation expressed in [rad/s] (f is the frequency expressed in [Hz])

and φ is the initial phase, expressed in [rad]. By assuming $a(t)$ belonging to the set of isofrequential sinusoids with given frequency f then, the waveform (2.1) can be uniquely determined by its amplitude a_m and it phase φ. Indeed, these two parameters define one and only one complex number given by

$$a = \frac{a_m}{\sqrt{2}} \exp(\mathtt{j}\varphi)\,. \tag{2.2}$$

On the other hand, the complex number a identifies one and only one waveform with pulsation ω, given by (2.1), i.e., there is a one-to-one relation between any complex number and the elements of the set of isofrequential sinusoids.
The complex number (2.2) is called the *phasor*, or phasorial representation, of the sine wave $a(t)$. Notice that the phasor magnitude is equal to the *root mean square* of $a(t)$ over one of its period $T := \frac{2\pi}{w} = \frac{1}{f}$, i.e.

$$\frac{a_m}{\sqrt{2}} = \sqrt{\frac{1}{T}\int_{t_0}^{t_0+T} a^2(t)dt}\,. \tag{2.3}$$

The phasorial representation turns out to be useful when the grid is working in steady state, i.e., at an ideal sinusoidal regime. In this case, both the currents (measured in amperes [A]) and the voltages (measured in volts [V]) are sinusoidal signals (with frequency $f = 50$ Hz in Europe and $f = 60$ Hz in the U.S.A) of the form

$$i(t) = i_m \sin(\omega t + \varphi_i)\,, \tag{2.4}$$

$$u(t) = u_m \sin(\omega t + \varphi_u)\,, \tag{2.5}$$

and they can be associated with the phasors $i = \frac{i_m}{\sqrt{2}} \exp(\mathtt{j}\varphi_i)$ and $u = \frac{u_m}{\sqrt{2}} \exp(\mathtt{j}\varphi_u)$, respectively.

Here we point out that we will make wide use of the equivalent phasorial representation since, in the problems considered, we usually assume the grid to be in steady state regime.

Impedance and admittance

We define *impedance* of a electrical element the ratio between the current passing through it, expressed by the phasor i, and the voltage drop between its ends, expressed by the phasor u, i.e.,

$$z = \frac{i}{u}\,, \quad \text{ohm } [\Omega]\,. \tag{2.6}$$

By definition, the impedance is in general a complex numbers whose real part $r := \Re(z)$ is called *resistance* and whose imaginary part $x := \Im(z)$ is called *reactance*, both of them measured in [Ω], i.e.,

$$z = r + \mathtt{j}x\,.$$

The inverse of the impedance is called *admittance*

$$y = \frac{i}{u}\,, \quad \text{siemens [S]}\,.$$

As the impendance, in general the admittance is a complex numbers whose real part $g := \Re(y)$ is called *conductance* and whose imaginary part $b := \Im(y)$ is called *susceptance*, both of them measured in [S], i.e.

$$y = g + \mathtt{j}b\,.$$

We use impedances/admittances to model electric lines, both overhead and underground, between buses[1], and shunt elements which connect buses to ground. For this reason we consider only *constant impedance devices* which exhibit a linear and constant relation between their voltage and current. However, for the sake of knowledge, we recall that certain materials and electronic devices exhibit a nonlinear relationship between current and voltage, that is, z is not constant but varies for different values of u and i.

AC Power

As above mentioned, in sinusoidal steady state alternating current (AC) circuits, voltage and current are described by sinusoidal signals. As for direct current (DC) circuit, it is convenient to give a definition of power which represents a measure of energy per unit of time, and thus it describes the rate of energy consumption or production. In particular the *instantaneous power* at every time instant is defined as the product between the voltage and the current, that is

$$\begin{aligned} p(t) &= v(t)i(t) && (2.7)\\ &= v_m \sin(\omega t + \varphi_u) i_m \sin(\omega t + \varphi_i) && (2.8)\\ &= \frac{v_m i_m}{2}\cos(\phi) - \frac{v_m i_m}{2}\cos(2\omega t + \varphi_u + \varphi_i) && (2.9) \end{aligned}$$

[1] As later described, a bus can be just thought as the point of connection of a device to the grid. In our graph representation, the buses will coincide with the nodes of the graph

where $\phi := \varphi_u - \varphi_i$. From Eq.(2.7) it is possible to observe how the instantaneous power results as the difference of two terms: the first is constant and depends on the phase difference ϕ while the second is a sine wave oscillating at double the frequency of the original signals. Although the instantaneous power represents the instantaneous consumption/production of energy a more meaningful quantity is represented by the average power over a certain period of time, usually a sinusoidal period, define as

$$\begin{aligned} p &= \frac{1}{T}\int_{t_0}^{t_0+T} p(t)dt \\ &= \frac{v_m i_m}{2}\cos(\phi) \end{aligned} \tag{2.10}$$

This quantity is called *real* or *active power*, is measured in watt [W] and corresponds to the power actually consumed by loads or generated by generators. By exploiting the phasorial representation for voltage and current, Eq.(2.10) can be rewritten as

$$\begin{aligned} p &= \frac{v_m}{\sqrt{2}}\frac{i_m}{\sqrt{2}}\Re\left(\exp(\mathtt{j}\varphi_u - \varphi_i)\right) \\ &= \Re(u\bar{i}) \end{aligned} \tag{2.11}$$

One could ask what the oscillating part of the instantaneous power, i.e., its second term, represents. Its average value over a period of time is zero and thus it does not perform any physical work over time. However, it still represents an energy "flow" over time, due to the exchange of energy between the electric and the magnetic fields and which flows back and forth along the circuit. This kind of power exchange is commonly related to what is called *reactive power*, measured in Volt-Ampere-Reactive [VAR] and defined as

$$q = \frac{v_m i_m}{2}\sin(\phi) \tag{2.12}$$

$$= \Im(u\bar{i}) \tag{2.13}$$

which represents the "imaginary" counterpart of the active power as defined in (2.11). Overall, the complex sum of active and reactive power is defined as *apparent power*, measured in Volt-Ampere [VA], i.e.,

$$s := u\bar{i} = p + \mathtt{j}q \tag{2.14}$$

Even though this definition of reactive and apparent power could seem artificial, it turns out that apparent power is important in the context of equipment capacity. Indeed, even if reactive power does not produce usable work, it still represents a flow of energy which

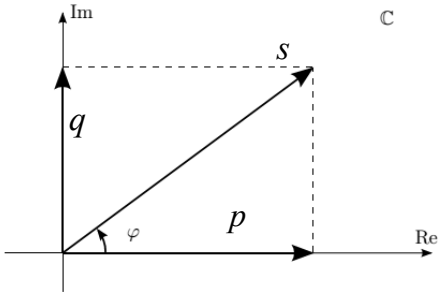

Figure 2.1: Representation in the complex plane of the apparent, active and reactive power.

is responsible for devices heating. Since usually the operating voltage of a given piece of equipment is quite constant, apparent power turns out to be a fair way to indicate the current flowing in the device which, usually, represents the reference quantity for devices' thermal capacity limits. For this reason, utility equipment ratings are typically given in VA.

2.2 Cyber-Physical system

The smart grid we envision will not only represent a physical system coinciding with the actual electric infrastructure. Rather, it will be a more complex system with computing and communication capabilities embedded in all types of objects and structures in the physical environment. Thanks to these additional capabilities it will be possible to bridge the cyber world of computing and communication with the actual physical world. Such systems, composed of these two "layers" are referred to as *cyber-physical systems* (Rajkumar, Lee, Sha, and Stankovic, 2010). Basically, they are physical and engineered systems whose operations are controlled, monitored, coordinated and integrated by a computing and communicating core. In this work, we envision a *smart network* as one of this cyber-physical systems, where:

- the **cyber layer** consists of intelligent agents, deployed in the grid, provided with actuation, sensing, communication, and computational capabilities;
- the **physical layer** consists of the power distribution infrastructure, including power lines, loads, microgenerators, and the distribution substations.

In general, in this kind of systems the cyber and the physical layer might not exactly coincides, as shown in Figure 2.2 where only a subset of nodes is equipped by smart devices. However, in the following, we assume that these two layers have the same

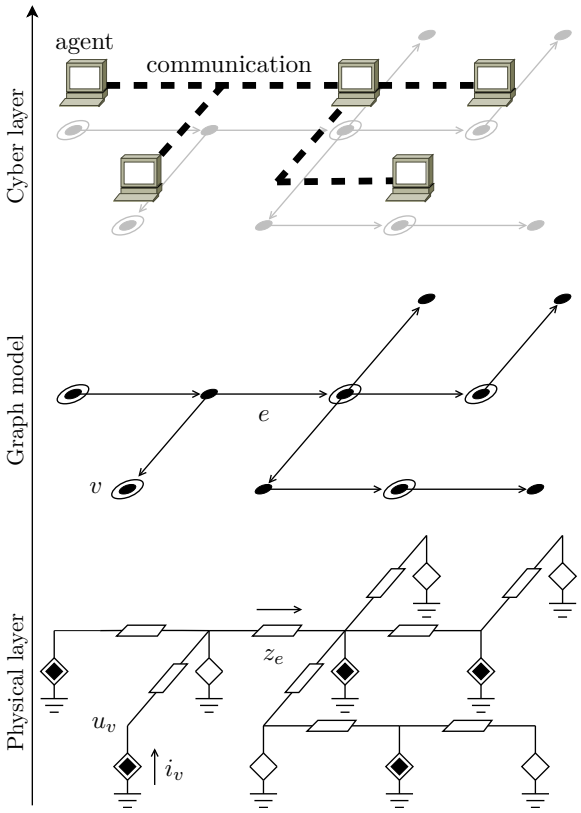

Figure 2.2: Schematic representation of the power distribution grid model. The lower panel shows the circuit representation of the *physical layer*. Different symbols represent different electrical devices in the grid. The middle panel illustrates the adopted *graph representation* for the same grid. The upper panel represents the *cyber layer*, where agents are connected via some communication infrastructure.

topology, i.e., graph structure. This does not necessarily mean that every agent has the same computing capabilities although all of them can at least communicate.

2.3 Cyber layer

The cyber layer is devoted to sensing, computing, communication and actuation tasks. In particular, we assume that some of the nodes (or possibly all of them) in the physical layer are equipped with what we refer to as an *agent*, that is a smart device characterized by some specifications. The agents are provided with:

- *sensing devices*: one example of sensing devices are *phasor measurement units* (PMUs) which measure, process and return current and voltage phasors (Phadke and Thorp, 2008; Barchi, Macii, and Petri, 2013; von Meier, Culler, McEachern, and Arghandeh, 2014);

- *computational devices*: exploited to implement the proposed algorithms;
- *actuation devices*: in order to control the grid, the agents must be able to physically actuate the grid. In particular in the last part of this thesis we consider devices able to inject a certain amount of reactive power in the grid within prescribed capability limits. Examples of such devices are the so called *synchronous condensers* (Bergen, 2009). However, renewable devices connected to the grid through power inverters are usually able to access to a certain amount of stored energy.

Last but not least, we include the communication capabilities and the knowledge the agents have access to. In particular the agents are characterized by:

- *communication capabilities*, necessary for the information exchange required for the algorithm execution. Possibly, agents can communicate via the same power lines (via *power line communication*).
- *local and partial knowledge of the electrical grid*, e.g. the impedance of the electric path between neighbor agents.

All these capabilities and equipment and, in particular, the communication topology of the cyber layer are modeled as an undirected communication graph $\mathcal{G}_c = (\mathcal{V}_c, \mathcal{E}_c)$ where the agents are identified by the node set $\mathcal{V}_c$ and in general labeled from 1 to n, while the communication edges are identified by the edge set $\mathcal{E}_c$.

Remark 2.3.1 (Communication modeling). As above mentioned, the communication among different entities/agents/nodes in the considered network is, in general, modeled as a graph $\mathcal{G}_c$. For what regards the more specific assumptions on the particular communication protocol used and on the communication non idealities considered, we will explicitly mention and outline them along the dissertation.

2.4 Electrical grid modeling

We model the physical layer (see Figure 2.2, lower panel) coinciding with the electrical infrastructure of a smart power distribution network, as a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, in which edges represent the power electric lines, i.e., the branches, and nodes represent the buses, i.e., loads and generators (see Figure 2.2, middle panel). Thus we associate with the electric grid the sets

- $\mathcal{V}$, that is the set of nodes (the buses), with cardinality n;
- $\mathcal{E}$, that is the set of edges (the electrical lines connecting them), with cardinality m.

The system is assumed to be in synchronous steady state, namely all the signals have the same frequency. Then, it can be described by the following quantities which coincide with the phasorial representation of all the signals:

- $\mathbf{u} \in \mathbb{C}^n$, where $u_h = |u_h| \exp(\mathtt{j} \angle u_h)$ is the grid voltage at node h;
- $\mathbf{i} \in \mathbb{C}^n$, where $i_h = |i_h| \exp(\mathtt{j} \angle i_h)$ is the current injected at node h;
- $\boldsymbol{\xi} \in \mathbb{C}^m$, where ξ_e is the current flowing on edge e;
- $\mathbf{s} = \mathbf{p} + \mathtt{j}\mathbf{q} \in \mathbb{C}^n$, where s_h, p_h and q_h are the complex, the active and the reactive power injected at node h. If $p_h > 0$ ($q_h > 0$) we say that node h is injecting active (reactive) power, if $p_h < 0$ ($q_h < 0$) we say that node h is absorbing active (reactive) power.

For the sake of ease of notation, throughout this manuscript we will use the symbol[2] ν to denote the voltage magnitudes and the symbol θ to denote the voltage angle.

Let A be the incidence matrix associated to $\mathcal{G}$. For every edge $e \in \mathcal{E}$ of $\mathcal{G}$, we define by z_e the impedance of the corresponding power line. In this sense, we model the grid power lines as series impedances. Similarly, we model the shunt elements connecting each bus to ground and denoted by their admittance z_h^{sh}. Note that differently from the line impedances which are referred to edges, the shunt elements are referred to nodes. However, it is convenient to associate to each shunt element a *virtual* edge e which in the incidence matrix starts at the corresponding node h, i.e., $a_{eh} = 1$, and does not end anywhere. With this slightly modified definition of incidence matrix and collecting all the impendance values z_e into the diagonal matrix $Z := \mathrm{diag}(z_e,\ e \in \mathcal{E})$, standard Ohm law and Kirchoff currents law can be written, respectively, as

$$A\mathbf{u} + Z\boldsymbol{\xi} = 0\,, \tag{2.15}$$

$$A^T\boldsymbol{\xi} + \mathbf{i} = 0\,. \tag{2.16}$$

From (2.16) and (2.15) we can also obtain the relation

$$\mathbf{i} = Y\mathbf{u} \tag{2.17}$$

where Y is the complex-valued weighted admittance matrix of the grid, i.e., $Y = A^T Z^{-1} A \in \mathbb{C}^{n\times n}$. Notice that the admittance matric Y is symmetric and sparse. In particular it

[2] As usual, to distinguish between scalar and vector quantities we use bold symbols to denote the latter.

reflects the sparsity pattern of the electric grid. This means that each row h of the matrix corresponds to a node of the graph and the indices of the non-zero elements of the row correspond to the nodes which are directly connected to node h, i.e., $\mathcal{N}_h$. In particular, let h be a bus of the network, we define $\mathcal{E}_h$ as the subset of $\mathcal{E}$ containing all the edges e connected to node h, both outgoing, i.e. (h,i), and ingoing edges, i.e. (i,h). The bus admittance matrix can be also written as

$$Y_{hk} = \begin{cases} \sum_{e\in\mathcal{E}_h} y_e + y_h^{sh}, \text{ if } h = k \\ -y_e, \text{ if } e = (h,k) \text{ or } e = (k,h) \end{cases} \tag{2.18}$$

where $y_e = \frac{1}{z_e}$ and $y_h^{sh} = \frac{1}{z_h^{sh}}$.

Finally, in order to underline the difference among loads and generators, we sometimes exploit the following block decomposition of the vector of voltages $\mathbf{u}$

$$\mathbf{u} = \begin{bmatrix} \mathbf{u}_g \\ \mathbf{u}_\ell \end{bmatrix}, \tag{2.19}$$

where $\mathbf{u}_g \in \mathbb{C}^{n_g}$ are the voltages at the microgenerators and $\mathbf{u}_\ell \in \mathbb{C}^{n_\ell}$ are the voltages at the loads. Similarly, it is possible to block decompose currents, powers and admittance matrix as

$$\mathbf{i} = \begin{bmatrix} \mathbf{i}_g \\ \mathbf{i}_\ell \end{bmatrix}, \quad \mathbf{s} = \begin{bmatrix} \mathbf{s}_g \\ \mathbf{s}_\ell \end{bmatrix} = \begin{bmatrix} \mathbf{p}_g \\ \mathbf{p}_\ell \end{bmatrix} + \begin{bmatrix} \mathbf{q}_g \\ \mathbf{q}_\ell \end{bmatrix}, \quad Y = \begin{bmatrix} Y_{gg} & Y_{g\ell} \\ Y_{\ell g} & Y_{\ell\ell} \end{bmatrix}.$$

2.5 Power flow problem and the Power Flow Equations - PFEs

Given the electric grid model, specified through the admittance matrix Y, and the voltage-current relation (2.17), the power flow problem is concerned with describing the operating state of an entire power system. Since an AC circuit in steady state is characterized by complex quantities, in order to completely determine the state of every node in the grid, either one complex or two real quantities per node need to be computed. In particular, if a grid contains n nodes, $2n$ are the quantities to be identified. Historically, the state of a power grid is chosen to be equal to the n buses complex voltages[3] and hereafter we will adopt this convention.

For the purpose of uniquely identify $2n$ quantities, an equivalent number of algebraic

[3] Observe that this is not the only choice since the currents could be chosen as state of the grid.

equations must be specified. By exploiting the definition of apparent power

$$\mathbf{s} = \mathrm{diag}(\mathbf{u})\bar{\mathbf{i}}\,,$$

together with the voltage-current relation (2.17), we obtain n complex algebraic equations which read as

$$\mathbf{s} = \mathrm{diag}(\mathbf{u})\overline{Y}\,\overline{\mathbf{u}}\,. \tag{2.20}$$

These are the so called *Power Flow Equations* (PFEs) which describe the relation between the buses through the branches of a power network and, ultimately, the power balance in an electric grid. It is sometimes convenient to split Eqs.(2.20) into their real and imaginary part obtaining $2n$ real algebraic equations which describe the relation between active and reactive powers and voltages. For completeness, for each $h \in \mathcal{V}$, these are

$$\begin{aligned} p_h =& \nu_h \cos\theta_h \sum_{k\in\mathcal{N}_h^+} \Big(\Re(Y_{hk})\nu_k \cos\theta_k - \Im(Y_{hk})\nu_k \sin\theta_k\Big) + \\ & \nu_h \sin\theta_h \sum_{k\in\mathcal{N}_h^+} \Big(\Im(Y_{hk})\nu_k \cos\theta_k + \Re(Y_{hk})\nu_k \sin\theta_k\Big)\,; \end{aligned} \tag{2.21}$$

$$\begin{aligned} q_h =& -\nu_h \cos\theta_h \sum_{k\in\mathcal{N}_h^+} \Big(\Im(Y_{hk})\nu_k \cos\theta_k + \Re(Y_{hk})\nu_k \sin\theta_k\Big) + \\ & \nu_h \sin\theta_h \sum_{k\in\mathcal{N}_h^+} \Big(\Re(Y_{hk})\nu_k \cos\theta_k - \Im(Y_{hk})\nu_k \sin\theta_k\Big)\,. \end{aligned} \tag{2.22}$$

If the PFEs describe the network relations, the behavior of each bus is specified by the particular model assumed. Classically, a bus can be modeled in three different ways:

1. **slack bus**: also known as *reference bus*, it is a node with a fixed voltage $u_h = cost$, i.e.,

$$\begin{aligned} \nu_h &= \mathrm{cost}\,, \\ \theta_h &= \mathrm{cost}\,. \end{aligned}$$

   Usually, the slack bus is labeled with the subscript 0, i.e., u_0, to highlight its role as reference bus. In the distribution grid the slack bus often coincides with the Point of Common Coupling (PCC), that is the point of connection between the distribution and the transmission grids and identified as u_{PCC}. It must be said that the slack bus is not necessary from a physical perspective but it is necessary from an algebraic one in order to solve the PFEs w.r.t. a reference voltage. For this

reason, in what follows we will never mention or explicitly point out the presence of a slack. However, the reader must be aware of the fact that one node is chosen as reference for solvability purposes.

2. **PV bus**: it is a node whose active power p_h (absorbed or injected) and whose voltage magnitude ν_h are fixed, i.e.,

$$\begin{aligned} p_h &= \Re(u_h \bar{i}_h) = \text{cost}\,, \\ \nu_h &= \text{cost}\,. \end{aligned}$$

3. **PQ bus**: it is a bus whose power (absorbed or injected) $s_h = p_h + \mathtt{j} q_h$ is fixed and independent on its voltage, i.e.,

$$\begin{aligned} p_h &= \Re(u_h \bar{i}_h) = \text{cost}\,, \\ q_h &= \Im(u_h \bar{i}_h) = \text{cost}\,. \end{aligned}$$

The three models above described are used to model loads and generators in a power grid. In particular, as common in power systems, we model generators as PV buses (Bergen, 2009) and loads as PQ buses (Pal, 1992; Machowski, Bialek, and Bumby, 2011)[4]. Moreover, one generator is chosen as slack bus[5].

By dividing the grid into n_g generators and n_ℓ loads ($n = n_g + n_\ell$), the $2n$ PFEs are characterized by:

- $4n$ variables, namely, n active powers, n reactive powers, n voltage magnitudes and n voltage angles;
- $2n_g + 2n_\ell$ fixed quantities, namely, n_g voltage amplitudes, n_g active powers, n_ℓ active and n_ℓ reactive powers;
- $2n$ remaining unknowns.

It must be noticed that, even though the PFEs are well posed, due to their high non linearity and complexity, no closed form solution exists. Indeed, the problem of power flow equations solvability represents an open and consistent problem in power system analysis (Cañizares, 2002; Eremia and Shahidehpour, 2013) which, due to the lack of theoretical

[4]It is worth noticing that our following results can be extended to standard ZIP load model (Bergen, 2009). However, we consider only the case of constant power loads.

[5]The choice of making one generator the slack bus is natural since generators are usually voltage regulated.

insights, is often tackled via numerical methods (Dobson and Lu, 1992; Hiskens, Davy, et al., 2001).

2.6 Reactive Power Flow Equations - RPFEs

When dealing with the control and optimization of the grid (Chapter 5) we consider mainly high voltage transmission grids. For this type of grid it is usually assumed that transmission lines are mainly inductive and then their resistive part is actually neglected. This is formally stated in the following

Assumption 2.6.1 (Highly inductive lines)**.** The transmission lines are mainly inductive and approximated as pure susceptances. Then, the admittance matrix Y collapses into a purely imaginary matrix, i.e.,

$$Y = \mathtt{j}B$$

where the entries of *symmetric susceptance matrix* $B \in \mathbb{R}^{n\times n}$ represents only the imaginary part of the corresponding element in Y.

Thanks to Assumption 2.6.1, Eqs.(2.21)–(2.22) can be simplified as follows

$$p_h = \sum_{k\in\mathcal{N}_h^+} B_{hk}\nu_h\nu_k \sin(\theta_h - \theta_k)\,; \tag{2.23a}$$

$$q_h = -\sum_{k\in\mathcal{N}_h^+} B_{hk}\nu_h\nu_k \cos(\theta_h - \theta_k)\,. \tag{2.23b}$$

where we made use of subtraction trigonometric formula to simplify the expressions.
A practical observation in power systems is that, for high voltage operated networks, it is usually true that the steady state voltage profile is characterized by small angles differences (Machowski et al., 2011) and sufficiently flat voltage amplitudes. In particular, it is possible to observe high correlation between active power and voltage phase differences ($\mathbf{p}\boldsymbol{\theta}$) from one side and between reactive power and voltage magnitudes ($\mathbf{q}\boldsymbol{\nu}$) from the other. Regarding the small voltage angle differences, these are treated as parameters (Thorp, Schulz, and Ilić-Spong, 1986) or even neglected Kaye and Wu (1984). This is formally described by the following

Assumption 2.6.2 (Decoupling assumption)**.** In steady state operating condition, for $\delta \in [0, \pi/2[$, the voltage angle differences are constant and such that

$$|\theta_h - \theta_k| \le \delta\,, \quad \forall (h,k) \in \mathcal{E}\,.$$

Note that under Assumption 2.6.2, from the form of Eq.(2.23b), it is possible to define an *effective susceptance matrix* by embedding the cosine terms into the original line susceptances. Assumption 2.6.2 leads to a decoupled analysis of the power flow equations. In particular, we are interested in the so called *Reactive Power Flow Equations*[6] (RPFEs) which, for each $h \in \mathcal{V}$ read as

$$q_h = -\sum_{k \in \mathcal{N}_h^+} B_{hk} \nu_h \nu_k \,,$$

or, in vector form

$$\mathbf{q} = -\operatorname{diag}(\boldsymbol{\nu}) B \boldsymbol{\nu} \,. \tag{2.24}$$

Equations (2.24) highlight the quadratic relation between the voltage magnitudes and the reactive powers and are the starting point for voltage stability and voltage collapse analysis in power systems (Thorp et al., 1986; Simpson-Porco, Dörfler, and Bullo, 2015).

2.7 Approximated solution of the RPFEs

In this section we introduce a linear (in the load profile) approximation for the solution of the RPFEs which will be extensively used in the last part of this thesis. The approximation is inspired by and almost coincides with the result in Gentile, Simpson-Porco, Dörfler, Zampieri, and Bullo (2014) where a similar approximation is derived for the case of distribution networks. A more complete and generic linear approximation for the solution of the complete coupled power flow equations has been presented in Bolognani and Zampieri (2015), again for the specific case of distribution grid. However, in the more recent Bolognani and Dörfler (2015) the authors generalize the result in Bolognani and Zampieri (2015) by introducing the best linear approximant of the power flow solution in any generic operating point, via implicit linearization of the power flow manifold. In this case the approximation does not rely on any restrictive assumption on the type of grid considered and holds in total generality. Moreover, the authors show how previous linearization can be considered as a special case of their result.
The linear approximation here recalled, has firstly been presented and is strictly related to the result in Todescato, Simpson-Porco, Dörfler, Carli, and Bullo (2015b) and moreover it characterizes only the solution of the RPFEs stated in the previous Section 2.6. We refer the interested reader to the above references for all the technical details on the

[6] These equations represent the reactive/imaginary counterpart of the maybe more familiar active/real power flow equations obtained assuming small angle differences and $\boldsymbol{\nu} = \mathbb{1}$. These lead to an expression of the active powers which is linear in the angle differences and independent from the voltage magnitude. This is also known as the DC power flow model (Andersson, 2008).

other approximations.

Let us recall the RPFEs of Eq.(2.24) which read as

$$\mathbf{q} = -\operatorname{diag}(\boldsymbol{\nu})B\boldsymbol{\nu}$$

Moreover, by considering the block partition introduced in Section 2.4, the susceptance matrix can be accordingly partitioned as

$$B = \begin{bmatrix} B_{gg} & B_{g\ell} \\ B_{\ell g} & B_{\ell\ell} \end{bmatrix}$$

and we refer to $B_{\ell\ell}$ as the *grounded susceptance matrix*. We make the following

Assumption 2.7.1 (Properties of $B_{\ell\ell}$)**.**

1. $-B_{\ell\ell}$ is a non singular symmetric M-matrix[7].
2. the graph associated to $B_{\ell\ell}$ is connected.

Assumption 2.7.1-1. is commonly verified in practice (Thorp et al., 1986) and always verified in the absence of line-charging and positive shunt susceptance, i.e., capacitors, to ground. Assumption 2.7.1-2. can be made without loss of generality, since connected components of the induced graph will be electrically isolated from one another by generator buses.
Now, by splitting the RPFEs into generators and loads we can write

$$\mathbf{q}_g = -\operatorname{diag}(\boldsymbol{\nu}_g)(B_{gg}\boldsymbol{\nu}_g + B_{g\ell}\boldsymbol{\nu}_\ell)\,, \tag{2.25a}$$

$$\mathbf{q}_\ell = -\operatorname{diag}(\boldsymbol{\nu}_\ell)(B_{\ell g}\boldsymbol{\nu}_g + B_{\ell\ell}\boldsymbol{\nu}_\ell)\,. \tag{2.25b}$$

Notice that only (2.25b) represents non trivial equations since, once solved for $\boldsymbol{\nu}_\ell$, Eqs.(2.25a) represent linear equations which can be easily solved for $\mathbf{q}_g$.
It is useful to introduce the concept of *open-circuit profile* which represents a particular non-trivial solution of the RPFEs corresponding to $\mathbf{q}_\ell = 0$.

Definition 2.7.2 (Open-circuit profile)**.** We define the *open-circuit voltage profile* as

$$\boldsymbol{\nu}_\ell^* := -B_{\ell\ell}^{-1}B_{\ell g}\boldsymbol{\nu}_g \tag{2.26}$$

[7] An M-matrix A is a matrix with negative off-diagonal elements and positive diagonal ones which can be expressed in the form $A = s\mathbb{I} - B$, with $b_{ij} \leq 0,\ s > \operatorname{sr}(B)$.

Notice that, thanks to Assumption 2.7.1, $\boldsymbol{\nu}_\ell^* > 0$ is always well defined. We sometimes refer to $\boldsymbol{\nu}_\ell^*$ as the *zero-load profile*.

Thanks to Assumption 2.7.1 and Definition 2.7.2 we can now write the load RPFEs as

$$\mathbf{q}_\ell = -\operatorname{diag}(\boldsymbol{\nu}_\ell) B_{\ell\ell} (\boldsymbol{\nu}_\ell - \boldsymbol{\nu}_\ell^*) \,.$$

highlighting the role of the zero-load profile as particular non trivial solution of the equations corresponding to $\mathbf{q}_\ell = 0$. Moreover, it is possible and convenient to further manipulate the equation as follows

$$\mathbf{q}_\ell = -4 \operatorname{diag}(\boldsymbol{\nu}_\ell)^{-1} \operatorname{diag}(\boldsymbol{\nu}_\ell^*) \left(\frac{1}{4} \operatorname{diag}(\boldsymbol{\nu}_\ell^*) B_{\ell\ell} \operatorname{diag}(\boldsymbol{\nu}_\ell^*) \right) (\operatorname{diag}(\boldsymbol{\nu}_\ell^*)^{-1} \boldsymbol{\nu}_\ell - \mathbb{1}) \,,$$

where we have simply extracted $\boldsymbol{\nu}_\ell^*$ and used the fact that diagonal matrices commute.

We now recall a definition from Todescato et al. (2015b); Simpson-Porco et al. (2015) which will be useful later on.

Definition 2.7.3 (Critical Load/Stiffness Matrix)**.** Given the grounded susceptance matrix $B_{\ell\ell}$ and the open circuit voltage profile $\boldsymbol{\nu}_\ell^*$ defined as in (2.26), we define the *critical load matrix* or *stiffness matrix* as

$$Q_{\text{crit}} := \frac{1}{4} \operatorname{diag}(\boldsymbol{\nu}_\ell^*) B_{\ell\ell} \operatorname{diag}(\boldsymbol{\nu}_\ell^*) \,. \tag{2.27}$$

Moreover, it is convenient in this context to define the *normalized voltage profile* as

$$\boldsymbol{v}_\ell = \operatorname{diag}(\boldsymbol{\nu}_\ell^*)^{-1} \boldsymbol{\nu}_\ell \tag{2.28}$$

Thanks to Definition 2.7.3 and the normalized voltage profile, the RPFEs are equivalent to

$$\mathbf{q}_\ell = -4 \boldsymbol{v}_\ell Q_{\text{crit}} (\boldsymbol{v}_\ell - \mathbb{1}) \,.$$

Now, by assuming $\mathbf{q}_\ell \simeq 0$ it is reasonable to think be the voltage profile $\boldsymbol{v}_\ell \simeq \mathbb{1}$, i.e., $\boldsymbol{\nu}_\ell \simeq \boldsymbol{\nu}_\ell^*$. Then, we can write

$$\mathbf{q}_\ell \simeq -\operatorname{diag}(\boldsymbol{\nu}_\ell^*) B_{\ell\ell} (|\mathbf{u}_\ell| - \boldsymbol{\nu}_\ell^*) \,,$$

and, to first order, the solution of the RPFEs is given by

$$\boldsymbol{v}_\ell \simeq \mathbb{1} - \frac{1}{4} Q_{\text{crit}}^{-1} \mathbf{q}_\ell \,, \tag{2.29}$$

or, equivalently,

$$\boldsymbol{\nu}_\ell \simeq \mathrm{diag}(\boldsymbol{\nu}_\ell^*)\left(\mathbb{1} - \frac{1}{4}Q_{\mathrm{crit}}^{-1}\mathbf{q}_\ell\right). \tag{2.30}$$

Some brief comments are now in order:

1. the matrix Q_{crit} takes its name from the fact that, as can be seen from Eqs.(2.29), since it maps the load profile into the voltage space, it quantifies the stiffness of the network and its capacity of absorbing load demand (Simpson-Porco et al., 2015).

2. the profile $\boldsymbol{v}_\ell$ represents the normalized profile w.r.t. the zero-load solution $\boldsymbol{\nu}_\ell^*$. In general this does not coincides with the *per unit* (p.u.) profile commonly defined in power systems which represents a scalar normalization w.r.t. a reference voltage, usually the slack bus. In Chapter 5 we will point out this difference and show that the normalized profile $\boldsymbol{v}_\ell$ is usually more informative than the p.u. profile. It is then true that, in absence of shunts element and assuming the only generator bus coincides with the PCC then, the profile $\boldsymbol{v}_\ell$ and the p.u. profile coincide.

We will come back deeper on these points in Chapter 5.

2.8 Electrical grid control layers

We have seen that one of the major features that is going to characterize future *smart grids* is the appearance of a large number of generators connected all over the network. This scenario poses a number of nontrivial challenges, together with exciting opportunities. The management of future smart grids requires that many control and optimization algorithms are executed at the same time: generation-demand matching protocols, energy market mechanisms, algorithms for optimal energy use and quality of service, and many others. The complexity of such scenario and the different time scales of these control tasks, suggest that a *layered* architecture should be adopted where different algorithms coexist at different levels and at different time scales. Lower level algorithms, in charge of controlling the specific physical devices, obtain references and commands from higher levels of the architecture. Higher level algorithms command many instances of the lower level ones, based on a simplified model of their behavior and on the aggregated data provided by the underlying layers. One possible application of this structure to the control of a smart microgrid has been depicted in Figure 2.3 (see also De Brabandere, Vanthournout, Driesen, Deconinck, and Belmans (2007) and references therein).

- **Tertiary control** algorithms (residing in the top layer) dispatch active power generation on the basis of economic reasons, measured and predicted aggregate

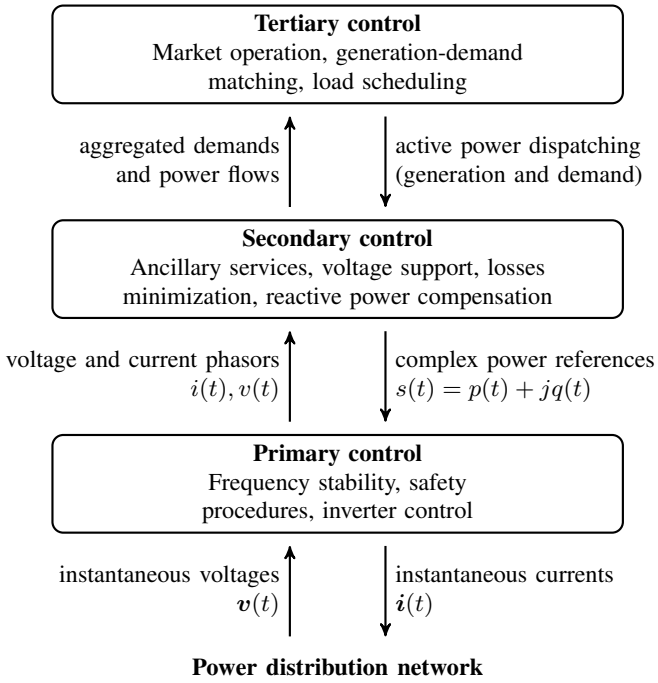

Figure 2.3: A possible layered architecture for the simultaneous execution of different algorithms in a smart microgrid.

demand, and availability of the microgenerators. The reference signals they send come from some optimization process.

- **Secondary control** algorithms, on the other hand, take care of sharing the commanded power references among controllable devices, while satisfying a number of operational constraints (voltage limits, stability, congestion avoidance) and optimizing the network operation (power losses minimization, voltage support, improved power quality). These algorithms require some knowledge of the grid parameters and of the system state.

- **Primary control** algorithms are executed on a local level (at the single inverter), with a few or no communication between the devices, and on a faster time scale; based on the reference signals they receive, they actuate the power converters, ensuring frequency stability, avoiding detrimental interactions, and ensuring safety of operation. The aim of primary control is to quickly stabilize the state of the grid, mostly in terms of voltage frequency and amplitude. The most popular approach is the droop control of the inverters (Chandorkar, Divan, and Adapa, 1993; Li and Kao, 2009; Simpson-Porco, Dörfler, and Bullo, 2012), which does not require direct communications between the agents, but rather exploits voltage frequency and

magnitude as information carrier.

The algorithms proposed in the following naturally reside in the secondary control layer and are characterized by slow time scale, thus justifying the neglect of fast transient behaviors.

3

Distributed Synchronization of Controllable Devices

"Anyone who has never made a mistake has never tried anything new."
A. Einstein

In this second part of the thesis the problem of device synchronization is considered. First we present the problem and some related literature. Then, two possible solutions are proposed. The first is based on a consensus strategy while the second is based on a gradient descent approach. Both the solutions are scalable, robust to delays and have proven exponential convergence rate under mild assumptions.

3.1 Introduction and related work

The proliferation of relatively inexpensive devices capable of communicating, computing, sensing, interacting with the environment and storing information is promising an unprecedented number of novel applications throughout the cooperation of these devices toward a common goal. These applications include swarm robotics, wireless sensor networks, smart energy grids, smart traffic networks, and smart camera networks. These applications also pose new challenges, of which *scalability* is one of the major ones.

Scalability is intended as the ability for an application to continue functioning without any dramatic performance degradation even if the number of devices involved keeps increasing. In particular, an application is scalable if it is not necessary to increase HW resources nor to adopt more complex SW algorithms in each device even if the total number of devices increases.

In this work we address the problem of designing algorithms that are capable to reconstruct the optimal estimate of the state of a device based on noisy relative measurements with respect to its neighbors in a connected network. In particular, we want to design distributed algorithms that allow each device to reconstruct its own state only from exchanging information with its neighbors, regardless of the size of the network. Moreover, these algorithms must be scalable, i.e. their computational complexity, bandwidth and memory requirements should be independent of the network size. Finally, the estimate provided should asymptotically converge to the solution of a centralized optimization problem.

In particular, the problem at hand in this part of the manuscript can be cast as the following unconstrained optimization problem:

$$\min_{x_1,\dots,x_n} \sum_{(i,j)\in\mathcal{E}}^{|\mathcal{E}|} f_{ij}(x_i - x_j)\,, \tag{3.1}$$

where $x_i \in \mathbb{R}^\ell$, $\mathcal{E}$ represents all the pair of nodes for which are available relative measurements and f_{ij} are convex functions. Many problems can be written in this framework such as sensor calibration Bolognani, Del Favero, Schenato, and Varagnolo (2010), camera localization Borra, Lovisari, Carli, Fagnani, and Zampieri (2012); Tron and Vidal (2009) and, in particular, clock synchronization Solis, Borkar, and Kumar (2006); Karp, Elson, Estrin, and Shenker (2003) and sensor localization Barooah and Hespanha (2005); Barooah (2007). For example, in the context of localization from vectorial relative distances in a plane, the cost functions f_{ij} are given by:

$$f_{ij}(x_i - x_j) = \|x_i - x_j - z_{ij}\|^2$$

where $z_{ij} \in \mathbb{R}^\ell$ is the noisy measurement of the relative (vector) distance of node i from node j, i.e.,

$$z_{ij} = x_i - x_j + w_{ij}\,. \tag{3.2}$$

As a consequence, the optimization problem in Eq. (3.1) becomes a distributed least-square problem. For the case of clock synchronization (Barooah and Hespanha, 2005) suppose that the internal clocks of n sensor nodes exhibit time shifts $x_1, x_2, \dots, x_n$ with

respect to a reference clock. A pair of neighboring sensors can measure their clock offsets with some error, by transmitting their local clock times to each other, resulting in a measurement model of the form of Eq.(3.2) (Karp et al., 2003). The goal would then be to estimate the time-shifts between all nodes and a reference clock to synchronize the clocks of all n nodes.

Several scalable distributed solutions to this problem are already available in the literature. In Barooah and Hespanha (2005); Barooah (2007) the authors propose a distributed Jacobi solution based on a synchronous implementation, which was later extended to account for asynchronous communication and packet losses Barooah and Hespanha (2005). The same approach has been independently proposed in Giridhar and Kumar (2006) in the context of distributed time synchronization in wireless sensor networks. Differently, in Bolognani et al. (2010) a broadcast consensus-based algorithm, which is suitable for asynchronous implementation, is proposed but the local estimates do not converge and exhibit an oscillatory behavior around the optimal value. A similar approach has been proposed in Rossi, Frasca, and Fagnani (2012); Ravazzi, Frasca, Ishii, and Tempo (2013) where the local ergodic average of the gossip asynchronous algorithm is proved to converge to the optimal value as $1/k$, where k is the number of iterations. An alternative approach based on the Kaczmarz method for the solution of general linear systems has been suggested in Zouzias and Freris (2013), however a practical asynchronous implementation for distributed localization from relative measurements which satisfies the specific edge and node activation probabilities dictated by the algorithm, is not given, moreover, no robustness analysis in terms of delays is provided.

In the next section we present the problem at hand. For simplicity we cast it in the framework of sensor network localization. However, as above underlined, the problem can be naturally exploited for time synchronization e.g., in power networks.

3.2 Problem formulation

The problem we deal with is that of estimating n variables $x_1, \dots, x_n$ from noisy measurements of the form

$$z_{ij} := x_i - x_j + w_{ij}, \qquad i, j \in \{1, \dots, n\} \ , \tag{3.3}$$

where w_{ij} is zero-mean measurement noise. Although all results in this work apply to general vector-valued variables, for sake of simplicity, here we assume that $x_i \in \mathbb{R}$, $i \in \{1, \dots, n\}$. This estimation problem can be naturally associated with a directed *measurement graph* $\mathcal{G}_m = (\mathcal{V}; \mathcal{E}_m)$. The vertex set $\mathcal{V}$ of the measurement graph consists

of the set of nodes while its edge set $\mathcal{E}_m$ consists of all of the ordered pairs of nodes (i, j) such that a noisy measurement of the form (3.3) between i and j is available to node i. The measurement errors on distinct edges are assumed uncorrelated.

Next we formally state the problem we aim at solving. Let $\mathbf{x} \in \mathbb{R}^n$ be the vector obtained by stacking together all the variables $x_1, \dots, x_n$, i.e., $\mathbf{x} = [x_1, \dots, x_n]^T$, and let $\mathbf{z} \in \mathbb{R}^m$ and $\mathbf{w} \in \mathbb{R}^m$, where $m = |\mathcal{E}_m|$, be the vectors obtained stacking together all the measurements z_{ij} and the noises w_{ij}, respectively. Additionally, let $R_{ij} > 0$ denote the covariance of the zero mean error w_{ij}, i..e, $R_{ij} = \mathbb{E}[w_{ij}^2]$, and let $R \in \mathbb{R}^{m \times m}$ be the diagonal matrix collecting in its diagonal the covariances of the noises w_{ij}, $(i, j) \in \mathcal{E}_m$, i.e., $R = \mathbb{E}[\mathbf{w}\mathbf{w}^T]$. Observe that Eq.(3.3) can be rewritten in a vector form as

$$\mathbf{z} = A\mathbf{x} + \mathbf{w}\,,$$

where A is the incidence matrix associated to $\mathcal{G}_m$. Now, define the set

$$\mathcal{X} := \arg\min_{\mathbf{x} \in \mathbb{R}^n} (\mathbf{z} - A\mathbf{x})^T R^{-1} (\mathbf{z} - A\mathbf{x})\,. \tag{3.4}$$

The goal is to construct an optimal estimate $\mathbf{x}_{\text{opt.}}$ of $\mathbf{x}$ in a least square sense, namely, to compute

$$\mathbf{x}_{\text{opt.}} \in \mathcal{X} \tag{3.5}$$

Assume the measurement graph $\mathcal{G}_m$ to be *weakly connected*, then it is well known (Barooah, 2007) that

$$\mathcal{X} = \left\{ \left(A^T R^{-1} A \right)^{\dagger} A^T R^{-1} \mathbf{z} + \alpha \mathbb{1} \right\}.$$

Moreover let

$$\mathbf{x}_{\text{opt., min}} := \left(A^T R^{-1} A \right)^{\dagger} A^T R^{-1} \mathbf{z}\,,$$

then $\mathbf{x}_{\text{opt., min}}$ is the minimum norm solution of (3.5), i.e.,

$$\mathbf{x}_{\text{opt., min}} = \min_{\mathbf{x}_{\text{opt.}} \in \mathcal{X}} \| \mathbf{x}_{\text{opt.}} \|$$

The matrix $A^T R^{-1} A$ is called in literature the *Weighted Generalized Grounded Laplacian* Barooah (2007).

Remark 3.2.1. Observe that, just with relative measurements, determining the x_i's is only possible up to an additive constant. This ambiguity might be avoided by assuming that a node (say node 1) is used as reference node, i.e., $x_1 = 0$.

The final goal is to compute the value of $\mathbf{x}_{\text{opt.}}$ in a distributed fashion exploiting only

a local exchange of information among neighboring nodes.

3.3 Communication protocol and failures model

Before presenting our solutions, we describe here the communications requirements and, in particular, the communication protocol considered and the model used for the communication non idealities. First of all, we assume the agents can communicate according to an undirected *communication graph* $\mathcal{G}_c = (\mathcal{V}, \mathcal{E}_c)$ where $(i,j) \in \mathcal{E}_c$ either if $(i,j) \in \mathcal{E}_m$ or $(j,i) \in \mathcal{E}_m$. In this context, $\mathcal{N}_i$ denotes the set of neighboring node of node i in the communication graph $\mathcal{G}_c$ rather than in $\mathcal{G}_m$. According to $\mathcal{G}_c$, the agents can communicate exploiting an *asynchronous broadcast* communication protocol where at each iteration only one node communicates its information to its neighbors while they are listening. Since the information coming from neighboring nodes is, by definition, not synchronous, and at each iteration only partial new information, i.e., coming from one neighboring node, is available, we equip each node with a certain amount of additional memory. In particular, for $j \in \mathcal{N}_i$, we denote by $\widehat{x}_j^{(i)}(k)$ the estimate of x_j kept in i's local memory at the end of the k-th iteration. If node j performed its last transmission to node i during h-th iteration, $h \leq k$, then $\widehat{x}_j^{(i)}(k) = \widehat{x}_j(h)$. Observe that $\widehat{x}_i^{(i)} \equiv \widehat{x}_i$ since the copy that node i would have of himself is always up-to-date and equal to its current estimate of x_i.
In next sections, we analyze the convergence properties and the robustness to delays and packet losses of the proposed solutions for two different scenarios which are formally described by the following definitions.

Definition 3.3.1 (Randomly persistent communicating network)**.** A network of n nodes is said to be a *randomly persistent communicating network* if there exists a n-upla $(\gamma_1, \ldots, \gamma_n)$ such that $\gamma_i > 0$, for all $i \in \{1, \ldots, n\}$, and $\sum_{i=1}^n \gamma_i = 1$, and such that, for all $k \in \mathbb{N}$,

$$\mathbb{P}\left[\text{the transmitting node at iteration } k \text{ is node } i\right] = \gamma_i.$$

Definition 3.3.2 (Uniformly persistent communicating network)**.** A network of n nodes is said to be a *uniformly persistent communicating network* if there exists a positive integer number τ such that, for all $k \in \mathbb{N}$, each node transmits the value of its estimate to its neighbors at least once within the time interval $[k, k+\tau)$.

Finally, to show resiliency of our proposed solutions to delays and packet losses in the communication channel we need to suitably model them. In particular, we show convergence of the algorithms provided that the network is *uniformly persistent*

communicating and the transmission delays and the frequencies of communication failures satisfy mild conditions which we formally describe next.

Assumption 3.3.3 (Bounded packet losses)**.** There exists a positive integer L such that the number of consecutive communication failures between every pair of neighboring nodes in the communication graph $\mathcal{G}_c$ is less than L.

Assumption 3.3.4 (Bounded delay)**.** Assume node i broadcasts its estimate to its neighbors at the beginning of iteration k, and, assume that, the communication link (i, j) does not fail. Then, there exists a positive integer D such that the information $\widehat{x}_i(k)$ is used by node j to perform its local update not later than iteration $k + D$.

Loosely speaking Assumption 3.3.3 implies that there can be no more than L consecutive packet losses between any pair of nodes i, j belonging to the communication graph. Differently, Assumption 3.3.4 consider the scenario where the received packets are not used instantaneously, but are subject to some delay no greater than D iterations.

In the next section we present our first solution which is an algorithm based on time varying consensus algorithm with memory.

3.4 An asynchronous consensus-based algorithm

Here we present the first algorithm proposed to solve the localization/synchronization problem of Section 3.2. The algorithm, which relies on linear consensus with memory, is based on the asynchronous broadcast communication protocol described in Section 3.3. Hereafter we refer to this algorithm as the *asynchronous Consensus-Like* (a-CL) algorithm. As already pointed out, since the actual value of neighboring estimates are not available at each iteration, we assume that each node stores in its local memory a copy of the neighbors' variables recorded from the last communication received. In particular, for $j \in \mathcal{N}_i$, $\widehat{x}_j^{(i)}(k)$ denotes the estimate of x_j kept in i's local memory at the end of the k-th iteration. We assume that before running the a-CL algorithm, the nodes exchange with their neighbors their relative measurements as well as the associated covariances. So every node has access to the measurements on the edges that are incident to it, whether the edge is directed to or away from it. Each node uses the measurements obtained initially for all future computations. The a-CL algorithm is formally described in Algorithm 1 where we have that

$$b_i := \epsilon \sum_{(i,j)\in\mathcal{E}_m} R_{ij}^{-1} z_{ij} - \epsilon \sum_{(j,i)\in\mathcal{E}_m} R_{ji}^{-1} z_{ji}\,,$$

and

$$p_{ij} := \begin{cases} \epsilon(R_{ij}^{-1} + R_{ji}^{-1}) & \text{if } (i,j) \in \mathcal{E}_m \text{ and } (j,i) \in \mathcal{E}_m \\ \epsilon R_{ij}^{-1} & \text{if } (i,j) \in \mathcal{E}_m \text{ and } (j,i) \notin \mathcal{E}_m \\ \epsilon R_{ji}^{-1} & \text{if } (j,i) \in \mathcal{E}_m \text{ and } (i,j) \notin \mathcal{E}_m \end{cases}$$

and finally,

$$p_{ii} := 1 - \sum_{j \in \mathcal{N}_i} p_{ij} \,,$$

being ϵ a positive constant *a-priori* assigned to the nodes.

Algorithm 1 The a-CL algorithm.

Require: For $i \in \{1, \dots, n\}$, node i stores in memory the measurements z_{ij}, z_{ji} and the covariances R_{ij}, R_{ji} for all $j \in \mathcal{N}_i$. Moreover node i stores in memory also the estimate $\widehat{x}_i$ of x_i and, for $j \in \mathcal{N}_i$ an estimate $\widehat{x}_j^{(i)}$ of $\widehat{x}_j$.

% `Initialization`

$\forall i \in \mathcal{V}$: initializes the estimate $\widehat{x}_i$ and the variables $\widehat{x}_j^{(i)}$, $j \in \mathcal{N}_i$, to arbitrary values.

for $k \in \mathbb{N}$ **do**

% `Transmission`

node i (randomly picked) wakes up and sends $\widehat{x}_i(k)$ to node j, $j \in \mathcal{N}_i$.

% `Update`

for $j \in \mathcal{N}_i$, node j **do**

1. sets $\widehat{x}_i^{(j)}(k+1) = \widehat{x}_i(k)$;
2. updates $\hat{x}_j$ as

$$\widehat{x}_j(k+1) := p_{jj}\widehat{x}_j(k) + \sum_{h \in \mathcal{N}_j} p_{jh}\widehat{x}_h^{(j)}(k+1) + b_j \,, \tag{3.6}$$

end for

end for

It should be noticed that Algorithm 1 has been described assuming that the communication channels are reliable, i.e, no packet losses occur, and that the communication delays are negligible, i.e., when node i perform a transmission, the estimate $\widehat{x}_i$ is instantaneously used by its neighbors. We will come back on these non-idealities in Section 3.6.

Next, we rewrite the updating step of the a-CL in a more compact way. Observe preliminarily that, under the assumption of reliable communications over the network, the broadcast protocol lets only two information about the estimate of x_i, $i \in \mathcal{V}$, to flow

through the network: specifically, $\widehat{x}_i(k)$, that is the actual value of the estimate $\widehat{x}_i$ at iteration k, and $\widehat{x}_i(t'_i(k))$, being $t'_i(k)$ the iteration during which node i has performed its last transmission up to iteration k of a-CL (that is, $\widehat{x}_i(t'_i(k))$ is the value of $\widehat{x}_i$ at its last communication round). Notice that, for $j \in \mathcal{N}_i$, $\widehat{x}_i^{(j)}(t'') = \widehat{x}_i(t'_i(k))$ for all t'' such that $t'_i(k) < t'' \leq k$.
Now let us define $x'_i(k) = \widehat{x}_i(k)$ and $x''_i(k) = \widehat{x}_i(t'_i(k))$ and, accordingly, let $\mathbf{x}'(k) = [x'_1(k), \ldots, x'_n(k)]^T$ and $\mathbf{x}''(k) = [x''_1(k), \ldots, x''_n(k)]^T$. Moreover let $Q_i \in \mathbb{R}^{2n \times 2n}$ be defined as

$$Q_i := \begin{bmatrix} Q_{11}^{(i)} & Q_{12}^{(i)} \\ Q_{21}^{(i)} & Q_{22}^{(i)} \end{bmatrix}, \tag{3.7}$$

where

$$\begin{aligned} Q_{11}^{(i)} &:= \sum_{h \notin \mathcal{N}_i} \mathbb{1}_h \mathbb{1}_h^T + \sum_{j \in \mathcal{N}_i} \left(p_{jj} \mathbb{1}_j \mathbb{1}_j^T + p_{ji} \mathbb{1}_j \mathbb{1}_i^T \right), \\ Q_{12}^{(i)} &:= \sum_{j \in \mathcal{N}_i} \mathbb{1}_j \left(\sum_{h \in \mathcal{N}_j / i} p_{jh} \mathbb{1}_h^T \right), \\ Q_{21}^{(i)} &:= \mathbb{1}_i \mathbb{1}_i^T, \\ Q_{22}^{(i)} &:= \mathbb{I} - \mathbb{1}_i \mathbb{1}_i^T, \end{aligned}$$

Observe that, for $i \in \{1, \ldots, n\}$, Q_i is a $2n$-dimensional stochastic matrix. Finally let

$$\boldsymbol{\beta}_i = \begin{bmatrix} \sum_{j \in \mathcal{N}_i} \mathbb{1}_j^T \mathbf{b} \\ \mathbb{0} \end{bmatrix},$$

where $\mathbf{b} = [b_1, \ldots, b_m]^T$. Assume, w.l.o.g., that node i is the node performing the transmission during the $(k+1)$-th iteration of the a-CL. Hence the updating step of a-CL can be written in vector form as

$$\begin{bmatrix} \mathbf{x}'(k+1) \\ \mathbf{x}''(k+1) \end{bmatrix} = Q_i \begin{bmatrix} \mathbf{x}'(k) \\ \mathbf{x}''(k) \end{bmatrix} + \boldsymbol{\beta}_i \,. \tag{3.8}$$

Now let us introduce the auxiliary variable

$$\boldsymbol{\xi}(k) = \begin{bmatrix} \mathbf{x}'(k) \\ \mathbf{x}''(k) \end{bmatrix} - \begin{bmatrix} \mathbf{x}_{\text{opt., min}} \\ \mathbf{x}_{\text{opt., min}} \end{bmatrix}.$$

By exploiting the fact that, for $i \in \{1, \dots, n\}$,

$$\begin{bmatrix} \mathbf{x}_{\text{opt., min}} \\ \mathbf{x}_{\text{opt., min}} \end{bmatrix} = Q_i \begin{bmatrix} \mathbf{x}_{\text{opt., min}} \\ \mathbf{x}_{\text{opt., min}} \end{bmatrix} + \boldsymbol{\beta}_i \,, \tag{3.9}$$

we have that the variable $\boldsymbol{\xi}$ satisfies the following $2n$-dimensional recursive equation

$$\boldsymbol{\xi}(k+1) = Q_i \boldsymbol{\xi}(k) \,. \tag{3.10}$$

Observe that $\widehat{\mathbf{x}}(k) \to \mathbf{x}_{\text{opt., min}} + \alpha \mathbb{1}$ if and only if $\boldsymbol{\xi}(k) \to \alpha \mathbb{1}$. Moreover, since Q_i is a stochastic matrix for any $i \in \{1, \dots, n\}$, we have that (3.10) represents a $2n$-dimensional time-varying consensus algorithm. Indeed in general, Eq.(3.10) can be rewritten as

$$\boldsymbol{\xi}(k+1) = Q_{\sigma(k)} \boldsymbol{\xi}(k) \,,$$

where $\sigma(k) \in \{1, \dots, n\}$ denotes the agent performing the transmission action during iteration k.
In next sections, we analyze the convergence properties and the robustness to delays and packet losses of the a-CL algorithm by studying system (3.10) resorting to the mathematical tools developed in the literature of the consensus algorithms.

3.5 Performance analysis of a-CL algorithm under randomly persistent communications

The following result, whose proof can be found in Appendix A.1, characterizes the convergence properties of the a-CL when the network is randomly persistent communicating (see Definition 3.3.1 of Section 3.3 for details).

Proposition 3.5.1. *Consider a* randomly persistent communicating network *of n nodes running the a-CL algorithm over a weakly connected measurement graph $\mathcal{G}_m$. Let $\mathcal{X}$ be defined as in (3.4). Let ϵ be such that $0 < \epsilon < 1/(2d_{\max} R_{\min}^{-1})$ where $d_{\max} := \max\{|\mathcal{N}_i|,\ i \in \mathcal{V}\}$ and $R_{\min} := \min\{R_{ij},\ (i,j) \in \mathcal{E}_m\}$. Moreover let $\widehat{x}_i$, $i \in \{1, \dots, n\}$, $\widehat{x}_j^{(i)}$, $j \in \mathcal{N}_i$, be initialized to any real number. Then the following facts hold true*

1. *the evolution $k \to \widehat{\mathbf{x}}(k)$ converges almost surely to an optimal solution $\mathbf{x}_{\text{opt.}} \in \mathcal{X}$, i.e., there exists $\alpha \in \mathbb{R}$ such that*

$$\mathbb{P}\left[\lim_{k \to \infty} \widehat{\mathbf{x}}(k) = \mathbf{x}_{\text{opt., min}} + \alpha \mathbb{1}\right] = 1 \,.$$

2. *the evolution* $k \to \widehat{\mathbf{x}}(k)$ *is exponentially convergent in mean-square sense, i.e., there exist* $c > 0$ *and* $0 \leq \rho < 1$ *such that*

$$\lim_{k\to\infty} \mathbb{E}\left[\|\widehat{\mathbf{x}}(k) - (\mathbf{x}_{\text{opt., min}} + \alpha\mathbb{1})\|^2\right] \leq c\rho^k \mathbb{E}\left[\|\widehat{\mathbf{x}}(0) - (\mathbf{x}_{\text{opt., min}} + \alpha\mathbb{1})\|^2\right] .$$

3.6 Robustness properties of the a-CL algorithm with respect to packet losses and delays

In Section 3.4 we have introduced the a-CL algorithm assuming that the communication channels are reliable, i.e., no packet losses occur, and that the transmission delays are negligible. In this section we relax these assumptions and we show that the a-CL algorithm still converges provided that the network is uniformly persistent communicating and the transmission delays and the frequencies of communication failures satisfy the mild boundedness conditions introduced in Section 3.3.

Clearly, in this more realistic scenario, it turns out that the implementation of the a-CL is slightly different from the description provided in Section 3.4. Specifically, consider the k-th iteration of the a-CL algorithm and, without loss of generality, assume node i is the transmitting node during this iteration. Due to the presence of packet losses and delays, it might happen that the set of updating nodes is, in general, different from the set $\mathcal{N}_i$. In fact, for $j \in \mathcal{N}_i$, node j does not perform any update since the packet $\widehat{x}_i(k)$ from node i is lost or simply because the update is delayed. Moreover there might be a node $h \notin \mathcal{N}_i$ which, during iteration k, decides to perform an update since it received a packet $\widehat{x}_s$, $s \in \mathcal{N}_h$, within the last D iterations. This scenario can be formally represented by the set of nodes $\mathcal{V}'(k) \subseteq \mathcal{V}$ which decide to perform an update at iteration k. Then, Eq.(3.6) can be rewritten as

$$\widehat{x}_j(k+1) := p_{jj}\widehat{x}_j(k) + \sum_{h\in\mathcal{N}_j} p_{jh}\widehat{x}_h(k'_h) + b_j, \tag{3.11}$$

for all $j \in \mathcal{V}'(k)$, where $k - (\tau L + D) \leq k'_h \leq k$, i.e. loosely speaking when an update is performed, the local estimate of the neighboring nodes cannot be older than $\tau L + D$ iterations[1]. Indeed, if $L = D = 0$, then we recover the standard a-CL algorithm where $\mathcal{V}'(k) = \mathcal{N}_i$.

The following result, whose proof can be found in Appendix A.2, characterizes the

[1]Recall we are assuming the network is uniformly persistent communicating, namely, for all $k \in \mathbb{N}$, each node performs at least one transmission within the time interval $[k, k + \tau)$.

convergence properties of the a-CL in presence of delays, packet losses and when the network is uniformly persistent communicating.

Proposition 3.6.1. *Consider a* uniformly persistent communicating network *of n nodes running the a-CL algorithm over a weakly connected measurement graph $\mathcal{G}_m$. Let Assumptions 3.3.3 and 3.3.4 be satisfied. Let ϵ be such that $0 < \epsilon < 1/(2d_{max}R_{min}^{-1})$ where $d_{\max} := \{|\mathcal{N}_i|,\ i \in \mathcal{V}\}$ and $R_{\min} := \min\{R_{ij},\ (i.j) \in \mathcal{E}_m\}$. Moreover let $\widehat{x}_i$, $i \in \{1, \dots, n\}$, $\widehat{x}_j^{(i)}$, $j \in \mathcal{N}_i$, be initialized to any real number. Then the following facts hold true*

1. *the evolution $k \to \widehat{\mathbf{x}}(k)$ asymptotically converges to an optimal estimate $\mathbf{x}_{\text{opt.}} \in \mathcal{X}$, i.e., there exists $\alpha \in \mathbb{R}$ such that*

$$\lim_{k\to\infty} \widehat{\mathbf{x}}(k) = \mathbf{x}_{\text{opt., min}} + \alpha\mathbb{1}\,;$$

2. *the convergence is exponential, namely, there exists $c > 0$ and $0 \leq \rho < 1$ such that*

$$\|\widehat{\mathbf{x}}(k) - (\mathbf{x}_{\text{opt., min}} + \alpha\mathbb{1})\| \leq c\rho^k \|\widehat{\mathbf{x}}(0) - (\mathbf{x}_{\text{opt., min}} + \alpha\mathbb{1})\|\,.$$

In the following we present our second solution based on a gradient descent approach.

3.7 An asynchronous gradient-based algorithm

Differently from Section 3.4 where a consensus based solution has been presented, in this section we propose a gradient-based solution to solve for the localization/synchronization problem given by Eq.(3.5). One fundamental difference from the a-CL algorithm regards the measurements at our disposal. In this case, rather than assuming that every node i can independently measure its own relative quantities z_{ij}, we assume that, if two nodes i and j are neighbors then, $z_{ij} = -z_{ji}$. That is the measurement graph $\mathcal{G}_m$ is undirected rather than directed and coincides with the communication graph $\mathcal{G}_c$ as defined in Section 3.2.

Again, we are interested in implementing a solution which is *distributed* as opposed to centralized and *asynchronous*, as opposed to synchronous in the communication protocol. In what follows we introduce a distributed algorithm which is based on a standard gradient descent strategy and which employs an *asynchronous broadcast* communication protocol as described in Section 3.3. We refer to this algorithm as the *asynchronous gradient-based localization* algorithm (denoted hereafter as a-GL algorithm). We assume that every node has access to the measurements on the edges that are incident to it, as well as the associated covariances. Additionally, in order to deal with possible delay

Algorithm 2 The a-GL algorithm.

Require: For $i \in \{1, \dots, n\}$, node i stores in memory the measurements z_{ij} and the covariances R_{ij} for all $j \in \mathcal{N}_i$. Moreover node i stores in memory also the estimate $\widehat{x}_i$ of x_i and, for $j \in \mathcal{N}_i$ an estimate $\widehat{x}_j^{(i)}$ of $\widehat{x}_j$.

% `Initialization`

$\forall i \in \mathcal{V}$: initializes the estimate $\widehat{x}_i$ and the variables $\widehat{x}_j^{(i)}$, $j \in \mathcal{N}_i$, to arbitrary values.

for $k \in \mathbb{N}$ **do**

% `Update`

node i (randomly picked) wakes up and updates its estimate $\widehat{x}_i$ as

$$\widehat{x}_i(k+1) = \widehat{x}_i(k) - \epsilon_i \sum_{j \in \mathcal{N}_i} \frac{\widehat{x}_i(k) - \widehat{x}_j^{(i)}(k) - z_{ij}}{R_{ij}} \,,$$

where ϵ_i is a suitable positive real number;

% `Transmission`

node i sends $\widehat{x}_i(k+1)$ to node j, $j \in \mathcal{N}_i$.

for $j \in \mathcal{N}_i$, node j **do**

updates its memory as $\widehat{x}_i^{(j)}(k+1) = \widehat{x}_i(k+1)$;

end for

end for

and/or packet losses, situation that we will analyze later in Sections 3.8–3.9, we assume that node i, $i \in \mathcal{V}$, stores in memory an estimate $\widehat{x}_i$ of x_i and, for $j \in \mathcal{N}_i$, an estimate $\widehat{x}_j^{(i)}$ of x_j. The a-GL algorithm is formally described in Algorithm 2.

Some explanations are now in order. Observe that the quantity

$$\sum_{j \in \mathcal{N}_i} \left(\widehat{x}_i - \widehat{x}_j^{(i)} - z_{ij} \right) / R_{ij}$$

represents the gradient computed with the respect to $\widehat{x}_i$ of the function

$$J_i = \frac{1}{2} \sum_{j \in \mathcal{N}_i} \frac{\left(\widehat{x}_i - \widehat{x}_j^{(i)} - z_{ij} \right)^2}{R_{ij}} \,.$$

Basically, node i updates the value of $\widehat{x}_i$ moving along a descent direction of the function

J_i. Notice that J_i does not increase if

$$0 < \epsilon_i \leq \left(\sum_{j \in \mathcal{N}_i} \frac{1}{R_{ij}} \right)^{-1},$$

and, in particular, if $\epsilon_i = \left(\sum_{j \in \mathcal{N}_i} 1/R_{ij} \right)^{-1}$ then the minimum of J_i is attained. Indeed in this case we have that

$$\widehat{x}_i = \left(\sum_{j \in \mathcal{N}_i} \frac{1}{R_{ij}} \right)^{-1} \left(\sum_{j \in \mathcal{N}_i} \frac{\widehat{x}_j^{(i)} + z_{ij}}{R_{ij}} \right), \tag{3.12}$$

which corresponds to the unique solution of the problem

$$\arg\min_{\widehat{x}_i} \frac{1}{2} \sum_{j \in \mathcal{N}_i} \frac{\left(\widehat{x}_i - \widehat{x}_j^{(i)} - z_{ij} \right)^2}{R_{ij}}.$$

Next we provide a convenient vector form description of the a-GL algorithm. Observe that, since for now, we are assuming that neither communication delays nor packet losses occur, $\widehat{x}_j^{(i)}(k) = \widehat{x}_j(k)$, $j \in \mathcal{N}_i$. Then we can rewrite (3.12) as

$$\widehat{x}_i(k+1) = \left(1 - \epsilon_i \sum_{j \in \mathcal{N}_i} \frac{1}{R_{ij}} \right) \widehat{x}_i(k) + \epsilon_i \sum_{j \in \mathcal{N}_i} \frac{\widehat{x}_j(k) + z_{ij}}{R_{ij}},$$

while $\widehat{x}_h(k+1) = \widehat{x}_h(k)$, $h \neq i$. Let us rewrite the above equation as

$$\widehat{x}_i(k+1) = p_{ii} \widehat{x}_i(k) + \sum_{j \in \mathcal{N}_i} p_{ij} \widehat{x}_j(k) + b_i, \tag{3.13}$$

where

$$p_{ij} = \begin{cases} 1 - \epsilon_i \sum_{j \in \mathcal{N}_i} 1/R_{ij} & \text{if } j = i, \\ \epsilon_i / R_{ij} & \text{if } j \neq i, j \in \mathcal{N}_i, \\ 0 & \text{otherwise,} \end{cases}$$

and where

$$b_i = \epsilon_i \sum_{j \in \mathcal{N}_i} \frac{z_{ij}}{R_{ij}}.$$

Let $P \in \mathbb{R}^{n \times n}$ be the matrix defined by the weights p_{ij} above introduced. Then the updating step at time k can be written in vector form as

$$\widehat{\mathbf{x}}(k+1) = \left(\mathbb{I} + \mathbb{1}_i \mathbb{1}_i^T (P - \mathbb{I}) \right) \widehat{\mathbf{x}}(k) + \boldsymbol{\beta}_i, \tag{3.14}$$

where $\widehat{\mathbf{x}}(k) = [\widehat{x}_1(k), \dots, \widehat{x}_n(k)]^T$ and where the vector $\boldsymbol{\beta}_i \in \mathbb{R}^n$ is defined as $\boldsymbol{\beta}_i = b_i \mathbb{1}_i$. Next we show that, by a proper change of variable, equation (3.14) can be rewritten as the iteration of a time-varying linear consensus algorithm. To do so, let

$$Q_i = \mathbb{I} + \mathbb{1}_i \mathbb{1}_i^T (P - \mathbb{I}),$$

and observe that, if

$$0 < \epsilon_i \leq \left(\sum_{j \in \mathcal{N}_i} \frac{1}{R_{ij}} \right)^{-1}, \qquad \forall\, i \in \mathcal{V},$$

then the matrix Q_i is a stochastic matrix for all $i \in \mathcal{V}$. Indeed all the elements of Q_i are nonnegative and, moreover, one can see that $Q_i \mathbb{1} = \mathbb{1}$. Now let us introduce the auxiliary variable

$$\boldsymbol{\xi}(k) = \widehat{\mathbf{x}}(k) - \mathbf{x}_{\text{opt., min}}.$$

By exploiting the fact that, for $i \in \{1, \dots, n\}$,

$$\mathbf{x}_{\text{opt., min}} = Q_i \mathbf{x}_{\text{opt., min}} + \boldsymbol{\beta}_i, \tag{3.15}$$

we have that the variable $\boldsymbol{\xi}$ satisfies the following recursive equation

$$\boldsymbol{\xi}(k+1) = Q_i \boldsymbol{\xi}(k). \tag{3.16}$$

Observe that $\widehat{\mathbf{x}}(k) \to \mathbf{x}_{\text{opt., min}} + \alpha \mathbb{1}$ if and only if $\boldsymbol{\xi}(h) \to \alpha \mathbb{1}$. Moreover, since Q_i is a stochastic matrix for any $i \in \{1, \dots, n\}$, we have that (3.16) represents a n-dimensional time-varying consensus algorithm.

Remark 3.7.1. It is worth to stress that the a-GL algorithm is a modified version of the algorithm proposed in Giridhar and Kumar (2006). The main differences are related to the communication protocol. Specifically, in Giridhar and Kumar (2006) when a node is activated, say i, firstly it interrogates its neighbors to obtain their estimates $\{\widehat{x}_j\}_{j \in \mathcal{N}_i}$; secondly, based on the information received, it updates its own estimate $\widehat{x}_i$. This implies that during this iteration of the algorithm there are $|\mathcal{N}_i| + 1$ transmitted packets (one packet is related to the broadcast request by node i while the other $|\mathcal{N}_i|$ packets are related to $\{\widehat{x}_j\}_{j \in \mathcal{N}_i}$ responses). Instead in the a-GL algorithm, there is just one packet broadcast during each iteration of the algorithm. This leads to a lighter, faster and energy-saving solution. Additionally in Giridhar and Kumar (2006) there is no robustness analysis against packet losses.

In next sections, we analyze the convergence properties and the robustness to delays

and packet losses of the a-GL algorithm.

3.8 Performance analysis in the randomly persistent communicating scenario

The following result characterizes the convergence properties of the a-GL algorithm when the network is a randomly persistent communicating network. The proof is reported in Appendix A.3.

Proposition 3.8.1. *Consider a* randomly persistent communicating network *of n nodes running the a-GL algorithm over a* connected measurement graph $\mathcal{G}_m$. *Assume the weights ϵ_i are such that*

$$0 < \epsilon_i \leq \left(\sum_{j \in \mathcal{N}_i} \frac{1}{R_{ij}} \right)^{-1}, \qquad \forall\, i \in \mathcal{V},$$

and assume that $\widehat{x}_i$, $i \in \mathcal{V}$, $\widehat{x}_j^{(i)}$, $j \in \mathcal{N}_i$, be initialized to any real number. Then the following facts hold true

1. *the evolution $k \to \widehat{\mathbf{x}}(k)$ converges almost surely to an optimal solution $\mathbf{x}_{\text{opt.}} \in \mathcal{X}$, i.e., there exists $\alpha \in \mathbb{R}$ such that*

$$\mathbb{P}\left[\lim_{k \to \infty} \widehat{\mathbf{x}}(k) = \mathbf{x}_{\text{opt., min}} + \alpha \mathbb{1} \right] = 1\,,$$

2. *the evolution $k \to \widehat{\mathbf{x}}(k)$ is exponentially convergent in mean-square sense, i.e., there exist $c > 0$ and $0 \leq \rho < 1$ such that*

$$\lim_{k \to \infty} \mathbb{E}\left[\|\widehat{\mathbf{x}}(k) - (\mathbf{x}_{\text{opt., min}} + \alpha \mathbb{1})\|^2 \right] \leq c \rho^h \mathbb{E}\left[\|\widehat{\mathbf{x}}(0) - (\mathbf{x}_{\text{opt., min}} + \alpha \mathbb{1})\|^2 \right].$$

3.9 Robustness to packet losses and delays in the uniformly persistent communicating scenario

In Section 3.7 we have introduced the a-GL algorithm under the assumptions that

- the communication channels are reliable, i.e., no packet losses occur; and
- the transmission delays are negligible.

In this section we consider a more realistic scenario where the above two assumptions are relaxed. We are still able to prove that the a-GL algorithm converges to an optimal solution provided that the network is uniformly persistent communicating and the transmission delays and the frequencies of communication failures satisfy the mild boundedness conditions introduces in Section 3.3.
Under Assumptions 3.3.3–3.3.4, it is in general true that $\widehat{x}_i^{(j)}(k) = \widehat{x}_j(k'_{ij})$ for some k'_{ij} such that $k - (\tau L + D) \leq k'_{ij} \leq k$. It turns out that the equation update (3.13) is, in general, modified as

$$\widehat{x}_i(k+1) = p_{ii}\widehat{x}_i(k) + \sum_{j \in \mathcal{N}_i} p_{ij}\widehat{x}_j(k'_{ij}) + b_i \,.$$

The following proposition, whose proof is reported in Appendix A.4, characterizes the convergence proprieties in presence of bounded packet losses and bounded delays.

Proposition 3.9.1. *Consider a* uniformly persistent communicating network *of n nodes running the a-GL algorithm over a* connected measurement graph $\mathcal{G}_m$. *Let Assumptions 3.3.3 and 3.3.4 be satisfied. Assume the weights ϵ_i are such that*

$$0 < \epsilon_i < \left(\sum_{j \in \mathcal{N}_i} \frac{1}{R_{ij}} \right)^{-1}, \qquad \forall\, i \in \mathcal{V},$$

and assume that $\widehat{x}_i$, $i \in \mathcal{V}$, $\widehat{x}_j^{(i)}$, $j \in \mathcal{N}_i$, be initialized to any real number. Then the following facts hold true

1. *the evolution $k \to \widehat{\mathbf{x}}(k)$ asymptotically converges to an optimal estimate $\mathbf{x}_{\text{opt.}} \in \mathcal{X}$, i.e., there exists $\alpha \in \mathbb{R}$ such that*

$$\lim_{k \to \infty} \widehat{\mathbf{x}}(k) = \mathbf{x}_{\text{opt.,\ min}} + \alpha \mathbb{1} \,;$$

2. *the convergence is exponential, namely, there exists $c > 0$ and $0 \leq \rho < 1$ such that*

$$\| \widehat{\mathbf{x}}(k) - (\mathbf{x}_{\text{opt.,\ min}} + \alpha \mathbb{1}) \| \leq c\rho^h \| \widehat{\mathbf{x}}(0) - (\mathbf{x}_{\text{opt.,\ min}} + \alpha \mathbb{1}) \|.$$

Observe that in Proposition 3.9.1 it is assumed that ϵ_i is strictly smaller than $\left(\sum_{j \in \mathcal{N}_i} \frac{1}{R_{ij}}\right)^{-1}$, while the result in Proposition 3.8.1 holds true also if the equality is satisfied. Next we provide a example showing that, if $\epsilon_i = \left(\sum_{j \in \mathcal{N}_i} \frac{1}{R_{ij}}\right)^{-1}$, then the optimal solution is not reached in presence of constant positive delays.

Example 3.9.2. Consider a network with $n = 2$ agents and the following cost function $J = \frac{1}{2R}(\widehat{x}_1 - \widehat{x}_2 - z)^2$, where z is the noisy measurement. Setting $\epsilon = R$ we have the following two update rules:

$$\begin{cases} \hat{x}_1(h) & = \hat{x}_2^{(1)}(h) + z \\ \hat{x}_2(h) & = \hat{x}_1^{(2)}(h) - z \end{cases} \tag{3.17}$$

where

$$\begin{cases} \hat{x}_1^{(2)}(h) & = \hat{x}_1(h-2) \\ \hat{x}_2^{(1)}(h) & = \hat{x}_2(h-2) \end{cases}$$

which means that the state of the other agent is known with one step delay. During the odd iterations node 1 makes the update, while during even iterations node 2 updates. Starting from initial conditions equal to zero and assuming that at the first iteration the agent 1 consider the state of the agent 2 equal to zero, following the update rule (3.17) we get

h	1	2	3	4	5	6	7	8	9	...
$\hat{x}_1$	z	z	z	z	0	0	z	z	z	...
$\hat{x}_2$	0	$-z$	$-z$	0	0	0	0	$-z$	$-z$	...

It can be seen from the above table that there is a three steps oscillatory behavior. The value of the cost function J keeps jumping from 0 to $\frac{z^2}{2R}$ so there is not convergence to the optimal set of solutions.

In what follows we provide some numerical comparison of the two solutions proposed.

3.10 Simulations

In this section we provide some numerical simulations to test the proposed distributed algorithms and to compare them with respect to existing solution in the literature. As simulation setup we consider a sensor network, rather than a smart grid, consisting of random geometric graph generated by choosing $n = 100$ nodes randomly placed in the interval $[0, 1]$. Two nodes are connected and take measurements if they are sufficiently close, i.e more specifically, both measurements z_{ij} and z_{ji} are available provided that $|x_i - x_j| \leq 0.15$. This choice resulted in networks with an average number of neighbours per node of about 7. Finally, every measurement was corrupted by Gaussian noise with covariance $\sigma^2 = 10^{-4}$.

We provide a numerical comparison with some well known algorithms proposed in literature which, for the sake of the completeness, we briefly recall.

The first algorithm that we considered is the Randomized Extended Kaczmarz, hereafter called **REK algorithm**, presented in Zouzias and Freris (2013), consisting of two different update steps. The first step is an orthogonal projection of the noisy measurements onto the column space of the incidence matrix A in order to bound the measurements error. The second step is similar to the standard Kaczmarz update. Since a distributed implementation is not formally presented in Zouzias and Freris (2013), we propose the following. More specifically, let $s \in \mathbb{R}^m$ be the current projection of the noisy measurements onto the column space of A. Similarly as above, we denote with a little abuse of notation the e-th entry of s with the corresponding edge, i.e. $s_e = s_{ij}$. Then, the REK algorithm proposed in Zouzias and Freris (2013) for general least-squares problems, reduces in our setting to randomly and independently selecting a node h and an edge (i, j) at each iteration k according to the following probabilities:

$$p_h = \frac{|\mathcal{N}_h| + 1}{2m}\,; \qquad\qquad p_{ij} = \frac{1}{m}\,.$$

and then to performing the following local updates:

$$\begin{aligned}
s_{\ell h}(k+1) &= s_{\ell h}(k) + \frac{\sum_{m\in\mathcal{N}_h}(s_{hm}(k) - s_{mh}(k))}{|\mathcal{N}_h|+1}\,, \quad \forall \ell \in \mathcal{N}_h\,,\\
s_{h\ell}(k+1) &= s_{h\ell}(k) - \frac{\sum_{m\in\mathcal{N}_h}(s_{hm}(k) - s_{mh}(k))}{|\mathcal{N}_h|+1}\,, \quad \forall \ell \in \mathcal{N}_h\,,\\
\widehat{x}_i(k+1) &= \widehat{x}_i(k) + \frac{z_{ij} - s_{ij}(k) - (\widehat{x}_i(k) - \widehat{x}_j(k))}{2}\,,\\
\widehat{x}_j(k+1) &= \widehat{x}_j(k) - \frac{z_{ij} - s_{ij}(k) - (\widehat{x}_i(k) - \widehat{x}_j(k))}{2}\,.
\end{aligned}$$

We point out that, since in the updating step only local information is required, the algorithm is implemented in a distributed fashion and it exactly requires $|\mathcal{N}_j| + 5$ communication rounds to perform an iteration. Specifically the first $|\mathcal{N}_j| + 2$ are due to the update of the variable s and the last 3 are needed to update $\widehat{x}_j$.

The second algorithm, denoted hereafter as **BC algorithm**, is proposed in Giridhar and Kumar (2006). It requires a coordinated broadcast communication protocol meaning that, during k-th iteration one node, say h, asks the variable $\widehat{x}_\ell$ to all its neighbors $\ell \in \mathcal{N}_h$. When it receives the current state values, it performs the following greedy local optimization based on the current status of the network:

$$\begin{aligned}
\widehat{x}_h(k+1) &= \arg\min_{\widehat{x}_h} \sum_{(i,j)\in\mathcal{E}}^{|\mathcal{E}|} \|\widehat{x}_i(k) - \widehat{x}_j(k) - z_{ij}\|^2\\
&= \tfrac{1}{2|\mathcal{N}_h|} \sum_{\ell\in\mathcal{N}_h} (2\widehat{x}_\ell(k) - z_{lh} + z_{hl})
\end{aligned}$$

Algorithm	Sent packets per iteration
a-CL	1
a-GL	1
BC	$\lvert\mathcal{N}_i\rvert + 1$
REK	$\lvert\mathcal{N}_i\rvert + 5$

Table 3.1: Number of sent packets per iteration for each algorithm.

Note that the number of communications performed during one iteration are $|\mathcal{N}_h| + 1$, since there is a broadcast packet sent by node h, and $|\mathcal{N}_h|$ packets sent by all its neighbors.

As first comparison between the algorithms proposed and those already presented in the literature, we report Table 3.1 which shows the number of sent packets per communication round at every iteration of each algorithm. This represents a measure of the energetic requirement for the algorithm. It can be seen that both the algorithm proposed are much "lighter" than the REK and the BC algorithm. In particular, we recall the fact that, our a-GL algorithm is inspired by the BC algorithm proposed in Giridhar and Kumar (2006). As reported in Remark 3.7.1, the main implementation difference between the two algorithm resides in the communication protocol. Indeed, in the BC algorithm, when a node is activated, first interrogates its neighbors and then updates its estimate. In the implementation of the a-GL algorithm, these actions are switched. This causes the BC algorithm to require a much larger number of sent packets per iteration. Figure 3.1 shows an illustrative representation of this fact. In particular it is reported the evolution of the error

$$J(k) = \log\left(\|A(\widehat{\mathbf{x}}(k) - \mathbf{x}_{\text{opt.}})\|\right) , \tag{3.18}$$

as function of the number of sent packet.

Figure 3.2 shows a comparison between the a-CL, the a-GL and the REK algorithm. We plot the error (3.18) as function of the algorithm iteration. It can be seen that all the strategies are converging to the optimal solution. In particular, the a-GL algorithm is the one showing best transient performance.

Next, we test the a-CL algorithm assuming that the network is randomly persistent communicating with uniform communication probabilities $(\gamma_1, \dots, \gamma_n)$, namely, $\gamma_1 = \dots = \gamma_N = 1/N$. Moreover the possibility of communication failure is taken into account. Specifically, supposing node i is transmitting, each node $j \in \mathcal{N}_i$ with a certain probability i.e., p_f, can not receive the sent packet. In Figure 3.3 we plotted the behavior of the error (3.18) for different values of the failure probability p_f. The plot reported is the

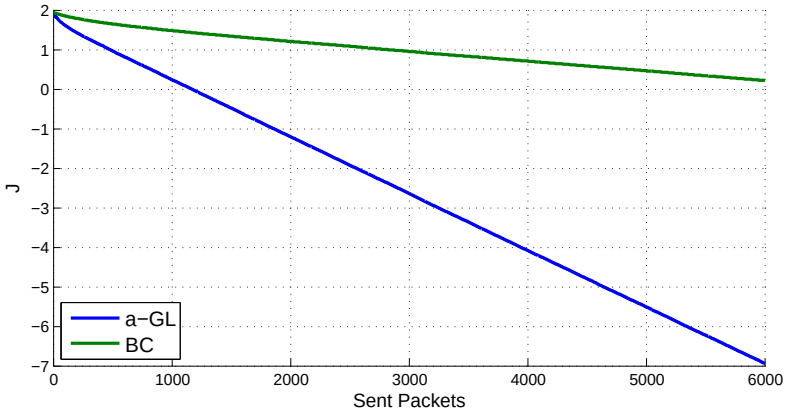

Figure 3.1: Comparison between the a-GL and the BC algorithm w.r.t. the number of sent packet

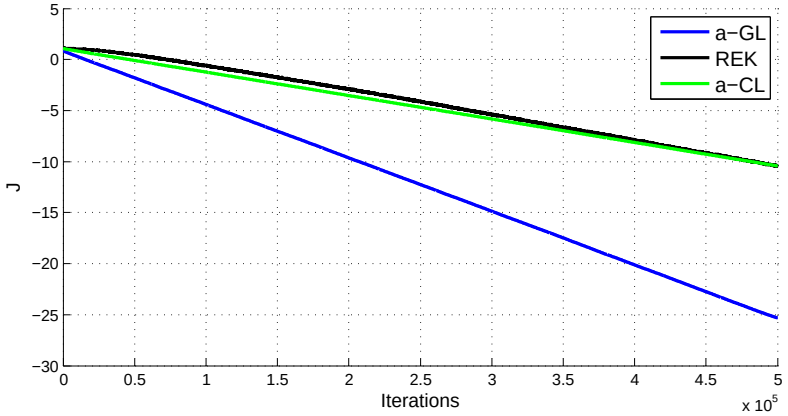

Figure 3.2: Comparison between the a-CL the a-GL and the REK algorithm w.r.t. the number of iteration.

result of the average over 1000 Monte Carlo runs, randomized with respect to both the measurement graph[2] and the initial conditions. Observe that the trajectory of J decreases exponentially.

Finally, we test the a-GL algorithm for different values of communication delays. From Figure 3.4 it is possible to observe the resiliency of the algorithm to communication non idealities. As expected, the convergence rate decreases as the delay increases. However, convergence is preserved.

[2]In performing our average we kept only the random geometric graphs which resulted to be connected.

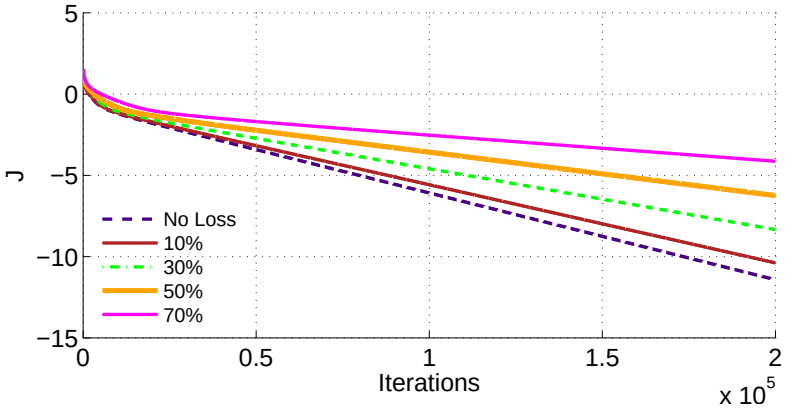

Figure 3.3: Behavior of J using the a-CL algorithm for a randomly persistent communicating network on a random geometric graph, for different values of the probability failure p_f.

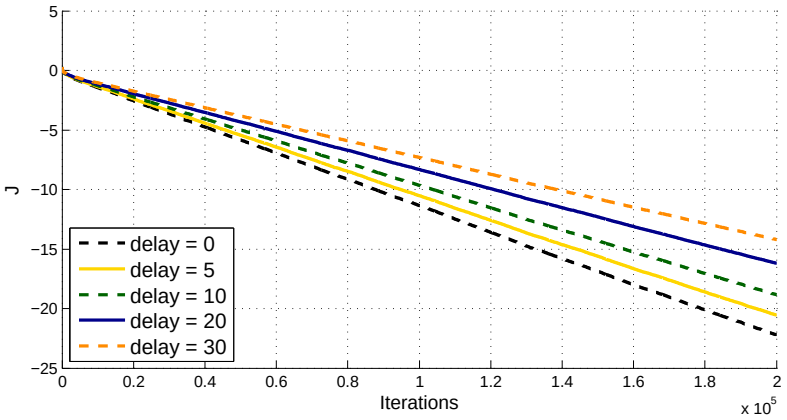

Figure 3.4: Behavior of J using the a-GL algorithm for a randomly persistent communicating network on a random geometric graph, for different values of the communication delay.

3.11 Conclusions

In this part of the manuscript we dealt with the synchronization problem among smart devices in a smart grid. In order to provide reliable and accurate control strategies, synchronization among smart devices might represent a critical step that must be carefully taken into account. In order to simplify the analysis, the problem can be usefully cast into the framework of sensor localization in a sensor network. In particular, we provide two different strategies to solve this problem. The first is a based on a linear consensus with memory. The second exploits a descent along a gradient direction. Both the proposed algorithms can be implemented in a completely distributed fashion. Moreover they are based on an asynchronous communication protocol, namely, asynchronous broadcast, in which, at every iteration, only one node wakes up and communicate with its neighboring

nodes. Both the a-CL and the a-GL show themselves to be resilient to communication non idealities. In particular, under mild boundedness assumptions on the packet losses and delay in the communication channel, we proved convergence of both the strategies to an optimal solution. All these features make the algorithms appealing and suitable for real world implementations. Finally, compared to solutions already present in the literature, the algorithms demonstrate themselves to be lighter in the energetic requirements and competitive in the convergence behavior.

4

Distributed State Estimation

"If you can't explain it simply, you don't understand it well enough."
A. Einstein

In this third part of the thesis the problem of state estimation is considered. First we present the problem and some related literature. Then, two algorithms to solve the problem are presented. The first is a partitioned based formulation of the well known ADMM algorithm. The second is a robust implementation of a generalized gradient method. Before concluding, a numerical comparison of the two approaches is presented.

4.1 Introduction and related work

One of the major aspects concerning the electric grid modernization process is the widespread deployment of dispersed measurement, monitoring, and actuation devices. This proliferation is creating large-scale cyber-physical systems which promise a new revolution in many fields. However, these systems require the development of new engineering design and computation paradigms due to the sheer amount of devices and data to be managed. For example, many problems have been shown to be cast as optimization problems. Distributed optimization has then become so important for mainly two different reasons: the first reason is that with the advent of Big Data, it is

unconceivable to run optimization algorithms on a single (super)-computer, but it is necessary to *parallelise* computation among many processors. The second reason is that many optimization problems are *sparse* by nature since correlation between data is local. One of the major difficulty to deal with distributed optimization using multiple processing units is to guarantee reliable synchronisation and communication since communication can be wireless and CPU execution times might not be known in advance as in the context of cloud-computing.

Several solutions alternative to the centralized computation of the state of a electric grid have been proposed in the literature. The most common approach is based in a two level architecture requiring a global coordinator Schweppe, Wildes, and Rom (1970); Cutsem and Ribbens-Pavella (1983); Iwamoto, Kusano, and Quintana (1989); Zhao and Abur (2005); Gómez Expósito, Abur, de la Villa Jean, and Goméz Quiles (2011); Korres (2011). In this multi-area approach the grid is split into macro-areas, each of them equipped with a data processor with communication and computational capabilities. Each processor estimates the state of the sub-area which has been assigned to it, excluding boundary bus measurements, and sends this estimate to the global coordinator. The global coordinator combines all these estimates making them compatible to the boundary bus measurements obtained at a interconnection level. Of note is that this approach becomes uninteresting in the case in which each macro area reduces to one single PMU, because the global coordinator would have to face the very same problem of a centralized state estimator.

Leader-less solutions include approximate algorithms developed from the optimality conditions involved Falcao, Wu, and Murphy (1995), Conejo, de la Torre, and Cañas (2007). Beyond requiring local observability, these algorithms are not always guaranteed to converge. The auxiliary principle is invoked in Ebrahimian and Baldick (2000), with the drawback that several paramenters have to be tuned. Also the distributed algorithms proposed in Pasqualetti, Carli, and Bullo (2012), Jiang, Vittal, and Heydt (2007), adopt the multi-area approach. Remarkably, the algorithm in Pasqualetti et al. (2012) is shown to be provably convergent in finite time to the optimal solution of a classical weighted least squares problem. However each area is envisioned to maintain a copy of the entire high-dimensional state vector, drastically increasing the communication and computation effort required at each iteration. In Jiang et al. (2007) the overall system is decomposed into subsystems, each of them provided with a slack bus monitored by a PMU. Each subsystem obtains its own local estimate, and these estimates are then coordinated based on the PMU measurements, re-estimating the boundary state variables at the aggregation level. However, the global solution obtained through this two-steps procedure is not guaranteed to coincide with the optimal centralized solution, and no analysis of potential

synchronization errors is included.
To address the conventional least-squares power system state estimation problem, the authors in Kekatos and Giannakis (2013) propose a novel algorithm which is a local version of the classical ADMM. When dealing with the exact non-linear model of the electric grid, the functional costs to be minimized are non-convex and, in general, the behavior of the ADMM is characterized by slow transient performance. For this reason, typically, the exact models are iteratively linearized via the Gauss-Newton method, or by resorting to the so-called DC approximation Abur and Gómez Expósito (2004). Either way, one arrives at classical linear models which are just first-order approximations of the true model. Even though the ADMM-based algorithm introduced in Kekatos and Giannakis (2013) exhibits good performance in simulations, a complete proof of its convergence is not provided.
As pointed out, although distributed optimization algorithms have a long history in the parallel and distributed computation literature, see, e.g. Bertsekas and Tsitsiklis (1989), it has mainly focused on synchronous algorithms. The first class of algorithms appearing in order to deal with asynchronous computations relies on primal sub-gradient or descent iterations as in Nedic and Ozdaglar (2009); Nedic, Ozdaglar, and Parrilo (2010); Marelli and Fu (2015). Finally, distributed algorithms based on Newton methods have been proposed to speed-up the computation (Zargham, Ribeiro, Ozdaglar, and Jadbabaie, 2014; Zanella, Varagnolo, Cenedese, Pillonetto, and Schenato, 2011).

Here, we consider one specific thrust, which is the deployment of low cost phasorial measurement units (e.g., PMUs) in the medium and low voltage power distribution grid. The presence of PMUs is quite uncommon in today's power distribution networks. However, this scenario has become the subject of recent research efforts in the power systems community, including a 3-years research project involving University of California together with the Power Standards Lab and Lawrence Berkeley National Lab von Meier et al. (2014). Previous investigations on the subject include the experimental deployment and the extensive measurement campaign in Albu, Neurohr, Apetrei, Silvas, and Federenciuc (2010), and the analysis proposed in Ochoa and Wilson (2011); Wache and Murray (2011); Ree, Centeno, Thorp, and Phadke (2010) for the possible applications of such measurement devices, also called μPMUs, in the power distribution grid.
There is a substantial consensus on the many opportunities offered by the deployment of a live network of μPMUs, including diagnostic routines (e.g. for unplanned islanding and fault detection), power flow estimation on the power lines (by knowing the phasorial nodal voltage differences across the entire grid), and feedback control strategies for

coordination of renewable resources, islanding, and volt/VAR compensation. However, there are a number of unresolved issues that need to be addressed, in order to make these applications possible. As the cost of PMUs depends on their sensing accuracy, noisy and low-end device will necessarily be preferred in order to keep the cost of large scale deployment acceptable. In particular, time synchronization between different PMUs is a major technological issue, which is generally tackled via GPS modules that can provide timestamping of the data. Following the approach that is typically adopted in wireless sensor networks Special issue on sensor networks and applications (2003); Bolognani et al. (2010), we propose a solution to these issues that exploits the large number of sensors and their communication capabilities to compensate for their modest performance. We propose a leader-less and distributed state estimation architecture, in which PMUs communicate with their corresponding area data aggregator which is responsible for the computation and communication with neighbor peers in order to improve the quality of their measurements. Such improved measurements can then be made available locally to the monitoring and control algorithms that may need them, in a transparent way (see Figure 4.1).

We design two different state estimation algorithms that can be adopted in order to implement the proposed architecture. In particular, we focus on some specific features that are of crucial importance in the scenario of medium voltage power distribution grids:

- we need to deal effectively with poorly synchronized measurements;
- we cannot assume large x/r impedance ratio of the power lines;
- we only allow measurements that can be performed by devices connected to the grid buses, and no power flow and current measurements on the power lines;
- we do not assume redundancy of measurement locations (instead, we exploit the fact of being able to sense both voltage and current injection at each bus);
- the computational effort must remain limited as the grid grows in size;
- we aim at leader-less algorithms (in which no grid supervisor or central data aggregator is present);
- we consider local data aggregator in the estimation architecture, as it might be a convenient architecture when combined with some hierarchical communication infrastructures De Craemer and Deconinck (2010). However, we point out that the case of peer-to-peer architecture, in which each PMU is provided with its own computational capabilities, and can communicate with its neighbor PMUs,

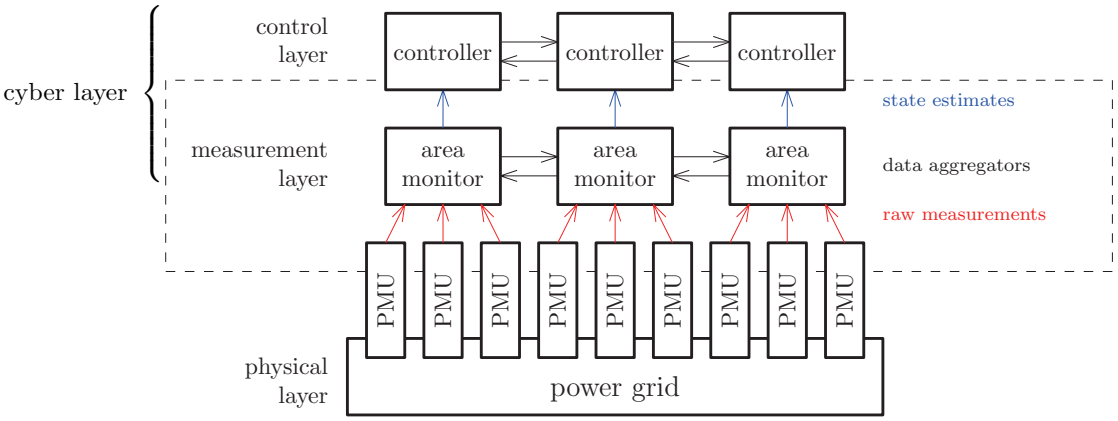

Figure 4.1: The panel shows a functional representation of the proposed architecture. Each bus is provided with a PMU, which measures bus voltage and current. These device communicate with their neighbors, in order to process the raw data and obtain a better estimate of the system state. They constitute the middle *measurement layer* that, together with the *control layer* form the *cyber layer*. The state estimates computed in this layer are then transmitted to the control layer, where different monitoring and optimization algorithm can coexist.

represents a special case of the proposed architecture where each area collapses to a single PMU.

We stress the fact that we focus on a class of quadratic optimization problems which can be encountered on a large variety of applications different from electric grid state estimation, e.g., multi-robot localization (Carron, Todescato, Carli, and Schenato (2014)) and Network Utility Maximization (Palomar and Chiang (2006)). These quadratic functions are obtained by casting the state estimation problem as a classical least-squares problem.

Finally, we point out that the first one of the two approaches that we propose (Bolognani, Carli, and Todescato, 2014) is an extension of the local ADMM algorithm of Kekatos and Giannakis (2013) to the minimization of quadratic functions. The algorithm we propose reduces to linear iterations and it is shown to be provably convergent. The second algorithm proposed (Todescato, Cavraro, Carli, and Schenato, 2015a), which is based on a generalized gradient descent strategy, is a provably convergent and distributed algorithm, under suitable assumptions on the step size, which is robust to the presence of packet losses in the communication channel. To the best of the authors knowledge, this is one of the first provably convergent algorithms in the presence of packet losses, since both ADMM algorithms and distributed sub gradient methods (DSM) require reliable communication. Interestingly, the proposed algorithm is also suitable for fully parallel computation, i.e. multiple agents can communicate and update their local variable

simultaneously, and for broadcast communication, i.e. nodes do not need to enforce a bidirectional communication such as in gossip algorithms, therefore very attractive from a practical point of view.

4.2 Problem formulation

In the following we present the problem at hand in its general form since it can be stated for a class of applications much broader than power systems state estimation. We will explicitly show later how the latter can be cast in the presented framework.

Consider a set of r agents $\mathcal{V}_c = \{1, \dots, r\}$, where each agent i is described by its state vector $\mathbf{x}_i \in \mathbb{R}^{n_i}$. Assume the agents can communicate among themselves through a bidirected strongly connected *communication graph* $\mathcal{G}_c = (\mathcal{V}_c, \mathcal{E}_c)$. As usual we denote the set of neighboring agents of systems i with $\mathcal{N}_i$ ($\mathcal{N}_i^+$ if we consider system i as well). Assume each agent collects a set of measurements $\mathbf{b}_i \in \mathbb{R}^{m_i}$ which are noisy linear combinations of its own state and those of its neighboring agents, i.e.,

$$\mathbf{b}_i = \sum_{j \in \mathcal{N}_i^+} A_{ij}\mathbf{x}_j + \mathbf{w}_i = A_{ii}\mathbf{x}_i + \sum_{j \in \mathcal{N}_i} A_{ij}\mathbf{x}_j + \mathbf{w}_i$$

where $A_{ij} \in \mathbb{R}^{m_i \times n_j}$ and where $\mathbf{w}_i$ is white noise of zero mean and variance R_i independent of the other $\mathbf{w}_j$.

We consider the problem of estimating the entire state of the network from the knowledge of the noisy measurements. By collecting all the agents state and measurements in the vectors

$$\mathbf{x} := \begin{bmatrix} \mathbf{x}_1 \\ \vdots \\ \mathbf{x}_r \end{bmatrix} \in \mathbb{R}^n, \qquad \mathbf{b} := \begin{bmatrix} \mathbf{b}_1 \\ \vdots \\ \mathbf{b}_r \end{bmatrix} \in \mathbb{R}^m, \tag{4.1}$$

where $n = \sum_i n_i$ and $m = \sum_i m_i$, respectively, it is possible to formulate the problem as a classical *weighted least square* problem. That is

$$\min_{\mathbf{x}} J(\mathbf{x}), \tag{4.2}$$

where

$$J(\mathbf{x}) = \frac{1}{2}(A\mathbf{x} - \mathbf{b})^T R^{-1}(A\mathbf{x} - \mathbf{b}). \tag{4.3}$$

The matrix $A \in \mathbb{R}^{m \times n}$ represents the measurements matrix, whose ij-th block is simply

defined as $[A]_{ij} = A_{ij}$, while $R \in \mathbb{R}^{m \times m}$ is the block diagonal matrix defined as

$$R = \text{blkdiag}\left\{R_1, \ldots, R_N\right\}, \tag{4.4}$$

which represents the noise variance.
From now on, we assume that $n \leq m$ and that A is full rank. Under these assumptions, it is well known that the solution of (4.2) is unique and given by

$$\mathbf{x}_{\text{opt.}} = (A^T R^{-1} A)^{-1} A^T R^{-1} \mathbf{b}\,. \tag{4.5}$$

To compute the value of $\mathbf{x}^*$ directly as in (4.5), one needs all the measurements, the matrix A and the noise variance R, i.e., full knowledge of the network is required. On the contrary, we aim at solving Problem (4.2) in a distributed fashion. To this end, note that it is possible to rewrite Problem (4.2) as

$$\min_{\mathbf{x}_1, \ldots, \mathbf{x}_N} \sum_{i=1}^{r} J_i(\mathbf{x}_i, \{\mathbf{x}_j\}_{j \in \mathcal{N}_i})\,, \tag{4.6}$$

where

$$J_i(\mathbf{x}_i, \{\mathbf{x}_j\}_{j \in \mathcal{N}_i}) = \frac{1}{2}(A_{ii}\mathbf{x}_i + \sum_{j \in \mathcal{N}_i} A_{ij}\mathbf{x}_j - \mathbf{b}_i)^T R_i^{-1}(A_{ii}\mathbf{x}_i + \sum_{j \in \mathcal{N}_i} A_{ij}\mathbf{x}_j - \mathbf{b}_i). \tag{4.7}$$

The above equation highlights the local dependence of each cost function J_i on information regarding only agent i and its neighbors $j \in \mathcal{N}_i$.

In the following, we present two distributed algorithms to solve optimization problems in the separable quadratic form (4.6). Firstly, we present a modified version of the ADMM algorithm (Kekatos and Giannakis, 2013) for quadratic cost which has been first presented in Bolognani et al. (2014). Secondly, a *Block-Jacobi* algorithm first presented in Todescato et al. (2015a). For both the algorithms a possible extension to deal with possible communication failures and delays will be given. However, only the second algorithm presented is resilient to communication failures and we will compare the performance of the two. Before moving on, we briefly step back in order to specify how a power grid state estimation problem can be cast in the framework presented.

4.3 Measurements model and area partitioning

Consider a power grid described by a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where $|\mathcal{V}| = n$. We are interested in estimating the state of the power grid which, in this particular application, consists of

the real and imaginary part of the voltage at each bus of the grid, i.e., in vector notation,

$$\mathbf{x} := \begin{bmatrix} \Re(\mathbf{u}) \\ \Im(\mathbf{u}) \end{bmatrix} .$$

In the following we assume that each electric bus $h \in \mathcal{V}$ is equipped with a *low-cost* measurement unit, e.g., low-cost PMU. In particular each PMU is able to collect measurements of the real and the imaginary part of voltage and current at bus $h \in \mathcal{V}$, that is:

$$\begin{aligned} \Re(\widetilde{u}_h) &= \Re(u_h) + w_{\Re(u_h)} , \\ \Im(\widetilde{u}_h) &= \Im(u_h) + w_{\Im(u_h)} , \\ \Re(\widetilde{i}_h) &= \Re(i_h) + w_{\Re(i_h)} , \\ \Im(\widetilde{i}_h) &= \Im(i_h) + w_{\Im(i_h)} , \end{aligned} \tag{4.8}$$

where the different measurement error terms are caused by independent physical causes and can therefore be assumed uncorrelated. We assume the PMUs are homogeneous and thus their errors exhibit the same probability distribution. In particular, we assume a normal distribution for the noise

$$\begin{aligned} w_{\Re(u_h)} &\sim \mathcal{N}(0, \sigma^2_{\Re(u)}) , \\ w_{\Im(u_h)} &\sim \mathcal{N}(0, \sigma^2_{\Im(u)}) , \\ w_{\Re(i_h)} &\sim \mathcal{N}(0, \sigma^2_{\Re(i)}) , \\ w_{\Im(i_h)} &\sim \mathcal{N}(0, \sigma^2_{\Im(i)}) . \end{aligned}$$

Observe that in the case the PMU would return measurements in polar coordinates, i.e. amplitude and phase of voltage and current, it is possible to obtain a representation which is equivalent to that of Eq.(4.8) by rotating the measurement from polar to rectangular coordinates. If this is the case however, the statistical description of the noise terms must be accordingly computed, since errors in magnitude and phase would project into both real and imaginary parts, leading to correlated noise.
By stacking the measurements and the errors into the vectors $\mathbf{b}$ and $\mathbf{w}$

$$\mathbf{b} := \begin{bmatrix} \Re(\widetilde{\mathbf{u}}) \\ \Im(\widetilde{\mathbf{u}}) \\ \Re(\widetilde{\mathbf{i}}) \\ \Im(\widetilde{\mathbf{i}}) \end{bmatrix} , \qquad \mathbf{w} = \begin{bmatrix} \mathbf{w}_{\Re(\mathbf{u})} \\ \mathbf{w}_{\Im(\mathbf{u})} \\ \mathbf{w}_{\Re(\mathbf{i})} \\ \mathbf{w}_{\Im(\mathbf{i})} \end{bmatrix} ,$$

and recalling Eq.(2.17) which describes the linear relation between voltages and currents

$$\mathbf{i} = Y\mathbf{u}\,,$$

where Y is the grid admittance matrix, it is clear that the measurement model can be rewritten as a linear function of the state that is

$$\mathbf{b} = A\mathbf{x} + \mathbf{w}\,,$$

where the measurement matrix A is equal to

$$A := \begin{bmatrix} \mathbb{I} & \mathbb{O} \\ \mathbb{O} & \mathbb{I} \\ \Re(Y) & -\Im(Y) \\ \Im(Y) & \Re(Y) \end{bmatrix}.$$

In order to cast the power system state estimation problem in the same framework described in Section 4.2 we need one more step. According to the partition and distributed estimation architecture described by Figure 4.1, we assume to divide the network into r non-overlapping macro-areas, namely $\mathcal{A}_1, \ldots, \mathcal{A}_r$, each of which consisting of a subset of buses. In particular we assume that

$$\mathcal{A}_i \subset \mathcal{V}\,, \qquad \mathcal{A}_i \cap \mathcal{A}_j = \emptyset\,, \qquad \bigcup_{i=1}^{r} \mathcal{A}_i = \mathcal{V}\,.$$

Each area is supervised by a dedicated monitor and data aggregator which

- can collect raw measurements coming from the buses which own to its supervised area;
- has only a local knowledge of the monitored area's topology, namely, it knows the grid topology pertaining its sub-area and it also knows which power lines connect the sub-area with other adjacent sub-areas (called neighboring areas);
- can communicate with neighboring areas monitors.

An example of partitioning is illustrated in Figure 4.2.

It is convenient, according to this partitioning, to build a communication graph $\mathcal{G}_c = (\mathcal{V}_c, \mathcal{E}_c)$ where $\mathcal{V}_c = \{1, \ldots, r\}$ identifies the set of areas/systems where each of them is uniquely determined by the corresponding index while $\mathcal{E}_c$ contains the communication links between neighboring monitors (see Figure 4.2). It is worth noticing that the case of

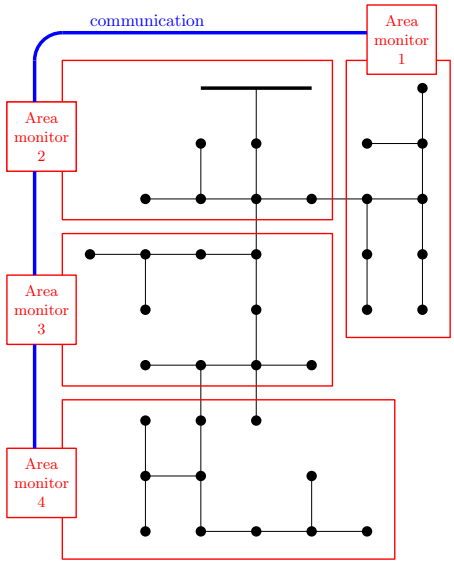

Figure 4.2: IEEE 37 bus test feeder divided into 4 non overlapping areas. The blue thick lines identify the communication link among neighboring area monitors.

a peer-to-peer architecture is one special case of the architecture that we just described. Such special case can be easily obtained just by considering "atomic" areas, each one encompassing one single electric bus.
Now, let us denote with $\mathbf{x}_i$, $\mathbf{b}_i$, $\mathbf{w}_i$ and A_{i*} the subset of the state vector, of the measurements vector, of the noise vector and measurement matrix respectively, corresponding to the i-th area and identified by the bus indexes contained in the set $\mathcal{A}_i$. Finally, with a little abuse and reloading the notation it is possible to define $\mathbf{x}$ and $\mathbf{b}$ as in Eq.(4.1) and the noise covariance matrix R as in Eq.(4.4) to eventually obtain the same formulation as that of Section 4.2.

4.4 Communication failures and packet losses

Later on along the chapter we will deal with the state estimation problem in the presence of communication non-idealities, e.g., failures, delays and packet losses. To model them and in particular the packet losses, it is convenient to introduce the indicator function

$$\gamma_j^{(i)}(t) = \begin{cases} 1 & \text{if } i \text{ received the information sent by } j \\ 0 & \text{otherwise.} \end{cases},$$

with the assumption that $\gamma_i^{(i)}(t) = 1$, since node i has always access to its local variables. We assume the following

Assumption 4.4.1. There exists a constant T such that, for all $t \geq 0$, for all $i \in \mathcal{V}_c$ and for all $j \in \mathcal{N}_i$,

$$\mathbb{P}\left[\{\gamma_j^{(i)}(t), \ldots, \gamma_j^{(i)}(t+T)\} = \{0, \ldots, 0\}\right] = 0\,.$$

Roughly speaking, Assumption 4.4.1 states that agent i receives, at least once, information coming from agent j within any window of T iterations of the algorithm. To avoid disambiguation, we will make clear whenever we are in the ideal and in the non-ideal communication case.

Next we present our first proposed solution which is based on a modified version of the classical ADMM algorithm.

4.5 A partition-based ADMM algorithm

The method we propose here is a partition-based version of the classical ADMM method which exploits the equivalence between problem in (4.6) and the following problem

$$\begin{aligned} \min_{\mathbf{x}} \quad & \sum_{i=1}^{r} J_i(\mathbf{x}_i^{(i)}, \{\mathbf{x}_j^{(i)}\}_{j\in\mathcal{N}_i}) \\ \text{subject to} \quad & \mathbf{x}_i^{(i)} = \mathbf{z}_i^{(i,j)};\ \mathbf{x}_j^{(i)} = \mathbf{z}_j^{(i,j)} \\ & \mathbf{x}_i^{(i)} = \mathbf{z}_i^{(j,i)};\ \mathbf{x}_j^{(i)} = \mathbf{z}_j^{(j,i)}, \qquad \forall\ j \in \mathcal{N}_i, \end{aligned} \tag{4.9}$$

where with the notation $\mathbf{x}_j^{(i)}$ we denote a copy of state $\mathbf{x}_j$ stored in memory by system i, and where the $\mathbf{z}$'s are auxiliary variables that are introduced by the ADMM algorithm. Observe that the connectedness of the graph $\mathcal{G}$ and the presence of the bridge variables $\mathbf{z}$'s ensures that the optimal solution of (4.9) is given by $\mathbf{x}_i^{(i)} = \mathbf{x}_{\text{opt.},i}$ and $\mathbf{x}_j^{(i)} = \mathbf{x}_{\text{opt.},j}$. The redundant constraints added in Problem (4.9) with the respect to Problem (4.6), allow to find the optimal solution through a distributed, iterative, partition-based implementation which optimizes the standard augmented Lagrangian defined, for $\rho > 0$, as

$$\begin{aligned} \mathcal{L} = \sum_{i=1}^{r} \Bigg\{ & J_i(\mathbf{x}_i^{(i)}, \{\mathbf{x}_j^{(i)}\}_{j\in\mathcal{N}_i}) + \sum_{j\in\mathcal{N}_i} \left[\boldsymbol{\lambda}_i^{(i,j)T}\left(\mathbf{x}_i^{(i)} - \mathbf{z}_i^{(i,j)}\right) + \boldsymbol{\lambda}_j^{(i,j)T}\left(\mathbf{x}_j^{(i)} - \mathbf{z}_j^{(i,j)}\right)\right] \\ & + \sum_{j\in\mathcal{N}_i} \left[\boldsymbol{\mu}_i^{(i,j)T}\left(\mathbf{x}_i^{(i)} - \mathbf{z}_i^{(j,i)}\right) + \boldsymbol{\mu}_j^{(i,j)T}\left(\mathbf{x}_j^{(i)} - \mathbf{z}_j^{(j,i)}\right)\right] + \frac{\rho}{2} \sum_{j\in\mathcal{N}_i} \Big[\|\mathbf{x}_i^{(i)} - \mathbf{z}_i^{(i,j)}\|^2 \\ & + \|\mathbf{x}_j^{(i)} - \mathbf{z}_j^{(i,j)}\|^2 + \|\mathbf{x}_i^{(i)} - \mathbf{z}_i^{(j,i)}\|^2 + \|\mathbf{x}_j^{(i)} - \mathbf{z}_j^{(j,i)}\|^2\Big] \Bigg\}. \end{aligned}$$

At each iteration of the algorithm, node i, $i \in \{1, \dots, r\}$, alternates dual ascent step on the Lagrange multipliers $\boldsymbol{\lambda}_i^{(i,j)}$, $\{\boldsymbol{\lambda}_j^{(i,j)}\}_{j\in\mathcal{N}_i}$ and $\boldsymbol{\mu}_i^{(i,j)}$, $\{\boldsymbol{\mu}_j^{(i,j)}\}_{j\in\mathcal{N}_i}$, with minimization steps on the variables $\mathbf{x}_i^{(i)}$, $\{\mathbf{x}_j^{(i)}\}_{j\in\mathcal{N}_i}$ and $\mathbf{z}_i^{(i,j)}$, $\{\mathbf{z}_j^{(i,j)}\}_{j\in\mathcal{N}_i}$. However, for the case where the functions $J_i's$ have the particular quadratic structure illustrated in (4.7), these optimization steps can be greatly simplified. Indeed in this case the partition-based ADMM algorithm reduces to a linear algorithm requiring, during each iteration of its implementation, only one communication round involving the $\mathbf{x}_i^{(i)}$, $\{\mathbf{x}_j^{(i)}\}_{j\in\mathcal{N}_i}$, $i \in \{1, \dots, r\}$, variables. To show that, it is convenient to introduce the following compact notation. Consider node i and, without loss of generality, assume $\mathcal{N}_i = \left\{j_1, \dots, j_{|\mathcal{N}_i|}\right\}$. Then let

$$X^{(i)} = \begin{bmatrix} \mathbf{x}_i^{(i)} \\ \left\{\mathbf{x}_j^{(i)}\right\}_{j\in\mathcal{N}_i} \end{bmatrix}, \qquad A_i = \begin{bmatrix} A_{ii} & A_{ij_1} & \dots & A_{ij_{|\mathcal{N}_i|}} \end{bmatrix},$$

$$M_i = \operatorname{diag}\left\{|\mathcal{N}_i|\mathbb{I}_{n_i},\ \mathbb{I}_{n_{j_1}}, \dots, \mathbb{I}_{n_{j_{|\mathcal{N}_i|}}}\right\}.$$

Additionally we introduce the following auxiliary variables,

$$G^{(i)} = \begin{bmatrix} G_i^{(i)} \\ G_{j_1}^{(i)} \\ \vdots \\ G_{j_{|\mathcal{N}_i|}}^{(i)} \end{bmatrix}, \qquad F^{(i)} = \begin{bmatrix} F_i^{(i)} \\ F_{j_1}^{(i)} \\ \vdots \\ F_{j_{|\mathcal{N}_i|}}^{(i)} \end{bmatrix}, \qquad B^{(i)} = \begin{bmatrix} B_i^{(i)} \\ B_{j_1}^{(i)} \\ \vdots \\ B_{j_{|\mathcal{N}_i|}}^{(i)} \end{bmatrix},$$

where $G_i^{(i)}, F_i^{(i)}, B_i^{(i)} \in \mathbb{R}^{n_i}$ and $G_{j_h}^{(i)}, F_{j_h}^{(i)}, B_{j_h}^{(i)} \in \mathbb{R}^{n_{j_h}}$. It turns out that $A_i \in \mathbb{R}^{m_i\times\gamma_i}$, $M_i \in \mathbb{R}^{\gamma_i\times\gamma_i}$ and $G^{(i)}, F^{(i)}, B^{(i)} \in \mathbb{R}^{\gamma_i}$, where $\gamma_i = n_i + \sum_{h=1}^{|\mathcal{N}_i|} n_{j_h}$.
The *partition-based ADMM algorithm* for quadratic functions is formally described in Algorithm 3. The standing assumption is that all the matrices $A_i^T Q_i A_i + M_i$, $i \in \{1, \dots, r\}$ are invertible.

The following proposition characterizes the performance of the above algorithm.

Proposition 4.5.1. *Consider the partition-based ADMM algorithm described above. Let ρ be any real number. Assume that the matrices $A_i^T Q_i A_i + M_i$, $i \in \{1, \dots, r\}$, are invertible. Then the trajectory $t \to \left\{X^{(i)}(t)\right\}$ converge exponentially to the optimal solution, namely, for $i \in \{1, \dots, r\}$, $\mathbf{x}_j^{(i)}(t) \to \mathbf{x}_{\text{opt.},j}$ for all $j \in \mathcal{N}_i$ and, in particular,*

$$\mathbf{x}_i^{(i)}(t) \to \mathbf{x}_{\text{opt.},i}.$$

The proof can be found in Appendix A.6

Remark 4.5.2. We point out that it is possible to provide a simplified a version of the

Algorithm 3 Partition Based ADMM algorithm.

Require: $\forall i \in \mathcal{V}_c$, store and initialize to 0 $X^{(i)}, G^{(i)}, F^{(i)}, B^{(i)}$.

1: **for** $t \in \mathbb{N}$ each $i \in \mathcal{V}_c$ **do**

2: sends $\mathbf{x}_i^{(i)}(t), \mathbf{x}_j^{(i)}(t)$ to $j \in \mathcal{N}_i$;

3: receives $\mathbf{x}_i^{(j)}(t), \mathbf{x}_j^{(j)}(t)$ from $j \in \mathcal{N}_i$;

4: updates the memory and the estimate as

$$G_i^{(i)}(t) = \frac{\rho}{2} \sum_{j \in \mathcal{N}_i} \left(\mathbf{x}_i^{(i)}(t) - \mathbf{x}_i^{(j)}(t) \right), \quad G_{j_h}^{(i)}(t) = \frac{\rho}{2} \left(\mathbf{x}_{j_h}^{(i)}(t) - \mathbf{x}_{j_h}^{(j_h)}(t) \right), \quad 1 \leq h \leq |\mathcal{N}_i|$$

$$F^{(i)}(t+1) = F^{(i)}(t) + G^{(i)}(t),$$

$$B^{(i)}(t+1) = 2\rho M_i X^{(i)}(t) - G^{(i)}(t) - 2F^{(i)}(t+1),$$

$$X^{(i)}(t+1) = \left[A_i^T R_i^{-1} A_i + M_i \right]^{-1} \left[A_i^T R_i^{-1} b_i + \frac{1}{2} B^{(i)}(t+1) \right].$$

5: **end for**

partition-based ADMM algorithm, where two monitors communicate with each other only the information related to those nodes which are on the boundary of their sub-areas of interest. To be more precise, monitor i receives from monitor j only those components of $\mathbf{x}_j^{(j)}$ (resp. of $\mathbf{x}_i^{(j)}$) such that the corresponding terms in A_{ij} (resp. in A_{ji}) are different from zero. This leads to a vector $X^{(i)}$ which is smaller in size, since monitor i keeps in memory only a copy of the states of the nodes of the other sub-regions which are on the boundary with sub-region A_i. In turn, also the vectors $G^{(i)}, F^{(i)}, B^{(i)}$ and the matrices M_i, A_i result to be smaller in size.

It is worth mentioning that this approach has been taken in Kekatos and Giannakis (2013). Here, for the sake of the notational simplicity, and because we are mostly interested in cases in which the size of each area is small (possibly just one agent), we have preferred not to provide the details of this implementation aspect.

As will be shown in the simulation section usually the ADMM algorithm exhibits a good transient behavior and fastly converges to the optimal solution of the corresponding centralized problem. Notice however that the algorithm does require reliable synchronous communications which makes it not resilient to packet losses and communication failures. In the following we present a possible naive variation of Algorithm 3 which can be exploited in order to deal with communication non idealities. The algorithm is known to be not convergent but is presented for comparison purposes. Indeed, the second algorithm hereafter presented is usually characterized by worse transient behavior but robustness to packet losses and failures and is amenable for an asynchronous implementation.

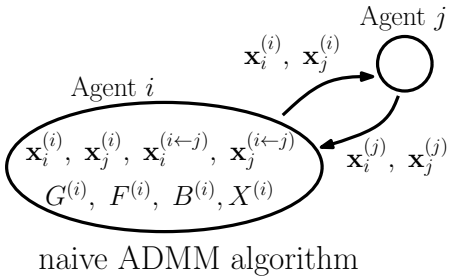

Figure 4.3: Communication scheme for the naive ADMM algorithm in the presence of packet losses and communication failures.

4.6 A partition-based ADMM algorithm with packet losses

Here we present a modified version of the partition-based ADMM algorithm to deal with communication non idealities. To model the packet losses, we make use of the indicator function introduced in Section 4.4 and we assume Assumption 4.4.1 holds. We adopt a very simple idea. More precisely, we use the last received information from a node if a packet loss occurs. To do so, we equip each agent with additional slots of memory. Specifically, agent i, for $j \in \mathcal{N}_i$, stores in memory the additional variables $\mathbf{x}_i^{(i\leftarrow j)}, \mathbf{x}_j^{(i\leftarrow j)}$, which are updated as follows:

$$\mathbf{x}_i^{(i\leftarrow j)}(t) = \begin{cases} \mathbf{x}_i^{(j)}(t) & \text{if } \gamma_j^{(i)}(t) = 1 \\ \mathbf{x}_i^{(i\leftarrow j)}(t-1) & \text{if } \gamma_j^{(i)}(t) = 0 \end{cases} \tag{4.10}$$

$$\mathbf{x}_j^{(i\leftarrow j)}(t) = \begin{cases} \mathbf{x}_j^{(j)}(t) & \text{if } \gamma_j^{(i)}(t) = 1 \\ \mathbf{x}_j^{(i\leftarrow j)}(t-1) & \text{if } \gamma_j^{(i)}(t) = 0 \end{cases}. \tag{4.11}$$

By exploiting these additional variables, it is possible to modify Algorithm 3 by updating $G_i^{(i)}(t)$ and $G_j^{(i)}(t)$ as

$$G_i^{(i)}(t) = \frac{\rho}{2} \sum_{j\in\mathcal{N}_i} \left(\mathbf{x}_i^{(i)}(t) - \mathbf{x}_i^{(i\leftarrow j)}(t)\right),$$
$$G_j^{(i)}(t) = \frac{\rho}{2} \left(\mathbf{x}_j^{(i)} - \mathbf{x}_j^{(i\leftarrow j)}\right).$$

while the remaining part of the algorithm is the same as before. We refer to this modified version of the partition-based ADMM algorithm as *naive ADMM algorithm*. An illustration of communication scheme and memory requirements are shown in Figure 4.3.

The following section contains the second solution proposed which relies on a general-

ized gradient descent strategy, namely a Block-Jacobi update.

4.7 Block-Jacobi algorithm

Consider the *generalized gradient* descent strategy

$$\mathbf{x}(t+1) = \mathbf{x}(t) - \epsilon D^{-1} \nabla J(\mathbf{x}(t)) \tag{4.12}$$

where $\nabla J(\mathbf{x}(t))$ is the gradient of J, i.e., $\nabla J(\mathbf{x}(t)) = [\partial J(\mathbf{x}(t))/\partial \mathbf{x}]^T$, D is a generic positive definite matrix and ϵ a suitable positive constant, usually referred as *step size*. The algorithm we propose is a particular case of (4.12), where D is a block diagonal matrix whose i-th diagonal block is defined as

$$D_i = \sum_{j \in \mathcal{N}_i^+} A_{ji}^T R_j^{-1} A_{ji} \,. \tag{4.13}$$

With a little abuse of notation, let us denote the Hessian of the cost function as $\nabla^2 J(\mathbf{x})$. From standard algebraic computations and given the quadratic structure of (4.7) for the cost J, it follows that $\nabla^2 J(\mathbf{x}) = A^T R^{-1} A$. The matrix $\nabla^2 J(\mathbf{x})$ can be partitioned as an $r \times r$ block matrix, where the i,j-th block $\left[\nabla^2 J(\mathbf{x})\right]_{ij}$ is given by $\frac{\partial J(\mathbf{x})}{\partial \mathbf{x}_i \partial \mathbf{x}_j}$. One can see that the block $[\nabla^2 J(\mathbf{x})]_{ij}$ is different from zero either if $j \in \mathcal{N}_i^+$ or if i and j are *two step neighbors* (i.e. there exists a agent k such that k is neighbor of both i and j). Furthermore, it can be shown that $D_i = [\nabla^2 J(\mathbf{x})]_{ii}$ is the i-th diagonal block of the cost function Hessian.
From (4.3), we can compute the gradient of the cost function

$$\nabla J(\mathbf{x}(t)) = A^T R^{-1} (A\mathbf{x}(t) - b) \,, \tag{4.14}$$

whose component associated with the i-th agent is

$$\left[\nabla J(\mathbf{x}(t))\right]_i = \sum_{j \in \mathcal{N}_i^+} A_{ji}^T R_j^{-1} \mathbf{g}_j(t+1) \,, \tag{4.15}$$

where the variable $\mathbf{g}_j(t+1)$ is defined as

$$\mathbf{g}_j(t+1) = \sum_{k \in \mathcal{N}_j^+} A_{jk} \mathbf{x}_k(t) - \mathbf{b}_j. \tag{4.16}$$

By plugging (4.13), (4.15) into (4.12), we can write the updating step performed by agent i as

$$\mathbf{x}_i(t+1) = \mathbf{x}_i(t) - \epsilon D_i^{-1} \sum_{j \in \mathcal{N}_i^+} A_{ji}^T R_j^{-1} \mathbf{g}_j(t+1), \tag{4.17}$$

which, in vector form, leads to

$$\mathbf{x}(t+1) = (\mathbb{I} - \epsilon D^{-1} A^T R^{-1} A)\mathbf{x}(t) + \epsilon D^{-1} A^T R^{-1} \mathbf{b}\,. \tag{4.18}$$

Observe that agent i, in order to perform (4.17), needs information coming from the neighbors of its neighbors, i.e. the two-step neighbors. As so, since we allow only communication between one step neighbors, to each iteration of the proposed algorithm it is necessary to perform two communications, the first to compute the $\mathbf{g}_i(t+1)$'s and the second to compute the $\mathbf{x}_i(t+1)$'s. The *distributed Block Jacobi algorithm* (denoted hereafter as the BJ algorithm) for quadratic functions is formally described as in Algorithm 4. Next, the convergence properties of the BJ algorithm are established.

Proposition 4.7.1. *Consider Problem (4.2) and the BJ algorithm. Assume*

$$\epsilon \leq \frac{2}{\|D^{-\frac{1}{2}} A^T R^{-1} A D^{-\frac{1}{2}}\|}\,. \tag{4.19}$$

Then, for any $\mathbf{x}(0) \in \mathbb{R}^n$, the trajectory $\mathbf{x}(t)$, generated by the BJ algorithm, converges exponentially fast to the minimizer of Problem (4.2), i.e.,

$$\|\mathbf{x}(t) - \mathbf{x}_{\text{opt.}}\| \leq c\rho^t\,,$$

for some constants $c > 0$ and $0 < \rho < 1$.

Proof. Consider the change of variables $\widetilde{\mathbf{x}} = \mathbf{x} - \mathbf{x}_{\text{opt.}}$. The cost function becomes

$$f(\widetilde{\mathbf{x}}) = \widetilde{\mathbf{x}}^T \frac{A^T R^{-1} A}{2} \widetilde{\mathbf{x}} + c,$$

while the evolution of $\widetilde{\mathbf{x}}$ is given by

$$\widetilde{\mathbf{x}}(t+1) = (\mathbb{I} - \epsilon D^{-1} A^T R^{-1} A)\widetilde{\mathbf{x}}(t)$$

By imposing $f(\widetilde{\mathbf{x}}(t+1)) - f(\widetilde{\mathbf{x}}(t)) < 0$, after some simple computations, it turns out that if equation (4.19) holds, then the algorithm reaches the minimizer of (4.2). ∎

Remark 4.7.2. It is worth noticing that to compute the step size upper bound of Eq.(4.19)

one needs complete knowledge of the network. It will be part of future research to find a possible distributed implementation of Eq.(4.19).

Algorithm 4 Distributed Block Jacobi algorithm.

Require: $\forall i \in \mathcal{V}_c$, store $A_{ij}, A_{ji}, R_j, j \in \mathcal{N}_i^+$.

1: **for** $t \in \mathbb{N}$ each $i \in \mathcal{V}_c$ **do**
2: sends $\mathbf{x}_i(t)$ to $j \in \mathcal{N}_i$;
3: receives $\mathbf{x}_j(t)$ from $j \in \mathcal{N}_i$;
4: updates $\mathbf{g}_i(t)$ by using (4.16)
5: sends $\mathbf{g}_i(t)$ to $j \in \mathcal{N}_i$;
6: receives $\mathbf{g}_j(t)$ from $j \in \mathcal{N}_i$;
7: updates $\mathbf{x}_i(t)$ by using (4.17)
8: **end for**

4.8 Robust Block-Jacobi algorithm

Algorithm 4 has been designed for the ideal case with no lossy communication. In the following, we generalize Algorithm 4 for the case with lossy communication, e.g. agent i could not receive information sent by some of its neighbors, due to communication failures. The modification of the algorithm is apparently naive, since we simply perform the same algorithm by using the last received data from its neighbors if a packet is not received.

Observe that, if agent i does not receive some of the packets transmitted by its neighbors, then it does not have the necessary information to perform the updates (4.16) and (4.17). To overcome this fact, similarly to what done in the case of the ADMM algorithm of Section 4.5, we assume agent i stores in memory the auxiliary variables $\mathbf{x}_j^{(i)}, \mathbf{g}_j^{(i)}$, $j \in \mathcal{N}_i$, which are equal, respectively, to the last packets $\mathbf{x}_j$ and $\mathbf{g}_j$ received by agent i from agent j; specifically, the dynamics of $\mathbf{x}_j^{(i)}, \mathbf{g}_j^{(i)}$ are

$$\mathbf{g}_j^{(i)}(t+1) = \begin{cases} \mathbf{g}_j(t) & \text{if } \gamma_j^{(i)}(t) = 1 \\ \mathbf{g}_j^{(i)}(t) & \text{if } \gamma_j^{(i)}(t) = 0 \end{cases} \tag{4.20}$$

$$\mathbf{x}_j^{(i)}(t+1) = \begin{cases} \mathbf{x}_i(t) & \text{if } \gamma_j^{(i)}(t) = 1 \\ \mathbf{x}_j^{(i)}(t) & \text{if } \gamma_j^{(i)}(t) = 0 \end{cases} . \tag{4.21}$$

As mentioned in the previous section, Algorithm 4 requires two communication rounds every iteration. In a lossy environment, in order to reduce the communication burden and the number of communication failures, we modify Algorithm 4 by letting the agents

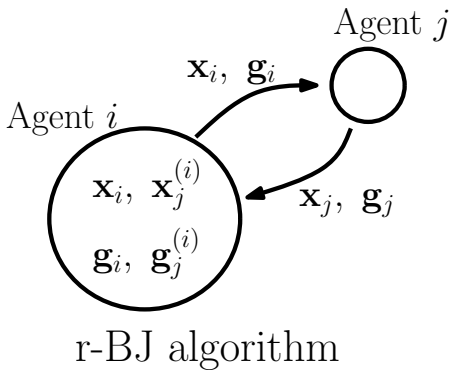

Figure 4.4: Communication scheme for the robust Block-Jacobi algorithm in the presence of packet losses and communication failures.

to communicate just once every iteration, transmitting together the $\mathbf{x}_i$'s and the $\mathbf{g}_i$'s. Agents i exploit $\mathbf{x}_j^{(i)}(t)$ and $\mathbf{g}_j^{(i)}(t)$ to update $\mathbf{g}_i$ and $\mathbf{x}_i$ as

$$\mathbf{g}_i(t+1) = \sum_{j \in \mathcal{N}_i^+} A_{ij} \mathbf{x}_j^{(i)}(t) - \mathbf{b}_i \,, \tag{4.22}$$

$$\mathbf{x}_i(t+1) = \mathbf{x}_i(t) - \epsilon D_i^{-1} \sum_{j \in \mathcal{N}_i^+} A_{ji}^T R_j^{-1} \mathbf{g}_j^{(i)}(t) \,. \tag{4.23}$$

As so, even in the scenario with no packet losses, this new algorithm does not exactly coincide with Algorithm 4, since a one-step delay is introduced in the computation of the variables $\mathbf{g}_i(t)$. The *robust block Jacobi algorithm* (hereafter referred to as r-BJ algorithm) for quadratic functions is formally described as in Algorithm 5.

Algorithm 5 Robust Block Jacobi algorithm.

Require: $\forall i \in \mathcal{V}_c$, store $A_{ij}, A_{ji}, R_j, j \in \mathcal{N}_i^+$.

for $t \in \mathbb{N}$ each $i \in \mathcal{V}_c$ **do**
  sends $\mathbf{x}_i(t), \mathbf{g}_i(t)$ to $j \in \mathcal{N}_i$;
  if $\gamma_j^{(i)}(t) = 1$ **then**
    receives $\mathbf{x}_j(t)$ and $\mathbf{g}_j(t)$ from $j \in \mathcal{N}_i$
  end if
  updates $\mathbf{g}_i(t)$ by using (4.22)
  updates $\mathbf{g}_j^{(i)}(t)$ by using (4.20)
  updates $\mathbf{x}_j^{(i)}(t)$ by using (4.21)
  updates $\mathbf{x}_i(t)$ by using (4.23)
end for

Figure 4.4 provides a pictorial representation of the stored and communicated variables by each node for both the r-BJ algorithm in presence of packet losses (compare with 4.3).

The convergence properties of the r-BJ algorithm are next established.

Proposition 4.8.1. *Let Assumption 4.4.1 hold. Consider Problem (4.2) and the r-BJ algorithm. There exists $\epsilon_{\max}$ such that, if $0 < \epsilon < \epsilon_{\max}$, then, for any $\mathbf{x}(0) \in \mathbb{R}^n$, the trajectory $\mathbf{x}(t)$, generated by the BJ algorithm, converges exponentially fast to the minimizer of Problem (4.2), i.e.,*

$$\|\mathbf{x}(t) - \mathbf{x}_{\text{opt.}}\| \leq c\rho^t$$

for some constants $c > 0$ and $0 < \rho < 1$.

The proof of Proposition 4.8.1 can be found in Appendix A.7 and basically relies on separation of time scales principle between the dynamics of the states $\mathbf{x}_i$'s and the auxiliary variables $\mathbf{x}_j^{(i)}$'s, $\mathbf{g}_i$'s and $\mathbf{g}_j^{(i)}$'s. Loosely speaking, if the update step-size parameter ϵ is small enough, the variation of the true states $\mathbf{x}_i$'s is so slow that, despite the lossy communication, the values of the copies $\mathbf{x}_j^{(i)}$'s and $\mathbf{g}_j^{(i)}$'s are essentially equal to the $\mathbf{x}_i$'s and the $\mathbf{g}_i$'s, respectively.

Remark 4.8.2 (Asynchronous communications). We would like to emphasize that this general model of packet losses includes as special cases asynchronous updates. In fact, asynchronous updates where only one node i updates its local variables based on the information received form its neighbors can be recovered by our algorithm assuming that all packets are lost except those from the neighbors of node i to node i itself. Also, our algorithm allows for multiple agents to communicate and perform updates at the same time, thus requiring no coordination. Finally, broadcast communication can be used since nodes do not need to establish reliable bidirectional communication as in gossip protocols.

4.9 Simulations

Here, we compare the Block Jacobi and the ADMM algorithms in both the ideal synchronous implementation with no packet losses and in the more realistic scenario with random packet losses.
The algorithms have been tested on the IEEE 123 nodes distribution grid benchmark assuming noise parameter in accordance with with the maximum measurement errors allowed by the IEEE standard C37.118-2005 (Martin, Hamai, Adamiak, Anderson, Begovic, Benmouyal, Brunello, Burger, Cai, Dickerson, Gharpure, Kennedy, Karlsson, Phadke, Salj, Skendzic, Sperr, Song, Huntley, Kasztenny, and Price, 2008), which specifies

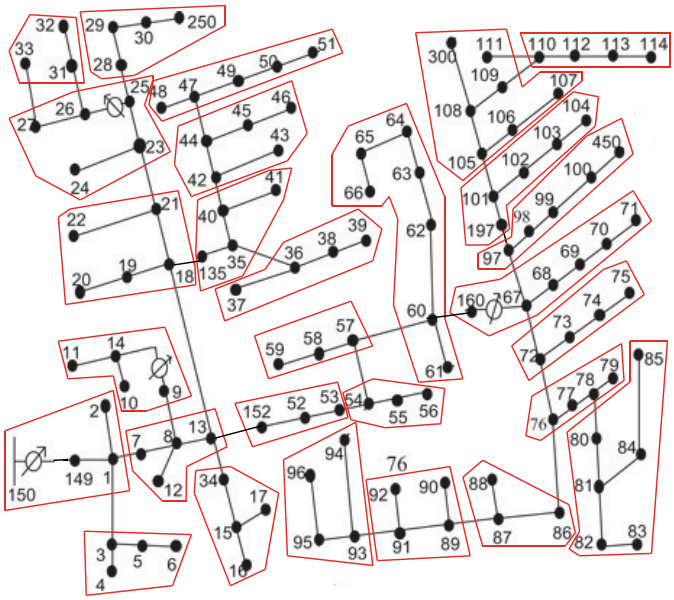

Figure 4.5: IEEE 123 node split into 28 non overlapping areas.

an aggregate constraint on the measurements error equal to

$$\frac{|\widetilde{u}_h - u_h|}{|u_h|} \leq 1\%\,, \qquad \frac{|\widetilde{i}_h - i_h|}{|i_h|} \leq 1\%\,.$$

For our purpose, the feeder has been divided into non overlapping areas as shown in Figure 4.5. The areas identify the agents of the communication graph $\mathcal{G}_c$. In particular, we assume there exists, for each area, a smart monitor able to sense the physical part of the grid which it has been assigned to, and to communicate with the monitors of the areas which are physically connected to it.

Here, the algorithms presented in Sections 4.5 and 4.7 are exploited to estimate the state of the electric grid which consists in the voltage real and imaginary parts at every node of the grid. We recall that we assume each node h of the grid is described by its bus voltage $u_h \in \mathbb{C}$ and its injected current $i_h \in \mathbb{C}$. Moreover, we assume to have at our disposal noisy measurements of the real and imaginary parts of voltage and current at every node of the grid as described in Section 4.3. These are retrieved with phasor measurement units (PMUs) placed at each node.

We partition A, b, x, w according to the splitting showed in Figure 4.5 and we formulate the state estimation problem as in (4.6).

Next, we compare the performance of the ADMM algorithm with the performance of the Block-Jacobi algorithm. In particular, we compare the algorithms in terms of the

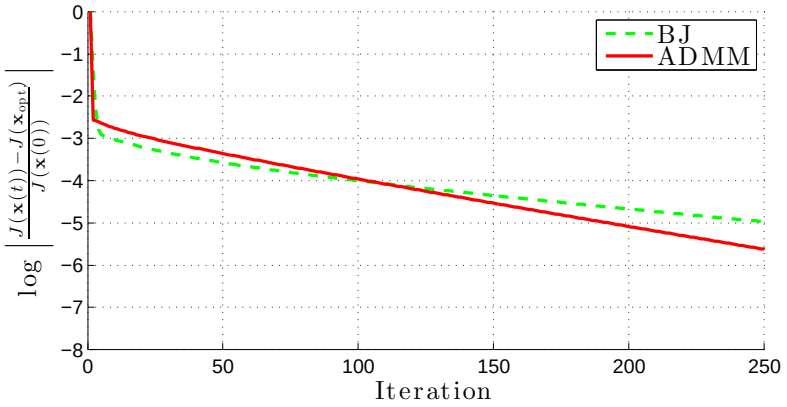

Figure 4.6: Comparison between the ADMM and the BJ algorithm with no packet losses, i.e. $\bar{\gamma} = 0$.

evolution of the normalized distance of the cost function from its minimal value, i.e.,

$$\left| \frac{J(\mathbf{x}(t)) - J(\mathbf{x}_{\text{opt.}})}{J(\mathbf{x}(0))} \right| , \tag{4.24}$$

where

$$J(\mathbf{x}(t)) = \frac{1}{2}(A\mathbf{x}(t) - \mathbf{b})^T R^{-1}(A\mathbf{x}(t) - \mathbf{b}) ,$$

being $\mathbf{x}(t)$ the estimated state at iteration t obtained by either the ADMM or the BJ algorithm.

In Figure 4.6, it is depicted the behavior of the two algorithms in the ideal scenario of reliable communications, i.e., no packet losses occur. The results reported have been obtained optimizing the performance of both algorithms over the parameters ρ and ϵ. We can see that the performance of the ADMM and BJ algorithms are comparable. We underline the fact that usually the ADMM better performs the BJ algorithm in the transient behavior. However, as can be seen from Figure 4.6, the outcome is case dependent.

A more realistic simulation is shown in Figure 4.7. Here, we compare the two algorithms in the lossy scenario. The parameters ϵ and ρ have been again manually optimized for best performance in both algorithms. To simulate the lossy communication channel, we assume that for all $i, j \in \mathcal{V}_c$ and for all $t \geq 0$

$$\mathbb{P}[\gamma_j^{(i)}(t) = 1] = 1 - \bar{\gamma},$$

where $\bar{\gamma}$ denotes the failure probability of the channel. Note that, according to the above probabilistic model, Assumption 4.4.1 is not necessarily satisfied, since we cannot ensure

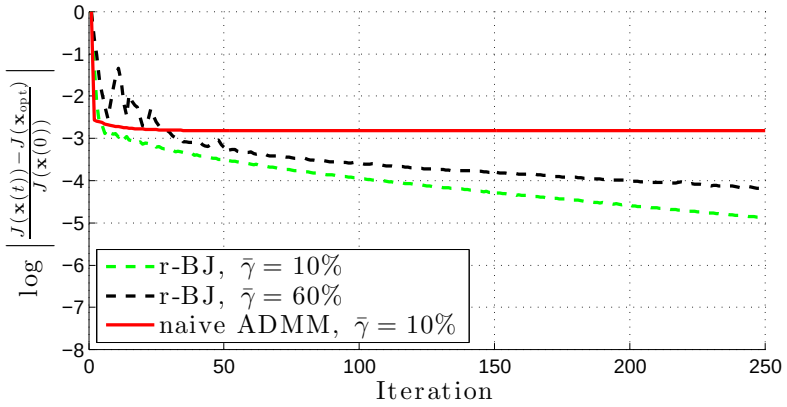

Figure 4.7: Comparison between the r-BJ and the naive ADMM algorithm for different packet loss probabilities $\bar{\gamma}$.

that within every T iterations every agent receives at least one packet from each of its neighbors. Nonetheless, the r-BJ algorithm still exhibit exponential convergence to the optimal solution even for high packet loss rates (60%). On the contrary, as expected, the ADMM fails to converge to the optimal point already in the presence of a moderate packet loss (10%).

4.10 Conclusions

In this part of the manuscript we dealt with the problem of state estimation in a smart power distribution grid which represents one first fundamental task to be solved in order to allow the possibility to tackle more challenging control and monitoring tasks. We presented two different possible solutions to solve the problem. The two solution are suitable to different scenarios. Both the strategies are based on a completely distributed and scalable communication architecture which, as particular case, allows the scenario of peer-to-peer communication where each electric bus is considered as a smart unit. In particular, the first is a modified version of the well known ADMM algorithm (Kekatos and Giannakis, 2013), suitable for the special case of quadratic cost function (Bolognani et al., 2014). It is fast but, as a limit, it requires synchronous and reliable communications. The second solution proposed is based on a generalized gradient descent strategy and in particular on a block-Jacobi type iteration (Todescato et al., 2015a). This second approach is resilient to communication failures and to asynchronous communication protocols.

5

Distributed Optimization: Stress Minimization via Optimal Reactive Power Control

"Live as if you were to die tomorrow. Learn as if you were to live forever."
M.K. Gandhi

In the last part of this thesis the problem of voltage support is considered. Usually in power systems, the voltage magnitudes are required to lie within predefined bounds to ensure good and safe operation of the grid. Here, we show that this security requirement, suggested by conventional wisdom, might be insufficient. Moreover, we propose a novel measure for the stress induced by the load profile on the network and formulate an optimization problem which maximizes the distance from voltage collapse through injection of reactive power.

5.1 Introduction and related work

Traditionally, the main purpose of voltage support is to maintain voltage magnitudes tightly within predetermined bounds (e.g., within 5% of some nominal level). Conventional wisdom suggests that such a tightly regulated voltage profile should also guarantee a secure system, operating far from static bifurcation instabilities such as voltage collapse.

Techniques for voltage support include shunt and static VAR compensation Baran and Wu (1989); Sode-Yome and Mithulananthan (2004), series compensation Moghavvemi and Faruque (2000), off-nominal transformer tap ratios Viawan, Sannino, and Daalder (2007), synchronous condensers Andersson (2008), and more inverters operating away from unity power factor Farivar, Neal, Clarke, and Low (2012); Na, Guannan, and Dahleh (2014). See Dixon, Moran, Rodriguez, and Domke (2005) for a survey on the topic.
A different approach, which represents a key direction in power system stability analysis, has been the development of indices quantifying a power network's proximity to voltage collapse. A broad overview of this large subfield can be found in Schlueter, Costi, Sekerke, and Forgey (1988); Van Cutsem (2000); Cañizares (2002); Van Cutsem and Vournas (1998); Eremia and Shahidehpour (2013). The existing approaches are largely based on numerical methods and lack of theoretical support. They often require either continuation power flow Hiskens et al. (2001) to identify the insolvability boundary, or repeated computation of loading margins in varying directions of parameter-space Dobson and Lu (1992).

In the following, first, we combine voltage support and distance to collapse, often analyzed separately, using the well known principle of reactive power injection. Indeed, as already stressed, usually, the ultimate goal in voltage support problems is the *security task* to confine the voltage magnitudes within predetermined bounds, as suggested by conventional engineering wisdom. Here, we follow an alternative approach: we define a particular measure for the *network stress*, i.e., the stress experienced by the network induced by the load profile. In particular, we begin our analysis from the recent article Simpson-Porco et al. (2015) where a sufficient and tight condition was presented for solvability of decoupled reactive power flow. This condition quantifies the proximity to voltage collapse by determining a nodal measure of network stress. Based on this condition we pursue a novel system-level formulation of optimal voltage support encoded as an optimization problem with *stress-minimization*, i.e., maximization of the distance to voltage collapse, as objective and subject to voltage security constraints. This approach allows us to match a *local* security requirement as well as a *system-level* stress-minimization objective encoding the distance to collapse. By exploiting an opportune linearized reformulation, our optimization formulation becomes convex and can be efficiently solved for the optimal injections. As second contribution, we also address the resource allocation and controller placement problem by regularizing our optimization problem with a convex proxy of the cardinality function. This sparsity-promoting formulation allows us to choose a desired trade-off between performance and a cost-effective solution. Finally, we present a possible distributed computation for the

solution of the stress minimization problem which is appealing to implement a distributed real-time feedback controller.

Compared to other approaches to voltage support problems our results do not rely on the assumption of a radial (i.e., acyclic) power grid topology Farivar et al. (2012); Na et al. (2014). This makes our approach appealing for both power transmission and distribution networks. Different from the reactive power compensation literature Bolognani and Zampieri (2013); Farivar et al. (2012); Bolognani, Carli, Cavraro, and Zampieri (2015) and from the voltage support literature Na et al. (2014), we seek stress minimization rather than optimal power flow (minimizing, e.g., losses) or voltage security tasks. Moreover, our formulation can nicely incorporate resource allocation and controller placement tasks.

5.2 Preliminaries

Before presenting the problem we deal with, for the sake of comprehension, we recall here some symbols and the notation used from the modeling Sections 2.5, 2.6 and 2.7. We refer the reader to the mentioned sections for a better clarification.

We model a smart grid as an undirected connected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where $|\mathcal{V}| = n$ and $|\mathcal{E}| = m$. We are interested in analyzing the relation between reactive power and voltage magnitude ($\mathbf{q}\boldsymbol{\nu}$). In particular we assume the following:

- highly inductive lines: according to Assumption 2.6.1, the electric lines are modeled as pure susceptances as commonly done in high voltage network;
- decoupled power flow: according to Assumption 2.6.2, the voltage angle differences between neighboring nodes in the electric graph $\mathcal{G}$ are negligible;
- node partitioning: according to the block partitioning introduced at the end of Section 2.4, we divide the set $\mathcal{V}$ into two subsets, namely $\mathcal{V}_\ell$ ($|\mathcal{V}_\ell| = n_\ell$) and $\mathcal{V}_g$ ($|\mathcal{V}_g| = n_g$), representing the load nodes, modeled as standard PQ nodes (see Section 2.5), and generator nodes, modeled as standard PV nodes (see Section 2.5), respectively.

Thanks to these and to Assumption 2.7.1 on the grounded suspectance matrix $B_{\ell\ell}$, the reactive power flow equations at the load nodes, described in Eq.(2.25b), read as

$$\mathbf{q}_\ell = -\operatorname{diag}(\boldsymbol{\nu}_\ell)(B_{\ell g}\boldsymbol{\nu}_g + B_{\ell\ell}\boldsymbol{\nu}_\ell)\,.$$

Moreover, recalling the Definitions 2.7.2–2.7.3 of the open circuit profile $\boldsymbol{\nu}_\ell^*$ and of the critical load matrix Q_{crit} which are respectively equal to

$$\boldsymbol{\nu}_\ell^* := -B_{\ell\ell}^{-1} B_{\ell g} \boldsymbol{\nu}_g \,, \qquad Q_{\text{crit}} := \frac{1}{4} \operatorname{diag}(\boldsymbol{\nu}_\ell^*) B_{\ell\ell} \operatorname{diag}(\boldsymbol{\nu}_\ell^*) \,.$$

the load RPFEs can be rewritten as

$$\mathbf{q}_\ell = -\operatorname{diag}(\boldsymbol{\nu}_\ell) B_{\ell\ell} (\boldsymbol{\nu}_\ell - \boldsymbol{\nu}_\ell^*) \,. \tag{5.1}$$

Finally, exploiting the normalized voltage profile $\boldsymbol{v}_\ell = \operatorname{diag}(\boldsymbol{\nu}_\ell^*)^{-1} \boldsymbol{\nu}_\ell$ defined in Eq.(2.28), Eq.(5.1) assumes the more compact form

$$\mathbf{q}_\ell = -4 \boldsymbol{v}_\ell Q_{\text{crit}} (\boldsymbol{v}_\ell - \mathbb{1}) \,. \tag{5.2}$$

We are now ready to present the problem at hand.

5.3 The voltage support problem

A common operational requirement is that the load bus voltage magnitudes must lie within a predefined percentage deviation, typically 5%, from a reference voltage. This tight clustering of voltages is due to the following reasons:

1. The loads and some system components are designed to operate with a voltage in a narrow region around the network base voltage;
2. A flat voltage profile minimizes the reactive flows and, consequently, minimizes the total power losses;
3. A flat profile usually reduces the sensitivity of the voltage profile respect to load changes (see Example 5.3.2);
4. Finally and most importantly, by conventional wisdom, a flat voltage profile indicates that the network is safe from voltage collapse.

We formalize this requirement by defining the *secure set*.

Definition 5.3.1 (Secure set)**.** Given a reference voltage $\nu_N \in \mathbb{R}_{>0}$, a percentage deviation $\alpha > 0$ and $\boldsymbol{\nu}_\ell^*$ as in (2.28), the *secure set* $\mathbb{V}$ is defined as

$$\mathbb{V} := \left\{ \boldsymbol{v}_\ell \in \mathbb{R}^{n_\ell} \;\middle|\; \frac{\|\operatorname{diag}(\boldsymbol{\nu}_\ell^*)\boldsymbol{v}_\ell - \nu_N \mathbb{1}\|_\infty}{\nu_N} \leq \alpha \right\} \,. \tag{5.3}$$

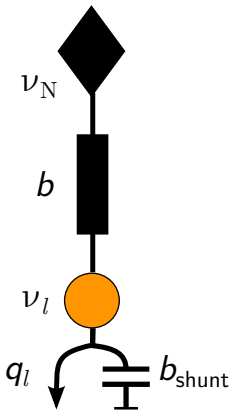

Figure 5.1: Example system. $\nu_N = 1$ [p.u.], $b = 4$ [S].

That is, if $\boldsymbol{v}_\ell \in \mathbb{V}$ is a solution to (5.2), then all voltages lie within α percent of the nominal voltage ν_N. While this represents a baseline operational requirement, under some circumstances it may not be sufficient to ensure safe grid operation. We present a simple example in which voltages remain within operational bounds, but the operating point is extremely sensitive to changes in load demands.

Example 5.3.2 (Security requirement inadequacy)**.** Consider the simple two-buses case study consisting of a load connected to a source at voltage $\nu_N = 1$, as illustrated in Figure 5.1. For the case where $b_{\text{shunt}} = 0$, Figure 5.2 plots the locus of solutions to (5.1) (blue solid blue) as q_ℓ is varied from 0 to Q_{crit}; this trace is often called a *nose curve*. Note that for a chosen q_ℓ, there may be two, one, or zero feasible solutions of (5.1). The boundary of the secure set is shown in dashed black. Also shown are the loading limits which ensure the high-voltage solution lies in the secure set (dashed orange), and the tangent line to the nose curve at the mid-point between the dashed orange lines (dashed magenta). This tangent line captures the sensitivity of the load voltage to changes in reactive power demand. From Figure 5.2, note that if q_ℓ is too large, the operating point does not lie within $\mathbb{V}$. A standard policy is then to support the voltage level by adjusting the shunt compensation, i.e., by increasing b_{shunt}. However, Figure 5.2 shows that the security requirement $v_\ell \in \mathbb{V}$ guarantees a "safe" distance to collapse, represented by the nose of the blue curve. Moreover, the sensitivity of the voltage to changes in load is small, meaning that relatively big changes in the load condition do not translate into big voltage changes. Conversely, in Figure 5.3 the security requirement is "dangerously" close to the nose of the curve. Moreover, the requirement alone does not discriminate the operating points within $\mathbb{V}$: indeed, it does not consider their proximity to the collapse point while, there are points (the left-most) which are preferable. Finally, the sensitivity

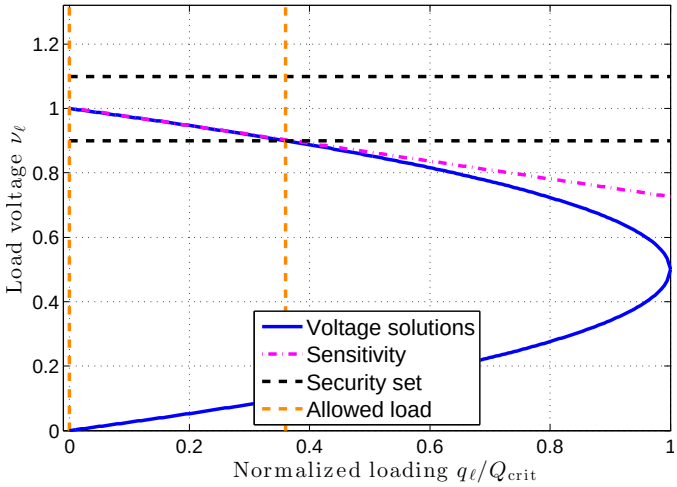

Figure 5.2: Absence of shunt, $b_{\text{shunt}} = 0$ [S]

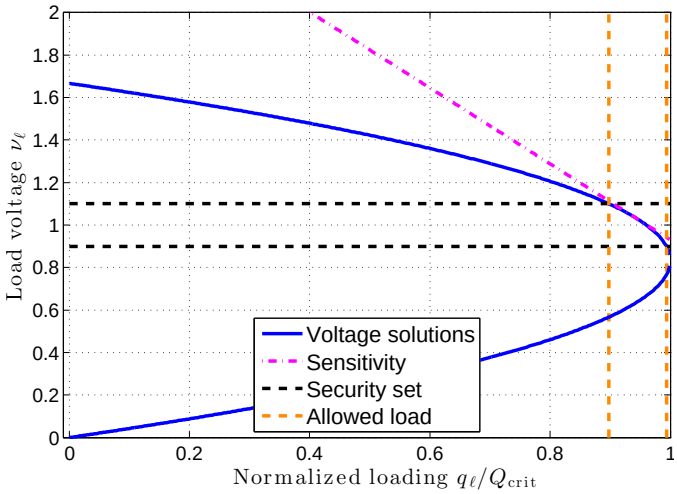

Figure 5.3: Presence of shunt, $b_{\text{shunt}} = 2.4$ [S].

line is steeper meaning that small changes in the load cause relatively big changes in the voltage. This affects the robustness of the network to small load changes. □

The previous analysis highlights that the security requirement $\boldsymbol{v}_\ell \in \mathbb{V}$ alone could be insufficient. Note that the most preferable operating point within $\mathbb{V}$ is the left-most voltage solution, identified by the intersection of the left orange line and the top black one. This is the feasible point farthest from voltage collapse. Moreover, by definition the left-most point on the blue curve represents the open-circuit $\boldsymbol{\nu}_\ell^*$ solution. Then, a simple intuition is that by minimizing the distance of the operating point from the open-circuit solution — constrained to the fact that the operating point must belong to $\mathbb{V}$ — we will maximize the voltage collapse stability margin of the network.

Based on the insights given by Example 5.3.2, we define the following measure quantifying the distance to collapse.

Definition 5.3.3 (Network Stress Measure)**.** Consider a power network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, described by the RPFEs (5.2), let the open-circuit profile $\boldsymbol{\nu}_\ell^*$ be as in (2.26) and $\boldsymbol{v}_\ell \in \mathbb{R}^{n_\ell}$ as in (2.28). Then, we define the *network stress measure* induced by the load as

$$J_{\text{stress}}(\boldsymbol{v}_\ell) := \|\boldsymbol{v}_\ell - \mathbb{1}\|_\infty \,. \tag{5.4}$$

Definition 5.3.3 is based on the intuition that the open-circuit profile $\boldsymbol{\nu}_\ell^*$ represents the network natural operating point in absence of loading, i.e., under "no stress". Conversely, when the network works close to the nose tip, i.e., the farthest point from $\boldsymbol{\nu}_\ell^*$ then, this is a "high-stress" scenario. In this sense, the stress function (5.4) quantifies the loading on the network conveniently expressed in the normalized profile $\boldsymbol{v}_\ell$.

In the following, we assume that a certain number of load buses can be equipped with additional controlled devices, e.g., synchronous condensers Andersson (2008) or photovoltaic panels connected to the grid through power inverters. We assume these devices can provide a controllable amount of reactive power support, and in the following we model them as controllable sources of reactive power $\mathbf{q}_{\text{ctrl}}$, subject to upper and lower operational bounds. Specifically, the RPFEs (5.2) are modified as

$$\mathbf{q}_\ell + \mathbf{q}_{\text{ctrl}} = -4\,\text{diag}(\boldsymbol{v}_\ell) Q_{\text{crit}} (\boldsymbol{v}_\ell - \mathbb{1}) \,, \tag{5.5}$$

where $\mathbf{q}_{\text{ctrl}} \in \mathbb{R}^{n_\ell}$ is such that $\mathbf{q}_{\text{ctrl,min}} \leq \mathbf{q}_{\text{ctrl}} \leq \mathbf{q}_{\text{ctrl,max}}$. If load bus $h \in \mathcal{V}_\ell$ is not equipped with a compensator, we set $\mathbf{q}_{\text{ctrl,min}} = \mathbf{q}_{\text{ctrl,max}} = 0$.

We now formulate our optimization problem of interest, which we refer to as the *Stress Minimization* problem.

Problem 5.3.4 (Stress Minimization)**.** *Let $J_{\text{stress}}(\boldsymbol{v}_\ell)$ be defined as in (5.4). Then the goal is to*

$$\begin{aligned} &\underset{\mathbf{q}_{\text{ctrl}} \in \mathbb{R}^{n_\ell}}{\text{minimize}} \quad J_{\text{stress}}(\boldsymbol{v}_\ell), \\ &\text{subject to} \quad \begin{cases} \boldsymbol{v}_\ell \in \mathbb{V} \,, \\ \mathbf{q}_{\text{ctrl,min}} \leq \mathbf{q}_{\text{ctrl}} \leq \mathbf{q}_{\text{ctrl,max}} \,, \\ \text{RPFEs (5.5)} \,. \end{cases} \end{aligned} \tag{5.6}$$

The main idea behind Problem 5.3.4 is that minimizing J_{stress} keeps the operating point of the network as far as possible from the tip of the nose curve, and thus far from

the point of voltage collapse. Observe that the previous security requirement $\boldsymbol{v}_\ell \in \mathbb{V}$ is incorporated into the problem as a hard constraint.

By being $\boldsymbol{v}_\ell$ related to $\mathbf{q}_{\text{ctrl}}$ through the quadratic equality constraints (5.5) Problem 5.3.4 is nonlinear and non-convex. In the following, we convexify this problem through the use of a power flow linearization.

5.4 Linear approximation

We recall here a suitable linearization, introduced at the end of Section 2.7, which had been first presented in Gentile et al. (2014). From (5.5), assuming $\|\mathbf{q}_\ell + \mathbf{q}_{\text{ctrl}}\| \sim 0$, we expect the normalized profile (2.28) to be $\boldsymbol{v}_\ell \simeq \mathbb{1}$ which would be the exact high voltage solution of (5.5) corresponding, by definition, to $\mathbf{q}_\ell + \mathbf{q}_{\text{ctrl}} = \mathbb{0}$. It follows that, by linearization around $\boldsymbol{v}_\ell = \mathbb{1}$, the solution of the RPFEs (5.5) is given by Eq.(2.29), i.e.,

$$\widehat{\boldsymbol{v}_\ell} = \mathbb{1} - \frac{1}{4} Q_{\text{crit}}^{-1} (\mathbf{q}_\ell + \mathbf{q}_{\text{ctrl}}) \,, \tag{5.7}$$

where, the contribution of the additional injection $\mathbf{q}_{\text{ctrl}}$ has been taken into account. That is, to first order the solution of (5.5) is given by a uniform component plus a deviation which is linear in the reactive injections.

5.5 Convexification of the Stress Minimization problem

Using (5.7), the cost (5.4) is approximated by

$$J_{\text{stress}}(\boldsymbol{v}_\ell) = \|\boldsymbol{v}_\ell - \mathbb{1}\|_\infty \propto \left\| Q_{\text{crit}}^{-1} (\mathbf{q}_\ell + \mathbf{q}_{\text{ctrl}}) \right\|_\infty . \tag{5.8}$$

Note that the approximated cost function is convex in the reactive power injections $\mathbf{q}_{\text{ctrl}}$. By exploiting (5.7) and thanks to some algebraic manipulations, it is possible to see that the security requirement $\widehat{\boldsymbol{v}_\ell} \in \mathbb{V}$ holds if and only if

$$\nu_N (1-\alpha)\, \text{diag}(\boldsymbol{\nu}_\ell^*)^{-1} \mathbb{1} \leq \widehat{\boldsymbol{v}_\ell} \leq \nu_N (1+\alpha)\, \text{diag}(\boldsymbol{\nu}_\ell^*)^{-1} \mathbb{1} \,. \tag{5.9}$$

Substituting for $\widehat{\boldsymbol{v}_\ell}$ from (5.7), (5.9) are equivalent to

$$\boldsymbol{\xi}_{\min} \leq -Q_{\text{crit}}^{-1} \mathbf{q}_{\text{ctrl}} \leq \boldsymbol{\xi}_{\max} \,, \tag{5.10}$$

where

$$\boldsymbol{\xi}_{\min} := 4 \left(\nu_N \,(1-\alpha)\, \text{diag}(\boldsymbol{\nu}_\ell^*)^{-1} \mathbb{1} - \mathbb{1} \right) + Q_{\text{crit}}^{-1} \mathbf{q}_\ell \,, \tag{5.11a}$$

$$\boldsymbol{\xi}_{\max} := 4\left(\nu_N\,(1+\alpha)\,\mathrm{diag}(\boldsymbol{\nu}_\ell^*)^{-1}\mathbb{1} - \mathbb{1}\right) + Q_{\mathrm{crit}}^{-1}\mathbf{q}_\ell\ . \tag{5.11b}$$

We have therefore expressed the security constraint $\widehat{\boldsymbol{v}_\ell} \in \mathbb{V}$ linearly in terms of the decision variables $\mathbf{q}_{\mathrm{ctrl}}$. The equivalent inequalities (5.10) are now linear in the $\mathbf{q}_{\mathrm{ctrl}}$'s.

We now present the convexified version of Problem 5.3.4.

Problem 5.5.1 (Convex Stress Minimization)**.** *Consider a power network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, governed by the RPFEs (5.5). Let ν_N, α, Q_{crit}, $\boldsymbol{\nu}_\ell^*$, $\mathbf{q}_{\mathrm{ctrl,min}}$ and $\mathbf{q}_{\mathrm{ctrl,max}}$ be as in the previous problem. Finally, define $\boldsymbol{\xi}_{\min}$ and $\boldsymbol{\xi}_{\max}$ as in (5.11a)–(5.11b), respectively. Then, the goal is*

$$\begin{aligned} &\underset{\mathbf{q}_{\mathrm{ctrl}}\in\mathbb{R}^{n_\ell}}{\text{minimize}} && \left\| Q_{\mathrm{crit}}^{-1}\left(\mathbf{q}_\ell + \mathbf{q}_{\mathrm{ctrl}}\right)\right\|_\infty, \\ &\text{subject to} && \begin{cases} \boldsymbol{\xi}_{\min} \le -Q_{\mathrm{crit}}^{-1}\mathbf{q}_{\mathrm{ctrl}} \le \boldsymbol{\xi}_{\max}, \\ \mathbf{q}_{\mathrm{ctrl,min}} \le \mathbf{q}_{\mathrm{ctrl}} \le \mathbf{q}_{\mathrm{ctrl,max}}. \end{cases} \end{aligned} \tag{5.12}$$

Remark 5.5.2 (On the stress measure). Aside from the linearization-based derivation in this subsection, the stress measure (5.12) is inspired by recent results Simpson-Porco et al. (2015) on the solvability of the decoupled reactive power flow equations (5.1). In Simpson-Porco et al. (2015) it has been shown that if $\|Q_{\mathrm{crit}}^{-1}\mathbf{q}_\ell\|_\infty < 1$, then (5.1) has a unique high-voltage solution safe from voltage collapse. For $\|Q_{\mathrm{crit}}^{-1}\mathbf{q}_\ell\|_\infty \ge 1$ voltage collapse may occur. From (5.8), we therefore see that $J_{\mathrm{stress}}(\boldsymbol{v}_\ell)$ quantifies the stress experienced by the network. □

Observe that in Problem 5.5.1 the cost (5.8) and the constraints (5.10) are convex in the decision variables. Indeed, the security constraints are linear in q and identify a polytope. Problem 5.5.1 can be efficiently approached via convex optimization techniques.

As final remark, it is worth noticing that both Problems 5.3.4 and 5.5.1 are offline centralized procedures which, as suggested by the formulation in Section 5.3, assume that either the full set of load buses or only an a priori assigned subset of them are equipped with controllable devices. The first scenario is impractical and economically unfeasible in large networks due to the large number of devices needed. The second scenario could likely lead to sub-optimal allocation of the resources if no specific allocation policies are used. In the following section, we ultimately refine the optimization problem in order to consider the allocation of the resources and simultaneously solve for the planning problem and the system-level stress minimization problem.

5.6 The Planning problem

Here, we propose a *sparsity-promoting* approach where, by tuning an additional parameter, the user is able to control the sparsity of the solution. In this way, we simultaneously solve the planning problem associated with the system-level stress minimization problem.

In order to account for the number of devices the cardinality function, $\mathrm{card}(\cdot)$, is a natural choice which, however, is discontinuous and non-convex. A convex approximation of $\mathrm{card}(\mathbf{q})$ is the *re-weighted* ℓ_1-norm (Boyd and Vandenberghe, 2004; Dörfler, Jovanović, Chertkov, and Bullo, 2014)

$$\|\mathrm{diag}(\mathbf{w}(\mathbf{q}))\mathbf{q}\|_1 = \sum_{h=1}^{n_\ell} w_h(q_h) q_h \,, \qquad w_h(q_h) := \frac{1}{|q_h| + \epsilon} \,, \tag{5.13}$$

where $0 < \epsilon \ll 1$. Adding equation (5.13) to the cost function (5.12), it is possible to formulate the following problem which we refer to as the *Sparse Stress Minimization* problem.

Problem 5.6.1 (Sparse Stress Minimization)**.** *Consider a power network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, governed by the RPFEs (5.5). Let ν_N, α, Q_{crit}, $\boldsymbol{\nu}_\ell^*$, $\mathbf{q}_{\mathrm{ctrl,min}}$, $\mathbf{q}_{\mathrm{ctrl,max}}$, $\boldsymbol{\xi}_{\mathrm{min}}$ and $\boldsymbol{\xi}_{\mathrm{max}}$ be defined as in the previous problem. Then, the goal is*

$$\begin{aligned} \underset{\mathbf{q}_{\mathrm{ctrl}} \in \mathbb{R}^{n_\ell}}{\text{minimize}} \quad & \left\| Q_{\mathrm{crit}}^{-1} \left(\mathbf{q}_\ell + \mathbf{q}_{\mathrm{ctrl}} \right) \right\|_\infty + \gamma \left\| \mathrm{diag}(\mathbf{w}(\mathbf{q}_{\mathrm{ctrl}})) \mathbf{q}_{\mathrm{ctrl}} \right\|_1 , \\ \text{subject to} \quad & \begin{cases} \boldsymbol{\xi}_{\mathrm{min}} \leq -Q_{\mathrm{crit}}^{-1} \mathbf{q}_{\mathrm{ctrl}} \leq \boldsymbol{\xi}_{\mathrm{max}}, \\ \mathbf{q}_{\mathrm{ctrl,min}} \leq \mathbf{q}_{\mathrm{ctrl}} \leq \mathbf{q}_{\mathrm{ctrl,max}}. \end{cases} \end{aligned} \tag{5.14}$$

The parameter γ in the cost function (5.14) can be used to promote sparsity of the solution. Obviously for $\gamma = 0$, Problem 5.6.1 reduces to Problem 5.5.1. By increasing the value of γ the user can force the solver to lean towards a more sparse solution. This automatically compels the solver to optimally allocate the resources in order to find the best trade-off between sparsity and system-level stress minimization.

5.7 Simulations: The planning problem and offline optimization

We now present a case study to show the effectiveness of planning and the offline optimization procedure proposed. The simulations, which refer to Problem (5.6.1), have been done in MATLAB using CVX (Grant and Boyd, 2014) to solve for the convex

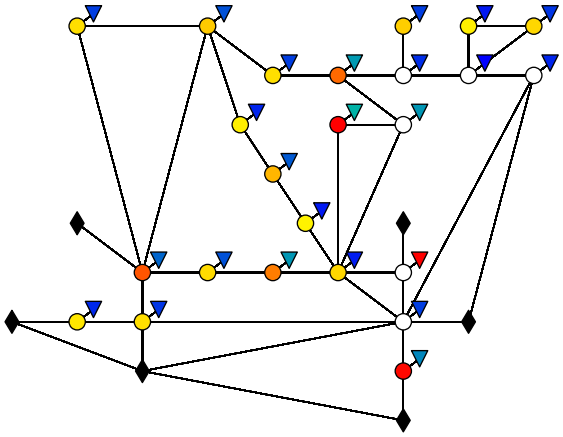

Figure 5.4: Placement for $\gamma = 0$

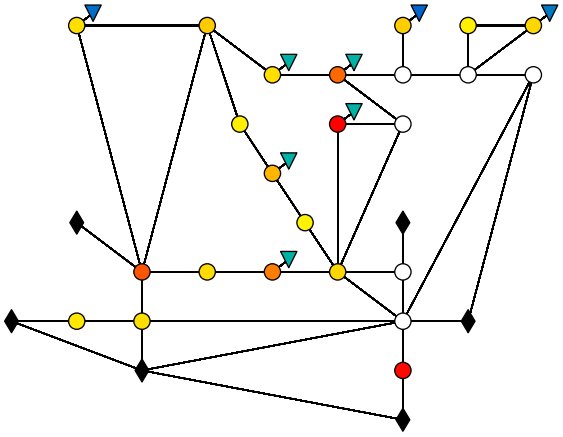

Figure 5.5: Placement for $\gamma = 4 \cdot 10^{-4}$

optimization. The plotted voltage profiles refer to the linearized solution (5.7) of the decoupled RPFEs (5.5). The test-bed consists of:

- IEEE 30 bus transmission grid;
- a reference voltage $\nu_N = 1$ [p.u.];
- a secure threshold $\alpha = 5\%$;
- capacity constraints $\mathbf{q}_{\text{ctrl,min}} = -0.5 \times \|\mathbf{q}_\ell\|_\infty$ and $\mathbf{q}_{\text{ctrl,max}} = 0.5 \times \|\mathbf{q}_\ell\|_\infty$.

Observe that consistent to power systems convention, we plot the profile $\boldsymbol{\nu}_\ell$ normalized with respect to the network base voltage ν_N instead of the normalized profile $\boldsymbol{v}_\ell$.

Figures 5.4–5.5–5.6 shows a graphical representation of the placement solution for increasing values of γ ($\gamma = 0$, $\gamma = 4 \times 10^{-4}$ and $\gamma = 8 \times 10^{-4}$, respectively). The color scheme for loads $\mathbf{q}_\ell$ and compensators $\mathbf{q}_{\text{ctrl}}$ is as follows:

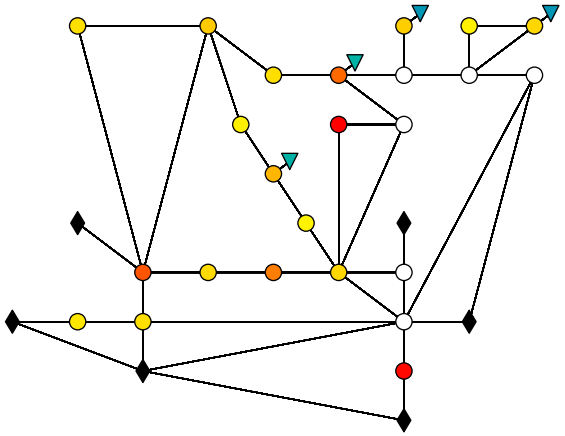

Figure 5.6: Placement for $\gamma = 8 \cdot 10^{-4}$

- reactive injections, i.e., positive values, has been plotted in blue-scale (from dark blue to light green, ■ − ■): the lighter the blue, the bigger the injection absolute value;
- reactive consumptions/absorptions, i.e., negative values, has been plotted in red-scale (from yellow to red, ■ − ■): the darker the red, the bigger the absolute value of the consumption;
- white node means zero injections/consumptions.

The symbols legend is as follows:

- black diamonds ♦ represent generators;
- circles ◯ represent loads;
- triangles ▽ represent reactive compensators.

First of all, for $\gamma = 0$ the solver places compensation everywhere. Moreover it can be seen that one compensator is red colored, meaning that it effectively absorbs reactive power. This occurs due to large reactive power injections at neighboring buses, which drive up voltage values across the network – additional reactive power must be absorbed to lower specific voltages and meet the security constraints. For an increasing value of γ sparsity promotion takes place. In particular, for a moderate value of the sparsity promoting parameter ($\gamma = 4 \times 10^{-4}$), the solver places all the compensators in the upper part of the grid where no generators are present and the voltage drop, due to the load, is stronger. By increasing the value of γ the sparsification process continues with the solver placing the compensation only in the top right part of the network where the heaviest

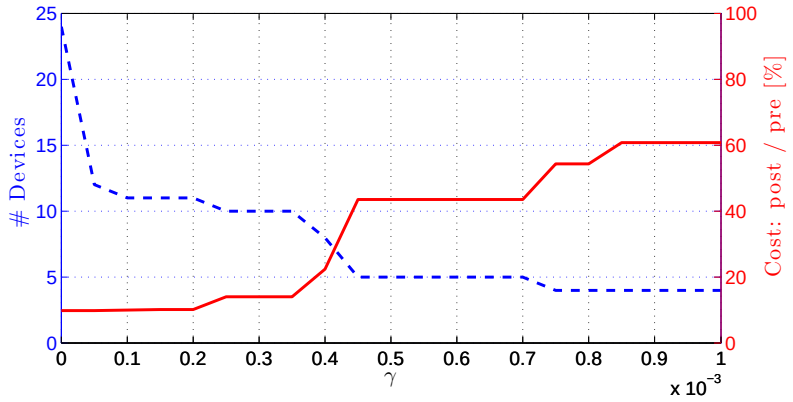

Figure 5.7: Behavior of the number of devices (left-blue axis) and of the ratio of the cost value after and before the optimization (right-red axis) as function of the sparsity parameter γ.

loading occurs.

Figure 5.7 shows, as a function of γ, the number of devices placed and the ratio between the value of the cost (5.14) after and before the optimization, namely

$$\frac{\left\|Q_{\text{crit}}^{-1}\left(\mathbf{q}_\ell + \mathbf{q}_{\text{ctrl}}\right)\right\|_\infty + \gamma \left\|\text{diag}(\mathbf{w}(\mathbf{q}_{\text{ctrl}}))\mathbf{q}_{\text{ctrl}}\right\|_1}{\left\|Q_{\text{crit}}^{-1}\mathbf{q}_\ell\right\|_\infty}.$$

It can be seen how, for increasing γ the final value of the cost increases since a smaller number of controllable units are less able to compensate the voltage profile. Figures 5.8 shows the linearized profiles $\boldsymbol{\nu}_\ell$ before and after the optimization for different γ. It can be seen that for increasing γ the profile $\boldsymbol{\nu}_\ell$ is less compensated, i.e., it is farther from the $\boldsymbol{\nu}_\ell^*$ profile. Finally, it is worth noting that stress minimization and classical voltage compensation do not coincide. Indeed, $\boldsymbol{\nu}_\ell^*$, in general, does not belong to $\mathbb{V}$.

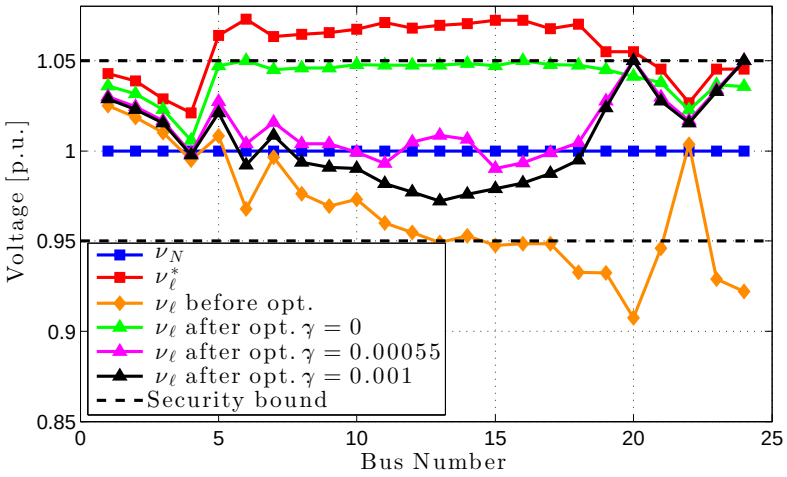

Figure 5.8: Voltage profiles for different values of γ.

5.8 Distributed online Stress Minimization through feedback control

The previous methods of Sections 5.5 and 5.6 are suitable only for offline optimization and planning. In this section we present a reformulation of Problem 5.5.1 which is suitable for an online distributed implementation. This is motivated by different reasons among which it is worth mentioning that:

- utilities could prefer not to share or widespread information to central operators because of privacy reasons;
- an online implementation can be naturally exploited as a distributed feedback controller in presence of time-varying loads, to reject disturbances, to increase the system robustness, and to track the optimal solution.

In the following, we assume that "smart agents" are embedded at all the grid's load buses. These are characterized by mild communication and computational capabilities. Moreover, they can communicate according to a communication graph which is designed to coincide with the electrical network. It is worth mentioning that, as will be clear later, even the load buses not equipped with a controllable compensator are required to share "smartness" capabilities. In the current formulation, the presence of the dense matrix Q_{crit}^{-1} in both the cost J_{stress} and the constraints (5.10) compromises the possibility to solve the stress minimization problem in a distributed fashion. Conversely, the matrix Q_{crit} is sparse and the graph induced by its sparsity pattern coincide with the electric graph connecting the loads. We take advantage of this structure to develop a distributed

algorithm to solve Problem 5.5.1. In this section we assume the planning of the resources has been solved.

Whereas the formulation (5.12) expressed the stress minimization compactly in injection coordinates $\mathbf{q}_{\text{ctrl}}$, now we derive the equivalent formulation in voltage coordinates to leverage on the sparsity of Q_{crit}. Let us start the analysis from (5.7) which reads as

$$\widehat{\boldsymbol{v}_\ell} = \mathbb{1} - \frac{1}{4} Q_{\text{crit}}^{-1}(\mathbf{q}_\ell + \mathbf{q}_{\text{ctrl}}) \ .$$

Then, we define the *deviation variable* $\mathbf{x}$ as

$$\mathbf{x} := -Q_{\text{crit}}^{-1}(\mathbf{q}_\ell + \mathbf{q}_{\text{ctrl}}) \,, \tag{5.15}$$

which represents the linear deviations of the voltages due to the load demand and the additional reactive input. Similar to what done in Section 5.4, from the definition of set $\widehat{\mathbb{V}}$ it is possible to obtain the security constraints expressed respect to $\mathbf{x}$. These are equal to

$$\mathbf{x}_{\min} \leq \mathbf{x} \leq \mathbf{x}_{\max} \,,$$

where

$$\mathbf{x}_{\min} := 4\left(\nu_N \,(1-\alpha)\, \text{diag}(\boldsymbol{\nu}_\ell^*)^{-1}\mathbb{1} - \mathbb{1}\right) \ ,$$

$$\mathbf{x}_{\max} := 4\left(\nu_N \,(1+\alpha)\, \text{diag}(\boldsymbol{\nu}_\ell^*)^{-1}\mathbb{1} - \mathbb{1}\right) \ .$$

From the definition of $\mathbf{x}$ it is clear that

$$\mathbf{q}_{\text{ctrl}} = -(Q_{\text{crit}}\mathbf{x} - \mathbf{q}_\ell) \,. \tag{5.17}$$

That is, since the matrix Q_{crit} is characterized by a sparsity pattern equivalent to that induced by the electric graph connecting the loads, the desired control inputs can be computed by means of a local exchange of information, namely the $\mathbf{x}$'s variables among electric neighbors. Additionally, from (5.17) it is easy to impose the capacity constraints, i.e., $\mathbf{q}_{\text{ctrl,min}} \leq \mathbf{q}_{\text{ctrl}} \leq \mathbf{q}_{\text{ctrl,max}}$. Then, Problem 5.6.1 is equivalently to

Problem 5.8.1 (Online Stress Minimization)**.**

$$\begin{aligned} \underset{\mathbf{x}\in\mathbb{R}^{n_\ell}}{\text{minimize}} \quad & \|\mathbf{x}\|_\infty \,, \\ \text{s.t.} \quad & \begin{cases} \mathbf{x}_{\min} \leq \mathbf{x} \leq \mathbf{x}_{\max}, \\ \mathbf{q}_{\text{ctrl,min}} \leq -(Q_{\text{crit}}\mathbf{x} - \mathbf{q}_\ell) \leq \mathbf{q}_{\text{ctrl,max}}. \end{cases} \end{aligned} \tag{5.18}$$

Now we point out three more issues related to Problem 5.8.1: (i) the ∞-norm is not everywhere differentiable and thus not suitable for a gradient-based iterative procedure; (ii) computing the cost in (5.18) requires knowledge of all x_i variables and (iii) in order to compute the derivative of the maximum function the index where the maximum is attained must be known. To overcome these issues, we propose a smooth approximation for the ∞-norm.

5.9 A smooth decomposable approximation of ∞-norm

Here, we present a continuously differentiable approximation for the ∞-norm which combines a smooth approximation for the maximum function, the softmax Bishop (2006), and a smooth approximation for the absolute value. This reads as

$$\begin{aligned} \tilde{f}_{\alpha,\epsilon}(\mathbf{x}) &:= \operatorname{softmax}_\alpha(|\mathbf{x}|^{1+\epsilon}), \quad 1 \ll \alpha, \ 0 < \epsilon \ll 1, \\ &= \frac{1}{\alpha} \log\left(\frac{1}{n_\ell} \sum\nolimits_{i=1}^{n_\ell} \exp\left(\alpha |x_i|^{1+\epsilon}\right)\right). \end{aligned} \tag{5.19}$$

The idea behind it is to exploit the super-linearity property of the exponential to let the maximum component of the vector x dominate the other components. The exponentiation is exploited to recover differentiability of the absolute value. The approximation $\tilde{f}_{\alpha,\epsilon}(x)$ approximates $\|\mathbf{x}\|_\infty$ in the following sense. The proof can be found in Appendix A.8.

Lemma 5.9.1 (Limit behavior of $\tilde{f}_{\alpha,\epsilon}$)**.** *Consider $\tilde{f}_{\alpha,\epsilon}$ as in (5.19). Then, it holds that*

$$\lim_{\substack{\alpha\to+\infty \\ \epsilon\to 0^+}} \tilde{f}_{\alpha,\epsilon}(\mathbf{x}) = \|\mathbf{x}\|_\infty .$$

Now, let us define

$$f_{\alpha,\epsilon}(\mathbf{x}) := n_\ell \exp(\alpha \tilde{f}_{\alpha,\epsilon}) = \sum_{i=1}^{n_\ell} \exp(\alpha |x_i|^{1+\epsilon}). \tag{5.20}$$

Notice that, due to the monotonicity of the log function in the softmax, it holds that

$$\operatorname*{argmin}_{\mathbf{x}} \tilde{f}_{\alpha,\epsilon}(\mathbf{x}) \equiv \operatorname*{argmin}_{\mathbf{x}} f_{\alpha,\epsilon}(\mathbf{x}). \tag{5.21}$$

Thanks to this simplification and in view of the distributed implementation, notice that the i-th component of the gradient vector of (5.20) is equal to

$$\frac{\partial f_{\alpha,\epsilon}(\mathbf{x})}{\partial x_i} = \alpha(1+\epsilon) \exp(\alpha |x_i|^{1+\epsilon}) |x_i|^\epsilon \operatorname{sgn}(x_i), \tag{5.22}$$

which depends only on the local state x_i of agent i. We will exploit this fact in our reformulation of Problem 5.8.1. Finally, the following lemma, whose proof may be found in Appendix A.9, characterizes the convexity of $f_{\alpha,\epsilon}(\mathbf{x})$

Lemma 5.9.2 (Strong convexity of $f_{\alpha,\epsilon}$)**.** *Consider the function $f_{\alpha,\epsilon} : \mathbb{R}^{n_\ell} \mapsto \mathbb{R}$ defined as in (5.20). Then, for all $0 < \epsilon \leq 1$ and for all $\alpha > 0$, $f_{\alpha,\epsilon}$ is strongly convex in $\mathbf{x}$.*

While Lemma 5.9.2 provides an asymptotic result, one finds that numerical issues are encountered for sufficient large (resp. small) values of α (resp. ϵ). In practice however, reasonably small values of α provide excellent results, with no noticeable numerical issues.

5.10 A distributed primal-dual feedback controller

We now reformulate Problem 5.8.1 by exploiting (5.20) and the equivalence in (5.21). By introducing the additional quantities

$$\Xi := \begin{bmatrix} \mathbb{I} \\ -\mathbb{I} \end{bmatrix}, \qquad \boldsymbol{\chi} := \begin{bmatrix} \mathbf{x}_{\max} \\ -\mathbf{x}_{\min} \end{bmatrix}, \qquad \boldsymbol{\varphi} := \begin{bmatrix} \mathbf{q}_{\text{ctrl,max}} \\ -\mathbf{q}_{\text{ctrl,min}} \end{bmatrix}, \tag{5.23}$$

the smooth approximation of Problem 5.8.1 reads as

Problem 5.10.1 (Smooth Stress Minimization)**.**

$$\begin{aligned} \underset{\mathbf{x}\in\mathbb{R}^{n_\ell}}{\text{minimize}} \quad & f_{\alpha,\epsilon}(\mathbf{x}), \\ \text{s.t.} \quad & \begin{cases} \Xi\mathbf{x} \leq \boldsymbol{\chi}\,, \\ \Xi(Q_{\text{crit}}\mathbf{x} - \mathbf{q}_\ell) \leq \boldsymbol{\varphi}\,. \end{cases} \end{aligned} \tag{5.24}$$

where, thanks to (5.23), the constraints are now in standard form.
One possible way to solve Problem 5.10.1 is the use of standard dual-ascent discrete-time algorithm. The Lagrangian function associated to Problem 5.10.1 is equal to

$$\mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = f(\mathbf{x}) + \boldsymbol{\lambda}^T (\Xi\mathbf{x} - \boldsymbol{\chi}) + \boldsymbol{\mu}^T (\Xi(Q_{\text{crit}}\mathbf{x} - \mathbf{q}_\ell) - \boldsymbol{\varphi}) \,.$$

Then, the dual-ascent consists of the iterative updates

$$\mathbf{x}(t+1) = \underset{\mathbf{x}}{\text{argmin}}\ \mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}(t), \boldsymbol{\mu}(t))\,, \tag{5.25a}$$

$$\boldsymbol{\lambda}(t+1) = [\boldsymbol{\lambda}(t) + \rho(\Xi\mathbf{x}(t+1) - \boldsymbol{\chi})]^+ \,, \tag{5.25b}$$

$$\boldsymbol{\mu}(t+1) = \Big[\boldsymbol{\mu}(t) + \rho(\Xi(Q_{\text{crit}}\mathbf{x}(t+1) - \mathbf{q}_\ell) - \boldsymbol{\varphi})\Big]^+ \tag{5.25c}$$

where ρ is the step size and $[\cdot]^+$ denotes the projection on the positive orthant, that is, $[a]^+ = a$, if $a > 0$, and 0 otherwise. We now state a convergence result whose proof can be found in Appendix A.10.

Proposition 5.10.2 (Convergence of the dual ascent algorithm)**.** *Consider Problem 5.10.1 and assume that Slater's condition holds, namely, a strictly feasible solution for (5.24) exists. Then, there exists $\overline{\rho}$ such that for any $\rho \leq \overline{\rho}$, the dual ascent algorithm (5.25) converges to the optimal solution of (5.24).*

Observe that the desired control injections can be computed by the corresponding agents from (5.17) as

$$\mathbf{q}_{\text{ctrl}}(t+1) = -(Q_{\text{crit}}\mathbf{x}(t+1) - \mathbf{q}_\ell)\,. \tag{5.26}$$

From (5.25), one can see that the proposed dual ascent algorithm is amenable of distributed implementation meaning that node i, to compute $x_i(t+1)$, $\mu_i(t+1)$, needs only information coming from neighboring nodes (with the sparsity pattern induced by the Q_{crit} matrix); while no exchange of information is needed to compute $\lambda_i(t+1)$. To be more precise, observe that (5.25a) is separable in the x_i variable; indeed, from the first order optimality condition must hold, for any $i \in \mathcal{V}_\ell$, that

$$\frac{\partial}{\partial x_i}\mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}(t), \boldsymbol{\mu}(t)) = 0\,.$$

which is equivalent to

$$\begin{aligned}\alpha(1+\epsilon)|x_i|^\epsilon \text{sgn}(x_i)\exp(\alpha|x_i|^{1+\epsilon}) = \\ \underbrace{-\sum_{j\in\mathcal{N}_i}\left([\Xi^T]_{ij}\lambda_j(t) + [Q_{\text{crit}}\Xi^T]_{ij}\mu_j(t)\right)}_{\zeta_i(t)}\,,\end{aligned} \tag{5.27}$$

where $\mathcal{N}_i := \{j \in \mathcal{V}_\ell \;:\; [Q_{\text{crit}}]_{ij} \neq 0\}$ represents the set of neighbors of node i. Observe that the right hand side of (5.27) is constant given the multipliers of the neighbors and that, thanks to strong convexity of f and then monotonicity of its first order derivative, (5.27) has always a unique real valued solution. Interestingly, for $\epsilon = 1$ a closed-form solution for (5.27) exists and is equal to

$$x_i(t+1) = \text{sgn}(\zeta_i(t))\frac{1}{\sqrt{2\alpha}}\sqrt{W\left(\frac{\zeta_i^2(t)}{2\alpha}\right)}\,, \tag{5.28}$$

where $W(\cdot)$ is the *Lambert W* or *ProductLog* function Corless, Gonnet, Hare, Jeffrey, and Knuth (1996) defined as the inverse of the function $g(z) = z\exp(z)$. Finally, note that,

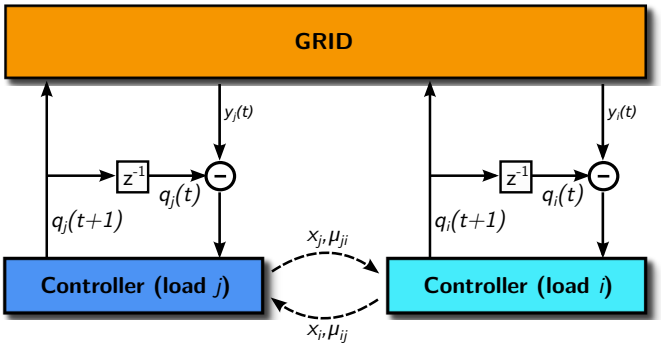

Figure 5.9: Illustrative scheme of the control scheme. Measurements $y(t)$ are gathered from the grid and used, together with $q(t)$, to update x. Neighbor controllers exchange local information. The block z^{-1} simply refers to the one-step delay operator.

each agent need to know some model information, namely its corresponding Q_{crit} entries which are related to the electric quantities connecting them to their neighbors. It is worth noticing the presence of the load demand $\mathbf{q}_\ell$ in both (5.25c) and (5.26). Usually, it is reasonable to have voltage and current monitoring at each bus. From this measurements it is possible to extract information about the total reactive load absorbed or injected at the bus. In particular, by defining $\mathbf{y} := \mathbf{q}_\ell + \mathbf{q}_{\text{ctrl}}$ as the aggregate contribution of the load together with the control input, from the bus monitoring it is possible to measure $\mathbf{y}$ rather than $\mathbf{q}_\ell$ independently. In order to compute (5.25c) and (5.26) it is sufficient to set

$$\mathbf{q}_\ell = \mathbf{y}(t) - \mathbf{q}_{\text{ctrl}}(t)\,,$$

where $\mathbf{y}(t)$ and $\mathbf{q}_{\text{ctrl}}(t)$ are the aggregate load measurements and the control input at the current t iteration. An illustration of the control loop is showed in Figure 5.9: the aggregate measurements are taken from the grid and are used together with the control input to compute the update $\mathbf{x}(t+1)$. Then, by using $\mathbf{x}(t+1)$, $\mathbf{y}(t)$ and $\mathbf{q}_{\text{ctrl}}(t)$, the new control input $\mathbf{q}_{\text{ctrl}}(t+1)$, used to actuate the grid, are computed.

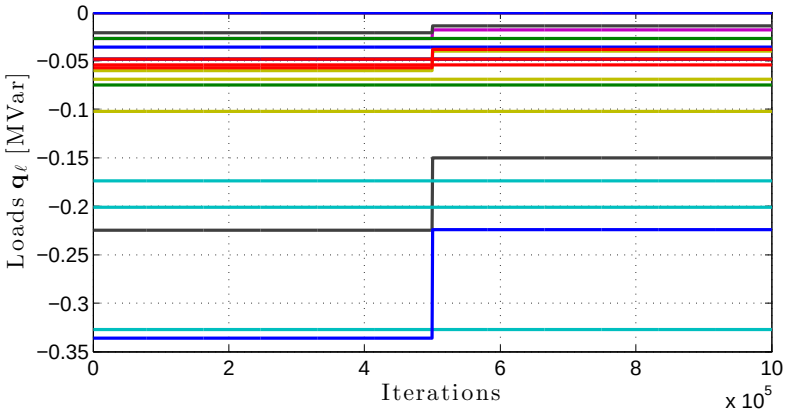

Figure 5.10: Dynamics of the loads $\mathbf{q}_\ell$. The different colored curves represent the time evolution of the loads at different buses.

5.11 Simulation: Distributed online feedback controller

We now present simulations illustrating the effectiveness of our distributed controller in the presence of time-varying loads. The setup is the same used in Section 5.7. However, differently from the previous set of simulations, we simulate the full coupled system which is solved at each iteration of the algorithm thanks to MATPOWER Zimmerman, Murillo-Sánchez, and Thomas (2011). We show the performance of the algorithm assuming we have access to aggregate measurements $\mathbf{y}$ for a fixed allocation of the resources. Specifically, there are 6 controllable units out of 24 total loads at the load buses number 1, 10, 15, 22, 23 and 24. Moreover, the controlled injections $\mathbf{q}_{\text{ctrl}}$ are saturated before actuating the electric grid, simulating the fact that they cannot exceed their capacity limits. The reactive loads randomly encounter, at half of the simulation time, a jump equal to 40% of their starting value (see Figure 5.10). All other parameters are held constant.

Figure 5.11 shows the evolution of the reactive injection returned by the algorithm. Figure 5.12 shows the corresponding evolution of bus voltages under the distributed controller (5.25)–(5.26). Interestingly the voltage solution of the full coupled PFEs, in steady-state, remains within the operational bounds. In particular the steady state difference between the solution of the full coupled PFEs and the linearized solution of the RPFEs given by (5.5) is of only 1.5%. This fact highlights the effectiveness of both the linearization and the control algorithm. Finally, Figure 5.13 shows the evolution of the error between the values of the injection $\mathbf{q}_{\text{ctrl}}(t)$ and the optimal value computed offline $\mathbf{q}_{\text{ctrl,opt}}$ as solution of Problem 5.5.1, as a function of the iterations, in logarithmic scale. Notice that the value of $\mathbf{q}_{\text{ctrl}}$ computed online converges to the optimal offline value even

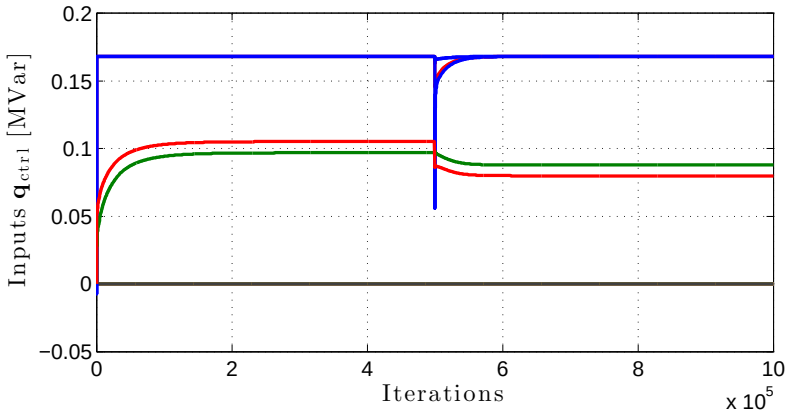

Figure 5.11: Dynamic of the control inputs $\mathbf{q}_{\mathrm{ctrl}}$. The different colored curves represent control inputs at different controlled buses.

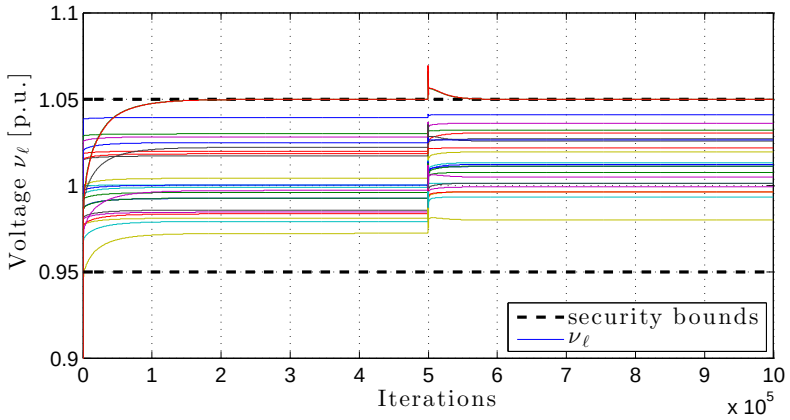

Figure 5.12: Evolution of the voltages corresponding to the coupled nonlinear power flow equations.

after the change in the loads.

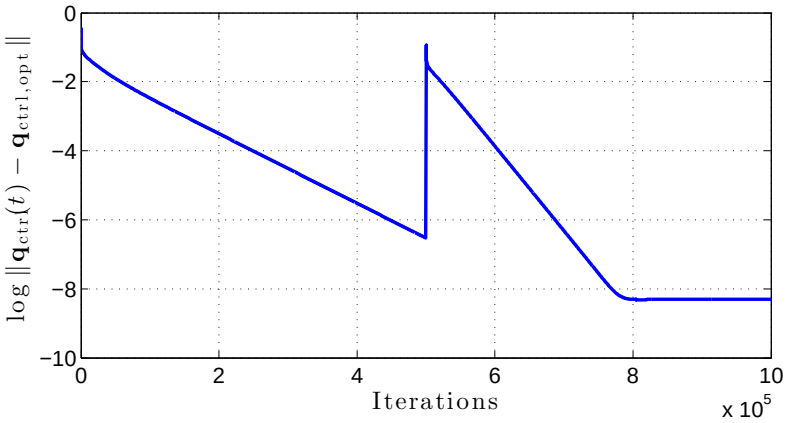

Figure 5.13: Evolution of the norm of the error between the online values of $\mathbf{q}_{\text{ctrl}}(t)$ and their optimal value $\mathbf{q}_{\text{ctrl,opt}}$ computed offline.

5.12 Conclusions

We considered the problem of voltage support via reactive power injections. Conversely to what suggested by conventional wisdom, we showed that the standard local security requirement of imposing the voltage magnitudes to lie within predefined bounds, might be inadequate. Building our analysis of this insights and the recent results on the solvability of the RPFEs, we proposed a novel optimization formulation whose cost function encodes the stress experienced by the grid, while the security requirements are imposed as constraints. Thanks to recent advances linearization of the reactive power flows, the problem becomes linear and convex and can be efficiently solved. In addition, we addressed the planning problem. Assuming that every load bus can be equipped by possibly expensive control unit might not a reasonable in large scale systems. We propose a reformulation of the stress minimization problem which, thanks to a regularization term, solves simultaneously for the optimal allocation of the resources and for the optimal injections. The final contribution regards a distributed online implementation to solve for the stress minimization problem. Thanks to a suitable reformulation, we presented a distributed algorithm, based on a dual decomposition, to solve for the stress minimization, which is appealing to implement a real-time feedback controller.

6
Conclusions

"I hated every minute of training, but I said, 'Don't quit. Suffer now and live the rest of your life as a champion.'"
Muhammad Ali

The last decades have foreseen a deep renovation of the energetic scenario. Energy is becoming more and more necessary for our lives but, as it is, the power production system is not environmentally and economically sustainable. The research community has the honor and the duty to seek for solutions which will lead to an improved global welfare from both perspectives. Technological advancements in renewable energy sources have demonstrated themselves to be promising and to represent the right path to follow to the final goal of a sustainable production system. However, if these methods will be exploited just as a mere replacement of old production plants, they are likely to be not sufficient in a future energetic scenario scattered of electric vehicles, large industries and smart buildings, all of them characterized by high quality and unpredictable high quantity of power demand. In such a complicated and tumultuous scenario, distributed energy resources (DERs) appear into place. However, they come at a price: network safety and stability. Here is where control theory and in particular, distributed optimization applied to large scale systems becomes of great importance. Indeed, if DERs have all the potential to overcome the issues due to high-quality and unpredictability of the demand,

they need to be suitably integrated into the grid to ensure and even increase network safety and avoid disastrous blackouts.

Innovation, technological advancements and research play a fundamental role in this renovating process. From the particular case of the power system research, *Smart Grids* represent an important concept offering a framework to diverse and equally important research fields, that must learn to coexist and complete themselves.

In this work, we analyzed different aspects regarding the control and optimization in a smart grid. Distributed implementability is the common denominator characterizing all the presented techniques. This aspect make them well suited for application in large scale systems. In particular, the manuscript is divided in four major parts.
The first part collects all the necessary preliminaries and modeling.
The second part discusses about the problem of devices synchronization. This is, in general, a quite broad topic in many fields of research, first of all in sensor networks. Interestingly enough, the problem can be cast in the localization framework of robotic networks as well. In this manuscript we follow this last approach. We presented two possible distributed solutions to the problem. The first is based on a consensus strategy; the second on a gradient descent one. Both the solutions are provably robust and resilient to communication failures and are amenable to be implemented using an asynchronous communication protocol.
The third part of this thesis regards the state estimation problem in smart grid. This is a fundamental task in control theory whose solution paves the path to face and tackle other control problems. We presented two possible distributed solutions. The first is based on a modified version of the classical ADMM algorithm. The second is based on a generalized gradient descent approach, namely a block-Jacobi procedure. Both the algorithms are built on a particular control and communication architecture which is flexible, decomposable and distributed.
The fourth and final part deals with a specific aspect in power systems: voltage support. To tackle this problem many different approach can be followed. We based our analysis on the well known principle of reactive power injection to support the voltage magnitudes which are usually required to lie within predefined bounds to ensure good operation and, as conventional wisdom suggests, safety of the grid. We showed that this *security requirement* is, in general, false. In particular, building our result on recent advances on reactive power flow solvability, we presented a novel measure for the stress induced over the network by the load profile to ultimately formulate an optimization problem which

minimizes the stress while is subject to the standard security constraints. Moreover, we considered the offline planning problem in order to simultaneously solve for the optimal allocation of the possibly scarce, due to the large scale of the system, resources and the optimal reactive injections. To complete the picture, we proposed a possible distributed implementation to solve for the optimization problem. This implementation is suitable to perform online control and can be implemented as a feedback controller for demand tracking in, e.g., a privacy preserving network.

Future developments regard the analysis of the problem assuming different decision variables, e.g., shunt/line capacitors. In this case, both the structure as well as the mathematical tractability of the problem deeply change. Even more interesting research directions would explicitly consider a lossy model for the power lines, comprehending conductances. This would ultimately leads to the study of the complete coupled power flow equations which represents the holy grail in power systems.

A

Appendix

"Be the change that you wish to see in the world."
M.K. Gandhi

A.1 Proof of Proposition 3.5.1

We present here the proof of Proposition 3.5.1. We recall the fact that we are considering a network of n nodes, running the a-CL algorithm under the *randomly persistent* communication scenario whose definition can be found in Section 3.3.
The proof of Proposition 3.5.1 is based on proving the convergence to consensus of (3.16) using the mathematical tools developed in Fagnani and Zampieri (2008).

Proposition. *Consider a* randomly persistent communicating network *of n nodes running the a-CL algorithm over a weakly connected measurement graph $\mathcal{G}_m$. Let $\mathcal{X}$ be defined as in (3.4). Let ϵ be such that $0 < \epsilon < 1/(2d_{\max}R_{\min}^{-1})$ where $d_{\max} := \max\{|\mathcal{N}_i|,\ i \in \mathcal{V}\}$ and $R_{\min} := \min\{R_{ij},\ (i,j) \in \mathcal{E}_m\}$. Moreover let $\widehat{x}_i$, $i \in \{1,\dots,n\}$, $\widehat{x}_j^{(i)}$, $j \in \mathcal{N}_i$, be initialized to any real number. Then the following facts hold true*

1. *the evolution $k \to \widehat{\mathbf{x}}(k)$ converges almost surely to an optimal solution $\mathbf{x}_{\text{opt.}} \in \mathcal{X}$,*

i.e., there exists $\alpha \in \mathbb{R}$ *such that*

$$\mathbb{P}\left[\lim_{k\to\infty} \widehat{\mathbf{x}}(k) = \mathbf{x}_{\text{opt., min}} + \alpha\mathbb{1}\right] = 1\,.$$

2. *the evolution* $k \to \widehat{\mathbf{x}}(k)$ *is exponentially convergent in mean-square sense, i.e., there exist* $c > 0$ *and* $0 \leq \rho < 1$ *such that*

$$\lim_{k\to\infty} \mathbb{E}\left[\|\widehat{\mathbf{x}}(k) - (\mathbf{x}_{\text{opt., min}} + \alpha\mathbb{1})\|^2\right] \leq c\rho^k \mathbb{E}\left[\|\widehat{\mathbf{x}}(0) - (\mathbf{x}_{\text{opt., min}} + \alpha\mathbb{1})\|^2\right]\,.$$

Proof. Let σ be the random process such that $\sigma(k)$ denotes the node performing the transmission action at the beginning of the $k+1$-th iteration. Clearly, in the randomized scenario we are considering, we have that, for $i \in \{1,\dots,n\}$, $\mathbb{P}[\sigma(k) = i] = \gamma_i$ for all k. Let

$$S(k) = \prod_{h=0}^{k} Q_{\sigma(h)}.$$

Observe that $S(k)$ inherits the same block structure of the matrices $\{Q_i\}_{i=1}^n$, namely we can write

$$S(k) = \begin{bmatrix} S_{11}(k) & S_{12}(k) \\ S_{21}(k) & S_{22}(k) \end{bmatrix}$$

As consequence of Theorem 3.1 in Fagnani and Zampieri (2008) the a-CL reaches almost surely consensus if and only if, for every i and j in $\mathcal{V}$

$$\mathbb{P}\left[\mathcal{E}_{ij}\right] = 1, \tag{A.1}$$

where

$$\mathcal{E}_{ij} = \{\exists \ell, \exists k \,|\, S_{i\ell}(k) S_{j\ell}(k) > 0\}\,.$$

Now observe that, since the measurement graph is weakly connected, then the communication graph is a connected undirected graph. This fact together with the fact the diagonal elements of $Q_{11}^{(i)}$ are all positive for any $i \in \{1,\dots,n\}$ implies that there exists almost surely $\bar{k}$ such that, for all $k' \geq \bar{k}$, all the elements of the matrix $S_{11}(k')$ are strictly greater than 0. Assume now, without loss of generality, that $\sigma(k') = i$, for $k' \geq k$. Then, since the i-th row of $S_{21}(k'+1)$ is equal to $\mathbb{1}_i \mathbb{1}_i^T S_{11}(k')$, it turns out that, all the elements of the i-th row of $S_{21}(k'+1)$ are strictly greater than 0. Moreover, it is easy to see that they will remain strictly greater than 0 also for any $k'' \geq k'$. Hence we can argue that, there exists almost surely, also a $\bar{k}'$ such that for all $k' \geq \bar{k}'$, all the elements of the matrix $S_{21}(k')$ are strictly greater than 0. It follows that the property stated in (A.1) is

satisfied for any $k \geq \bar{k}'$ and for any $\ell \in \{1, \dots, n\}$. This concludes the proof of item *1.*

Concerning item *2.*, we again resort to the results in Fagnani and Zampieri (2008). Let $\Omega = \mathbb{I} - \frac{1}{2n}\mathbb{1}\mathbb{1}^T$ where in this expression we assume that $\mathbb{I}$ is the $2n$-dimensional identity matrix and the vector $\mathbb{1}$ is $2n$-dimensional. From the results in Fagnani and Zampieri (2008), it follows that to study the rate of convergence of $\mathbb{E}\left[\|\boldsymbol{\xi}(k) - \alpha\mathbb{1}\|^2\right]$ is equivalent to study the convergence rate of $\mathbb{E}\|\Omega\boldsymbol{\xi}(k)\|^2$ and in particular of the linear recursive system

$$\Delta(t+1) = \mathbb{E}\left[Q_{\sigma(0)}^T \Delta(t) Q_{\sigma(0)}\right]$$

where $\Delta(0) = \Omega$. Observe that $\Delta(t)$ is the evolution of a linear dynamical system which can be written in the form

$$\Delta(t+1) = \Lambda(\Delta(t))$$

where $\Lambda : \mathbb{R}^{2n\times 2n} \to \mathbb{R}^{2n\times 2n}$ is given by

$$\Lambda(M) = \mathbb{E}\left[Q_{\sigma(0)}^T M Q_{\sigma(0)}\right] .$$

As highlighted in Fagnani and Zampieri (2008), the linear operator Λ can be represented by the matrix $L = \mathbb{E}[Q_{\sigma(0)} \otimes Q_{\sigma(0)}]^T$. Following the proof of Proposition 4.3 of Fagnani and Zampieri (2008), one can see that L^T is a primitive stochastic matrix which, therefore, has the eigenvalue 1 with algebraic multiplicity 1. Moreover, $L^T(\mathbb{1}\otimes\mathbb{1}) = (\mathbb{1}\otimes\mathbb{1})$ and $(\mathbb{1}\otimes\mathbb{1})(\Omega\otimes\Omega) = 0$, from which it follows that $\mathbb{E}\|\Omega\xi(k)\|^2 \leq c\,\mathrm{esr}(L^T)\mathbb{E}\|\Omega\xi(0)\|^2$. ■

A.2 Proof of Proposition 3.6.1

We present here the proof of Proposition 3.6.1. We recall the fact that we are considering a network of n nodes, running the a-CL algorithm under the *uniform persistent* communication scenario whose definition can be found in Section 3.3. Moreover, we assume Assumptions 3.3.3–3.3.4, which describe the type of packet losses and delays in the communication channel considered, hold.

Proposition. *Consider a* uniformly persistent communicating network *of n nodes running the a-CL algorithm over a weakly connected measurement graph $\mathcal{G}_m$. Let Assumptions 3.3.3 and 3.3.4 be satisfied. Let ϵ be such that $0 < \epsilon < 1/(2d_{max}R_{min}^{-1})$ where $d_{\max} := \{|\mathcal{N}_i|,\ i \in \mathcal{V}\}$ and $R_{\min} := \min\{R_{ij},\ (i.j) \in \mathcal{E}_m\}$. Moreover let $\widehat{x}_i$, $i \in \{1, \dots, n\}$, $\widehat{x}_j^{(i)}$, $j \in \mathcal{N}_i$, be initialized to any real number. Then the following facts hold true*

1. *the evolution $k \to \widehat{\mathbf{x}}(k)$ asymptotically converges to an optimal estimate $\mathbf{x}_{\text{opt.}} \in \mathcal{X}$,*

i.e., there exists $\alpha \in \mathbb{R}$ *such that*

$$\lim_{k\to\infty} \widehat{\mathbf{x}}(k) = \mathbf{x}_{\text{opt., min}} + \alpha \mathbb{1}\,;$$

2. *the convergence is exponential, namely, there exists* $c > 0$ *and* $0 \leq \rho < 1$ *such that*

$$\|\widehat{\mathbf{x}}(k) - (\mathbf{x}_{\text{opt., min}} + \alpha\mathbb{1})\| \leq c\rho^k \|\widehat{\mathbf{x}}(0) - (\mathbf{x}_{\text{opt., min}} + \alpha\mathbb{1})\|\,.$$

Proof. The proof follows from the statement of Proposition 1 in Nedic and Ozdaglar (2008) which for convenience has been reported in Appendix A.5 as Proposition A.5.5. To apply Proposition A.5.5 we show that a-CL algorithm in presence of delays and packet losses can be rewritten as a consensus with delays that satisfies Assumptions A.5.1, A.5.2, A.5.3, and A.5.4 reported in Appendix A.5.
To this aim, let $\delta_j(k) = \widehat{x}_j(k) - [\mathbf{x}_{\text{opt., min}}]_j$. Recalling that $\mathbf{x}_{\text{opt.,min}} = P\mathbf{x}_{\text{opt., min}} + \mathbf{b}$ where P is the stochastic matrix obtained form the terms p_{ij} and, according to (3.11) we have that, if $j \in \mathcal{V}'(k)$

$$\delta_j(k+1) := p_{jj}\delta_j(k) + \sum_{h\in\mathcal{N}_j} p_{jh}\delta_h(k'_h), \tag{A.2}$$

otherwise

$$\delta_j(k+1) = \delta_j(k).$$

The above equations describe a consensus algorithm on the variables $\delta_1, \ldots, \delta_n$ which satisfies Assumptions A.5.1, A.5.2, A.5.3 and A.5.4 reported in the Appendix. Indeed Assumption A.5.1 on the weights is trivially satisfied. Assumption A.5.2 follows from the facts that the communication graph $\mathcal{G}_c$ is connected, the network is uniformly persistent communicating and from Assumptions 3.3.3 and 3.3.4. Assumption A.5.3 is a consequence of the fact that the network is uniformly persistent communicating and Assumption 3.3.3; in our setup we have $B = L\tau$. Finally Assumption A.5.4 follows from Assumption 3.3.4 and equation (A.3). Hence the variables $\delta_1, \ldots, \delta_n$ converge exponentially to a consensus value α which, in turn, implies that $\widehat{\mathbf{x}}$ converge exponentially to $\mathbf{x}_{\text{opt., min}} + \alpha\mathbb{1}$. ■

A.3 Proof of Proposition 3.8.1

We report here the proof of Proposition 3.8.1 which states the following.

Proposition. *Consider a* randomly persistent communicating network *of* n *nodes run-*

ning the a-GL algorithm over a connected measurement graph $\mathcal{G}_m$. *Assume the weights* ϵ_i *are such that*

$$0 < \epsilon_i \leq \left(\sum_{j \in \mathcal{N}_i} \frac{1}{R_{ij}} \right)^{-1}, \qquad \forall\, i \in \mathcal{V},$$

and assume that $\widehat{x}_i$, $i \in \mathcal{V}$, $\widehat{x}_j^{(i)}$, $j \in \mathcal{N}_i$, *be initialized to any real number. Then the following facts hold true*

1. *the evolution* $k \to \widehat{\mathbf{x}}(k)$ *converges almost surely to an optimal solution* $\mathbf{x}_{\text{opt.}} \in \mathcal{X}$, *i.e., there exists* $\alpha \in \mathbb{R}$ *such that*

$$\mathbb{P}\left[\lim_{k \to \infty} \widehat{\mathbf{x}}(k) = \mathbf{x}_{\text{opt., min}} + \alpha \mathbb{1} \right] = 1\,,$$

2. *the evolution* $k \to \widehat{\mathbf{x}}(k)$ *is exponentially convergent in mean-square sense, i.e., there exist* $c > 0$ *and* $0 \leq \rho < 1$ *such that*

$$\lim_{k \to \infty} \mathbb{E}\left[\|\widehat{\mathbf{x}}(k) - (\mathbf{x}_{\text{opt., min}} + \alpha \mathbb{1})\|^2 \right] \leq c\rho^h \mathbb{E}\left[\|\widehat{\mathbf{x}}(0) - (\mathbf{x}_{\text{opt., min}} + \alpha \mathbb{1})\|^2 \right].$$

Proof. Consider the case $0 < \epsilon << \left(\sum_{j \in \mathcal{N}_i} \frac{1}{R_{ij}} \right)^{-1}$ then we can apply Corollary 3.2 of Fagnani and Zampieri (2008). Indeed for all $i \in \mathcal{V}$, we have that Q_i has all the diagonal terms strictly greater than zero. Moreover

$$\bar{Q} = \mathbb{E}\left[Q_i\right] = \sum_{i=1}^{n} \gamma_i Q_i = \sum_{i=1}^{n} \gamma_i \left[\mathbb{I} + \mathbb{1}_i \mathbb{1}_i^T (P - \mathbb{I}) \right] = \mathbb{I} + \Gamma(P - \mathbb{I}) = (\mathbb{I} - \Gamma) + \Gamma P$$

where $\Gamma = diag\{\gamma_1 \dots \gamma_n\}$. Observe that $p_{ij} \neq 0$ if $(i,h) \in \mathcal{E}_m$. Since $0 < \gamma_i < 1$ then, if $p_{ij} \neq 0$ then $q_{ij} \neq 0$. Additionally $\bar{Q}_{ii} > 0 \forall i$. Hence, from the fact that $\mathcal{G}_m$ is connected we have that $\bar{Q}$ is primitive. Corollary 3.2 in Fagnani and Zampieri (2008) implies that $\boldsymbol{\xi}(k) \to \boldsymbol{\xi}$ almost surely. Moreover, from the results of Section IV of Fagnani and Zampieri (2008) it follows that the convergence is exponential in mean-square. If there exists i such that $\epsilon_i = \left(\sum_{j \in \mathcal{N}_i} \frac{1}{R_{ij}} \right)^{-1}$, we can not longer apply Corollary 3.2 of Fagnani and Zampieri (2008) since in this case Q_i has the i-th term of the main diagonal which is equal to zero. We reason as follows. Assume that, at time k, the node transmitting is node i. Observe that, if $\widehat{x}_i(k+1) \neq \widehat{x}_i(k)$ then $J(k+1) < J(k)$. Hence we can apply Theorem 4.5 of Bullo, Carli, and Frasca (2010) to deduce that $\widehat{\mathbf{x}}(k) \to \mathcal{X}$. It remains to prove that $\widehat{\mathbf{x}}(k) \to \mathbf{x}_{\text{opt.}} \in \mathcal{X}$. Consider again the evolution $\boldsymbol{\xi}(k+1) = Q_i(k)\boldsymbol{\xi}(k)$. Let us write $\boldsymbol{\xi}(k) = \boldsymbol{\xi}_\perp(k) + \alpha(k)\mathbb{1}$, it is known that $\mathbb{1}^T \boldsymbol{\xi}_\perp(k) = 0$ and $\boldsymbol{\xi}_\perp(k) \to 0$ which leads

us to write

$$\boldsymbol{\xi}(k+1) = Q_i(k)\boldsymbol{\xi}(k) = Q_i(k)\left(\boldsymbol{\xi}_\perp(k) + \alpha(k)\mathbb{1}\right) = \boldsymbol{\xi}_\perp(k+1) + \alpha(k+1)\mathbb{1}\,,$$

and making the difference of the two last term we get

$$\left(\alpha(k+1) - \alpha(k)\right)\mathbb{1} + \boldsymbol{\xi}_\perp(k+1) - Q_i(k)\boldsymbol{\xi}_\perp(k) = 0$$

then, multiplying the left side for $\frac{1}{n}\mathbb{1}^T$

$$\alpha(k+1) - \alpha(k) = \frac{1}{n}\mathbb{1}^T Q_i(k)\boldsymbol{\xi}_\perp(k)\,.$$

From section IV of Fagnani and Zampieri (2008) we know that $\mathbb{E}\left[||\boldsymbol{\xi}_\perp(k)||^2\right] < c\rho^{-t}$. Then $\mathbb{E}\left[||\alpha(k+1) - \alpha(k)||^2\right] < \bar{c}\rho^{-t}$ and therefore $\alpha(h) \to \alpha$ constant. Then, in mean square we have that $\boldsymbol{\xi}(k) \to \alpha\mathbb{1}$ which is equivalent to $\widehat{\mathbf{x}}(k) \to \mathbf{x}_{\text{opt., min}} + \alpha\mathbb{1}$. ■

A.4 Proof of Proposition 3.9.1

We report here the proof of Proposition 3.9.1 which states the following.

Proposition. *Consider a* uniformly persistent communicating network *of n nodes running the a-GL algorithm over a* connected measurement graph $\mathcal{G}_m$. *Let Assumptions 3.3.3 and 3.3.4 be satisfied. Assume the weights ϵ_i are such that*

$$0 < \epsilon_i < \left(\sum_{j\in\mathcal{N}_i} \frac{1}{R_{ij}}\right)^{-1}, \qquad \forall\, i \in \mathcal{V},$$

and assume that $\widehat{x}_i$, $i \in \mathcal{V}$, $\widehat{x}_j^{(i)}$, $j \in \mathcal{N}_i$, be initialized to any real number. Then the following facts hold true

1. *the evolution $k \to \widehat{\mathbf{x}}(k)$ asymptotically converges to an optimal estimate $\mathbf{x}_{\text{opt.}} \in \mathcal{X}$, i.e., there exists $\alpha \in \mathbb{R}$ such that*

$$\lim_{k\to\infty} \widehat{\mathbf{x}}(k) = \mathbf{x}_{\text{opt., min}} + \alpha\mathbb{1}\,;$$

2. *the convergence is exponential, namely, there exists $c > 0$ and $0 \le \rho < 1$ such that*

$$\|\widehat{\mathbf{x}}(k) - (\mathbf{x}_{\text{opt., min}} + \alpha\mathbb{1})\,\| \le c\rho^h \|\widehat{\mathbf{x}}(0) - (\mathbf{x}_{\text{opt., min}} + \alpha\mathbb{1})\,\|.$$

Proof. The proof follows from the statement of Proposition 1 in Nedic and Ozdaglar (2008) which for convenience has been reported in Appendix A.5 as Proposition A.5.5. To apply Proposition A.5.5 we show that a-CL algorithm in presence of delays and packet losses can be rewritten as a consensus with delays that satisfies Assumptions A.5.1, A.5.2, A.5.3, and A.5.4 reported in Appendix A.5.

Consider the a-GL algorithm in presence of delays and packet losses. Let $\delta_i(k) = \widehat{x}_i(k) - [\mathbf{x}_{\text{opt.,min}}]_i$ and assume node i is performing the update at iteration k. Recalling that $\mathbf{x}_{\text{opt., min}} = P\mathbf{x}_{\text{opt., min}} + \mathbf{b}$ and, according to Eq. (A.4) we have that,

$$\delta_i(k+1) := p_{ii}\delta_i(k) + \sum_{j\in\mathcal{N}_i} p_{ij}\delta_j(k'_{ij})\,. \tag{A.3}$$

while for $j \neq i$

$$\delta_j(k+1) = \delta_j(k)\,.$$

The above equations describe a consensus algorithm on the variables $\delta_1, \ldots, \delta_n$ which satisfies the Assumptions A.5.1÷A.5.4. Indeed Assumption A.5.1 on the weights is trivially satisfied. Assumption A.5.2 follows from the facts that the communication graph $\mathcal{G}_c$ (which in the setup considered coincides with $\mathcal{G}_m$) is connected, the network is uniformly persistent communicating and from Assumptions 3.3.3 and 3.3.4 on the boundedness of the communication non idealities. Assumption A.5.3 is a consequence of the fact that the network is uniformly persistent communicating and Assumption 3.3.3; in our setup we have $B = L\tau$. Finally Assumption A.5.4 follows from Assumption 3.3.4 and equation (A.3). Hence the variables $\delta_1, \ldots, \delta_n$ converge exponentially to a consensus value α which, in turn, implies that $\widehat{\mathbf{x}}$ converge exponentially to $\mathbf{x}_{\text{opt., min}} + \alpha\mathbb{1}$. ∎

A.5 Additional Materials

In this appendix we review the result stated in Proposition 1 in Nedic and Ozdaglar (2008). In Nedic and Ozdaglar (2008), the authors consider the following consensus algorithm with delays[1]

$$x^i(k+1) = \sum_{j=1}^{m} a^i_j(k)x^j(k - t^i_j(k)) \tag{A.4}$$

[1]We adopt the notations of paper Nedic and Ozdaglar (2008).

where x^i denotes the state of node i, $i \in \{1, \ldots, m\}$, the scalar $t_j^i(k)$ is nonnegative and it represents the delay of a message from agent j to agent i, while the scalar $a_j^i(k)$ is a nonnegative weight that agent i assigns to a delayed estimate $x^j(s)$ arriving from agent j at time k. It is assumed that the weights $a_j^i(k)$ satisfy the following assumption.

Assumption A.5.1. There exists a scalar η, $0 < \eta < 1$ such that

1. $a_i^i(k) \geq \eta$ for all $k \geq 0$;
2. $a_j^i(k) \geq \eta$ for all $k \geq 0$, and all agents j whose (potentially delayed) information $x^j(s)$ reaches agent i during the k-th iteration;
3. $a_j^i(k) = 0$ for all $k \geq 0$ and j otherwise.
4. $\sum_{j=1}^m a_j^i(k) = 1$ for all i and k.

For any k let the information exchange among the agents may be represented by a directed graph $(\mathcal{V}, \mathcal{E}_k)$, where $\mathcal{V} = \{1, \ldots, m\}$ with the set $\mathcal{E}_k$ of directed edges given by $\mathcal{E}_k = \left\{(j,i)|a_j^i(k) > 0\right\}$. The authors impose a connectivity assumption on the agent system, which is stated as follows.

Assumption A.5.2. The graph $(\mathcal{V}, \mathcal{E}_\infty)$ is connected, where $\mathcal{E}_\infty$ is the set of edges (j,i) representing agent pairs communicating directly infinitely many times, i.e., $\mathcal{E}_\infty = \{(j,i)|(j,i) \in \mathcal{E}_k \text{ for infinitely many indices } k\}$.

Additionally it is assumed that the intercommunication intervals are bounded for those agents that communicate directly. Specifically,

Assumption A.5.3. There exists an integer $B \geq 1$ such that for every $(j,i) \in \mathcal{E}_\infty$, agent j sends information to its neighbor i at least once every B consecutive iterations.

Finally, it is assumed that the delays $t_j^i(k)$ in delivering a message from an agent j to any neighboring agent i is uniformly bounded at all times. Formally

Assumption A.5.4. Let the following hold:

1. $t_i^i(k) = 0$ for all agents i and all $k \geq 0$.
2. $t_j^i(k) = 0$ for all agents j communicating with agent i directly and whose estimates x^j are not available to agent i during the k-th iteration.
3. There is an integer B_1 such that $0 \leq t_j^i(k) \leq B_1 - 1$ for all agents i, j, and all k.

The result illustrated in Proposition 1 of Nedic and Ozdaglar (2008) is recalled in the following Proposition.

Proposition A.5.5. *Let Assumption A.5.1, A.5.2, A.5.3, A.5.4 hold. Then the sequences* $\left\{x^{(i)}(k)\right\}$, $i = 1, \ldots, m$, *generated by Equation (A.4) converge exponentially to a consensus.*

A.6 Proof of Proposition 4.5.1

In the following, to show the convergence of the evolution of Algorithm 3 to the optimal solution of Problem (4.2), we first show convergence of a more general partition based ADMM algorithm for the generic case of strictly convex cost under the assumption that the corresponding optimization problem has a unique solution. Then, we show that, for the particular case of quadratic cost, Algorithm 3 coincides with the more general formulation.

Let us consider the Lagrangian associated with Problem (4.9) defined as

$$\begin{aligned}\mathcal{L} = \sum_{i=1}^{r} \Bigg\{ & J_i(\mathbf{x}_i^{(i)}, \{\mathbf{x}_j^{(i)}\}_{j\in\mathcal{N}_i}) + \sum_{j\in\mathcal{N}_i} \left[\boldsymbol{\lambda}_i^{(i,j)T}\left(\mathbf{x}_i^{(i)} - \mathbf{z}_i^{(i,j)}\right) + \boldsymbol{\lambda}_j^{(i,j)T}\left(\mathbf{x}_j^{(i)} - \mathbf{z}_j^{(i,j)}\right)\right] \\ & + \sum_{j\in\mathcal{N}_i} \left[\boldsymbol{\mu}_i^{(i,j)T}\left(\mathbf{x}_i^{(i)} - \mathbf{z}_i^{(j,i)}\right) + \boldsymbol{\mu}_j^{(i,j)T}\left(\mathbf{x}_j^{(i)} - \mathbf{z}_j^{(j,i)}\right)\right] + \frac{\rho}{2}\sum_{j\in\mathcal{N}_i}\Big[\|\mathbf{x}_i^{(i)} - \mathbf{z}_i^{(i,j)}\|^2 \\ & + \|\mathbf{x}_j^{(i)} - \mathbf{z}_j^{(i,j)}\|^2 + \|\mathbf{x}_i^{(i)} - \mathbf{z}_i^{(j,i)}\|^2 + \|\mathbf{x}_j^{(i)} - \mathbf{z}_j^{(j,i)}\|^2\Big]\Bigg\}.\end{aligned}$$

Let us define the following local quantities

$$X^{(i)} = \begin{bmatrix} \mathbf{x}_i^{(i)} \\ \left\{\mathbf{x}_j^{(i)}\right\}_{j\in\mathcal{N}_i} \end{bmatrix}; \qquad Z^{(i)} = \begin{bmatrix} \left\{\mathbf{z}_i^{(i,j)}\right\}_{j\in\mathcal{N}_i} \\ \left\{\mathbf{z}_j^{(i,j)}\right\}_{j\in\mathcal{N}_i} \end{bmatrix};$$

$$\Lambda^{(i)} = \begin{bmatrix} \left\{\boldsymbol{\lambda}_i^{(i,j)}\right\}_{j\in\mathcal{N}_i} \\ \left\{\boldsymbol{\lambda}_j^{(i,j)}\right\}_{j\in\mathcal{N}_i} \end{bmatrix}, \qquad \mathcal{M}^{(i)} = \begin{bmatrix} \left\{\boldsymbol{\mu}_i^{(i,j)}\right\}_{j\in\mathcal{N}_i} \\ \left\{\boldsymbol{\mu}_j^{(i,j)}\right\}_{j\in\mathcal{N}_i} \end{bmatrix}.$$

Let t denote the iteration index, then the ADMM cycles through three steps:

(i) Dual ascent step on the $\Lambda's$ and $\mathcal{M}'s$ variables: Node i updates the variables

$\Lambda^{(i)}$ and $\mathcal{M}^{(i)}$ through a gradient ascent of $\mathcal{L}$ with step size ρ; precisely,

$$\begin{aligned}
\boldsymbol{\lambda}_i^{(i,j)}(t+1) &= \boldsymbol{\lambda}_i^{(i,j)}(t) + \rho\left(\mathbf{x}_i^{(i)}(t) - \mathbf{z}_i^{(i,j)}(t)\right), \\
\boldsymbol{\lambda}_j^{(i,j)}(t+1) &= \boldsymbol{\lambda}_j^{(i,j)}(t) + \rho\left(\mathbf{x}_j^{(i)}(t) - \mathbf{z}_j^{(i,j)}(t)\right), \\
\boldsymbol{\mu}_i^{(i,j)}(t+1) &= \boldsymbol{\mu}_i^{(i,j)}(t) + \rho\left(\mathbf{x}_i^{(i)}(t) - \mathbf{z}_i^{(j,i)}(t)\right), \\
\boldsymbol{\mu}_j^{(i,j)}(t+1) &= \boldsymbol{\mu}_j^{(i,j)}(t) + \rho\left(\mathbf{x}_j^{(i)}(t) - \mathbf{z}_j^{(j,i)}(t)\right).
\end{aligned}$$

(ii) Update of $X's$ variables: Node i updates the variable $X^{(i)}$ minimizing the augmented Lagrangian while keeping all the other variables fixed, namely,

$$\begin{aligned}
X^{(i)}(t+1) = \arg\min_{X^{(i)}} \Bigg\{ & J_i\left(\mathbf{x}_i^{(i)}, \{\mathbf{x}_j^{(i)}\}_{j\in\mathcal{N}_i}\right) + \sum_{j\in\mathcal{N}_i}\Big[\boldsymbol{\lambda}_i^{(i,j)T}(t+1)\left(\mathbf{x}_i^{(i)} - \mathbf{z}_i^{(i,j)}(t)\right) \\
& + \boldsymbol{\lambda}_j^{(i,j)T}(t+1)\left(\mathbf{x}_j^{(i)} - \mathbf{z}_j^{(i,j)}(t)\right)\Big] + \sum_{j\in\mathcal{N}_i}\Big[\boldsymbol{\mu}_i^{(i,j)T}(t+1)\left(\mathbf{x}_i^{(i)} - \mathbf{z}_i^{(j,i)}(t)\right) \\
& + \boldsymbol{\mu}_j^{(i,j)T}(t+1)\left(\mathbf{x}_j^{(i)} - \mathbf{z}_j^{(j,i)}(t)\right)\Big] + \frac{\rho}{2}\sum_{j\in\mathcal{N}_i}\Big[\|\mathbf{x}_i^{(i)} - \mathbf{z}_i^{(i,j)}(t)\|^2 \\
& + \|\mathbf{x}_j^{(i)} - \mathbf{z}_j^{(i,j)}(t)\|^2 + \|\mathbf{x}_i^{(i)} - \mathbf{z}_i^{(j,i)}(t)\|^2 + \|\mathbf{x}_j^{(i)} - \mathbf{z}_j^{(j,i)}(t)\|^2\Big]\Bigg\}.
\end{aligned}$$

(iii) Update of $Z's$ variables: Node i updates the variable $Z^{(i)}$ minimizing the augmented Lagrangian while keeping all the other variables fixed, namely,

$$\begin{aligned}
Z^{(i)}(t+1) = \arg\min_{Z^{(i)}} \Bigg\{ & \sum_{j\in\mathcal{N}_i}\Big[\boldsymbol{\lambda}_i^{(i,j)T}(t+1)\left(\mathbf{x}_i^{(i)}(t+1) - \mathbf{z}_i^{(i,j)}\right) \\
& + \boldsymbol{\lambda}_j^{(i,j)T}(t+1)\left(\mathbf{x}_j^{(i)}(t+1) - \mathbf{z}_j^{(i,j)}\right)\Big] + \sum_{j\in\mathcal{N}_i}\Big[\boldsymbol{\mu}_j^{(j,i)T}(t+1)\left(\mathbf{x}_j^{(j)}(t+1) - \mathbf{z}_j^{(i,j)}\right) \\
& + \boldsymbol{\mu}_i^{(j,i)T}(t+1)\left(\mathbf{x}_i^{(j)}(t+1) - \mathbf{z}_i^{(i,j)}\right)\Big] + \frac{\rho}{2}\sum_{j\in\mathcal{N}_i}\Big[\|\mathbf{x}_i^{(i)}(t+1) - \mathbf{z}_i^{(i,j)}\|^2 \\
& + \|\mathbf{x}_j^{(i)}(t+1) - \mathbf{z}_j^{(i,j)}\|^2 + \|\mathbf{x}_j^{(j)}(t+1) - \mathbf{z}_j^{(i,j)}\|^2 + \|\mathbf{x}_i^{(j)}(t+1) - \mathbf{z}_i^{(i,j)}\|^2\Big]\Bigg\}.
\end{aligned}$$

Proposition A.6.1. *Consider the partition-based ADMM algorithm described above. Let ρ be any real number. Then the trajectory $t \to \left\{X^{(i)}(t)\right\}$ converge exponentially to the optimal solution, namely, for $i \in \{1,\dots,r\}$, $\mathbf{x}_j^{(i)}(t) \to \mathbf{x}_{\text{opt.},j}$ for all $j \in \mathcal{N}_i$ and, in particular,*

$$\mathbf{x}_i^{(i)}(t) \to \mathbf{x}_{\text{opt.},i}\,.$$

Proof. Let X, Z, Λ and $\mathcal{M}$ be the vectors obtained by stacking together the vectors

$\left\{X^{(i)}\right\}_{i\in\mathcal{V}_c}$, $\left\{Z^{(i)}\right\}_{i\in\mathcal{V}_c}$, $\left\{\Lambda^{(i)}\right\}_{i\in\mathcal{V}_c}$ and $\left\{\mathcal{M}^{(i)}\right\}_{i\in\mathcal{V}_c}$, respectively, namely,

$$X = \begin{bmatrix} X^{(1)} \\ X^{(2)} \\ \vdots \\ X^{(r)} \end{bmatrix}, \quad Z = \begin{bmatrix} Z^{(1)} \\ Z^{(2)} \\ \vdots \\ Z^{(r)} \end{bmatrix}, \quad \Lambda = \begin{bmatrix} \Lambda^{(1)} \\ \Lambda^{(2)} \\ \vdots \\ \Lambda^{(r)} \end{bmatrix}, \quad \mathcal{M} = \begin{bmatrix} \mathcal{M}^{(1)} \\ \mathcal{M}^{(2)} \\ \vdots \\ \mathcal{M}^{(r)} \end{bmatrix}.$$

Now consider constraints in (4.9). From their linear structure of Eqs.(4.9), it follows that there exists suitable matrices A and B such that they can be rewritten as

$$AX + BZ = 0,$$

where the matrix A is such that A^TA is invertible. Hence Problem (4.9) can be equivalently formulated as

$$\begin{aligned} \min_{X} \quad & F(X) \\ \text{subject to} \quad & AX + BZ = 0 \end{aligned} \tag{A.5}$$

where $F(X) = \sum_{i=1}^{r} J_i(X^{(i)})$ is a convex function in X. Observe that, from the assumption that the solution is unique and from the connectness of the graph $\mathcal{G}_c$, it follows that Problem (A.5) admits an unique solution $X_{\text{opt.}}$ such that $\mathbf{x}^{(i)}_{\text{opt.},i} = \mathbf{x}^{(i)}_{\text{opt.},i}$, for all $j \in \mathcal{N}_i$, $i \in \mathcal{V}_c$. Moreover, Problem (A.5) can be solved by the standard ADMM algorithm illustrated in Bertsekas and Tsitsiklis (1989) which consists on the following three steps

(i) Dual ascent step on the Λ and $\mathcal{M}$ variables:

$$\begin{bmatrix} \Lambda(t+1) \\ \mathcal{M}(t+1) \end{bmatrix} = \begin{bmatrix} \Lambda(t) \\ \mathcal{M}(t) \end{bmatrix} + \rho\left(AX(t) + BZ(t)\right).$$

(ii) Update of X variable:

$$X(t+1) = \arg\min_{X} \left\{F(X) + \begin{bmatrix}\Lambda^T(t+1) & \mathcal{M}^T(t+1)\end{bmatrix}\left(AX + BZ(t)\right)\right\}.$$

(iii) Update of Z variable:

$$Z(t+1) = \arg\min_{Z} \left\{\begin{bmatrix}\Lambda^T(t+1) & \mathcal{M}^T(t+1)\end{bmatrix}\left(AX(t+1) + BZ\right)\right\}.$$

It is easy to see that the above steps correspond to the steps (i), (ii), (iii) of the partition-based ADMM algorithm previously described.

Proposition 4.2 in Bertsekas and Tsitsiklis (1989) guarantees, that under the assumptions that F is convex and the matrix A^TA is invertible, the trajectory $t \to X(t)$ converges to the optimal solution $X_{\text{opt.}}$. This concludes the proof. ■

For the case where the functions $J_i's$ have the particular quadratic structure illustrated in (4.3) it is possible to show that Algorithm 3 is equivalent to the partition-based ADMM algorithm described above. To do so, we next introduce the following lemmas.

Lemma A.6.2. *The update of the variable* $\mathbf{z}_k^{(i,j)}$, $k \in \{i,j\}$, *is given by*

$$\mathbf{z}_k^{(i,j)}(t+1) = \frac{\boldsymbol{\lambda}_k^{(i,j)}(t+1) + \boldsymbol{\mu}_k^{(j,i)}(t+1)}{2\rho} + \frac{\mathbf{x}_k^{(i)}(t+1) + \mathbf{x}_k^{(j)}(t+1)}{2}\,.$$

Proof. Without loss of generality assume that $k = i$. The value $\mathbf{z}_i^{(i,j)}(t+1)$ is computed by setting to zero the gradient of the function

$$f(\mathbf{z}_i^{(i,j)}) = \boldsymbol{\lambda}_i^{(i,j)T}(t+1)\left(\mathbf{x}_i^{(i)}(t+1) - \mathbf{z}_i^{(i,j)}\right) + \boldsymbol{\mu}_i^{(j,i)T}(t+1)\left(\mathbf{x}_i^{(j)}(t+1) - \mathbf{z}_i^{(i,j)}\right) + \\ + \frac{\rho}{2}\|\mathbf{x}_i^{(i)}(t+1) - \mathbf{z}_i^{(i,j)}\|^2 + \frac{\rho}{2}\|\mathbf{x}_i^{(j)}(t+1) - \mathbf{z}_i^{(i,j)}\|^2.$$

We have

$$\frac{\partial f(\mathbf{z}_i^{(i,j)})}{\partial \mathbf{z}_i^{(i,j)}} = -\boldsymbol{\lambda}_i^{(i,j)}(t+1) - \boldsymbol{\mu}_i^{(j,i)}(t+1) - \rho\left(\mathbf{x}_i^{(i)}(t+1) - \mathbf{z}_i^{(i,j)}\right) - \\ \rho\left(\mathbf{x}_i^{(j)}(t+1) - \mathbf{z}_i^{(i,j)}\right)\,.$$

From $\frac{\partial f(\mathbf{z}_i^{(i,j)})}{\partial \mathbf{z}_i^{(i,j)}} = 0$ we get the statement of the Lemma. ■

Lemma A.6.3. *If* $\boldsymbol{\lambda}_k^{(i,j)}(0) = -\boldsymbol{\mu}_k^{(j,i)}(0)$, $k \in \{i,j\}$, *then*

$$\boldsymbol{\lambda}_k^{(i,j)}(t) = -\boldsymbol{\mu}_k^{(j,i)}(t)\,,$$

for $t > 0$.

Proof. The statement of the Lemma can be proved by induction. Let $\boldsymbol{\lambda}_k^{(i,j)}(\ell) = -\boldsymbol{\mu}_k^{(j,i)}(\ell)$,

for $\ell = 0, \dots, t-1$. Then the updates take the form

$$\begin{aligned}
\boldsymbol{\lambda}_k^{(i,j)}(t) &= \boldsymbol{\lambda}_k^{(i,j)}(t-1) + \rho\left(\mathbf{x}_k^{(i)}(t-1) - \mathbf{z}_k^{(i,j)}(t-1)\right) \\
&= \boldsymbol{\lambda}_k^{(i,j)}(t-1) \\
&\quad + \rho\left(\mathbf{x}_k^{(i)}(t-1) - \frac{\boldsymbol{\lambda}_k^{(i,j)}(t-1) + \boldsymbol{\mu}_k^{(j,i)}(t-1)}{2\rho} - \frac{\mathbf{x}_k^{i}(t-1) + \mathbf{x}_k^{(j)}(t-1)}{2}\right) \\
&= \boldsymbol{\lambda}_k^{(i,j)}(t-1) + \rho\frac{\mathbf{x}_k^{i}(t-1) - \mathbf{x}_k^{(j)}(t-1)}{2}
\end{aligned}$$

where the second equality follows from the previous Lemma, while the second equality comes from the inductive hypothesis. In a similar way one can obtain

$$\boldsymbol{\mu}_k^{(j,i)}(t) = \boldsymbol{\mu}_k^{(j,i)}(t-1) + \rho\frac{\mathbf{x}_k^{j}(t-1) - \mathbf{x}_k^{(i)}(t-1)}{2}\,,$$

that, together with the inductive hypothesis, implies that $\boldsymbol{\lambda}_k^{(i,j)}(t) = -\boldsymbol{\mu}_k^{(j,i)}(t)$. ∎

Lemma A.6.4. *If* $\boldsymbol{\lambda}_k^{(i,j)}(0) = -\boldsymbol{\mu}_k^{(j,i)}(0)$, $k \in \{i, j\}$, *then*

$$\mathbf{z}_k^{(i,j)}(t) = \mathbf{z}_k^{(j,i)}(t)\,,$$

for $t \geq 0$.

Proof. From Lemma A.6.2 and Lemma A.6.3, we have

$$\mathbf{z}_k^{(i,j)}(t) = \frac{\boldsymbol{\lambda}_k^{(i,j)}(t) + \boldsymbol{\mu}_k^{(j,i)}(t)}{2\rho} + \frac{\mathbf{x}_k^{(i)}(t) + \mathbf{x}_k^{(j)}(t)}{2} = \frac{\mathbf{x}_k^{(i)}(t) + \mathbf{x}_k^{(j)}(t)}{2} = \mathbf{z}_k^{(j,i)}(t)\,.$$

∎

Lemma A.6.5. *If* $\boldsymbol{\lambda}_k^{(i,j)}(0) = \boldsymbol{\mu}_k^{(i,j)}(0)$, $k \in \{i, j\}$, *then*

$$\boldsymbol{\lambda}_k^{(i,j)}(t) = \boldsymbol{\mu}_k^{(i,j)}(t)\,,$$

for $t \geq 0$.

Proof. The Lemma can be prove by induction. Let us assume that $\boldsymbol{\lambda}_k^{(i,j)}(\ell) = \boldsymbol{\mu}_k^{(i,j)}(\ell)$ for $\ell = 0, \dots, t-1$. From Lemma A.6.2 and Lemma A.6.3, we have that

$$\mathbf{z}_k^{(i,j)}(t) = \frac{\mathbf{x}_k^{(i)}(t) + \mathbf{x}_k^{(j)}(t)}{2}$$

and, in turn, that

$$\boldsymbol{\lambda}_k^{(i,j)}(t) = \boldsymbol{\lambda}_k^{(i,j)}(t-1) + \rho\left(\mathbf{x}_k^{(i)}(t-1) - \frac{\mathbf{x}_k^{(i)}(t-1) + \mathbf{x}_k^{(j)}(t-1)}{2}\right),$$
$$\boldsymbol{\mu}_k^{(i,j)}(t) = \boldsymbol{\mu}_k^{(i,j)}(t-1) + \rho\left(\mathbf{x}_k^{(i)}(t-1) - \frac{\mathbf{x}_k^{(j)}(t-1) + \mathbf{x}_k^{(i)}(t-1)}{2}\right).$$

■

From Lemmas A.6.2 and A.6.3 we get the following corollary.

Corollary A.6.6. *If for $t \geq 0$, $\boldsymbol{\lambda}_k^{(i,j)}(t) = -\boldsymbol{\mu}_k^{(j,i)}(t) = \boldsymbol{\mu}_k^{(i,j)}(t) = -\boldsymbol{\lambda}_k^{(j,i)}(t)$, $k \in \{i,j\}$, then*

$$\mathbf{z}_k^{(i,j)}(t+1) = \mathbf{z}_k^{(j,i)}(t+1) = \frac{\mathbf{x}_k^{(i)}(t+1) + \mathbf{x}_k^{(j)}(t+1)}{2},$$
$$\boldsymbol{\lambda}_k^{(i,j)}(t+1) = \boldsymbol{\lambda}_k^{(i,j)}(t) + \frac{\rho}{2}\left(\mathbf{x}_k^{(i)} - \mathbf{x}_k^{(j)}\right).$$

The above Lemmas allow us to simplify the expression of the augmented Lagragian and, precisely, we can write that

$$\mathcal{L} = \sum_{i=1}^{r}\Bigg\{J_i(\mathbf{x}_i^{(i)}, \{\mathbf{x}_j^{(i)}\}_{j\in\mathcal{N}_i}) + \sum_{j\in\mathcal{N}_i}\left[2\boldsymbol{\lambda}_i^{(i,j)T}\left(\mathbf{x}_i^{(i)} - \mathbf{z}_i^{(i,j)}\right) + 2\boldsymbol{\lambda}_j^{(i,j)T}\left(\mathbf{x}_j^{(i)} - \mathbf{z}_j^{(i,j)}\right)\right]$$
$$+ \rho\sum_{j\in\mathcal{N}_i}\left[\|\mathbf{x}_i^{(i)} - \mathbf{z}_i^{(i,j)}\|^2 + \|\mathbf{x}_j^{(i)} - \mathbf{z}_j^{(i,j)}\|^2\right]\Bigg\}.$$

We have the following Lemma.

Lemma A.6.7. *The minimization over the vector $X^{(i)}$ is given by*

$$X_i^{(i)}(t+1) = \arg\min_{X^{(i)}}\left\{J_i(X^{(i)}) + \rho\left(X^{(i)}\right)^T M_i X^{(i)} - \left(X^{(i)}\right)^T B^{(i)}(t+1)\right\},$$

where $B^{(i)}(t+1)$ and M_i are defined as in the description of the algorithm.

Proof.

$$\begin{aligned}
&\arg\min_{X^{(i)}} \Bigg\{ J_i(X_i^{(i)}) + \sum_{j\in\mathcal{N}_i} \left[2\boldsymbol{\lambda}_i^{(i,j)T} \left(\mathbf{x}_i^{(i)} - \mathbf{z}_i^{(i,j)} \right) + 2\boldsymbol{\lambda}_j^{(i,j)} \left(\mathbf{x}_j^{(i)} - \mathbf{z}_j^{(i,j)} \right) \right] \\
&\quad + \rho \sum_{j\in\mathcal{N}_i} \left[\|\mathbf{x}_i^{(i)} - \mathbf{z}_i^{(i,j)}\|^2 + \|\mathbf{x}_j^{(i)} - \mathbf{z}_j^{(i,j)}\|^2 \right] \Bigg\} = \\
&\arg\min_{X^{(i)}} \Bigg\{ J_i(X^{(i)}) + 2\left(F^{(i)}(t+1) \right)^T X^{(i)} + \rho\, |\mathcal{N}_i|\, \|\mathbf{x}_i^{(i)}\|^2 + \rho \sum_{j\in\mathcal{N}_i} \|\mathbf{x}_j^{(i)}\|^2 + \\
&\quad - 2\rho \left(\mathbf{x}_i^{(i)} \right) \sum_{j\in\mathcal{N}_i} \mathbf{z}_i^{(i,j)}(t) - 2\rho \sum_{j\in\mathcal{N}_i} \left(\mathbf{x}_j^{(i)} \right)^T \mathbf{z}_j^{(i,j)}(t) \Bigg\}
\end{aligned}$$

where

$$F^{(i)}(t) = \begin{bmatrix} \sum_{j\in\mathcal{N}_i} \boldsymbol{\lambda}_i^{(i,j)}(t) \\ \boldsymbol{\lambda}_{j_1}^{(j_1,i)}(t) \\ \vdots \\ \boldsymbol{\lambda}_{j_{|\mathcal{N}_i|}}^{(j_{|\mathcal{N}_i|},i)}(t) \end{bmatrix}.$$

Let

$$M_i = \operatorname{diag}\left\{ |\mathcal{N}_i|\, \mathbb{I}_{m_i}, \mathbb{I}_{m_{j_1}}, \ldots, \mathbb{I}_{m_{j_{|\mathcal{N}_i|}}} \right\}.$$

We have that

$$\begin{aligned}
&J_i(X^{(i)}) + 2\left(F^{(i)}(t+1) \right)^T X^{(i)} + \rho\, |\mathcal{N}_i|\, \|\mathbf{x}_i^{(i)}\|^2 + \rho \sum_{j\in\mathcal{N}_i} \|\mathbf{x}_j^{(i)}\|^2 + \\
&\quad - 2\rho \left(\mathbf{x}_i^{(i)} \right) \sum_{j\in\mathcal{N}_i} \mathbf{z}_i^{(i,j)}(t) - 2\rho \sum_{j\in\mathcal{N}_i} \left(\mathbf{x}_j^{(i)} \right)^T \mathbf{z}_j^{(i,j)}(t) = \\
&J_i(X^{(i)}) + 2\left(F^{(i)}(t+1) \right)^T X^{(i)} + \rho \left(X^{(i)} \right)^T M_i X^{(i)} + -2\rho \left(\mathbf{x}_i^{(i)} \right)^T \sum_{j\in\mathcal{N}_i} \frac{\mathbf{x}_i^{(i)}(t) + \mathbf{x}_i^{(j)}(t)}{2} + \\
&\quad - 2\rho \sum_{j\in\mathcal{N}_i} \left(\mathbf{x}_j^{(i)} \right)^T \frac{\mathbf{x}_j^{(i)}(t) + \mathbf{x}_j^{(j)}(t)}{2}
\end{aligned}$$

We can write

$$
\begin{aligned}
&-2\rho\left(x_i^{(i)}\right)\sum_{j\in\mathcal{N}_i}\frac{\mathbf{x}_i^{(i)}(t)+\mathbf{x}_i^{(j)}(t)}{2}-2\rho\sum_{j\in\mathcal{N}_i}\left(\mathbf{x}_j^{(i)}\right)^T\frac{\mathbf{x}_i^{(i)}(t)+\mathbf{x}_i^{(j)}(t)}{2}=\\
&-\rho\left(X^{(i)}\right)^T M_iX^{(i)}(t)-\rho\left(\mathbf{x}_i^{(i)}\right)^T\sum_{j\in\mathcal{N}_i}\mathbf{x}_i^{(j)}(t)-\rho\sum_{j\in\mathcal{N}_i}\left(\mathbf{x}_j^{(i)}\right)^T\mathbf{x}_j^{(j)}(t)=\\
&-2\rho\left(X^{(i)}\right)^T M_iX^{(i)}(t)-\rho\left(\mathbf{x}_i^{(i)}\right)^T\sum_{j\in\mathcal{N}_i}\left(\mathbf{x}_i^{(j)}(t)-\mathbf{x}_i^{(i)}(t)\right)\\
&\qquad\qquad-\rho\sum_{j\in\mathcal{N}_i}\left(\mathbf{x}_j^{(i)}\right)^T\left(\mathbf{x}_j^{(j)}(t)-\mathbf{x}_j^{(i)}(t)\right)=\\
&-2\rho\left(X^{(i)}\right)^T M_iX^{(i)}(t)+\left(X_i^{(i)}\right)^T G^{(i)}(t)
\end{aligned}
$$

where $G^{(i)}$ is defined as

$$
\begin{aligned}
G_i^{(i)}(t)&=\rho\sum_{j\in\mathcal{N}_i}\left(\mathbf{x}_i^{(i)}(t)-\mathbf{x}_i^{(j)}(t)\right),\\
G_{j_h}^{(i)}(t)&=\rho\left(\mathbf{x}_{j_h}^{(i)}(t)-\mathbf{x}_{j_h}^{(j_h)(t)}\right),\quad 1\le h\le|\mathcal{N}_i|.
\end{aligned}
$$

Summarizing we have that

$$
\begin{aligned}
X_i^{(i)}(t+1)=\arg\min_{X^{(i)}}\Big\{&J_i(X^{(i)})+\rho\left(X^{(i)}\right)^T M_iX^{(i)}+\left(2F^{(i)}(t+1)\right)^T X^{(i)}\\
&-2\rho\left(X^{(i)}\right)^T M_iX^{(i)}(t)+\left(X_i^{(i)}\right)^T G^{(i)}(t)\Big\}.
\end{aligned}
$$

Hence

$$
X_i^{(i)}(t+1)=\arg\min_{X^{(i)}}\left\{J_i(X^{(i)})+\rho\left(X^{(i)}\right)^T M_iX^{(i)}-\left(X^{(i)}\right)^T B^{(i)}(t+1)\right\},
$$

where

$$
B^{(i)}(t+1)=2\rho M_iX^{(i)}(t)-G^{(i)}(t)-2F^{(i)}(t+1).
$$

■

A.7 Proof of Proposition 4.8.1

The proof of Proposition 4.8.1 relies on the time scale separation of the dynamic of the x_i's and of the auxiliary variables $\mathbf{x}_j^{(i)}$'s, $\mathbf{g}_i$'s and $\mathbf{g}_j^{(i)}$'s, and fully exploits the following

Lemma A.7.1. *Consider the dynamical system*

$$\begin{bmatrix}\mathbf{x}(t+1)\\ \mathbf{y}(t+1)\end{bmatrix} = \begin{bmatrix}\mathbb{I} & \epsilon B\\ C(t) & F(t)\end{bmatrix}\begin{bmatrix}\mathbf{x}(t)\\ \mathbf{y}(t)\end{bmatrix} \tag{A.6}$$

Let the following assumptions hold

1. $\forall \mathbf{x}, \exists! \mathbf{y} : \mathbf{y} = C(t)\mathbf{x} + F(t)\mathbf{y}, \forall t$. *As a consequence, we can write* $\mathbf{y} = G\mathbf{x}$;
2. *the system*
$$\mathbf{z}(t+1) = F(t)\mathbf{z}(t) \tag{A.7}$$
is asymptotically stable;
3. *the system*
$$\dot{\boldsymbol{\xi}}(t) = -BG\boldsymbol{\xi}(t) \tag{A.8}$$
is asymptotically stable.

Then, there exists $\epsilon_{\max}$, *with* $0 < \epsilon < \epsilon_{\max}$ *such that the origin is an asymptotically stable equilibrium for the system* (A.6).

We are now ready to prove Proposition 4.8.1 which states the following.

Proposition. *Let Assumption 4.4.1 hold. Consider Problem* (4.2) *and the r-BJ algorithm. There exists* $\epsilon_{\max}$ *such that, if* $0 < \epsilon < \epsilon_{\max}$, *then, for any* $\mathbf{x}(0) \in \mathbb{R}^n$, *the trajectory* $\mathbf{x}(t)$, *generated by the BJ algorithm, converges exponentially fast to the minimizer of Problem* (4.2), *i.e.,*

$$\|\mathbf{x}(t) - \mathbf{x}_{\text{opt.}}\| \leq c\rho^t$$

for some constants $c > 0$ *and* $0 < \rho < 1$.

Proof. Let us define

$$\mathbf{g}_{\text{opt.}} := A\mathbf{x}_{\text{opt.}} - \mathbf{b}\,, \quad \mathbf{g}^{(i)}_{\text{opt.},j} := \mathbf{g}_{\text{opt.},j}\,, \quad \mathbf{x}^{(i)}_{\text{opt.},j} = \mathbf{x}_{\text{opt.},j}\,,$$

and consider the change of variables

$$\begin{aligned}\widetilde{\mathbf{x}} &= \mathbf{x} - \mathbf{x}_{\text{opt.}}\,, & \widetilde{\mathbf{x}}^{(i)}_j &= \mathbf{x}^{(i)}_j - \mathbf{x}_{\text{opt.},j}\\ \widetilde{\mathbf{g}} &= \mathbf{g} - \mathbf{g}_{\text{opt.}}\,, & \widetilde{\mathbf{g}}^{(i)}_j &= \mathbf{g}^{(i)}_j - \mathbf{g}_{\text{opt.},j}\end{aligned} \tag{A.9}$$

Let us collect all the auxiliary variables $\widetilde{\mathbf{x}}^{(i)}_j$'s, $\widetilde{\mathbf{g}}$'s and $\widetilde{\mathbf{g}}^{(i)}_j$'s in the vector $\widetilde{\mathbf{y}}$. Then,

Algorithm 5 dynamic can be expressed as

$$\begin{bmatrix}\widetilde{\mathbf{x}}(t+1)\\ \widetilde{\mathbf{y}}(t+1)\end{bmatrix} = \begin{bmatrix}\mathbb{I} & \epsilon B\\ C(t) & F(t)\end{bmatrix}\begin{bmatrix}\widetilde{\mathbf{x}}(t)\\ \widetilde{\mathbf{y}}(t)\end{bmatrix} \tag{A.10}$$

The proof's aim is to show that the system (A.10) satisfies the hypotheses of Lemma A.6. In (A.10), $\widetilde{\mathbf{x}}$ will be the variable with a slow dynamic, while $\widetilde{\mathbf{y}}$ will be the variable with a fast dynamic.

Now, fix the slow variable value $\widetilde{\mathbf{x}} = \bar{\mathbf{x}}$. It can be shown that the vector $\widetilde{\mathbf{y}}(\bar{\mathbf{x}})$ that stacks the values $\widetilde{\mathbf{x}}_j^{(i)} = \bar{\mathbf{x}}_j$, $\widetilde{\mathbf{g}} = A\bar{\mathbf{x}}$, $\widetilde{\mathbf{g}}_j^{(i)} = \widetilde{\mathbf{g}}_j$, satisfies the condition $\widetilde{\mathbf{y}}(\bar{\mathbf{x}}) = C(t)\bar{\mathbf{x}} + D(t)\widetilde{\mathbf{y}}(\bar{\mathbf{x}}),\ \forall t, \bar{\mathbf{x}}$. Furthermore, it can be easily found a matrix G such that $\widetilde{\mathbf{y}}(\bar{\mathbf{x}}) = G\bar{\mathbf{x}}$.

For what concerns the dynamic of the fast variables $\widetilde{\mathbf{y}}$, because of Assumption 4.4.1, we have that after a time lower or equal to $2T+1$, $\widetilde{\mathbf{y}}$ will reach the value $\widetilde{\mathbf{y}}(\bar{\mathbf{x}}) = G\bar{\mathbf{x}}$. In fact, exploiting (4.20), (4.21) and the change of variables (A.9), in the worst case, T iteration are necessary to have $\widetilde{\mathbf{x}}_j^{(i)} = \bar{\mathbf{x}}_j, \forall i$. After that, one iteration is necessary to compute $\widetilde{\mathbf{g}} = A\bar{\mathbf{x}}$ and finally T iteration are necessary to have $\widetilde{\mathbf{g}}_j^{(i)} = \widetilde{\mathbf{g}}_j, \forall i$. As a result, the fast variable dynamic is exponentially stable, reaching the equilibrium in a finite number of iteration. That is, fixed $\bar{\mathbf{x}}$, we have

$$\widetilde{\mathbf{y}}(t) = G\bar{\mathbf{x}},\ \forall t \geq 2T+1 \tag{A.11}$$

Furthermore, from (A.6) we have that

$$\begin{aligned}\widetilde{\mathbf{y}}(t) &= \prod_{k=0}^{t-1} F(k)\mathbf{y}(0) + \prod_{k=1}^{t-1} F(k)C(0)\bar{\mathbf{x}} + \cdots + \prod_{k=2}^{t-1} F(k)C(1)\bar{\mathbf{x}} + \ldots C(0)\bar{\mathbf{x}}\\ &= \prod_{k=0}^{t-1} F(k)\mathbf{y}(0) + \Phi\bar{\mathbf{x}}\end{aligned} \tag{A.12}$$

Since equation (A.11) is satisfied for every initial condition $\mathbf{y}(0)$, it turns out that

$$\prod_{k=0}^{t-1} F(k)\mathbf{y}(0) = 0$$

for every $\mathbf{y}(0)$ and for every sequence $F(0), \ldots, F(t-1)$. As a consequence, the system (A.7) is asymptotically stable.

Now, consider the dynamical system

$$\dot{\boldsymbol{\xi}}(t) = -BG\boldsymbol{\xi}(t)\,. \tag{A.13}$$

It can be shown that

$$BG = D^{-1}A^T R^{-1}A$$

and thus, choosing as a Lyapunov function

$$V(\boldsymbol{\xi}) = \boldsymbol{\xi}^T \frac{A^T R^{-1} A}{2} \boldsymbol{\xi}\,,$$

we can see that system (A.13) is asymptotically stable. As a result, system (A.10) satisfies the hypotheses of Lemma A.6, and thus there exists $\epsilon_{\max}$, with $0 < \epsilon < \epsilon_{\max}$ such that, by using the robust block Jacobi Algorithm 5,

$$\lim_{t\to\infty} \mathbf{x}(t) = \mathbf{x}_{\text{opt.}}\,.$$

■

A.8 Proof of Lemma 5.9.1

Lemma. *Consider $\tilde{f}_{\alpha,\epsilon}$ as in (5.19). Then, it holds that*

$$\lim_{\substack{\alpha\to+\infty \\ \epsilon\to 0^+}} \tilde{f}_{\alpha,\epsilon}(\mathbf{x}) = \|\mathbf{x}\|_\infty\,.$$

Proof. In the following, we will prove the result by first taking the limit for α followed by the limit for ϵ. It is easy to show, thanks to a Taylor series expansion of $\tilde{f}_{\alpha,\epsilon}$ around $\epsilon = 0$, that exchanging the order of the limits does not change the result.
Consider the function

$$\tilde{f}_{\alpha,\epsilon}(\mathbf{x}) = \frac{1}{\alpha}\log\left(\frac{1}{n}\sum_i \exp(\alpha|x_i|^{1+\epsilon})\right)\,.$$

Let us define $|\mathbf{x}|_{\max}^{1+\epsilon} := \max_i |x_i|^{1+\epsilon}$. It is possible to rewrite

$$\tilde{f}_{\alpha,\epsilon}(\mathbf{x}) = |\mathbf{x}|_{\max}^{1+\epsilon} + \frac{1}{\alpha}\log\left(\frac{1}{n}\sum_i \exp\left(\alpha(\underbrace{|x_i|^{1+\epsilon} - |x|_{\max}^{1+\epsilon}}_{\leq 0})\right)\right)\,.$$

Now, since the exponent in the second term is always < 0 except for the components where the maximum is attained which are exactly equal to 0, we have

$$\lim_{\alpha\to+\infty} \tilde{f}_{\alpha,\epsilon} = |\mathbf{x}|_{\max}^{1+\epsilon}\,.$$

Finally, the Taylor series expansion around $\epsilon = 0^+$ of $|\mathbf{x}|_{\max}^{1+\epsilon}$ is equal to

$$|\mathbf{x}|_{\max} + \sum_n \frac{1}{n!} |\mathbf{x}|_{\max} \log^n(|x|_{\max}) \epsilon^n \underset{\epsilon \to 0^+}{\longrightarrow} |\mathbf{x}|_{\max} .$$

which concludes the proof. ■

A.9 Proof of Lemma 5.9.2

Lemma (Strong convexity of $f_{\alpha,\epsilon}$)**.** *Consider the function $f_{\alpha,\epsilon} : \mathbb{R}^{n_\ell} \mapsto \mathbb{R}$ defined as in (5.20). Then, for all $0 < \epsilon \leq 1$ and for all $\alpha > 0$, $f_{\alpha,\epsilon}$ is strongly convex in $\mathbf{x}$.*

Proof. To prove strong convexity $f_{\alpha,\epsilon}(\mathbf{x})$ we exploit the second order characterization of strong convexity which states that a function is strongly convex if and only if

$$\nabla^2_{\mathbf{x}} f_{\alpha,\epsilon}(\mathbf{x}) - m\mathbb{I} \text{ is positive semidefinite.}$$

for some $m > 0$. The Hessian of $f_{\alpha,\epsilon}$ is indeed a diagonal matrix whose i-th diagonal entry is equal to

$$\underbrace{\alpha(1+\epsilon)\exp(\alpha|x_i|^{1+\epsilon})}_{>0} \left(\epsilon \frac{|x_i|^\epsilon}{|x_i|} + \alpha(1+\epsilon)|x_i|^{2\epsilon} \right).$$

Note that, the entire expression can fail to be positive only if the second term in the right hand side fails to be positive. Moreover, the only possible point of failure is $x_i = 0$ where we encounter a $\frac{0}{0}$ limit. However, being $|x_i|^\epsilon = o(|x_i|)$ for any finite value $0 < \epsilon \leq 1$, we have that

$$\frac{\partial^2 f_{\alpha,\epsilon}(\mathbf{x})}{\partial x_i^2} \underset{|x_i| \to 0}{\longrightarrow} +\infty .$$

Now, since each diagonal component is bounded below by a positive $m_i > 0$, by taking $m := \min_i \ m_i$ we can conclude for strong convexity. ■

A.10 Proof of Proposition 5.10.2

Proposition (Convergence of the dual ascent algorithm)**.** *Consider Problem (5.24) and assume that Slater's condition holds, namely, a strictly feasible solution for (5.24) exists. Then, there exists $\overline{\rho}$ such that for any $\rho \leq \overline{\rho}$, the dual ascent algorithm (5.25) converges to the optimal solution of (5.24).*

Proof. First of all, we recall that assuming Slater's conditions hold there is zero duality gap zero, i.e. solving the primal problem (5.24) is equivalent to solve its dual problem which is defined as

$$\begin{aligned} \underset{\lambda}{\text{maximize}} \quad & d(\boldsymbol{\lambda}) := \inf_{\mathbf{x}} \; f_{\alpha,\epsilon}(\mathbf{x}) + \boldsymbol{\lambda}^T \Big(A\mathbf{x} - \mathbf{b} \Big) \,, \\ \text{s.t.} \quad & \boldsymbol{\lambda} \geq 0 \,. \end{aligned} \tag{A.14}$$

where $A := \begin{bmatrix} \Xi^T & (\Xi Q_{\text{crit}})^T \end{bmatrix}^T$ and $b := \begin{bmatrix} \chi^T & \varphi^T \end{bmatrix}^T$. Moreover, thanks to strong convexity of $f_{\alpha,\epsilon}$, the solution is unique. Thanks to Proposition 6.1.1 in Bertsekas (2008), it follows that d is everywhere continuously differentiable and, moreover

$$\nabla_\lambda d(\boldsymbol{\lambda}) = A\mathbf{x}^*_{\boldsymbol{\lambda}} - \mathbf{b}$$

where $\mathbf{x}^*_{\boldsymbol{\lambda}} := \text{argmin}_{\mathbf{x}} \; f_{\alpha,\epsilon}(\mathbf{x}) + \boldsymbol{\lambda}^T g(\mathbf{x})$. In addition, the dual ascent algorithm (5.25) coincides with a gradient projected Bertsekas (2008) applied to (A.14) which, thanks to Proposition 2.3.2 in Bertsekas (2008), is known to converge, for sufficiently small step sizes, namely $0 < \rho < \frac{2}{L} = \overline{\rho}$, if $\nabla_{\boldsymbol{\lambda}} d$ is Lipschitz continuous with Lipschitz constant L. To prove Lipschitz continuity of $\nabla_{\boldsymbol{\lambda}} d$, observe that it is linear respect to $\mathbf{x}^*_{\boldsymbol{\lambda}}$. Then, we just need Lipschitz continuity of $\mathbf{x}^*_{\boldsymbol{\lambda}}$ respect to $\boldsymbol{\lambda}$. From the definition of $x^*_{\boldsymbol{\lambda}}$ and thanks to first order optimality condition, it holds that

$$\nabla_{\mathbf{x}} f_{\alpha,\epsilon}(\mathbf{x}^*_{\boldsymbol{\lambda}}) = -A^T \boldsymbol{\lambda} \,.$$

By defining $\boldsymbol{\zeta} := -A^T \boldsymbol{\lambda}$ and recalling from (5.22) that each component of $\nabla_{\mathbf{x}} f$ is only a function of x_i, we have that

$$\frac{\partial f_{\alpha,\epsilon}(\mathbf{x}^*_{\boldsymbol{\lambda}})}{\partial x_i} = \zeta_i \,,$$

and it is possible to reduce the analysis to show Lipschitz continuity of the i-th component of $\mathbf{x}^*_{\boldsymbol{\lambda}}$ respect to ζ_i. Now, by being $f_{\alpha,\epsilon}$ twice continuously differentiable and thanks to the inverse function theorem, $\partial f_{\alpha,\epsilon}/\partial x_i$ is invertible, namely

$$[\mathbf{x}^*_{\boldsymbol{\lambda}}]_i = \left(\frac{\partial f_{\alpha,\epsilon}}{\partial x_i} \right)^{-1} (\zeta_i) \,,$$

its inverse is continuously differentiable and moreover

$$\frac{\partial \left((\partial f_{\alpha,\epsilon}/\partial x_i)^{-1} \right)}{\partial x_i} = \left(\frac{\partial (\partial f_{\alpha,\epsilon}/\partial x_i)}{\partial x_i} \right)^{-1} < \frac{1}{m_i} \,,$$

where the last inequality holds since $f_{\alpha,\epsilon}$ is strongly convex. Then, $\mathbf{x}^*_{\boldsymbol{\lambda}}$ is Lipschitz

continuous respect to $\boldsymbol{\zeta}$ and so respect to $\boldsymbol{\lambda}$ being the former a linear function of the latter. ■

References

Abur A. and Gómez Expósito A. *Power system state estimation: theory and implementation.* Marcel Dekker, 2004.

Albu M. M., Neurohr R., Apetrei D., Silvas I., and Federenciuc D. Monitoring voltage and frequency in smart distribution grids. a case study on data compression and accessibility. In *IEEE Power and Energy Society General Meeting*, 2010.

Andersson G. Modelling and analysis of electric power systems. *EEH-Power Systems Laboratory, Swiss Federal Institute of Technology (ETH), Zürich, Switzerland*, 2008.

Baran M. and Wu F. Optimal capacitor placement on radial distribution systems. *Power Delivery, IEEE Transactions on*, 4(1):725–734, 1989.

Barchi G., Macii D., and Petri D. Synchrophasor estimators accuracy: A comparative analysis. *IEEE Transactions on Instrumentation and Measurement*, 62(5):963–973, 2013.

Barooah P. *Estimation and Control with Relative Measurements: Algorithms and Scaling Laws.* PhD thesis, University of California, Santa Barbara, 2007.

Barooah P. and Hespanha J. P. Distributed estimation from relative measurements in sensor networks. In *Proceedings of the 2nd International Conference on Intelligent Sensing and Information Processing*, Dec. 2005.

Bergen A. R. *Power systems analysis.* Pearson Education India, 2009.

Bertsekas D. P. *Nonlinear Programming.* Athena Scientific, 2 edition, 2008.

Bertsekas D. P. and Tsitsiklis J. N. *Parallel and Distributed Computation: Numerical Methods.* Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1989. ISBN 0-13-648700-9.

Bishop C. *Pattern recognition and machine learning*, volume 1. springer, 2006.

Bolognani S., Carli R., Cavraro G., and Zampieri S. Distributed reactive power feedback control for voltage regulation and loss minimization. *Automatic Control, IEEE Transaction on*, 60(4):966–981, 2015.

Bolognani S., Carli R., and Todescato M. State estimation in power distribution networks with poorly synchronized measurements. In *Decision and Control (CDC), 2014 IEEE 53rd Annual Conference on*, pages 2579–2584, December 2014.

Bolognani S. and Dörfler F. Fast power system analysis via implicit linearization of the power flow manifold. In *Allerton Conference on Communication, Control, and Computing*, 2015.

Bolognani S. and Zampieri S. A distributed control strategy for reactive power compensation in smart microgrids. *Automatic Control, IEEE Transaction on*, 58(11): 2818–2833, 2013.

Bolognani S. and Zampieri S. On the existence and linear approximation of the power flow solution in power distribution networks. *Power Systems, IEEE Transactions on*, (99):1–10, 2015.

Bolognani S., Del Favero S., Schenato L., and Varagnolo D. Consensus-based distributed sensor calibration and least-square parameter identification in wsns. *International Journal of Robust and Nonlinear Control*, 20(2):176–193, January 2010.

Borra D., Lovisari E., Carli R., Fagnani F., and Zampieri S. Autonomous calibration algorithms for networks of cameras. In *Proceedings of the Americal Control Conference, ACC'12.*, July 2012.

Boyd S. and Vandenberghe L. *Convex Optimization*. Cambridge University Press, 2004. ISBN 0521833787.

Bullo F., Carli R., and Frasca P. Gossip coverage control for robotic networks: dynamical systems on the space of partitions. *SIAM Journal on Control and Optimization*, 2010.

Cañizares C. Voltage stability assessment: concepts, practices and tools. Technical Report PES-TR9, IEEE Power System Stability Subcommittee, 2002.

Carron A., Todescato M., Carli R., and Schenato L. An asynchronous consensus-based algorithm for estimation from noisy relative measurements. *IEEE Transactions on Control of Network Systems*, 1(3):283 – 295, 2014.

Chandorkar M. C., Divan D. M., and Adapa R. Control of parallel connected inverters in standalone ac supply systems. *IEEE Transactions on Industry Application*, 1993.

Conejo A. J., de la Torre S., and Cañas M. An optimization approach to multiarea state estimation. *IEEE Transactions on Power Systems*, 22(1):213–231, February 2007.

Corless R., Gonnet G., Hare D., Jeffrey D., and Knuth D. On the lambertw function. *Advances in Computational mathematics*, 5(1):329–359, 1996.

Cutsem T. and Ribbens-Pavella M. Critical survey of hierarchical methods for state estimation of electric power systems. *IEEE Transactions on Power Apparats and Systems*, PAS-102(10):3415–3424, October 1983.

De Brabandere K., Vanthournout K., Driesen J., Deconinck G., and Belmans R. Control of microgrids. In *Power Engineering Society General Meeting, 2007. IEEE*, pages 1–7, June 2007.

De Craemer K. and Deconinck G. Analysis of state-of-the-art smart metering communication standards. In *Proceedings of the 5th Young Researchers Symposium*, Leuven, Belgium, March 2010.

Dixon J., Moran L., Rodriguez J., and Domke R. Reactive power compensation technologies: State-of-the-art review. *Proceedings of the IEEE*, 93(12):2144–2164, Dec 2005.

Dobson I. and Lu L. Computing an optimum direction in control space to avoid stable node bifurcation and voltage collapse in electric power systems. *IEEEE Transactions of Automatic Control*, 37(10):1616–1620, 1992.

Dörfler F., Jovanović M., Chertkov M., and Bullo F. Sparsity-promoting optimal wide-area control of power networks. *IEEE Transactions on Power Systems*, 29(5): 2281–2291, 2014.

Ebrahimian R. and Baldick R. State estimation distributed processing. *IEEE Transactions Power Syst.*, 15(4):1240–1246, November 2000.

Eremia M. and Shahidehpour M., editors. *Handbook of Electrical Power System Dynamics: Modeling, Stability, and Control*. Wiley-IEEE Press, 2013. ISBN 978-1-118-49717-3.

Fagnani F. and Zampieri S. Randomized consensus algorithms over large scale networks. *IEEE Journal on Selected Areas in Communications*, 26(4):634–649, 2008.

Falcao D. M., Wu F. F., and Murphy L. Parallel and distributed state estimation. *IEEE Transaction on Power Systems*, 10(2):724–730, May 1995.

Farivar M., Neal R., Clarke C., and Low S. Optimal inverter var control in distribution systems with high PV penetration. In *IEEE Power & Energy Society General Meeting*, pages 1–7, San Diego, CA, USA, July 2012.

Gentile B., Simpson-Porco J. W., Dörfler F., Zampieri S., and Bullo F. On reactive power flow and voltage stability in microgrids. In *American Control Conference*, Portland, OR, USA, 2014.

Giridhar A. and Kumar P. Distributed clock synchronization over wireless networks: Algorithms and analysis. In *Decision and Control, 2006 45th IEEE Conference on*, pages 4915–4920, Dec 2006.

Gómez Expósito A., Abur A., de la Villa Jean A., and Goméz Quiles C. A multilevel state estimation paradigm for smart grids. *Proc. IEEE*, 99(6):952–976, June 2011.

Grant M. and Boyd S. CVX: Matlab software for disciplined convex programming, version 2.1. http://cvxr.com/cvx, oct 2014.

Harris Williams C. Transmission & distribution infrastructure. In *A Harris Williams & Co. White Paper*, Summer 2014.

Hiskens I., Davy R. J., and others . Exploring the power flow solution space boundary. *Power Systems, IEEE Transactions on*, 16(3):389–395, 2001.

Ipakchi A. and Albuyeh F. Grid of the future. *Power and Energy Magazine, IEEE*, 7(2):52–62, 2009.

Iwamoto S., Kusano M., and Quintana V. Hierarchical state estimation using a fast rectangular-coordinate method. *IEEE Transactions on Power Systems*, 4(3): 870–880, August 1989.

Jiang W., Vittal V., and Heydt G. A distributed state estimator utilizing synchronized phasor measurements. *IEEE Transactions on Power Systems*, 22(2):563–571, May 2007.

Jiayi H., Chuanwen J., and Rong X. A review on distributed energy resources and microgrid. *Renewable and Sustainable Energy Reviews*, 12(9):2472–2483, 2008.

Karp R., Elson J., Estrin D., and Shenker S. Optimal and global time synchronization in sensornets. *Center for Embedded Network Sensing*, 2003.

Kaye R. and Wu F. Analysis of linearized decoupled power flow approximations for steady-state security assessment. *Circuits and Systems, IEEE Transactions on*, 31(7): 623–636, 1984.

Kekatos V. and Giannakis G. Distributed robust power system state estimation. *IEEE Transactions on Power Systems*, 28(2):1617–1626, May 2013.

Korres G. N. A distributed multiarea state estimation. *IEEE Transactions on Power Systems*, 26(1):73–84, February 2011.

Li Y. W. and Kao C.-N. An accurate power control strategy for power-electronics-interfaced distributed generation units operating in a low-voltage multibus microgrid. *IEEE Transactions on Power Electronics*, 2009.

Liserre M., Sauter T., and Hung J. Y. Future energy systems: Integrating renewable energy sources into the smart power grid through industrial electronics. *IEEE Industrial Electronics Magazine*, pages 18–37, 2010.

Machowski J., Bialek J., and Bumby J. *Power system dynamics: stability and control*. John Wiley & Sons, 2011.

Marelli D. E. and Fu M. Distributed weighted least-squares estimation with fast convergence for large-scale systems. *Automatica*, 51:27–39, 2015.

Martin K. E., Hamai D., Adamiak M., Anderson S., Begovic M., Benmouyal G., Brunello G., Burger J., Cai J., Dickerson B., Gharpure V., Kennedy B., Karlsson D., Phadke A. G., Salj J., Skendzic V., Sperr J., Song Y., Huntley C., Kasztenny B., and Price E. Exploring the IEEE standard C37.118-2005 synchrophasors for power systems. *IEEE Transactions on Power Delivery*, 23(4): 1805–1811, October 2008.

Moghavvemi M. and Faruque M. Effects of FACTS devices on static voltage stability. In *TENCON 2000. Proceedings*, volume 2, pages 357–362. IEEE, 2000.

Na L., Guannan Q., and Dahleh M. Real-time decentralized voltage control in distribution networks. In *Allerton Conference on Communications, Control and Computing*, pages 582–588, Monticello, IL, USA, September 2014.

Nedic A. and Ozdaglar A. Convergence rate for consensus with delays. *Journal of Global Optimization*, 47(3):437–456, 2008.

Nedic A. and Ozdaglar A. Distributed subgradient methods for multi-agent optimization. *Automatic Control, IEEE Transactions on*, 54(1):48–61, 2009.

Nedic A., Ozdaglar A., and Parrilo P. A. Constrained consensus and optimization in multi-agent networks. *Automatic Control, IEEE Transactions on*, 55(4):922–938, 2010.

Ochoa L. F. and Wilson D. H. Using synchrophasor measurements in smart distribution networks. In *Proc. of the 21st International Conference on Electricity Distribution*, 2011.

Pal M. K. Voltage stability conditions considering load characteristics. *Power Systems, IEEE Transactions on*, 7(1):243–249, 1992.

Palomar D. P. and Chiang M. A tutorial on decomposition methods for network utility maximization. *Selected Areas in Communications, IEEE Journal on*, 24(8): 1439–1451, 2006.

Pasqualetti F., Carli R., and Bullo F. Distributed estimation via iterative projections with application to power network monitoring. *Automatica*, 48(5):747–758, 2012.

Phadke A. G. and Thorp J. S. *Synchronized phasor measurements and their applications*. Springer, 2008.

Rajkumar R., Lee I., Sha L., and Stankovic J. Cyber-physical systems: The next computing revolution. In *Design Automation Conference (DAC), 2010 47th ACM/IEEE*, pages 731–736, June 2010.

Ravazzi C., Frasca P., Ishii H., and Tempo R. A distributed randomized algorithm for relative localization in sensor networks. In *Proceedings of European Conference on Control (ECC'13)*, 2013.

Ree J. D. L., Centeno V., Thorp J. S., and Phadke A. G. Synchronized phasor measurement applications in power systems. *IEEE Transactions on Smart Grid*, 1(1): 20–27, June 2010.

Rossi W., Frasca P., and Fagnani F. Transient and limit performance of distributed relative localization. In *Decision and Control (CDC), 2012 IEEE 51st Annual Conference on*, pages 2744–2748, 2012.

Schlueter R. A., Costi A. G., Sekerke J. E., and Forgey H. L. Voltage stability and security assessment. Technical Report EPRI EL-5967 Project 1999-8, Division of Engineering Research, Michigan State University, August 1988.

Schweppe F. C., Wildes J., and Rom D. Power system static state estimation: Parts I, II, and III. *IEEE Transactions on Power Apparatus and Systems*, PAS-89: 120–135, January 1970.

Simpson-Porco J. W., Dörfler F., and Bullo F. Voltage collapse in complex power grids. March 2015. Submitted.

Simpson-Porco J. W., Dörfler F., and Bullo F. Droop-controlled inverters in microgrids are kuramoto oscillators. In *IFAC Workshop on Distributed Estimation and Control in Networked Systems, Santa Barbara, California, USA*, 2012.

Sode-Yome A. and Mithulananthan N. Comparison of shunt capacitor, svc and statcom in static voltage stability margin enhancement. *International Journal of Electrical Engineering Education*, 41(2):158–171, 2004.

Solis R., Borkar V., and Kumar P. R. A new distributed time synchronization protocol for multihop wireless networks. In *45th IEEE Conference on Decision and Control (CDC'06)*, pages 2734–2739, San Diego, December 2006.

Special issue on sensor networks and applications . *Proceedings of the IEEE*, 91 (8), August 2003.

Thorp J., Schulz D., and Ilić-Spong M. Reactive power-voltage problem: conditions for the existence of solution and localized disturbance propagation. *International Journal of Electrical Power & Energy Systems*, 8(2):66–74, 1986.

Todescato M., Cavraro G., Carli R., and Schenato L. A robust block-jacobi algorithm for quadratic programming under lossy communications. In *IFAC Workshop on Distributed Estimation and Control in Networked Systems (NecSys)*, September 2015a.

Todescato M., Simpson-Porco J., Dörfler F., Carli R., and Bullo F. Optimal voltage support and stress minimization in power networks. *IEEE Conference on Decision and Control (CDC'15)*, December 2015b.

Tron R. and Vidal R. Distributed image-based 3-d localization of camera sensor networks. In *Proceedings of the 49th IEEE Conference on Decision and Control CDC'09*, pages 901–908, 2009.

Twidell J., Weir T., and others . *Renewable energy resources*. Routledge, 2015.

Van Cutsem T. Voltage instability: phenomena, countermeasures, and analysis methods. *Proceedings of the IEEE*, 88(2):208–227, 2000.

Van Cutsem T. and Vournas C. *Voltage Stability of Electric Power Systems*. Springer, 1998.

Viawan F., Sannino A., and Daalder J. Voltage control with on-load tap changers in medium voltage feeders in presence of distributed generation. *Electric power systems research*, 77(10):1314–1322, 2007.

Von Meier A. *Electric power systems: a conceptual introduction*. John Wiley & Sons, 2006.

von Meier A., Culler D., McEachern A., and Arghandeh R. Micro-synchrophasors for distribution systems. In *IEEE 5th Innovative Smart Grid Technologies Conference, Washington, DC*, 2014.

Wache M. and Murray D. C. Application of synchrophasor measurements for distribution networks. In *IEEE Power and Energy Society General Meeting*, 2011.

Zanella F., Varagnolo D., Cenedese A., Pillonetto G., and Schenato L. Newton-raphson consensus for distributed convex optimization. In *Decision and Control and European Control Conference (CDC-ECC), 2011 50th IEEE Conference on*, pages 5917–5922. IEEE, 2011.

Zargham M., Ribeiro A., Ozdaglar A., and Jadbabaie A. Accelerated dual descent for network flow optimization. *Automatic Control, IEEE Transactions on*, 59 (4):905–920, 2014.

Zhao L. and Abur A. Multiarea state estimation using synchronized phasor measurements. *IEEE Transactions on Power Systems*, 20(2):611–617, May 2005.

Zimmerman R. D., Murillo-Sánchez C. E., and Thomas R. J. MATPOWER: Steady-state operations, planning, and analysis tools for power systems research and education. *Power Systems, IEEE Transaction on*, 26(1):12–19, 2011.

Zouzias A. and Freris N. Randomized extended Kaczmarz for solving least-squares. Technical report, University of Toronto, 2013.