Università degli Studi di Padova

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE
Corso di Laurea in Ingegneria dell'Automazione

Tesi di Laurea Magistrale

Distributed Algorithms for State Estimation in a Low Voltage Distribution Network

Candidato
Marco Todescato

Relatore
prof. Ruggero Carli

Anno Accademico 2011-2012

Contents

Chapter 1

Introduction

1.1 Extended Abstract

Nowadays it is increasingly attended that nations become aware of energetic and environmental impact. In particular politics because of the community demand, are intended to find solutions that minimize ejection and consequently the greenhouse effect. To reach this aim has its rise the **smart technology**, that is, tech provided with intelligent control modules able to lean performances optimization and a perfect integration with the already present components. From the environmental point of view, in particular, it is talked about **green technology**, that is, smart tech capable to exploit sustainable source of energy.

One of the most hot topic concerns the electric power grid. Traditional grids are of great dimensions and controlled by central units. That is because they are responsible of production and distribution of large geographical areas. Additionally they are usually placed far away from consumers because of security and cheapness reasons. The contribution given to the production by renewable sources of energy is limited and the power flows are one way: from producer to user. This makes the users subjected to the economic conditions of the dealer.

A **Smart grid** is an electric grid capable of smartly exploiting alternative sources of energy. The main aim of *smart grids* is to spread throughout the country the electric energy production, in particular of low voltage. The purpose is to make the consumers totally or partially independent from an energetic point of view, that is, they should be able to manage their own requirements. Through the smart interaction between alternative sources (as windpower, photovoltaic, hydroelectric, etc.) and appropriate storage devices, the aim would be to let *neighborhood micro grids* to manage their energy demand and become self sufficient. Furthermore, users could be able

to sell their own production. This would make bidirectional the power flows. As a consequence, consumers would obtain purchasing power and producers would offer a better service quality. Additionally the quality control of service would become distributed, making the energetic system robust and scalable.

To achieve the previously mentioned purpose, the electric grids need a deep renovation process. In particular, the modernization of the low voltage power distribution networks will consist in the deployment of a large amount of ICT[1], which is currently not present, in the form of dispersed measurement, monitoring, actuation and control devices.

One example in this sense is the coordinated control of the power inverters of the microgeneration devices connected to the low voltage grid. When properly controlled, these devices can provide valuable ancillary services like reactive power compensation, voltage support, automatic generation control, optimal power flow computation, etc.

One of the main bottleneck in the actuaction of this kind of control strategies in the low voltage power distribution network is the need for accurate voltage phasor measurements across the grid. Specifically, to achieve the aim of control is necessary to handle with the voltage phasor at every node[2] of the grid, namely the **state** of the grid.

Phasor measurement units (PMU) can provide these measurements, but their cost is generally unacceptable for large scale deployment. In particular, time synchronization between different PMUs is a major technological issue, and it is generally tackled via a GPS module that can provide timestamping of the data.

A first contribution of this thesis lies in evaluating the effects of PMU measurement errors for measurement-driven control strategies, adopting the reactive power control proposed in [14] as a prototype. The analysis shows that the quality of standard PMUs is largely insufficient.

Tackling this issue via a large scale deployment of GPS or of more advanced measurements technology is very unlikely in the near future, because of the low end nature of the power distribution networks. The number of devices is large, reliability is relatively low, and the cost of deployment must necessarily be limited.

We propose a solution to this problem that exploits the large number of sensors and their communication capabilities. Specifically, we present two **distributed** and **scalable** estimation algorithms capable of improving the quality of the voltage measurements via exchange of data with other PMUs

[1] Information and Communication Technology

[2] A *Node* can represent either an household appliance in a domestic microgrid or an entire house demand in a urban grid.

and via distributed processing of the raw data.

We assume that the power distribution grid is divided into a number of areas. The PMUs beloning to each area transmit their voltage measurements to an *area monitor*. Area monitors can communicate and they are instructed to process the collected measurement in a distributed way.

Similar algorithms have already been proposed in the literature, especially for medium voltage and high voltage power grids, see [5], [6], [7], [9] [10]. The two solutions proposed, however, exhibit some notable original features which make them particularly suited for the scenario of low voltage power distribution grids:

- they can deal effectively with poorly synchronized measurements, introducing time sync errors in the measurement model;
- they only require measurements that can be performed by the devices at their point of connection, instead of power flow and current measurements on the power lines, which are generally not available in low voltage grids;
- the computational effort is very limited and remains constant if the grid grows in size;
- they are completely leader-less (no grid supervisor is present).

In order to present the two proposed distributed algorithms, we first introduce a model for the power grid, which includes a convenient modelling of the measurement errors in which time sync error are explicitly considered. Based on this model, we detail the least-square problem that has to be solved in order to find the maximum-likelyhood estimation of the grid state. For the solution of such optimization problem, we propose two different approaches. The first approach is a distributed Jacobi-like algorithm. The algorithm is completely leader-less, and each monitor has to solve an extremely simple optimization problem, for which a closed form solution is provided. The second approach is a distributed implementation of the Alternating Direction Method of Multipliers ([11]). This contribution is of particular interest *per se*: we show how ADMM can be implemented in a scalable way [13], in which every agent is only required to store a portion of the entire state of the systems. We show how both these algorithms are effective in improving the quality of the measurements, providing a consistent and accurate estimate of the node voltages.

Finally, we consider a power network control application. Specifically, we take a power losses minimization distributed algorithm. This algorithm ensures optimal control if no noisy measurements are provided. On the contrary, using noisy measurements it does not work. We show how, using the estimated state, it leads to an optimal behavior.

1.2 State of the Art

Since Power Networks State Estimation represents the starting point to implement a desirable network control, it has been fully treated in literature. Firstly it has been analyzed through centralized techniques. Afterwards researchers focused on distributed solution since the increasing in network size, the always more relevant computational effort, the networks topology privacy and the robustness to failures become strictly urgent.

The main aim of the estimation is to adequately filter the raw measurements with the purpose to achieve a better knowledge of a desire quantity, namely the state. This is very important due to the fact that measurements could be very noisy. Therefore, they cannot be straightly used to control the network. Indeed, presence of outliers, measurement errors and noisy measures, corrupting the real value of the state, make absolutely unusable the control. As a byproduct, the estimation could be efficiently used to do *fault detection* and *bad data detection*. This is, respectively, to detect fault of the network and to identify particularly bad measures (outliers), for instance, due to instrumentation faults or corruption through the connection lines.

In [1] the authors firstly present the principal electric components to introduce a suitable network model. Secondly, it is fully explained the centralized *weighted least squares* estimation supposing to deal with measurements affected by gaussian noise.

In [2], [3] e [4] it is firstly developed an exact network model, secondly an approximated one and finally the authors deal with the implementation of a centralized static least square estimation modeling the noise as a gaussian random variable. Finally it is suggested how to implement a real-time version of the algorithm proposed and a bad data detection.

In [5] the authors proposed a multi area distributed two-level estimation. Firstly the single area, using just inner measures, estimates its own knowledge of the state. Secondly, a central unit deals with the task of coordinate the single areas estimations via an additional set of pseudo-measurements take by *Phasor Measurement Units* (PMUs). Similar method is described both in [6] and [7].

In [8] it is proposed a technique that, after a preliminary decomposition of the net in smaller subnetworks, place the measurement units with the aim of optimizing their number and costs.

In [9], similar to the two-level implementation of [5], [6] and [7], the author proposes a method to deal with a distributed state estimation via only

local measures. Thanks to the exchange of borders information between neighboring areas and a central coordination unit it is finally reached the wide range estimation.

In [10] a complete leader-less algorithm is proposed. Coordination and estimation are carried out only via local exchange of information.

In [11] the authors develop a fully distributed mean square algorithm. This leads a Wireless Sensor Network (WSN), in which the algorithm is tested, to adaptively reach the state estimation with single-hop neighbors exchanges of messages. The optimization problem is solved exploiting the Alternating Direction Method of Multipliers (ADMM).

In [12] an approach able to parallelize optimal power flow is presented. The proposed distributed scheme can be use to coordinate an heterogeneous collection of utilities. Three mathematical decomposition coordination methods are introduced to implement the proposed distributed sheme: the Auxiliary Problem Principle (APP); the Predictor-Corrector Proximal Multiplier Method (PCPM); the Alternating Direction Method (ADM).

In [13] is proposed a modification of the standard Alternating Direction Multiplier Method formulation in order to obtain a scalable version. The resulting algorithm is completely distributed and scalable.

In [14] the authors firstly propose an appropriate model for a low voltage microgrid, secondly they develop a completely distributed algorithm to appropriately command a sub set of microgenerators to achieve an optimal distribution losses minimization.

1.3 Structure

The thesis is organized as follows:

- Chapter 2 firstly presents a general model for an electric grid. Consecutively, is presented a specific low voltage microgrid model more suitable for the analyzed topic.
- Chapter 3 fully presents the problem to deal with. Specifically the estimation problem and its importance. A centralized solution to the problem is developed. Finally one of the main topic about Power Networks Control is presented to better motivate the estimation procedure.
- Chapter 4 introduces and develops two completely distributed and scalable techniques to achieve the target. Specifically a first Jacobi-like algorithm is proposed. Secondly an ADMM based solution is developed.
- Chapter 5 presents a full set of tests to validate the algorithms proposed.
- Chapter 6 gathers the main features of the algorithms proposed.

Notations

A. *State variables*

v_i Magnitude of the voltage at the i^{th} node.

θ_i Phase of the voltage at the i^{th} node.

V Vector containing all voltages' magnitude.

Θ Vector containing all voltages' phase.

i_i Magnitude of the current at the i^{th} node.

ϕ_i Phase of the current at the i^{th} node.

I Vector containing the magnitude of the current.

Φ Vector containing all currents' phase.

x_i Real of the voltage at the i^{th} node.

y_i Imaginary of the voltage at the i^{th} node .

X Vector containing all the real parts.

Y Vector containing all the imaginary parts.

$\mathbf{X} = [X\ Y]^T$ Vector of all the state variables.

B. *Measures*

v_i^m Magnitude of the voltage at the i^{th} node.

θ_i^m Phase of the voltage at the i^{th} node.

V^m Vector containing all the magnitude of the voltages measured.

Θ^m Vector containing all the phases of the voltages measured.

s_i Real of the voltage at the i^{th} node.

r_i Imaginary of the voltage at the i^{th} node.

S Vector of the real parts of the voltage.

R Vector of the imaginary parts of the voltage.

i_i^m Magnitude of the current at the i^{th} node.

ϕ_i^m Phase of the current at the i^{th} node.

h_i Real of the current at the i^{th} node.

k_i Imaginary of the current at the i^{th} node.

H Vector of the real part.

K Vector of the real part.

C. *Standard deviations*

σ_v Standard deviation of the voltage magnitude error.

σ_θ Standard deviation of the voltage phase error.

σ_i Standard deviation of the current magnitude error.

σ_ϕ Standard deviation of the current phase error.

D. *Functions*

$J(\cdot)$ Objective Cost Function.

$f(\cdot)$ Current magnitude nonlinear function of the state.

$g(\cdot)$ Current phase nonlinear function of the state.

$|\cdot|$ Both the absolute value of a quantity or the cardinality of a set depending on the context.

$\cdot^T$ Transpose of a vector or matrix.

$\bar{\cdot}$ Complex conjugate of a complex quantity.

$\cdot^*$ Both complex conjugate and transpose.

E. *Matrixes and Vectors*

I_n Identity matrix $\in \mathbb{R}^{n\mathbf{x}n}$.

$\mathbb{1}$ Vector whose element are all equal to one.

$\mathbb{1}_i$ Canonical vector whose elements are all equal to zero except for that in position i.

Chapter 2

Grid Modeling

In this chapter we firstly introduce the principal components constituting an electric grid. Secondly, we describe a general model, commonly adopted, of an electric grid ([1]). Finally we deal with the specific model ([14]) exploited in this thesis.

2.1 Components of an Electric Grid

An electric grid consists of a series of electric components such as transmission lines, loads, generators, transformers and capacitors. It is usually assumed the power system to operate in the steady state under balance conditions. This implies that all bus loads and branch flows will be three phase and balanced, all transmission lines are fully transposed and all other series or shut devices are symmetrical in the three phases [1]. These assumptions allow the use of the single phase positive sequence equivalent circuit for modeling the entire system. The following component models are commonly used in representing the network.

2.1.1 Transmission Lines

Transmission lines are usually represented by a two-port π-model characterized by a series impedance of $R + jX$ and a total line charging susceptance of $j2B$ corresponding to the equivalent circuit of figure 2.1

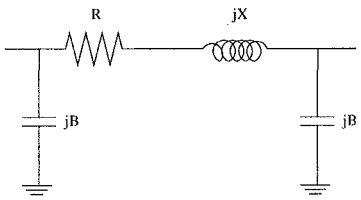


Figure 2.1: Equivalent circuit for a transmission line

2.1.2 Shunt Capacitors or Reactors

Shunt devices are represented by their susceptance at the corresponding bus whose sign determines the type of shunt element. They can be used for voltage and/or reactive power control.

Consider an admittance $Y = jB$ characterized the susceptance B

$$B = \begin{cases} \omega L \\ -\frac{1}{\omega C} \end{cases}$$

the sign of B determines the type of shunt element so a positive value corresponds to a capacitor, alternatively a negative one to a reactor.

2.1.3 Transformers

Transformers can be modeled, as shown in figure 2.2, as series impedance in series with an ideal transformers, where a represents the tap ratio which can be a real value if the transformer is an in phase device or complex if is a phase shifting device; m and k are the buses connected to the transformer.

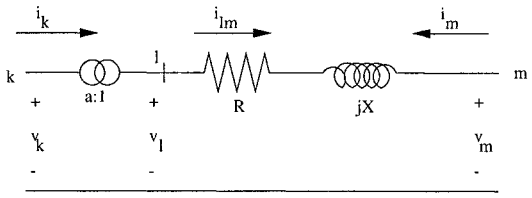


Figure 2.2: Equivalent circuit for a transformers

It is easy to see that the nodal equations of the two-port circuit, for the more general case of a complex value of a, are

$$\begin{bmatrix} i_k \\ i_m \end{bmatrix} = \begin{bmatrix} \frac{y}{|a|^2} & -\frac{y}{\bar{a}} \\ -\frac{y}{a} & y \end{bmatrix} \begin{bmatrix} v_k \\ v_m \end{bmatrix}$$

where $\bar{a}$ is the complex conjugate of a and y represents the admittance of the $l - m$ branch.

2.1.4 Loads and Generators

Loads and Generators are modeled respectively as negative or positive complex power injection and therefore have no effect on the network model.

2.2 General Network Model

The above component models can be used to build the network model of the system that is the admittance matrix Y describing the Kirchhoff's current law at each bus:

$$I = \begin{bmatrix} i_1 \\ \vdots \\ i_N \end{bmatrix} = \begin{bmatrix} Y_{11} & \cdots & Y_{1N} \\ \vdots & & \vdots \\ Y_{N1} & \cdots & Y_{NN} \end{bmatrix} \begin{bmatrix} v_1 \\ \vdots \\ v_N \end{bmatrix} = YV$$

where i_k and v_k are the current injection and voltage phasors at bus k; Y_{km} is the (m, k) entry of the Y matrix representing the total admittance between nodes m and k.

2.3 Smart Grid Exploited Model

For our purpose the model used is a bit different from the general one proposed in the previous section and almost similar to the one described in [14].

We first introduce some preliminaries about graph theory that will turn out to be useful in the description of our model.

2.3.1 Mathematical Preliminaries

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \sigma, \tau)$ be a directed graph, where $\mathcal{V}$ is the set of nodes ($|\mathcal{V}| = n$), $\mathcal{E}$ is the set of edges ($|\mathcal{E}| = r$) and $\sigma,\ \tau : \mathcal{E} \to \mathcal{V}$ are two functions such that the edge $e \in \mathcal{E}$ goes from node $\sigma(e)$ to node $\tau(e)$.
Two edges e, e' are said to be *consecutive* if

$$\{\sigma(e), \tau(e)\} \cap \{\sigma(e'), \tau(e')\}$$

is not empty. A *path* is a sequence of consecutive edges. It is possible to describe the graph through its incidence matrix $A \in \mathbb{R}^{r \times n}$ defined as follows:

$$A_{ei} = \begin{cases} \text{-1} & \text{if } i = \sigma(e); \\ 1 & \text{if } i = \tau(e); \\ 0 & \text{otherwise.} \end{cases}$$

A graph is connected if exists a path connecting every pair of nodes. If this is the case the vector $\mathbb{1}$ is the only one owning to the null space $ker A$.

2.3.2 Model

Let us firstly define a **smart grid** (or microgrid depending on its dimensions) as a portion of the power distribution network, described above, that is connected to the power transmission network in one point, the PCC (point

of common coupling), and hosts a number of loads and power generators. We consider the grid as a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ whose edges $\mathcal{E}$ represent the power lines and nodes $\mathcal{V}$ both the loads and generators. Figure 2.3 shows the correspondence between the electric and graph formulation for the grid.

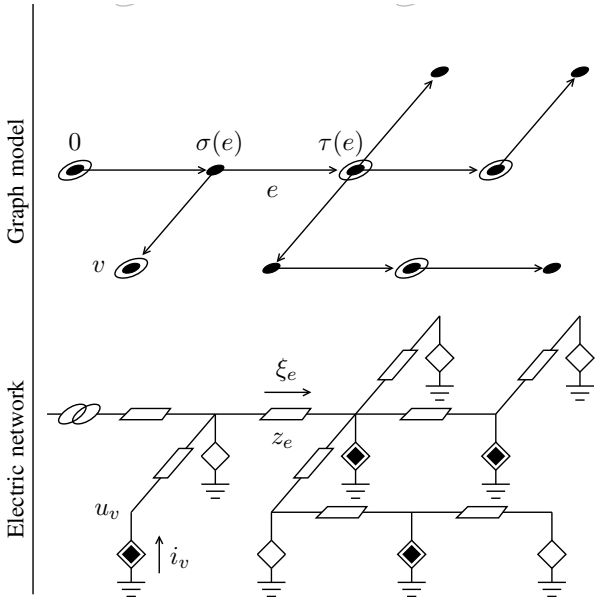


Figure 2.3: Lower Panel: Electric point of view for the grid. Black diamonds represent loads, while white diamonds represent microgenerators.
Upper panel: graph interpretation of the grid. Circled nodes correspond to microgenerators.

We limit our study to the steady state behavior of the system, as mention above. This let us represent all the signal via a complex number $y = |y|e^{j\angle y}$, since they are sinusoidal wave of the same frequency. The absolute value $|y|$ represents the signal root mean square and the argument $\angle y$ represents its phase with respect to an arbitrary global reference (usually that of the PCC).
The notation introduced above let us define the steady state of the system as:

- $\mathbf{v} = Ve^{j\Theta} \in \mathbb{C}^n$, where $v_i e^{j\theta_i}$ is the complex voltage of the i^{th} node;
- $\mathbf{i} = Ie^{j\Phi} \in \mathbb{C}^n$, where $i_i e^{j\phi_i}$ is the complex current of the i^{th} node;
- $\xi \in \mathbb{C}^r$, where ξ_e is the current flowing in the edge e.

It is useful to highlight the electric component specifically considered in our model clarifying the differences existing between our model and that of section 2.1.

Power lines are commonly described via the π-model characterized by, see figure 2.4,

- $z_e = r_e + jx_e = \frac{1}{g_e + jb_e}$: line impedance;
- y_e^{sh}: shunt admittance.

We neglect the shunt devices and consider only the line impedance z_e.

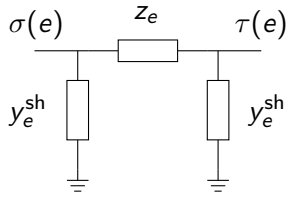


Figure 2.4: Equivalent circuit for a power lines

It is easy to see that the power line is described by the equation

$$\xi_e = \frac{v_{\sigma(e)} - v_{\tau(e)}}{z_e}$$

The PCC (point of common coupling) is modeled as a constant voltage generator

$$v_{PCC} = V_N e^{j\theta_0} \tag{2.1}$$

where V_N is the nominal voltage and θ_0 is an arbitrary reference angle.

Loads are considered to require a given amount of active and reactive power for example depending on the voltage amplitude v_i. Examples of this are constant impedance loads and constant power loads.

Finally, dealing with a low voltage power distribution network, *transformers*, both *tap changers* and *phase shifters*, are neglected.

To describe in a unique way all different loads considered it is useful to exploit the exponential model in which each node (except the PCC) is modeled via a law relating the voltage v_i and current i_i. Specifically

$$v_i \bar{i}_i = s_i \left| \frac{v_i}{V_N} \right|^{\eta_i} \tag{2.2}$$

where s_i is the nominal complex power and η_i is a characteristic parameter. More specifically s_i is the value of the complex power that the node would inject into the grid if the voltage at its point of connection is equal to V_N. Its value belongs to the $\{\Re(s_i) < 0\}$ halfplane meaning that positive active

power is supplied to the device if this is a load; on the contrary belongs to the $\{\Re(s_i) > 0\}$ halfplane meaning that positive active power is injected into the grid if the device connected is a generator. The parameter η_i identify the device typology: for example constant power, constant current and constant impedance loads are described respectively by $\eta_i = 0, 1, 2$.
See that generators fit this model for a parameter $\eta_i = 0$.

Once introduced our network elements it can be noted that the mentioned steady state quantities are characterized by the following constraints

$$A^T \xi + \mathbf{i} = 0 \tag{2.3}$$

$$A\mathbf{v} + Z\xi = 0 \tag{2.4}$$

where A is the incidence matrix introduced above; $Z = diag\{z_e; e \in \mathcal{E}\}$ represents the matrix of line impedances.
Equation (2.3) represents the Kirchhoff's current law at the nodes while equation (2.4) describes the voltage drop on the edges.
Equation (2.3) and eq.(2.4) yield together to the system of linear equation

$$\mathbf{i} = A^T Z^{-1} A\mathbf{v} = L\mathbf{v} \tag{2.5}$$

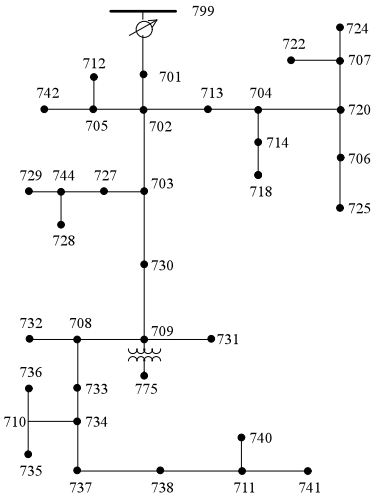
where L represents the *weighted Laplacian* matrix in graph theory, the *nodal admittance* matrix in power system analysis.

The matrices derived fully describe the grid and individuate our model.

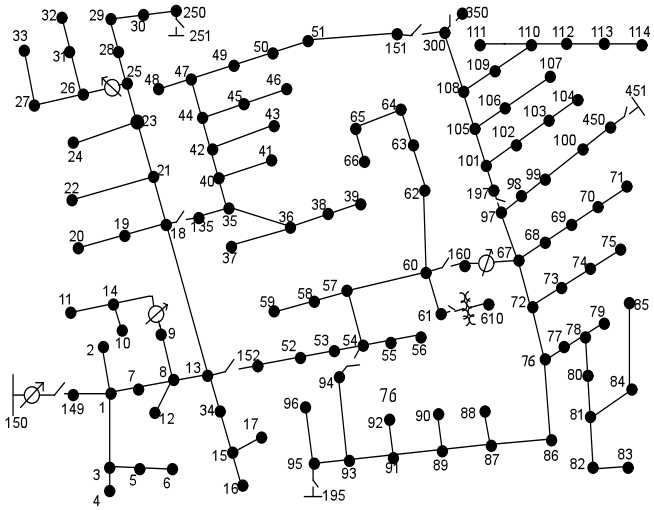
The three equations (2.1), (2.2) and (2.5) individuate a system of non linear equation to be solved to determine the steady state of the grid starting from the knowledge of the grid topology (identify by L) and the power demand required by the nodes. This topic is extensively covered in literature known as *Power Flow Analysis*. To our purpose is completely indifferent how the grid is solved and we will assume on the following to exploit some algorithm that performs it.

2.4 Testing setup: Ieee test Feeders

It is useful to introduce the specific test setup used through this thesis. All algorithm presented in the following has been tested on either one or both the `Ieee 37 nodes[15] or 123 nodes[16] Radial Distribution Test Feeder`. More specifically the graphs describing the mentioned test feeder are presented in figure 2.5.



(a) 37 nodes test feeder graph



(b) 123 nodes test feeder graph

Figure 2.5: Test Feeders graphs

Chapter 3

Electric grid State Estimation

In this chapter we introduce the problem of interest for this thesis. Specifically, the estimation of the state of an electric power grid. Firstly, we provide a rigorous formulation of the problem. Secondly, we review its closed form solution. Finally, we analyze the importance of the estimation exploiting it in a *Reactive Power Compensation Control* (see [14]) specific algorithm.

3.1 Model and Problem Formulation

3.1.1 Measurements Model

Consider a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ representing the grid, as described in chapter 2, where $\mathcal{V}$ is the set of n nodes and $\mathcal{E}$ is the set of r edges.
It is assumed that every node can measure its current and voltage divided into magnitude and phase, i.e.,

$$\begin{aligned} v_i^m &= v_i + e_{v_i}; \qquad e_{v_i} \sim \mathcal{N}(0, \sigma_v^2); \\ \theta_i^m &= \theta_i + e_{\theta_i}; \qquad e_{\theta_i} \sim \mathcal{N}(0, \sigma_\theta^2); \\ i_i^m &= i_i + e_{i_i}; \qquad e_{i_i} \sim \mathcal{N}(0, \sigma_i^2); \\ \phi_i^m &= \phi_i + e_{\phi_i}; \qquad e_{\phi_i} \sim \mathcal{N}(0, \sigma_\phi^2); \end{aligned}$$

where e_{v_i}, e_{θ_i}, e_{i_i} and e_{ϕ_i} represent the error introduced by the measure itself[1]. All the measurements are assumed to be independent from each other. Collecting all the measurements in vectors one can write

$$\begin{cases} V^m := V + \mathbf{e}_V; \\ \Theta^m := \Theta + \mathbf{e}_\Theta; \\ I^m := I + \mathbf{e}_I; \\ \Phi^m := \Phi + \mathbf{e}_\Phi \end{cases}$$

[1] In a real set up every node of the grid is equipped by a PMU(**P**hasor **M**easurement **U**nit).

where, we recall, (see notations page 13)

$$V^m = \begin{bmatrix} v_1^m \\ \vdots \\ v_N^m \end{bmatrix}; V = \begin{bmatrix} v_1 \\ \vdots \\ v_N \end{bmatrix}; e_V = \begin{bmatrix} e_{v_1} \\ \vdots \\ e_{v_N} \end{bmatrix};$$

and where Θ^m, Θ, e_Θ, I^m, I, e_I, Φ^m, Φ and e_Φ are defined similarly.
Now let us define the noise vector $\mathbf{e} = [\mathbf{e}_V\ \mathbf{e}_\Theta\ \mathbf{e}_I\ \mathbf{e}_\Phi]^T$. Then the correlation matrix R for the noise is

$$R = E[\mathbf{e}\mathbf{e}^T] = \begin{bmatrix} \sigma_v^2 I_n & & & \\ & \sigma_\theta^2 I_n & & \\ & & \sigma_i^2 I_n & \\ & & & \sigma_\phi^2 I_n \end{bmatrix}$$

We define the **state** of the grid as the voltage magnitude and phase at every node. Then, it is well known that

$$\begin{cases} V^m = V + \mathbf{e}_V; \\ \Theta^m = \Theta + \mathbf{e}_\Theta; \\ I^m = f(V, \Theta) + \mathbf{e}_I; \\ \Phi^m = g(V, \Theta) + \mathbf{e}_\Phi \end{cases} \tag{3.1}$$

where, generally, $f(\cdot)$ and $g(\cdot)$ represent non linear current's dependance on the state.

Synchronization noise

In our setup, there is another kind of noise which is relevant, the *synchronization* noise e_{sync}. We model it as a normal random variable

$$e_{sync} \sim \mathcal{N}(0, \sigma_{sync}^2)$$

This noise is due to the fact that different nodes, in general, carry out the measurement in different time instants because of some unsynchronization. The noise enters as an additive term in the phase measurement causing a phase shifting. It is important to underline that the synchronization noise is the same for measures taken by the same node because it is assumed that quantities relating the same node are always synchronized.

Formally, we can write, for $i \in \mathcal{V}$,

$$\begin{aligned} v_i^m &= v_i + e_{v_i}; \\ \theta_i^m &= \theta_i + e_{\theta_i} + e_{sync_i}; \\ i_i^m &= i_i + e_{i_i}; \\ \phi_i^m &= \phi_i + e_{\phi_i} + e_{sync_i}; \end{aligned}$$

where the value of e_{sync_i} is the synchronization error of node i.
Accordingly, we define the noise vector $\mathbf{e}$

$$\mathbf{e} = \begin{bmatrix} \mathbf{e}_V \\ \mathbf{e}_\Theta + \mathbf{e}_{sync} \\ \mathbf{e}_I \\ \mathbf{e}_\Phi + \mathbf{e}_{sync} \end{bmatrix}$$

The corresponding correlation matrix R becomes equal to

$$R = E[\mathbf{e}\mathbf{e}^T] = \begin{bmatrix} \sigma_v^2 I_n & & & \\ & (\sigma_\theta^2 + \sigma_{sync}^2) I_n & & \sigma_{sync}^2 I_n \\ & & \sigma_i^2 I_n & \\ & \sigma_{sync}^2 I_n & & (\sigma_\phi^2 + \sigma_{sync}^2) I_n \end{bmatrix}$$

where the out-of-diagonal blocks represent the correlation between voltage and current measurements at the same node due to the synchronization noise.

3.1.2 Problem Formulation

Generally the state estimation problem of an electric grid is posed as a *Weighted Least Squares Problem* (see [1],[4]). We follow the same approach. According to the measurements defined in the previous subsection, we introduce the following cost function

$$J(V,\Theta) = \begin{bmatrix} V^m & \Theta^m & I^m & \Phi^m \end{bmatrix} R^{-1} \begin{bmatrix} V^m \\ \Theta^m \\ I^m \\ \Phi^m \end{bmatrix} \tag{3.2}$$

In case of absence of synchronization noise the function can be explicitly written as

$$\begin{aligned} J(V,\Theta) \;=\; & \sum_{i=1}^{n} \Big\{ \frac{1}{\sigma_v^2}(v_i^m - v_i)^2 + \frac{1}{\sigma_\theta^2}(\theta_i^m - \theta_i)^2 + \frac{1}{\sigma_i^2}(i_i^m - f(V,\Theta))^2 + \\ & + \frac{1}{\sigma_\phi^2}(\phi_i^m - g(V,\Theta))^2 \Big\} \end{aligned}$$

Eventually, if measurements of other nature are available, i.e. power (real and reactive) injected, power flow exc..., they could be added to expression in (3.2).
As a matter of fact, we want to underline that a novelty of this work stays just in discarding them and using only current and voltage measures.

To estimate the state means to find the value $(\hat{V}, \hat{\Theta})$ of (V,Θ) that minimizes the objective cost function, that is, to find the solution of the optimization problem

$$\underset{V,\Theta}{minimize}\; J(V,\Theta) \tag{3.3}$$

We observed that the nonlinear dependance of I and Φ on V and Θ casts the above optimization problem into the *nonlinear unconditioned* class of problems. To solve it there are lots of iterative solver; for instance, all methods based on the augmented Lagrangian technique.
However, this kind of algorithm can suffer of:

- non convergence;
- convergence to local ad not global minima;
- long running time to achieve the convergence.

In this work to deal with the optimization problem in (3.3), we pursue an approach which is based on a suitable model linearization of the electric grid. Interestingly we will see how the linear model will lead to a convenient closed form solution.

3.1.3 Model Linearization

Recall the relation between the complex value current and voltage of the grid, i.e.,

$$\mathbf{i} = L\mathbf{u} \tag{3.4}$$

where $L = A^T Z^{-1} A$ represents the admittance matrix of the grid; being A and Z, respectively, the incidence and inductance matrix of the grid (see chapter 2).

The basic idea to obtain a linear model is based on expressing the quantities of interest as function of the real and imaginary part of the voltage instead of magnitude and phase. Splitting relation (3.4) into real and imaginary part leads to, for a single node, recalling the meaning of h, k, s and r from page 13,

$$h + jk = [\Re(L) + j\Im(L)] * (s + jr) = [\Re(L)s - \Im(L)r] + j[\Im(L)s + \Re(L)r]$$

that can be readily rewritten in a matrix form as

$$\begin{bmatrix} h \\ k \end{bmatrix} = \begin{bmatrix} \Re(L) & -\Im(L) \\ \Im(L) & \Re(L) \end{bmatrix} \begin{bmatrix} s \\ r \end{bmatrix}$$

Collecting all nodes values, the whole measures model becomes (see notations page 13)

$$\begin{bmatrix} S \\ R \\ H \\ K \end{bmatrix} = \begin{bmatrix} I_n & 0 \\ 0 & I_n \\ \Re(L) & -\Im(L) \\ \Im(L) & \Re(L) \end{bmatrix} \begin{bmatrix} X \\ Y \end{bmatrix} + \begin{bmatrix} \mathbf{e}_S \\ \mathbf{e}_R \\ \mathbf{e}_H \\ \mathbf{e}_K \end{bmatrix} \Longrightarrow Z = \mathcal{H}\mathbf{X} + \mathbf{e} \tag{3.5}$$

where $\mathbf{e}_S$, $\mathbf{e}_R$, $\mathbf{e}_H$ and $\mathbf{e}_K$ denotes the noises of the measures with respect to the real and imaginary parts; Z denotes the measures vector and $\mathcal{H}$ denotes the model matrix and $\mathbf{e}$ the noises vector.

Note that (3.5) represents a linear model for the measures with respect to the new state variables, i.e., the real and imaginary part of the nodes voltage.

To handle with this writing it is necessary to express the noise in a suitable form, starting from the knowledge of the standard deviation of magnitude and phase measures. The noise expressed in this new form is in fact no more uncorrelated because, in general, part of the magnitude and phase noise will be reprojected into both real and imaginary part.

To better understand this fact consider two generic vectors

$$x = \rho e^{j\psi} = \rho(cos\psi + jsin\psi);$$

$$\tilde{x} = \tilde{\rho} e^{j\tilde{\psi}} = (\rho + \delta\rho)e^{j(\psi+\delta\psi)} = (\rho + \delta\rho)(cos(\psi + \delta\psi) + jsin(\psi + \delta\psi))$$

where, clearly, $\delta\rho$ and $\delta\psi$ represent a sort of error affecting the exact values ρ and ψ. It is possible to rewrite $\tilde{x}$, exploiting some trigonometric relations, as

$$\begin{aligned} \tilde{x} &= \rho(cos\psi + jsin\psi)cos\delta\psi + \delta\rho(cos\psi + jsin\psi)cos\delta\psi - \\ &\quad -\rho(sin\psi - jcos\psi)sin\delta\psi - \delta\rho(sin\psi - jcos\psi)sin\delta\psi \end{aligned}$$

To simplify the expression above we assume $\delta\psi$ small enough and, using the McLaurin expansion for the sine and cosine functions, we get

$$\begin{aligned} \tilde{x} &\simeq \rho(cos\psi + jsin\psi)\Big(1 - \frac{\delta\psi^2}{2}\Big) + \delta\rho(cos\psi + jsin\psi)\Big(1 - \frac{\delta\psi^2}{2}\Big) - \\ &\quad -\rho(sin\psi - jcos\psi)\delta\psi - \delta\rho(sin\psi - jcos\psi)\delta\psi \end{aligned}$$

Finally, taking a first order approximation, we get

$$\begin{aligned} \tilde{x} &\simeq \rho(cos\psi + jsin\psi) + \delta\rho(cos\psi + jsin\psi) - \rho(sin\psi - jcos\psi)\delta\psi \\ &\simeq x + (\delta\rho\ cos\psi - \rho\ sin\psi\ \delta\psi) + j(\delta\rho\ sin\psi + \rho\ cos\psi\ \delta\psi) \end{aligned}$$

that can be rewritten in a matrix form as

$$\tilde{x} \simeq x + \begin{bmatrix} cos\psi & -sin\psi \\ sin\psi & cos\psi \end{bmatrix} \begin{bmatrix} \delta\rho \\ \rho\ \delta\psi \end{bmatrix} \tag{3.6}$$

Equation (3.6) highlights the approximation exploited to project the error measured in phase and magnitude into real and imaginary component.

Figure 3.1 shows an illustrative example: because of the noise, the quantity is distorted in its magnitude and phase values whose reference system is the green one. A variation in the real and imaginary part is expressed in the red reference system. The approximation introduced is equivalent to a rotation of the green reference system over the red one.

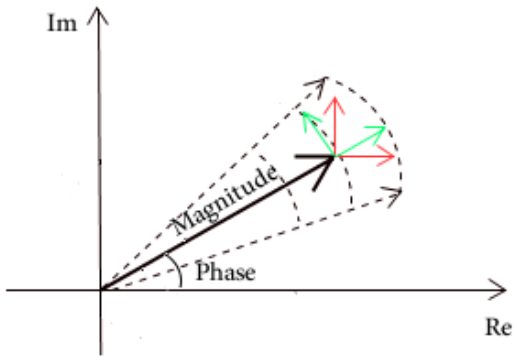


Figure 3.1: Distortion introduced by the noisy measures

Let us go back to our specific context. Let ρ, ψ, $\tilde{\rho}$, $\tilde{\psi}$, e_{mag} and e_{phase} be, respectively, the exact values, the measured value and the error value of magnitude and phase[2], then

$$\begin{cases} e_{Re} = e_{mag}cos(\psi) - e_{phase}sin(\psi)\rho \\ e_{Im} = e_{mag}sin(\psi) + e_{phase}cos(\psi)\rho \end{cases}$$

where e_{Re} and e_{Im} are obviously the error values of the real and imaginary parts.
Since the exact values are unknown, another approximation is introduced substituting their values by the measured ones. This leads to

$$\begin{cases} e_{Re} = e_{mag}cos(\tilde{\psi}) - e_{phase}sin(\tilde{\psi})\tilde{\rho} \\ e_{Im} = e_{mag}sin(\tilde{\psi}) + e_{phase}cos(\tilde{\psi})\tilde{\rho} \end{cases}$$

Adding the eventual synchronization error e_{sync}, the noise becomes equal to

$$\begin{cases} e_{Re} = e_{mag}cos(\tilde{\psi}) - e_{phase}sin(\tilde{\psi})\tilde{\psi} - e_{sync}sin(\tilde{\psi})\tilde{\rho} \\ e_{Im} = e_{mag}sin(\tilde{\psi}) + e_{phase}cos(\tilde{\psi})\tilde{\psi} + e_{sync}cos(\tilde{\psi})\tilde{\rho} \end{cases}$$

The noise correlation matrix changes due to the correlation between the real and imaginary part and assumes the form

$$R = \begin{bmatrix} \sigma^2_{\Re(V)}I_n & \sigma_{\Re(V)\Im(V)}I_n & \sigma_{\Re(V)\Re(I)}I_n & \sigma_{\Re(V)\Im(I)}I_n \\ \sigma_{\Im(V)\Re(V)}I_n & \sigma^2_{\Im(V)}I_n & \sigma_{\Im(V)\Re(I)}I_n & \sigma_{\Im(V)\Im(I)}I_n \\ \sigma_{\Re(I)\Re(V)}I_n & \sigma_{\Re(I)\Im(V)}I_n & \sigma^2_{\Re(I)}I_n & \sigma_{\Re(I)\Im(I)}I_n \\ \sigma_{\Im(I)\Re(V)}I_n & \sigma_{\Im(I)\Im(V)}I_n & \sigma_{\Im(I)\Re(I)}I_n & \sigma^2_{\Im(I)}I_n \end{bmatrix}$$

[2]For a better comprehension, to relate this specific case to the one presented in the example above set $\delta\rho = e_{mag}$ and $\delta\psi = e_{phase}$

where for all $i \in \{1...n\}$, the diagonal block are equal to

$$\begin{aligned}
\sigma^2_{\Re(V)} &= \sigma_v^2 cos^2\theta + \sigma_\theta^2 (v_i^m)^2 sin^2\theta + \sigma^2_{sync}(v_i^m)^2 sin^2\theta; \\
\sigma^2_{\Im(V)} &= \sigma_v^2 sin^2\theta + \sigma_\theta^2 (v_i^m)^2 cos^2\theta + \sigma^2_{sync}(v_i^m)^2 cos^2\theta; \\
\sigma^2_{\Re(I)} &= \sigma_i^2 cos^2\phi + \sigma_\phi^2 (i_i^m)^2 sin^2\phi + \sigma^2_{sync}(i_i^m)^2 sin^2\theta; \\
\sigma^2_{\Im(I)} &= \sigma_i^2 cos^2\phi + \sigma_\phi^2 (i_i^m)^2 cos^2\phi + \sigma^2_{sync}(i_i^m)^2 cos^2\theta;
\end{aligned}$$

representing the autocorrelation between quantities. The cross correlation between the real and imaginary part of the voltage is

$$\sigma_{\Re(V)\Im(V)} = \big(\sigma_v^2 + (\sigma_\theta^2 + \sigma^2_{sync})(v_i^m)^2\big) sin\theta cos\theta = \sigma_{\Im(V)\Re(V)}.$$

Similarly the correlation between real and imaginary part of the current is

$$\sigma_{\Re(I)\Im(I)} = \big(\sigma_i^2 + (\sigma_\phi^2 + \sigma^2_{sync})(i_i^m)^2\big) sin\phi cos\phi = \sigma_{\Im(I)\Re(I)}.$$

Finally the correlation due to the synchronization noise is

$$\begin{aligned}
\sigma_{\Re(V)\Re(I)} &= \sigma^2_{sync} v_i^m i_i^m sin\theta sin\phi = \sigma_{\Re(I)\Re(V)}; \\
\sigma_{\Re(V)\Im(I)} &= -\sigma^2_{sync} v_i^m i_i^m sin\theta cos\phi = \sigma_{\Im(I)\Re(V)}; \\
\sigma_{\Im(V)\Re(I)} &= -\sigma^2_{sync} v_i^m i_i^m cos\theta sin\phi = \sigma_{\Re(I)\Im(V)}; \\
\sigma_{\Im(V)\Im(I)} &= \sigma^2_{sync} v_i^m i_i^m cos\theta cos\phi = \sigma_{\Im(I)\Im(V)}.
\end{aligned}$$

3.1.4 Closed Form Solution

Consider the linear model in (3.5), i.e,

$$Z = \mathcal{H}\mathbf{X} + \mathbf{e}$$

It is possible to rewrite the objective cost function $J(V, \Theta)$ as

$$J(\mathbf{X}) = \big[Z - \mathcal{H}\mathbf{X}\big]^T R^{-1} \big[Z - \mathcal{H}\mathbf{X}\big] \tag{3.7}$$

This cost function, being a linear function of the decision variable $\mathbf{X}$, reduces to the classical linear weighted least squares problem. It is well known that, if the matrix $(\mathcal{H}^T R^{-1}\mathcal{H})$ is not singular, then the optimal solution $\hat{\mathbf{X}}$ can be obtain in a closed form as

$$\hat{\mathbf{X}} = (\mathcal{H}^T R^{-1}\mathcal{H})^{-1}\mathcal{H}^T R^{-1} Z \tag{3.8}$$

Dealing with Gaussian additive noise this solution coincides with the *maximum likelihood estimation.*

3.2 On the Correlation Matrix R and Non Singularity Condition of $(\mathcal{H}^T R^{-1}\mathcal{H})$

To guarantee the application of the algorithm is necessary, as previously said, that the matrix $(\mathcal{H}^T R^{-1}\mathcal{H})$ is not singular.

First of all let us consider the correlation matrix R.

3.2.1 Correlation matrix R

Absence of Synchronization Noise

Supposing absence of synchronization noise between nodes the correlation matrix is definite positive hence invertible. This is true since for every node, the eigenvalues of its corresponding sub-block

$$R_i = \begin{bmatrix} R_{i_{11}} & \\ & R_{i_{22}} \end{bmatrix}$$

where

$$R_{i_{11}} = \begin{bmatrix} \sigma_V^2 cos^2\theta + \sigma_\theta^2 (v_i^m)^2 sin^2\theta & \big(\sigma_V^2 - \sigma_\theta^2 (v_i^m)^2\big) sin\theta cos\theta \\ \big(\sigma_V^2 - \sigma_\theta^2 (v_i^m)^2\big) sin\theta cos\theta & \sigma_V^2 cos^2\theta + \sigma_\theta^2 (v_i^m)^2 sin^2\theta \end{bmatrix}$$

$$R_{i_{22}} = \begin{bmatrix} \sigma_I^2 cos^2\phi + \sigma_\phi^2 (i_i^m)^2 sin^2\phi & \big(\sigma_I^2 - \sigma_\phi^2 (i_i^m)^2\big) sin\phi cos\phi \\ \big(\sigma_I^2 - \sigma_\phi^2 (i_i^m)^2\big) sin\phi cos\phi & \sigma_I^2 cos^2\phi + \sigma_\phi^2 (i_i^m)^2 sin^2\phi \end{bmatrix}$$

can be easily computed and are equal to

$$\Lambda_{R_i} = \{\sigma_V^2 \ ; \ \sigma_\theta^2 (v_i^m)^2 \ ; \ \sigma_I^2 \ ; \ \sigma_\phi^2 (i_i^m)^2 \ \}$$

Since all these terms are positive then R is positive definite and so invertible.

Presence of Synchronization Noise

Let us now consider the presence of synchronization noise between nodes. The correlation matrix drastically changes its structure. Indeed every node is characterized by

$$R_i = \begin{bmatrix} R_{i_{11}} & R_{i_{12}} \\ R_{i_{21}} & R_{i_{22}} \end{bmatrix}$$

where

$$R_{i_{11}} = \begin{bmatrix} \sigma_V^2 cos^2\theta + (\sigma_\theta^2 + \sigma_{sync}^2)(v_i^m)^2 sin^2\theta & \big(\sigma_V^2 - (\sigma_\theta^2 + \sigma_{sync}^2)(v_i^m)^2\big) sin\theta cos\theta \\ \big(\sigma_V^2 - (\sigma_\theta^2 + \sigma_{sync}^2)(v_i^m)^2\big) sin\theta cos\theta & \sigma_V^2 cos^2\theta + (\sigma_\theta^2 + \sigma_{sync}^2)(v_i^m)^2 sin^2\theta \end{bmatrix}$$

$$R_{i_{22}} = \begin{bmatrix} \sigma_I^2 cos^2\phi + (\sigma_\phi^2 + \sigma_{sync}^2)(i_i^m)^2 sin^2\phi & \big(\sigma_I^2 - (\sigma_\phi^2 + \sigma_{sync}^2)(i_i^m)^2\big) sin\phi cos\phi \\ \big(\sigma_I^2 - (\sigma_\phi^2 + \sigma_{sync}^2)(i_i^m)^2\big) sin\phi cos\phi & \sigma_I^2 cos^2\phi + (\sigma_\phi^2 + \sigma_{sync}^2)(i_i^m)^2 sin^2\phi \end{bmatrix}$$

$$R_{i_{12}} = R_{i_{21}} = \begin{bmatrix} \sigma^2_{sync} v_i^m i_i^m sin\theta sin\phi & -\sigma^2_{sync} v_i^m i_i^m sin\theta cos\phi \\ -\sigma^2_{sync} v_i^m i_i^m cos\theta sin\phi & \sigma^2_{sync} v_i^m i_i^m cos\theta cos\phi \end{bmatrix}$$

To study such a matrix is not easy since it is not characterized by a special structure. A simple condition that guarantees positiveness and so invertibility, is to have R be diagonal dominant and so $|\sum_{j\neq i}[R]_{ij}| < [R_{ii}]$ $\forall i$. Indeed in this case by *Gershgorin theorem*, the eigenvalues of R are in the right half complex plane i.e. $\Re > 0$. Actually this is not necessary since R has to be invertible and not even positive definite. Anyway a simple numerical analysis shows an *almost probable invertibility* assuring R to be not singular in most practical conditions.

Assuming R to be invertible, is then necessary to analyze the $\mathcal{H}R^{-1}\mathcal{H}$ matrix since if is singular the solving algorithm cannot be implemented.

3.2.2 About $\mathcal{H}^T R^{-1}\mathcal{H}$

Both in presence and in absence of synchronization noise the matrix does not assume a suitable form to be study with some algebraic method. Anyway a deep empirical analysis shows not only its invertibility but its definite positivity as well. This assure the applicability of the algorithm proposed in most practical situations.

3.3 The Importance of State Estimation in Power Networks Control

To deal with a useful control strategy it is firstly necessary to appropriately filter the raw measurements. This is due to the fact that raw data are usually not directly usable by the algorithm, for instance, because:

- they are too inaccurate;
- can collect outliers.

This leads to the use of filter data able to let the control algorithm efficiently work.
To this point of view, the estimation represents the filtering of the measurements to achieve a better knowledge of the real state of the grid. Therefore it is the first necessary step to deal with a good control strategy.

To explore the importance of estimation in power networks control we consider one of its most interesting topic, namely, the minimization of power distribution losses. To achieve this aim one strategy is to appropriately control the microgenerators connected to the grid. Specifically, driving the amount of reactive power injection into the grid[14].
Let us briefly introduce the problem, exploiting what done in [14], to better understand it.

3.3.1 Problem Formulation

Consider a grid modeled as in chapter 2. As a metric for optimality of reactive power flows, active power losses on lines are chosen. The total active power losses on the edges are then equal to

$$J^{tot} \triangleq \sum_{e \in \mathcal{E}} |\xi_e|^2 \Re(z_e) = \bar{u}^T \Re(L) u$$

It is assumed to be possible to command only a subset $\mathcal{C} \subset \mathcal{V}$ of nodes of the grid, i.e. the set of microgenerators, supposing that only a part of them are equipped with some sort of intelligence, as shown in figure 3.2 (**upper panel**). In addition to this, let us assume be possible to decide only the amount of reactive power injected, as the decision on the amount of active power follows imperative economic criteria.
The resulting problem is then

$$\underset{q_v, v \in \mathcal{C}}{minimize}\ J^{tot}$$

where q_v represents the reactive power injected at nodes $v \in \mathcal{C}$. To appropriately drive the microgenerators, the algorithm provided in [14] implements

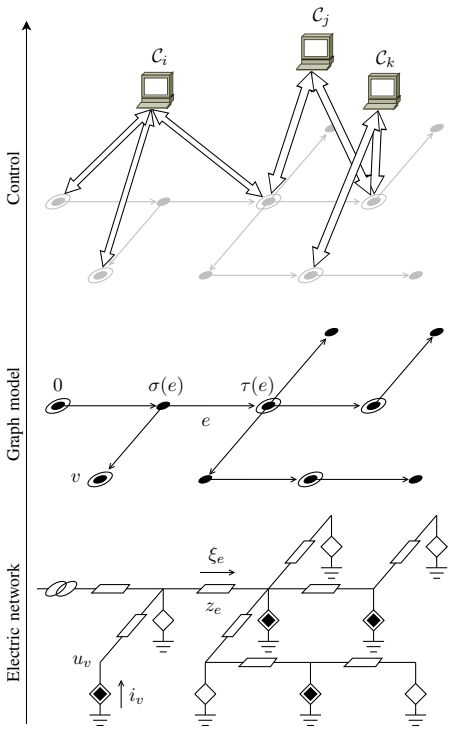


Figure 3.2: Different schematic view of the grid. **upper panel**: division of microgenerators into overlapping clusters to implement the distributed algorithm. **middle panel**: graph representation. Circled nodes represent microgenerators. **lower panel**: circuit representation. Black diamonds are microgenerators and white are loads.

a distributed optimal reactive power compensation. Specifically the microgenerators are assumed to be organized into overlapping groups, namely *clusters*, each of which coordinated by a cluster header equipped with some intelligence unit (see figure 3.2).

3.3.2 Algorithm and Estimation

To our purpose, we are not directly interested on how to control the microgenerators. Indeed, we want to analyze how the use of filtered data instead of raw data affect the control algorithm. To this end let us assume to deal with a sort of black box algorithm that, receiving in input the grid state both exact, measured or estimated, is able to compute the optimal compensation to achieve losses minimization.
We will show later in chapter 5, that the algorithm performances using the estimated state are comparable to those obtained using the grid real state even in situation in which raw measurements completely corrupt the algorithm efficiency.

Chapter 4

Distributed and Scalable Estimation Solutions

We now propose two completely **distributed** and **scalable** algorithms for solving the estimation problem. We will refer to the first one as *The Distributed Estimator* and to the second one as *The ADMM Estimator*.

Let us firstly introduce a suitable decomposition of the grid into appropriate subareas. Afterwards, we specifically deal with the two solution proposed.

Let us briefly recall what moves the choice of implementing this kind of algorithms. Dealing with a central elaboration unit which collect all the measurements and coordinate the estimation leads to huge communication and computational effort growing in size with the grid size. Moreover if communication faults occur, the entire grid state will not be computed. Distributed and scalable solution improve computability as well as robustness.

4.1 Multi Area Decomposition

Let us divide the grid into a subset of m non overlapping microgrid. Suppose adjacent areas being connected through tie lines called *border lines*, as shown in figure 4.1.

Every area $a \in [1, ..., m]$, with its subset of adjacent areas $b \in \Omega_a \subset [1, ..., m]$, will be then characterized by:

$\mathbf{X_a}$ internal state;

Z_a internal measures;

L_a internal inductance matrix (describing the internal topology);

Z_{ab} measures of the nodes of area b that direct communicate with some node of area a;

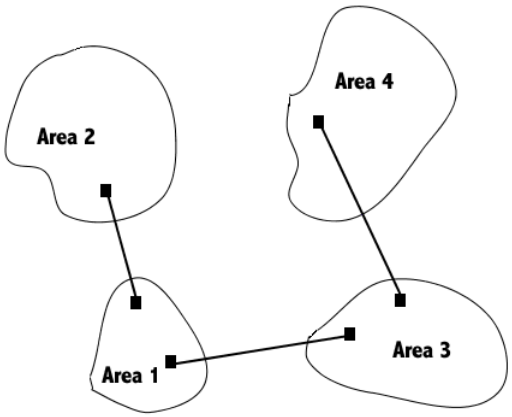


Figure 4.1: Grid divided into non overlapping areas

L_{ab} inductance matrix between area a and $b \in \Omega_a$ (describes the communication topology).

Figure 4.2 well explain the quantities introduced.

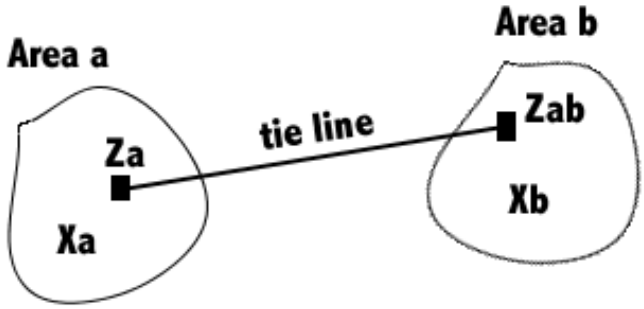


Figure 4.2: Information relating two adjacent areas

Let us explicitly consider the components relating every area in the cost function in (3.7). The introduced decomposition let us rewrite the cost function as

$$\begin{aligned} J(\mathbf{X}) &= \left[Z - \mathcal{H}\mathbf{X}\right]^T R^{-1} \left[Z - \mathcal{H}\mathbf{X}\right] \\ &= \sum_{a=1}^{m} J_a(\mathbf{X_a}) + \sum_{a=1}^{m} \sum_{b \in \Omega_a} J_{ab}(\mathbf{X_a}, \mathbf{X_b}) \end{aligned} \tag{4.1}$$

where $J_a(\mathbf{X_a})$ is the part of the cost function concerning with the inner state of area a and $J_{ab}(\mathbf{X_a}, \mathbf{X_b})$ with the *border state*.
The minimization problem of function in (4.1) is now divided into smaller multiple subproblems. Anyway, it could still not be solved in a distributed way. Indeed, the optimal solution needs a flow of information concerning the entire grid to be computed.

It is necessary to separate the subproblems to let every area to solve its inner estimation independently.

4.2 The Distributed Estimator

Let us firstly recall some preliminaries about the *Jacobi* iterative method for solving linear systems. This will turn to be useful in our first solution proposed.

4.2.1 Jacobi Method

Let us consider an invertible matrix $A \in \mathbb{R}^{n\times n}$ and a vector $b \in \mathbb{R}^n$. Consider the system of linear equation

$$Ax = b$$

where $x \in \mathbb{R}^n$ is unknown and to be determined. The i^{th} equation can be explicitly written as

$$\sum_{j=1}^{n} a_{ij}x_j = b_i$$

where $a_{ij} = [A]_{ij}$ are the entries of the matrix, x_j and b_i are respectively the j^{th} and i^{th} component of the corresponding vectors. Assuming $a_{ii} \neq 0$ and solving for the i^{th} component of x we obtain

$$x_i = -\frac{1}{a_{ii}}\left[\sum_{j\neq i} a_{ij}x_j - b_i\right] \tag{4.2}$$

If all components x_j, $j \neq i$, are known, then the remaining component x_i can be determined by equation (4.2). The procedure can be done simultaneously for every component leading to an iterative distributed algorithm of the form:

Starting from x(0) for t = 1,2,... evaluate x(t) using the iterative step

$$x_i(t+1) = -\frac{1}{a_{ii}}\left[\sum_{j\neq i} a_{ij}x_j(t) - b_i\right] \tag{4.3}$$

The algorithm produces an infinite sequence $\{x(t)\}$ that, if converges, tends to a limit that is a solution of the considered linear system as can be easily seen taking the limit of both side of equation (4.3)

Let us now consider a multivariable linear system of the form

$$\begin{bmatrix} A_{11} & A_{12} & \dots & A_{1n} \\ & \vdots & & \\ & \dots & & A_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$

Similarly, this can be managed with the Jacobi procedure, which for the i^{th} block of x leads to

$$x_i(t+1) = -A_{ii}^{-1}\left[\Big(\sum_{j\neq i}\big[A - diag(A)\big]_{ij}\Big)x_j(t) - b_i\right] \tag{4.4}$$

Note, from expression (4.4), that the i^{th} component, x_i, of the "state" is computed only using its respective "measurements" b_i and the component of the state x_j that directly "talk" to it, i.e. its adjacent component of the state. This is because if $A_{ik} = 0$, the state x_k do not intervene in computing x_i.

4.2.2 Local Informations

At the beginning of the chapter was underlined how to implement a distributed algorithm it is necessary to separate the cost function in (3.2) into pieces concerning only local informations. However, because of the presence of border lines between adjacent areas, every area depends on at least its neighbor states. Nevertheless, the Jacobi procedure of subsection 4.2.1, highlights that, from the knowledge of neighbor states it is possible to recursively compute the inner state by equation (4.4).

It is then necessary to let the i^{th} area to know its j^{th}, $j \in \Omega_i$, neighbor states. Specifically, only the state of the nodes $j \in \Omega_i$ that directly communicate with it, i.e. the *border nodes*.
Since neighbor states are not known a priori what done is simply to let adjacent areas to send to each other their border state, estimated at the current time instant. This is equal to consider the border estimated state as pseudo optimal measures from which every area starts to compute its inner optimal state.

Thanks to the exchange of the border states it is possible to consider fixed their values, $\mathbf{X_b} \equiv \hat{\mathbf{X}}_\mathbf{b}$, so that the global estimation problem

$$\underset{\mathbf{X_a}\dots\mathbf{X_m}}{minimize} \sum_{a=1}^{m} J_a(\mathbf{X_a}) + \sum_{a=1}^{m}\sum_{b\in\Omega_a} J_{ab}(\mathbf{X_a}, \mathbf{X_b})$$

can be separated into a collection of problems of the form

$$\underset{\mathbf{X_a}}{minimize}\ J_a(\mathbf{X_a}) + \sum_{b\in\Omega_a} J_{ab}(\mathbf{X_a}, \hat{\mathbf{X}}_\mathbf{b})$$

each of which concerning only a single area. This is just what is needed to implement a completely distributed algorithm.

4.2.3 Distributed Algorithm

It is now presented the first solution proposed. Let us firstly describe the specific model considered for every area.

Sub-model description

Consider a linear model of the grid similar of that introduced in subsection 3.1.3. Suppose the grid divided into m areas according with section 4.1. It is possible to suitably sort the state vector, composed of the real and imaginary part of the voltage nodes, as

$$\mathbf{X} = \begin{bmatrix} \mathbf{X_1} \\ \mathbf{X_2} \\ \vdots \\ \mathbf{X_m} \end{bmatrix} = \begin{bmatrix} X_1 \; Y_1 \mid X_2 \; Y_2 \mid \cdots \mid X_m \; Y_m \end{bmatrix}^T$$

This lets to highlight the single area state. Accordingly, the inductance matrix L becomes

$$L = \begin{bmatrix} L_{11} & L_{12} & \cdots & L_{1m} \\ L_{21} & L_{22} & \cdots & L_{2m} \\ & & \vdots & \\ L_{m1} & L_{m2} & & L_{mm} \end{bmatrix}$$

where L_{ij} represents the part of L concerning area i and j; it identifies the communication edges as well as the admittance line values.
Similarly, we have that the measures Z and noise $\mathbf{e}$ can be expressed as

$$Z = \begin{bmatrix} Z_1 \\ Z_2 \\ \vdots \\ Z_m \end{bmatrix} = \begin{bmatrix} S_1 \\ R_1 \\ H_1 \\ K_1 \\ -- \\ \vdots \\ -- \\ S_m \\ R_m \\ H_m \\ K_m \end{bmatrix}; \quad \mathbf{e} = \begin{bmatrix} \mathbf{e}_1 \\ \mathbf{e}_2 \\ \vdots \\ \mathbf{e}_m \end{bmatrix} = \begin{bmatrix} \mathbf{e}_{S_1} \\ \mathbf{e}_{R_1} \\ \mathbf{e}_{H_1} \\ \mathbf{e}_{K_1} \\ -- \\ \vdots \\ -- \\ \mathbf{e}_{S_m} \\ \mathbf{e}_{R_m} \\ \mathbf{e}_{H_m} \\ \mathbf{e}_{K_m} \end{bmatrix}$$

It follows a correlation matrix R, for the noise vector $\mathbf{e}$, equal to

$$R = E[\mathbf{e}\mathbf{e}^T] = \begin{bmatrix} R_1 & & & \\ & R_2 & & \\ & & \ddots & \\ & & & R_m \end{bmatrix}$$

R_i has a structure similar to that seen in section 3.1.3 but it concernes only measurements taken from the same area.

Accordingly to the state sorting, the matrix of model (3.5) will be characterized by a similar structure. Specifically

$$\mathcal{H} = \begin{bmatrix} \mathcal{H}_1 \\ \vdots \\ \mathcal{H}_m \end{bmatrix}$$

Thanks to this sorting it is then possible to outline a specific linear model for every area $i \in [1, \dots, m]$ of the form

$$Z_i = \mathcal{H}_i \mathbf{X} + \mathbf{e}_i = \mathcal{H}_{ii} \mathbf{X_i} + \sum_{j \neq i} \mathcal{H}_{ij} \mathbf{X_j} + \mathbf{e}_i;$$

Note that the matrix model is equal to

$$\begin{aligned} \mathcal{H}_i &= \begin{bmatrix} \mathcal{H}_{i1} & \cdots & \mathcal{H}_{ii} & \cdots & \mathcal{H}_{im} \end{bmatrix} \\ &= \left[\begin{array}{cc|c|cc|c|cc} 0 & 0 & & I_{n_i} & 0 & & 0 & 0 \\ 0 & 0 & & 0 & I_{n_i} & & 0 & 0 \\ \Re(L_{i1}) & -\Im(L_{i1}) & \cdots & \Re(L_{ii}) & -\Im(L_{ii}) & \cdots & \Re(L_{im}) & -\Im(L_{im}) \\ \Im(L_{i1}) & \Re(L_{i1}) & & \Im(L_{ii}) & \Re(L_{ii}) & & \Im(L_{im}) & \Re(L_{im}) \end{array}\right] \end{aligned}$$

whose block $\mathcal{H}_{ii}$ is relative to the inner state and blocks $\mathcal{H}_{ij}$ to other areas state.

Distributed Solution

In section 3.1.4 it has been shown that the optimal global solution to the centralized problem is equal to

$$\hat{\mathbf{X}} = \left(\mathcal{H}^T R^{-1} \mathcal{H}\right)^{-1} \mathcal{H}^T R^{-1} Z$$

Let us recall the sorting introduced in the previous subsection

$$\mathcal{H} = \begin{bmatrix} \mathcal{H}_1 & \cdots & \mathcal{H}_m \end{bmatrix}^T$$

Exploiting the Jacobi procedure of subsection 4.2.1, the closed form solution, in a Jacobi point of view, can be rewritten as

$$\hat{\mathbf{X}}_\mathbf{i}(t+1) = \left(\mathcal{H}_{ii}^T R_{ii}^{-1} \mathcal{H}_{ii}\right)^{-1} \mathcal{H}_{ii}^T R_{ii}^{-1} \left(Z_i - \sum_{j \neq i} \mathcal{H}_{ij} \hat{\mathbf{X}}_\mathbf{j}(t)\right) \tag{4.5}$$

This expression can be easily managed to obtain

$$\begin{aligned}
\hat{\mathbf{X}}_\mathbf{i}(t+1) &= \Big(\mathcal{H}_{ii}^T R_{ii}^{-1}\mathcal{H}_{ii}\Big)^{-1}\mathcal{H}_{ii}^T R_{ii}^{-1}\Big(Z_i - \sum_{j\neq i}\mathcal{H}_{ij}\hat{\mathbf{X}}_\mathbf{j}(t) - \mathcal{H}_{ii}\hat{\mathbf{X}}_\mathbf{i}(t) + \mathcal{H}_{ii}\hat{\mathbf{X}}_\mathbf{i}(t)\Big) \\
&= \Big(\mathcal{H}_{ii}^T R_{ii}^{-1}\mathcal{H}_{ii}\Big)^{-1}\mathcal{H}_{ii}^T R_{ii}^{-1}\Big(Z_i - \sum_{j=1}^{m}\mathcal{H}_{ij}\hat{\mathbf{X}}_\mathbf{j}(t) + \mathcal{H}_{ii}\hat{\mathbf{X}}_\mathbf{i}(t)\Big) \\
&= \hat{\mathbf{X}}_\mathbf{i}(t) + \Big(\mathcal{H}_{ii}^T R_{ii}^{-1}\mathcal{H}_{ii}\Big)^{-1}\mathcal{H}_{ii}^T R_{ii}^{-1}\Big(Z_i - \sum_{j=1}^{m}\mathcal{H}_{ij}\hat{\mathbf{X}}_\mathbf{j}(t)\Big) \\
&= \hat{\mathbf{X}}_\mathbf{i}(t) + D_{ii}\Big(Z_i - \sum_{j=1}^{m}\mathcal{H}_{ij}\hat{\mathbf{X}}_\mathbf{j}(t)\Big) \qquad (4.6)
\end{aligned}$$

The iterative step described by equation (4.6) shows how the single area state estimation depends on its inner measures and its border nodes estimation.

This let us handling only with local informations and to implement a completely distributed algorithm:

1. each area i receive the border estimated state form adjacent areas $j \in \Omega_i$;
2. estimates its inner state using equation 4.6;
3. sends to area j the estimated value of node $i \in \Omega_j$.

Collecting all areas state it is then possible to write the iteration step in a compact way as

$$\mathbf{X}(t+1) = \big(I - D\mathcal{H}\big)\mathbf{X}(t) + DZ = M\mathbf{X}(t) + DZ \qquad (4.7)$$

where

$$I = \begin{bmatrix} I_{n_1} & & \\ & \ddots & \\ & & I_{n_m} \end{bmatrix}; D = \begin{bmatrix} D_{11} & & \\ & \ddots & \\ & & D_{mm} \end{bmatrix};$$

being n_i the number of state variables of area i.

Note that expression (4.7) allows to compute the states of all areas together. Obviously in a real implementation in which every area is responsible of its own computation the compact form does not make sense. The script is useful in a simulation environment.

As mention above, for the i^{th} area, $\hat{\mathbf{X}}_\mathbf{j}$, $j \in \Omega_i$ represent the current estimation of border nodes. So exchanging only local informations at every iteration step, the algorithm implemented become completely distributed.

To avoid misunderstanding, is important to notice that the algorithm described does not represent an exactly Jacobi version of the centralized algorithm since to implement a Jacobi algorithm is, in general, necessary, for every area, a two-ops knowledge. This means that every area should receive information from its neighbor and from the neighbor of its neighbor. The algorithm implemented concerns more with considering the border state as pseudo-measures and solving the local sub problem in a centralized way. It could be seen as an approximation of a Jacobi algorithm.

4.2.4 Convergence analysis

Consider equation (4.7). Note that the evolution of the system is equal to

$$\begin{aligned}
\mathbf{X}(1) &= M\mathbf{X}(0) + DZ \\
\mathbf{X}(2) &= M\mathbf{X}(1) + DZ = M^2\mathbf{X}(0) + MDZ + DZ \\
\mathbf{X}(3) &= M\mathbf{X}(2) + DZ = M^3\mathbf{X}(0) + M^2DZ + MDZ + DZ \\
&\vdots \\
\mathbf{X}(k) &= M\mathbf{X}(k) + DZ = M^k\mathbf{X}(0) + \Big(\sum_{i=0}^{k} M^i\Big)DZ
\end{aligned} \tag{4.8}$$

that, if the "iteration" matrix M is stable, i.e. $\rho(M) < 1$, being

$$M^k \underset{k\to\infty}{\longrightarrow} 0$$

leads to, supposing $(I - M)$ invertible,

$$\mathbf{X}(\infty) = M^\infty\mathbf{X}(0) + \Big(\sum_{i=0}^{\infty} M^i\Big)DZ = \Big(I - M\Big)^{-1}DZ; \quad \forall\, \mathbf{X}(0); \tag{4.9}$$

This is the limit of a geometric series of ratio M.

Consequently, a necessary and sufficient convergence condition is

$$\rho(M) < 1$$

An alternative only sufficient and more practical condition follows observing the matrix M. This, from its definition, is equal to

$$M = I - D\mathcal{H} = \begin{bmatrix} 1 & & \\ & \ddots & \\ & & 1 \end{bmatrix} - \begin{bmatrix} D_{11} & & \\ & \ddots & \\ & & D_{mm} \end{bmatrix} \begin{bmatrix} \mathcal{H}_{11} & \cdots & \mathcal{H}_{1m} \\ & \vdots & \\ \mathcal{H}_{m1} & \cdots & \mathcal{H}_{mm} \end{bmatrix}$$

where $D_{ii} = \Big(\mathcal{H}_{ii}^T R_{ii}^{-1} \mathcal{H}_{ii}\Big)^{-1} \mathcal{H}_{ii}^T R_{ii}^{-1}$. It is easy to note that

$$D\mathcal{H} = \begin{bmatrix} I & \star & \cdots \\ & \ddots & \\ \star & \cdots & I \end{bmatrix}$$

and consequently

$$M = I - D\mathcal{H} = \begin{bmatrix} 0 & \star & \cdots \\ & \ddots & \\ \star & \cdots & 0 \end{bmatrix}$$

It is well known, for the *Gershgorin theorem*, that the eigenvalues of a matrix A are confined in an area of the complex plane resulting by the union of circles centered, $\forall i$, in $[A]_{ii}$ with a radius of $\sum_{j \neq i}[A]_{ij}$. As M has a zero diagonal, to accomplish convergence it is sufficient but not necessary that $\sum_{j \neq i}[M]_{ij} < 1$.

Note that this is a sufficient condition so even if the sum of out of diagonal elements for each row is not < 1 *it is still possible to have convergence.*

4.3 The ADMM Estimator

Let us now introduce the second solution proposed: *The ADMM Estimator*. The algorithm is based on the Alternating Direction Multiplier Method. This represent an optimization technique based on the iterative solution of an augmented Lagrangian problem.

It is well known how the classical ADMM, because of its nature, can be implemented in a distributed way. However, the flow of informations through different areas do not concern local informations but global informations. This does not make the algorithm to be scalable.

The novelty of our solution is based on [13]. We will show how to implement a local and scalable ADMM algorithm to solve the estimation problem. Tests results show how the iterative procedure converges to the optimal solution of equation (3.8). Specifically, the convergence has not been rigorously demonstrated but the tests carried out suggest a good behavior of the algorithm.

Let us firstly introduce the classical ADMM procedure. Afterwards we present the scalable solution proposed.

4.3.1 Classical ADMM Algorithm

Consider a system of matrixes of the form

$$Y = \begin{bmatrix} y_1 \\ \vdots \\ y_N \end{bmatrix}; A = \begin{bmatrix} A_{11} & \cdots & A_{1N} \\ \vdots & & \vdots \\ A_{N1} & \cdots & A_{NN} \end{bmatrix}; X = \begin{bmatrix} x_1 \\ \vdots \\ x_N \end{bmatrix}; R = \begin{bmatrix} R_1 & & \\ & \ddots & \\ & & R_N \end{bmatrix}$$

It is possible, for each row or block of rows, to write the quadratic function

$$f_i(X) = \big(y_i - \sum_{j=1}^{N} A_{ij}x_j\big)^T R_i^{-1} \big(y_i - \sum_{j=1}^{N} A_{ij}x_j\big)$$

To link this formulation with our topic note that it can be thought that quantities owing to the same i^{th} block correspond to quantities owing to different area of a grid. Specifically, consider a multiareas division of the grid as that shown above. The introduced matrixes and vectors, recalling the above notation, can then be thought to be equal to

$$Y \equiv Z; \quad A \equiv \mathcal{H}; \quad X \equiv \mathbf{X}; \quad R \equiv R$$

Collecting all function f_i, $\forall i$, is easy to get

$$F(X) = \sum_{i=1}^{N} f_i(X) = \big(Y - AX\big)^T R^{-1} \big(Y - AX\big) \tag{4.10}$$

whose optimal solution, $\hat{X}$, is the well known

$$\hat{X} = \left(A^T R^{-1} A\right)^{-1} A^T R^{-1} Y$$

Let us define $X^{(i)}$, $i \in [1, \ldots, N]$, as the i^{th} copy of the vector X, owning to area i. Thanks to this it is possible to rewrite the minimization problem referred to equation in (4.10) as

$$\underset{X^{(1)} \cdots X^{(N)}}{minimize} \sum_{i=1}^{N} f_i\left(X^{(i)}\right) \quad s.t. \quad X^{(i)} = X^{(j)} \ \forall j \in \mathcal{N}_i$$

where $\mathcal{N}_i$ represents the subset of indices of areas adjacent to area i, comprising area i itself.

It is then possible to solve the minimization problem through the augmented Lagrangian technique, introducing some redundant bonds that allow us to manage the solution with the ADMM algorithm, as

$$\underset{X^{(1)} \cdots X^{(N)}}{minimize} \sum_{i=1}^{N} f_i\left(X^{(i)}\right) \quad s.t. \quad X^{(i)} = z_{ij};\ X^{(i)} = z_{ji} \ \forall j \in \mathcal{N}_i$$

which leads to a Lagrangian function equal to

$$\begin{aligned} \mathcal{L} \ = \ & \sum_{i=1}^{N} f_i\left(X^{(i)}\right) + \sum_{i=1}^{N} \sum_{j \in \mathcal{N}_i} \lambda_{ij}^T \left(X^{(i)} - z_{ij}\right) + \mu_{ij}^T \left(X^{(i)} - z_{ji}\right) + \\ & + \frac{c}{2} \sum_{i=1}^{N} \sum_{j \in \mathcal{N}_i} ||X^{(i)} - z_{ij}||^2 + ||X^{(i)} - z_{ji}||^2 \end{aligned}$$

This function can be solved through ADMM algorithm which consist of three main updating steps:

1.
$$\begin{cases} \lambda_{ij}(t) = \lambda_{ij}(t-1) + c\left(X^{(i)}(t) - z_{ij}(t)\right) \\ \mu_{ij}(t) = \mu_{ij}(t-1) + c\left(X^{(i)}(t) - z_{ji}(t)\right) \end{cases} \tag{4.11}$$

2.
$$X^{(i)}(t+1) = \underset{X^{(i)}}{argmin}\ \mathcal{L}(X, z(t), \lambda(t), \mu(t)) \tag{4.12}$$

3.
$$z_{ij}(t+1) = \underset{z_{ij}}{argmin}\ \mathcal{L}(X(t+1), z, \lambda(t), \mu(t)) \tag{4.13}$$

Considering some suitable manipulations of the updating step, see[11] and appendix A.1, it is possible to rewrite the algorithm in a simpler way consisting of only two updating step:

1.
$$\lambda_{ij}(t) = \lambda_{ij}(t-1) + \frac{c}{2}\big(X^{(i)}(t) - X^{(j)}(t)\big) \tag{4.14}$$

2.
$$X^{(i)}(t+1) = \underset{X^{(i)}}{argmin}\ \mathcal{L}(X, \lambda(t)) \tag{4.15}$$

The state update, exploiting a first order optimal condition (see appendix A.2), can finally be rewritten, for our specific quadratic problem, in a closed form, as

$$\begin{aligned} X^{(i)}(t+1) &= \underset{X^{(i)}}{argmin}\ \mathcal{L}(X, \lambda(t)) \\ &= \big[A^{(i)T} R_i^{-1} A^{(i)} + c|\mathcal{N}_i| I_n\big]^{-1} A^{(i)T} R_i^{-1} y_i(t) + \\ &+ \frac{1}{2}\big[A^{(i)T} R_i^{-1} A^{(i)} + c|\mathcal{N}_i| I_n\big]^{-1}\big(c \sum_{j\in\mathcal{N}_i} X^{(i)}(t) + X^{(j)}(t)\big) - \\ &- \frac{1}{2}\big[A^{(i)T} R_i^{-1} A^{(i)} + c|\mathcal{N}_i| I_n\big]^{-1}\big(c \sum_{j\in\mathcal{N}_i} \lambda_{ij}(t) - \lambda_{ji}(t)\big) \end{aligned} \tag{4.16}$$

where $A^{(i)} = \begin{bmatrix} A_{i1} & \dots & A_{iN} \end{bmatrix}$ indicates the i^{th} row block of matrix A.

Since every area estimates the entire grid state, the flow of informations between areas is global and not local.

On the contrary, we are interested in appropriately exploit the ADMM algorithm to let every area compute its inner state $X^{(i)}$in a distributed and local fashion, i.e. only from the knowledge of its adjacent areas state.

4.3.2 Scalable ADMM Algorithm

Let us start from the usual state of the art formulation of ADMM minimization problem

$$\underset{X^{(1)}\cdots X^{(N)}}{minimize} \sum_{i=1}^{N} f_i\big(X^{(i)}\big) \quad s.t. \quad X^{(i)} = z_{ij};\ X^{(i)} = z_{ji}\ \forall j \in \mathcal{N}_i \tag{4.17}$$

We remind that $X^{(1)}, \dots, X^{(N)}$ are local copies of X

$$X = \begin{bmatrix} x_1 \\ \vdots \\ x_N \end{bmatrix}$$

where x_i represents the inner state of area i.

To force the exchange of only local information between adjacent areas we introduce the projector matrix P_i (see [13])

$$P_i = \begin{bmatrix} O_{n_1} & & & & \\ & \ddots & & & \\ & & I_{n_i} & & \\ & & & \ddots & \\ & & & & O_{n_N} \end{bmatrix}$$

which is a diagonal matrix with the i^{th} block equal to an identity matrix of dimension n_i and zeros elsewhere. n_i represents the dimension of the state of area i. The matrix P_i lets to extract only the component x_i from vector X, indeed,

$$P_i X = \begin{bmatrix} 0 \\ \vdots \\ x_i \\ \vdots \\ 0 \end{bmatrix}$$

Similarly it is possible to define the joint projector

$$P_{ij} = \begin{cases} P_i & i = j \\ P_i + P_j & i \neq j \end{cases}$$

Note that, using the projector matrixes, it is possible to rewrite the initial problem as

$$\begin{aligned} &\underset{X^{(1)} \dots X^{(N)}}{minimize} \sum_{i=1}^{N} f_i\big(X^{(i)}\big) \\ &s.t. \quad P_{ij}X^{(i)} = P_{ij}z_{ij};\ P_{ij}X^{(i)} = P_{ij}z_{ji}\ \forall j \in \mathcal{N}_i \end{aligned} \tag{4.18}$$

where the P matrixes let to involve only local information between adjacent areas.

What to do is to apply the ADMM procedure to the problem in (4.18). This lets, see appendix A.3, to write the updating step for the local copy of $X^{(i)}$ as

$$X^{(i)}(t+1) = \underset{X^{(i)}}{argmin}\ f\big(X^{(i)}\big) + X^{(i)T} M_i X^{(i)} - X^{(i)T} B_i(t+1) \tag{4.19}$$

where

$$\begin{aligned} B_i(t+1) &= 2M_i X^{(i)}(t) + U_i(t) + \Lambda_i(t) && (4.20)\\ M_i &= c\sum_{j\in\mathcal{N}_i} P_{ij} && (4.21)\\ U_i(t) &= c\sum_{j\in\mathcal{N}_i} P_{ij}\big(X^{(i)}(t) - X^{(j)}(t)\big) && (4.22)\\ \Lambda_i(t) &= \begin{cases} 0 & t=0 \\ \Lambda_i(t-1) + U_i(t) & t\geq 1 \end{cases} && (4.23) \end{aligned}$$

and c is an arbitrary constant setting the convergence rate.

In our specific quadratic problem, equation (4.19) leads, exploiting the first order optimality conditions, to a closed form solution equals to

$$X^{(i)}(t+1) = \big[2\big(A_i^T R_i^{-1} A_i + M_i\big)\big]^{-1}\big[2A_i^T R_i^{-1} Y_i + B_i(t+1)\big] \qquad (4.24)$$

Similarly to what seen in the classical formulation, this represents our specific update step.

Note that if $P_i = I\,\forall i$, the algorithm turn to be equal to the state of the art formulation (4.17).

What we now want to do is to underline the locality of this particular formulation. Indeed, since in M_i, U_i, Λ_i and so, in B_i are considered only local information, i.e. $j \in \mathcal{N}_i$, the remaining parts of the vector could be neglected in the computation. This makes the algorithm fully scalable and local.

Scalability Property

Let us introduce the following notation to stress out the scalability property. Let $X_j^{(i)}$ be the j^{th} component of the i^{th} local copy $X^{(i)}$ of X. Namely

$$X_j^{(i)} = x_j \in area\ i$$

The update equation in (4.19) can be rewritten as

$$\begin{aligned} X^{(i)}_{k\in\mathcal{N}_i}(t+1) &= \underset{\{x_k, k\in\mathcal{N}_i\}}{argmin}\ f_i\big(\{x_k, k\in\mathcal{N}_i\}\big) + x_i^T\big(c\mathcal{N}_i x_i - \big[B_i(t+1)\big]_i\big) + \\ &\quad c\sum_{k\in\mathcal{N}_i/i} x_k^T\big(x_k - \big[B_i(t+1)\big]_k\big) \qquad (4.25) \end{aligned}$$

where $\big[B_i(t)\big]_k$ are set according to

$$\begin{aligned} \big[M_i X^{(i)}(t)\big]_k &= \begin{cases} c\mathcal{N}_i X_i^{(i)} & k = i \\ c X_k^{(i)} & k \in \mathcal{N}_i/i \end{cases} \\ \big[U_i(t)\big]_k &= \begin{cases} c\sum_{j\in\mathcal{N}_i/i}\big(X_i^{(j)}(t) - X_i^{(i)}(t)\big) & k = i \\ c\big(X^{(j)_i(t) - X_i^{(i)}(t)}\big) & k \in \mathcal{N}_i/i \end{cases} \end{aligned} \tag{4.26}$$

Note that in both equations (4.25) and (4.26) are used only local informations.

This makes the algorithm distributed and scalable. Indeed, since only local informations are used if the grid grows in size only areas adjacent to added areas will update their information exchange.

Chapter 5

Testing Results

We are going to test the proposed algorithms through several simulations executed over the `IEEE Test Feeders` [15], [16], whose graphs are shown in figure 5.1.

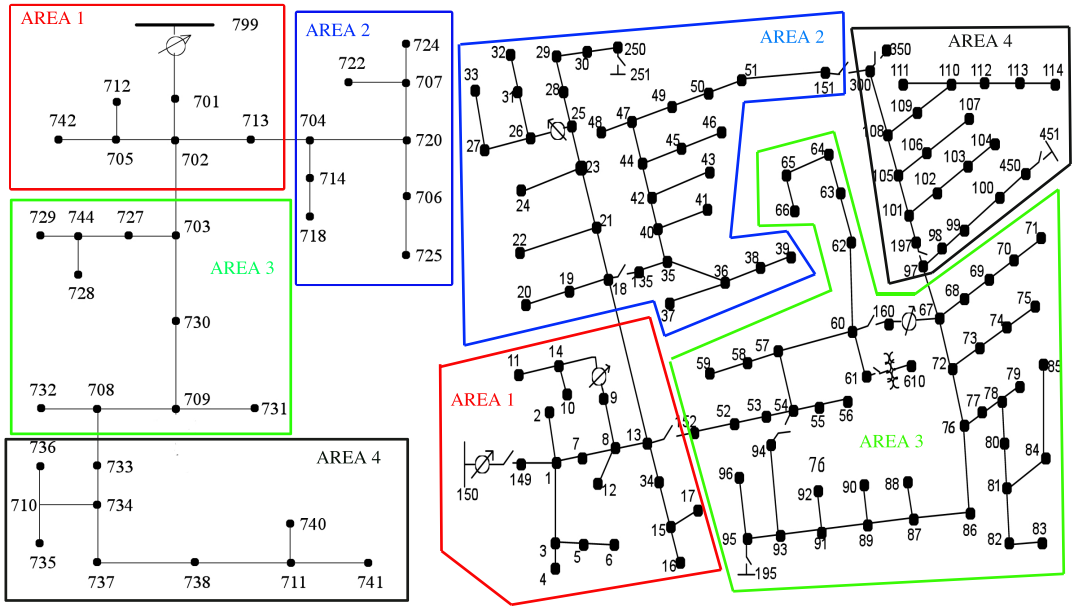


Figure 5.1: IEEE test Feeders graphs

Note that both the grids have been divided into four non overlapping areas according with section 4.1.

All tests have been carried out using MATLAB R2011b on a MAC OS X based Computer with core2duo processor clocking at 2.53GHz and 4GB of RAM.

5.1 Noise sizing

Let us assumed to measure both voltage and current, divided into amplitude and phase, at every node[1]. Such measurements are generated from a solution of the power flow problem, i.e. the complex voltage and current at every node, corrupting it with gaussian additive noise. This lets every node to be characterized by a collection of values equal to:

$$\begin{cases} v_m = v + e_v; & e_v \sim \mathcal{N}(0, \sigma_V^2); \\ \theta_m = \theta + e_\theta + e_{sync}; & e_\theta \sim \mathcal{N}(0, \sigma_\Theta^2),\ e_{sync} = \mathcal{N}(0, \sigma_{sync}^2); \\ i_m = i + e_i; & e_i \sim \mathcal{N}(0, \sigma_I^2); \\ \phi_m = \phi + e_\phi + e_{sync}; & e_\phi \sim \mathcal{N}(0, \sigma_\Phi^2),\ e_{sync} = \mathcal{N}(0, \sigma_{sync}^2); \end{cases}$$

More specifically, as default standard deviation values, it is assumed that:

- $\sigma_V = 10^{-2} PCC_{VoltageAmplitue}[Volt]$: it means to let the measures to have a standard deviation equal to 1% the PCC voltage amplitude. To better understand it for a PCC voltage amplitude of $4.16KV$ it means to have a measurement error of $\simeq 10V$ on average;
- $\sigma_I = 10^{-2} i_{max}[Ampere]$: as seen for the voltage measures, it means to let the current amplitude measurement being equal to 1% of the maximum current on average;
- $\sigma_\theta = \sigma_\phi = 10^{-2}[rad]$: for a $50Hz$ signal it means to measure a phase with an almost maximum error of $\simeq 100\mu s$.

The value of the synchronization noise standard deviation depends on what kind of synchronization unit is assumed to be in use. For example, if a GPS unit is considered a reasonable value for the synchronization standard deviation is represented by $\sigma_{sync} = 10^{-4} \div 10^{-5}[rad]$ that corresponds to a maximum error of $\simeq 1 \div 0.1\mu s$. On the contrary if just a basic synchronization algorithm is considered then, a value of $\sigma_{sync} = 10^{-2} \div 10^{-1}[rad]$ could be reasonable, corresponding to a maximum error of $\simeq 100\mu s \div 1ms$.

Note that these values represent the default testing values. It implies that smaller values lead to equal or better but absolutely not worse performances of the algorithms. The tests are carried out over several values of the standard deviation to highlight the algorithms performances in different setup.

However, to avoid misunderstanding, each test will specify the corresponding values used.

In addition to this we assume to test all estimation algorithm (section 5.2÷5.5) over the 123 nodes test feeder [16] and the Power Losses Minimization Algorithm (section 5.6) over the 37 nodes test feeder [15].

[1] In a real implementation every node is equipped with a PMU which takes the measurements.

5.2 Centralized Estimator

The centralized estimation is the starting point of our study representing the optimal global solution to the estimation problem. It is then first analyze its performance over different noise standard deviation.

Note that this estimator assumes the presence of a central unit over the entire network, able to collect and process all nodes measurements.

5.2.1 Performance for default values of noise standard deviation

It is first presented the performance of the centralized estimation algorithm subject to default values for the noise standard deviation, assuming absence of synchronization error. Figure 5.2 shows the distance of measurements and estimated state from the exact state value in p.u.[2] Note that the estimation

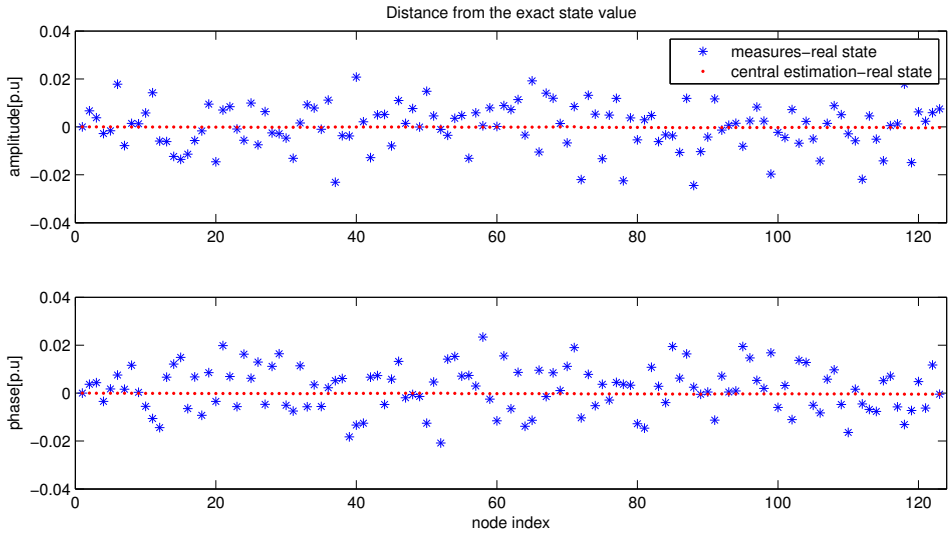


Figure 5.2: Distances of the measures and of the centralize solution from the exact state value ($\sigma_V = 10^{-2} PCC_{VoltageAmplitude}[Volt]$; $\sigma_I = 10^{-2} i_{max}$; $\sigma_\theta = \sigma_\phi = 10^{-2}[rad]$; $\sigma_{sync} = 0[rad]$).

greatly improve the knowledge of the state. In particular, table 5.1 reports a collection of quantities to quantify the improvement.

Observe how the estimated state is one and two orders of magnitude closer to the real state value, respectively in correspondence with the smallest and the greatest measured quantity. In addition to this the estimated state present only one order of magnitude variation between the worst and best

[2] $1p.u.$ voltage amplitude value $= PCC_{VoltageAmplitude}$; $1p.u$ phase value $= 1rad$.

	Measures	Estimation
min Amplitude Distance	$6.765 \cdot 10^{-5}$	$7.103 \cdot 10^{-6}$
Max Amplitude Distance	$2.447 \cdot 10^{-2}$	$4.051 \cdot 10^{-4}$
min Phase Distance	$3.350 \cdot 10^{-4}$	$2.154 \cdot 10^{-5}$
Max Phase Distance	$2.337 \cdot 10^{-2}$	$4.474 \cdot 10^{-4}$

Table 5.1: Maximum and minimum (p.u) amplitude and phase value of measures and estimated state from exact state.

distance from the real state. Instead, the measurements present a variation of three orders.

5.2.2 Performance for $\sigma_V = PCC_{VoltageAmplitude}[Volt]$

We now present the performances of the centralize estimation algorithm in correspondence of an increase of two order of magnitude of the voltage amplitude noise standard deviation. This leads to a maximum error equal to $4.16KV$ in a network with a PCC of $4.16KV$. Such an error could be considered more as a fault than as a real measure. Anyway the estimation algorithm works well as can be seen in figure 5.3.

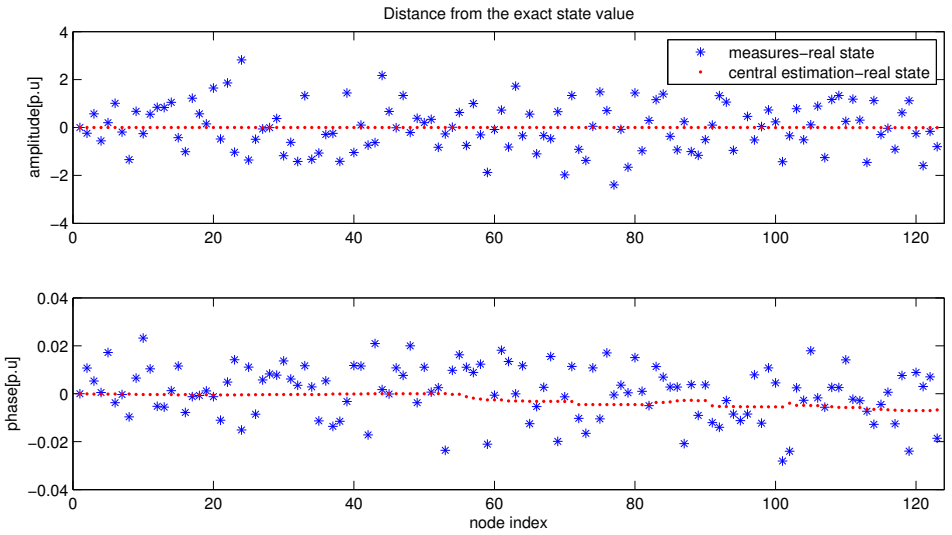


Figure 5.3: Distances of the measures and of the centralize solution from the exact state value ($\sigma_V = PCC_{VoltageAmplitude}[Volt]$; $\sigma_I = 10^{-2} i_{max}$; $\sigma_\theta = \sigma_\phi = 10^{-2}[rad]$; $\sigma_{sync} = 0[rad]$).

As previously, table 5.2 reports the maximum and minimum distance of amplitude and phase for the measures and the estimation. This time the

	Measures	Estimation
min Amplitude Distance	$6.878 \cdot 10^{-3}$	$6.369 \cdot 10^{-5}$
Max Amplitude Distance	2.820	$4.445 \cdot 10^{-3}$
min Phase Distance	$6.183 \cdot 10^{-5}$	$5.422 \cdot 10^{-7}$
Max Phase Distance	$2.809 \cdot 10^{-2}$	$7.074 \cdot 10^{-3}$

Table 5.2: Maximum and minimum (p.u) amplitude and phase value of measures and estimated state from exact state.

estimation improves the knowledge of the state of three order of magnitude correspondingly to the maximum voltage amplitude value. Of course the estimation corresponding to this set of measures is worse than the previous one presenting an average distance from the exact state one order of magnitude greater. This is outlined comparing table 5.1 with table 5.2.

5.2.3 Performance for $\sigma_\theta = \sigma_\phi = 10^{-1}[rad]$

This set of measures present default values for the standard deviations excepting for phase measurements. This is fixed equal to $\sigma_\theta = \sigma_\phi = 10^{-1}[rad]$. As previously mentioned, this corresponds to a maximum error of $1ms$ for a $50Hz$ signal.

Note from figure 5.4 the estimation improvements.

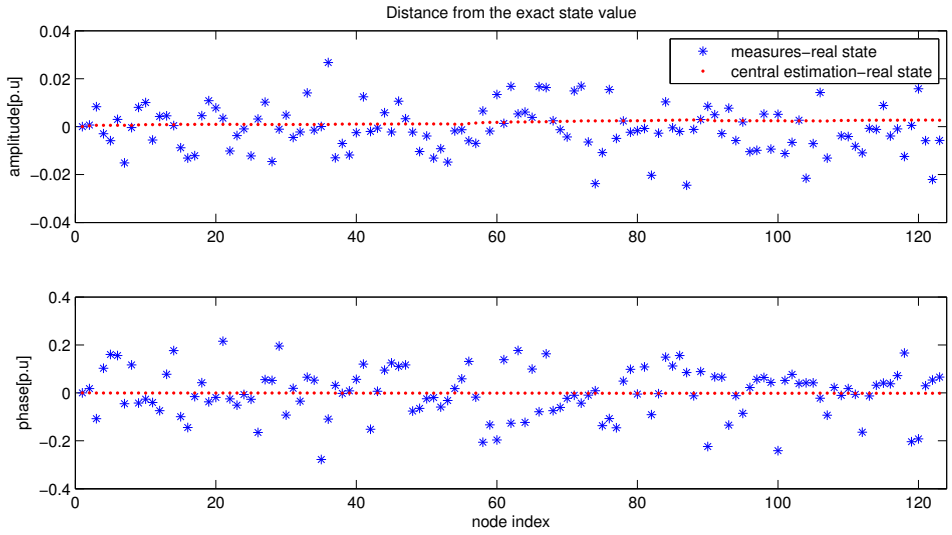


Figure 5.4: Distances of the measures and of the centralize solution from the exact state value ($\sigma_V = 10^{-2} PCC_{VoltageAmplitude}[Volt]$; $\sigma_I = 10^{-2} i_{max}$; $\sigma_\theta = \sigma_\phi = 10^{-1}[rad]$; $\sigma_{sync} = 0[rad]$).

5.2.4 Performance for $\sigma_{sync} = 10^{-1}[rad]$

It is finally reported the performance of the algorithm corresponding to a synchronization noise standard deviation equal to $\sigma_{sync} = 10^{-1}[rad]$. This means to handle with a poorly synchronized measurement units that specifically cannot synchronize the measure under $\simeq 1ms$. Figure 5.5 shows the comparison between the distance of the measures and of the estimated state from the exact state.

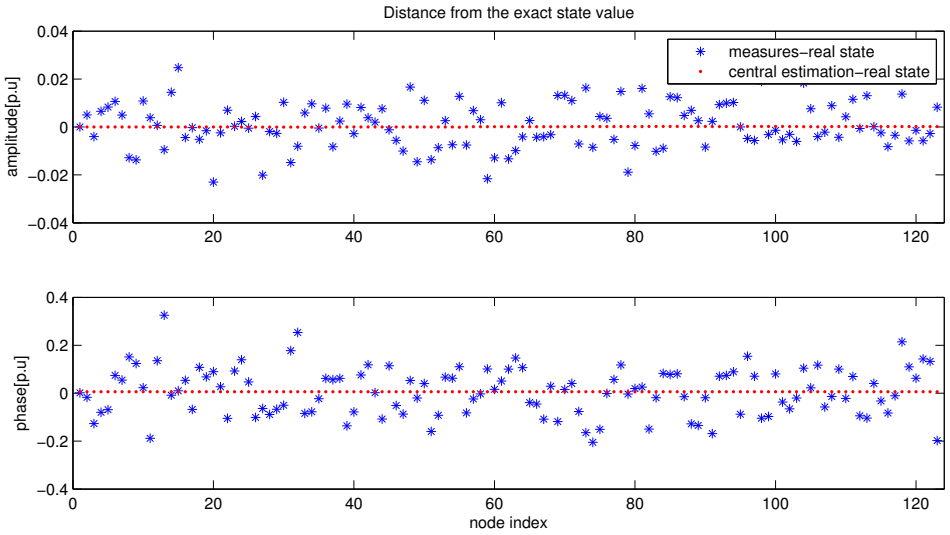


Figure 5.5: Distances of the measures and of the centralize solution from the exact state value ($\sigma_V = 10^{-2} PCC_{VoltageAmplitude}[Volt]$; $\sigma_I = 10^{-2} i_{max}$; $\sigma_\theta = \sigma_\phi = 10^{-2}[rad]$; $\sigma_{sync} = 10^{-1}[rad]$).

Table 5.3 collects the maximum and minimum distance from it.

	Measures	Estimation
min Amplitude Distance	$1.549 \cdot 10^{-5}$	$7.464 \cdot 10^{-7}$
Max Amplitude Distance	$2.472 \cdot 10^{-2}$	$2.346 \cdot 10^{-4}$
min Phase Distance	$1.145 \cdot 10^{-3}$	$6.113 \cdot 10^{-3}$
Max Phase Distance	$3.254 \cdot 10^{-1}$	$6.409 \cdot 10^{-3}$

Table 5.3: Maximum and minimum (p.u) amplitude and phase value of measures and estimated state from exact state.

Note, comparing table 5.3 with table 5.1 how the improvement due to the estimation is similar even in presence of a poor synchronization. Of course the performance in absence of synchronization noise are more accurate but a synchronization error of $1ms$ is absolutely not negligible.

5.2.5 Computational Effort

All tests shown how the centralize estimator works well in a wide range of different noise standard deviation values. Tests carried out with greater values show that the estimation becomes not significant to an useful usage with respect to the measurements. However greater noise values, being grossly inaccurate, could be less significant in a real environment.

In addition to its wide range of usage, is important to remark that thanks to its closed form solution the algorithm requires very low computational effort and is solved only in one iteration. The average time required to solve the 123 nodes test feeder is equal to

$$T_{AverageCentralEstimation} \simeq 10^{-2}[s]$$

Anyway representing a centralize and not distributed nor local estimation solution, the algorithm needs the knowledge of the entire grid topology, is absolutely not scalable and requiring a flow of information through all the network is not robust to failure.

5.3 The Distributed Estimator

In this section we report the testing results of the first distributed and local algorithm proposed. To test it, the grid has been divided in four sub areas as shown in figure 5.1. The communications lines, which are bidirectional, are established between: areas 1 and 2; 1 and 3; 3 and 4.
The default noise standard deviation values are the same of those seen for the centralized estimatot, equal to:

- $\sigma_V = 10^{-2} PCC_{VoltageAmplitue}[Volt]$;
- $\sigma_I = 10^{-2} i_{max}[Ampere]$;
- $\sigma_\theta = \sigma_\phi = 10^{-2}[rad]$;
- $\sigma_{sync} = 0[rad]$

The tests highlights the algorithm performances showing its response to different noise standard deviation values. The results are compared with both the real state value and the central estimated value.

5.3.1 Performance for default values of noise standard deviation

The performance of the algorithm for default values of the standard deviation is shown in figure 5.6 in which is reported the distance from the real state.

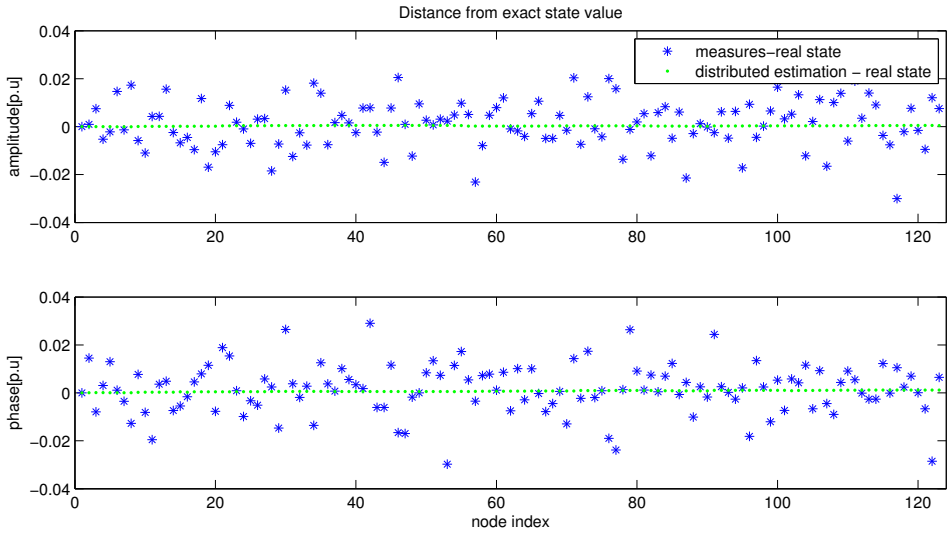


Figure 5.6: Distances of the measures and of the distributed solution from the exact state value. ($\sigma_V = 10^{-2} PCC_{VoltageAmplitude}[Volt]$; $\sigma_I = 10^{-2} i_{max}$; $\sigma_\theta = \sigma_\phi = 10^{-2}[rad]$; $\sigma_{sync} = 0[rad]$)

As can be seen the estimation largely improves the knowledge of the state with respect to the measurements. In addition to this, figure 5.7 shows a comparison between the central and the distributed estimations respectively obtained with the estimator proposed. Remind that the distributed algorithm converges to an approximate solution and because of this will never be exactly equal to the centralize optimal solution. Anyway the order of magnitude of their distance from the real state value is comparable.

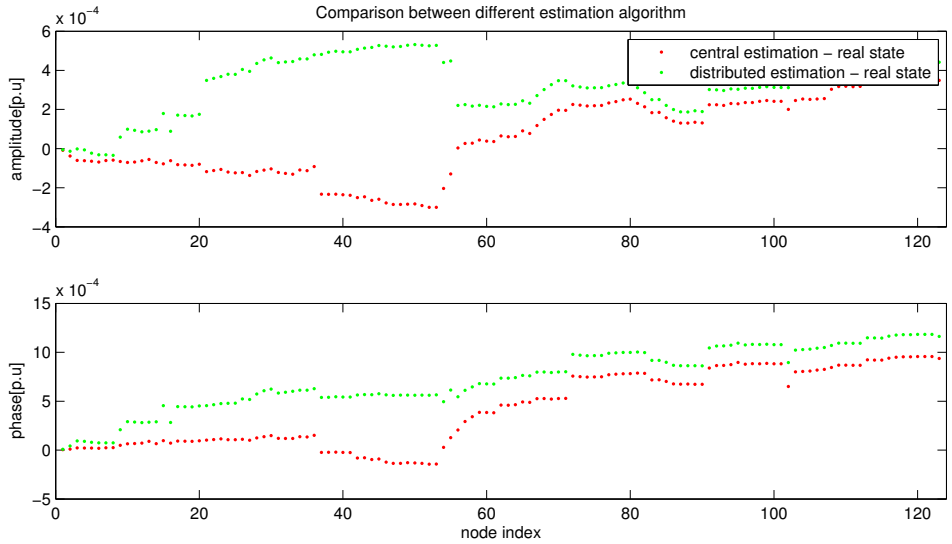


Figure 5.7: Distances of the centralize and of the distributed solution from the exact state value. ($\sigma_V = 10^{-2} PCC_{VoltageAmplitude}[Volt]$; $\sigma_I = 10^{-2} i_{max}$; $\sigma_\theta = \sigma_\phi = 10^{-2}[rad]$; $\sigma_{sync} = 0[rad]$)

Note how the difference of the two estimation is relatively small compared with the measurements. Table 5.4 sums up some results that help the comparison between the algorithms with respect to the real state value and the measurements.

	Measures	Central	Distributed
min Amplitude Distance	$1.582 \cdot 10^{-4}$	$3.309 \cdot 10^{-6}$	$1.727 \cdot 10^{-6}$
Max Amplitude Distance	$3.005 \cdot 10^{-2}$	$3.988 \cdot 10^{-4}$	$5.315 \cdot 10^{-4}$
min Phase Distance	$3.123 \cdot 10^{-6}$	$1.007 \cdot 10^{-5}$	$4.222 \cdot 10^{-5}$
Max Phase Distance	$2.974 \cdot 10^{-2}$	$9.572 \cdot 10^{-4}$	$1.183 \cdot 10^{-3}$

Table 5.4: Maximum and minimum (p.u) amplitude and phase value of measures and estimated state from exact state.

Note how, comparing their maximum and minimum distance from the real state, the performances of the two algorithms are very similar.

5.3.2 Performance for $\sigma_V = 10^{-1} PCC_{VoltageAmplitude}[Volt]$

For a variation in the value of the voltage amplitude standard deviation noise fixed equal to $\sigma_V = 10^{-1} PCC_{VoltageAmplitude}[Volt]$, the performance of the distributed algorithm is reported in figure 5.8. Remind that such a standard deviation value correspond to an average error equal to almost the 10% of the PCC voltage amplitude value, one order of magnitude greater than the default value early analyzed.

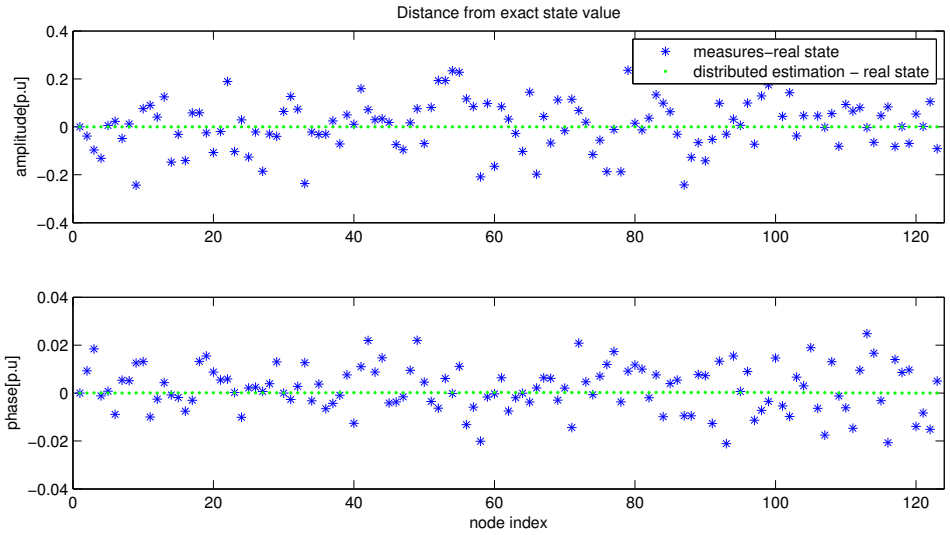


Figure 5.8: Distances of the measures and of the distributed solution from the exact state value. ($\sigma_V = 10^{-1} PCC_{VoltageAmplitude}[Volt]$; $\sigma_I = 10^{-2} i_{max}$; $\sigma_\theta = \sigma_\phi = 10^{-2}[rad]$; $\sigma_{sync} = 0[rad]$)

Again, figure 5.9 shows the comparison of the central and distributed algorithms. Differently to figure 5.7, note how the relative distance, mainly of the estimated phase, between the two solutions increase. This is obviously due to the greater noise that makes harder the estimation for the distributed approximate algorithm.

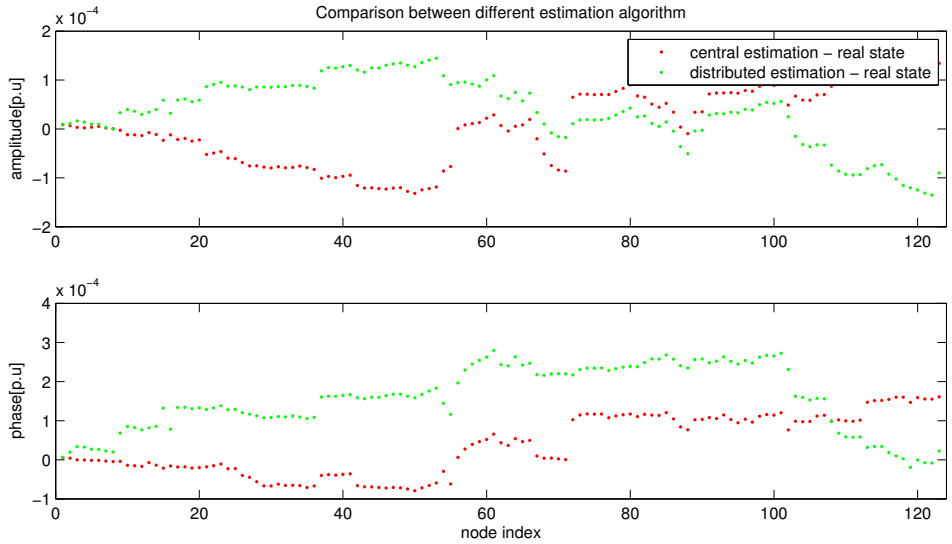


Figure 5.9: Distances of the centralize and of the distributed solution from the exact state value. ($\sigma_V = 10^{-1} PCC_{VoltageAmplitude}[Volt]$; $\sigma_I = 10^{-2} i_{max}$; $\sigma_\theta = \sigma_\phi = 10^{-2}[rad]$; $\sigma_{sync} = 0[rad]$)

Anyway the minimum and maximum distance from the real state of the algorithms remains of the same order of magnitude as shown in table 5.5. This suggests that even if the distance from the optimal solution is lightly worse the overall performance of the distributed algorithm is still remarkable.

	Measures	Central	Distributed
min Amplitude Distance	$9.786 \cdot 10^{-4}$	$3.809 \cdot 10^{-7}$	$1.275 \cdot 10^{-7}$
Max Amplitude Distance	$2.434 \cdot 10^{-1}$	$1.456 \cdot 10^{-4}$	$1.445 \cdot 10^{-4}$
min Phase Distance	$2847 \cdot 10^{-6}$	$2.714 \cdot 10^{-7}$	$5.675 \cdot 10^{-7}$
Max Phase Distance	$2.480 \cdot 10^{-2}$	$1.608 \cdot 10^{-4}$	$2.797 \cdot 10^{-4}$

Table 5.5: Maximum and minimum (p.u) amplitude and phase value of measures and estimated state from exact state.

5.3.3 Performance for $\sigma_\theta = \sigma_\phi = 10^{-1}[rad]$

The algorithm was tested in presence of a greater phase noise as well. Its performances are shown in figure 5.10 in which is easy to see that the algorithm works well.

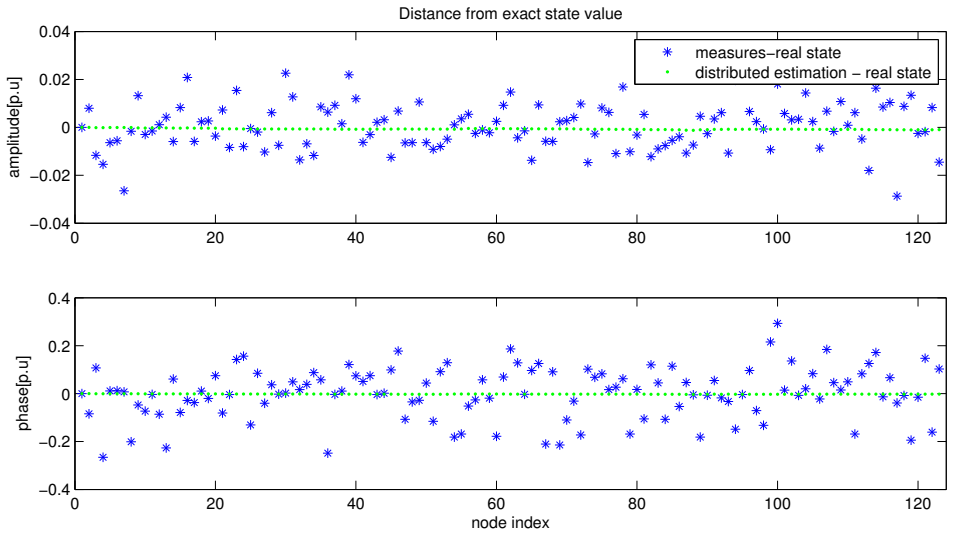


Figure 5.10: Distances of the measures and of the distributed solution from the exact state value. ($\sigma_V = 10^{-2} PCC_{VoltageAmplitude}[Volt]$; $\sigma_I = 10^{-2} i_{max}$; $\sigma_\theta = \sigma_\phi = 10^{-1}[rad]$; $\sigma_{sync} = 0[rad]$)

Its distance from the optimal central solution does not appreciably change with respect to other standard deviation values and is not reported.

5.3.4 Presence of Synchronization noise

Let us analyze the presence of synchronization noise. The centralize estimator is able to work even with poor synchronized measurements, see section 5.2. In particular section 5.2.4 shows its performance with a noise characterized by a standard deviation value of $\sigma_{sync} = 10^{-1}[rad]$ that correspond to an error of $\simeq 1ms$ for a $50Hz$ signal. This means that algorithm could work with basic synchronization algorithm as well.

On the contrary, the behavior of the distributed estimation is deeply different. The algorithm becomes unstable in presence of a relatively small synchronization noise, i.e. with a standard deviation of $\sigma_{sync} = 10^{-3}[rad]$ corresponding to an error of $\simeq 10\mu s$. Such a behavior leads to an employment of the distributed algorithm only if at least a GPS synchronization unit is present. Indeed, a GPS unit leads to an error of $\simeq 0.1\mu s$ that corresponds to a standard deviation of $\sigma_{sync} = 10^{-5}[rad]$.

Anyway, up to a value of $\sigma_{sync} = 10^{-4}[rad]$, the distributed estimator works well as shown in figure 5.11 and figure 5.12 in which are shown the distances from the real state value with respect to the measurements and to the centralize estimation respectively.

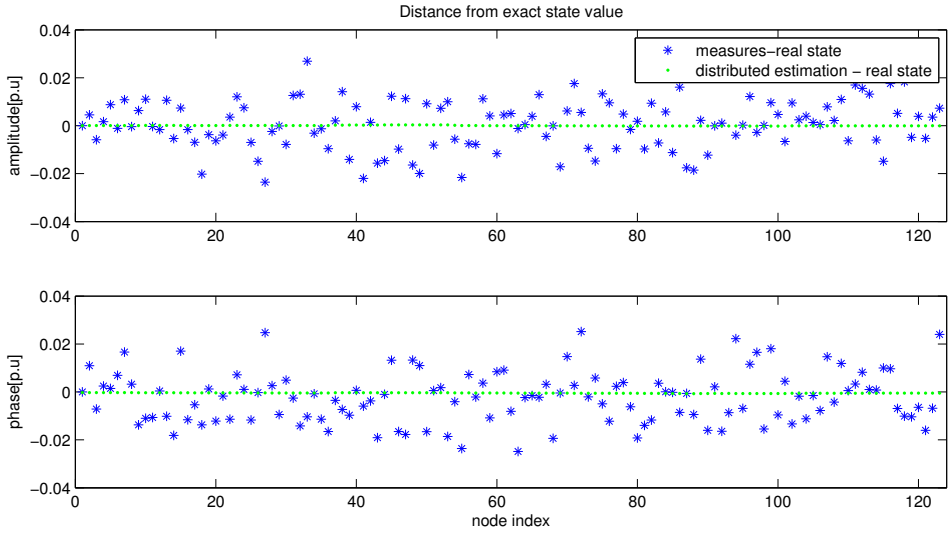


Figure 5.11: Distances of the measures and of the distributed solution from the exact state value. ($\sigma_V = 10^{-2} PCC_{VoltageAmplitude}[Volt]$; $\sigma_I = 10^{-2} i_{max}$; $\sigma_\theta = \sigma_\phi = 10^{-2}[rad]$; $\sigma_{sync} = 10^{-4}[rad]$)

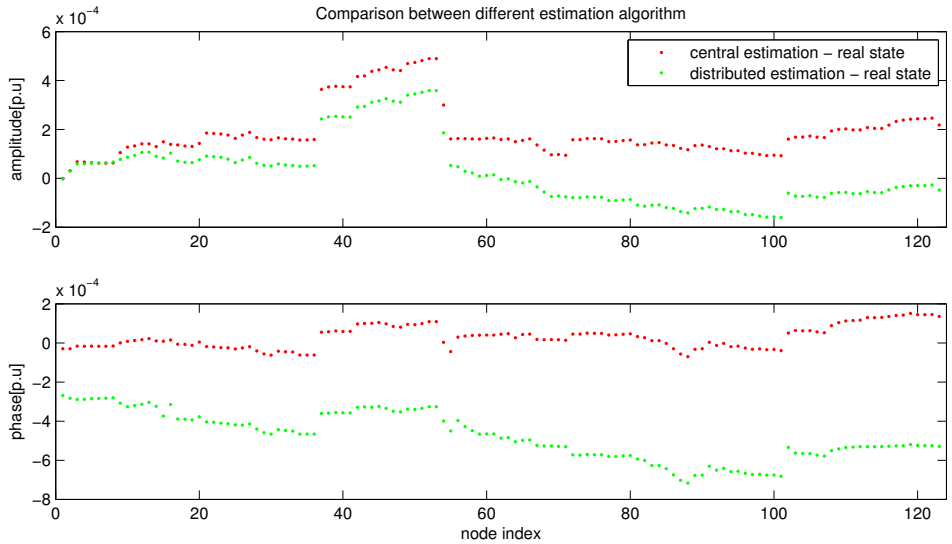


Figure 5.12: Distances of the centralize and of the distributed solution from the exact state value. ($\sigma_V = 10^{-2} PCC_{VoltageAmplitude}[Volt]$; $\sigma_I = 10^{-2} i_{max}$; $\sigma_\theta = \sigma_\phi = 10^{-2}[rad]$; $\sigma_{sync} = 10^{-4}[rad]$)

To better understand the behavior of the algorithm to an increase of the synchronization standard deviation, table 5.6 reports the values of the amplitude and phase sample standard deviation of the centralize and distributed estimated state from the real state value. This is computed as

$$Std_{sample} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(Estimated\ State_i - Exact\ State_i)^2}$$

where N is the number of nodes and $Estimated\ State_i$, $Exact\ State_i$ represent the estimated and exact state value of node i.

	Amplitude Sample Std		Phase Sample Std	
$\sigma_{sync}[rad]$	Central	Distributed	Central	Distributed
10^{-5}	$2.089 \cdot 10^{-4}$	$2.899 \cdot 10^{-4}$	$2.116 \cdot 10^{-4}$	$4.016 \cdot 10^{-4}$
10^{-4}	$1.753 \cdot 10^{-4}$	$2.953 \cdot 10^{-4}$	$2.212 \cdot 10^{-4}$	$6.945 \cdot 10^{-4}$
10^{-3}	$1.809 \cdot 10^{-4}$	$7.549 \cdot 10^{-4}$	$5.732 \cdot 10^{-4}$	$1.045 \cdot 10^{-2}$
10^{-2}	$2.109 \cdot 10^{-4}$	$2.684 \cdot 10^{143}$	$1.111 \cdot 10^{-3}$	$7.119 \cdot 10^{-1}$
10^{-1}	$6.032 \cdot 10^{-3}$	∞	$2.460 \cdot 10^{-2}$	1.143

Table 5.6: Amplitude and Phase sample standard deviation (p.u.) of the estimation from the real state corresponding to different sync noise standard deviation values.

It can be seen how for a value of $\sigma_{sync} = 10^{-3}[rad]$ the sample standard deviation value of the phase begin to increase. For a value of $\sigma_{sync} = 10^{-2}[rad]$ and above, the amplitude sample standard deviation highlights the instability of the distributed algorithm.

5.3.5 Computational Effort

The tests show how the distributed approximated estimator works well in a wide range of different noise standard deviation. Its performances are comparable with those of the optimal central estimation except for the case of presence of synchronization noise. The distributed algorithm requires more synchronization.

Excepting this case, the performance and the computational effort of the distributed solution is almost similar to the central one. Differently from the centralized algorithm it takes different iterations to converge but its running time remains almost constant since the algorithm is local and so scalable.

Table 5.7 reports the average number of iteration to convergence and the average running time for an execution on a 123 nodes test feeder divided in 4 areas almost of the same inner state dimension.

# iterations	Time[s]
$\simeq 10^3$	$\simeq 3 \cdot 10^{-3}$

Table 5.7: Number of iteration and running time of the distributed estimator tested on the 123 nodes test feeder

Thanks to the scalability, the computational time, for each area, remains constant adding different areas to the grid. The central estimator, on the contrary, increase its computational effort almost linearly with an increasing in size.

5.4 The ADMM Estimator

The setup used for testing the second algorithm proposed is equivalent to that used for testing the distributed algorithm shown in section 5.3. So the grid in considered divided into four sub areas. There is only one main difference to that setup:

- in the Admm classical (global) algorithm, every area estimates the entire network state. This means that even if every area communicates only with adjacent areas (distributed property), adjacent areas send to each other information belonging all network areas (globality);
- in the Admm scalable (local) solution proposed, on the contrary, adjacent areas estimate only neighbor areas state and so exchange only local information.

This does not affect the communication lines but only the type of information exchanged.

Finally, the noise default standard deviation values are equal to those seen above equal to:

- $\sigma_V = 10^{-2} PCC_{VoltageAmplitue}[Volt]$;
- $\sigma_I = 10^{-2} i_{max}[Ampere]$;
- $\sigma_\theta = \sigma_\phi = 10^{-2}[rad]$;
- $\sigma_{sync} = 0[rad]$

5.4.1 Convergence between *global* and *local* Admm algorithms

Before testing the scalable local algorithm proposed, a simple convergence analysis between the two algorithm is performed. Obviously, only the inner state of single areas are compared since the same area estimates different states in the global and local version.

In particular figure 5.13 reports the distance between the entire network estimated state[3] obtained from the global Admm algorithm and from the local one. Note that the order of magnitude of the difference is 10^{-8}. This can be considered almost as a computational error.

[3]The entire state is obtain appending different areas state.

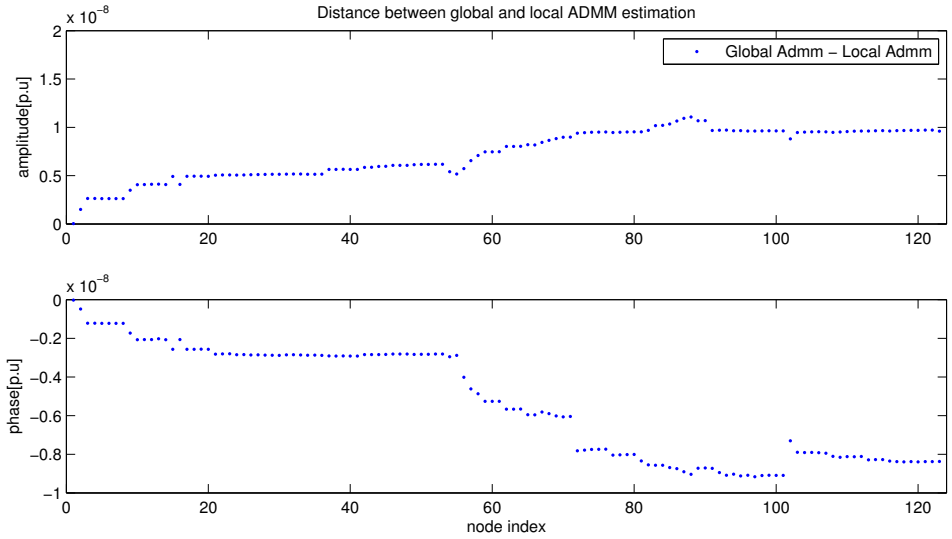


Figure 5.13: Distances between global and local Admm estimate state ($\sigma_V = 10^{-2} PCC_{VoltageAmplitude}[Volt]$; $\sigma_I = 10^{-2} i_{max}$; $\sigma_\theta = \sigma_\phi = 10^{-2}[rad]$; $\sigma_{sync} = 0[rad]$)

In addition to this is important to underline that the Admm algorithm converges to the central optimal solution. Figure 5.14 shows the distance of both Admm algorithms, global and local, from the optimal solution obtain with the centralize estimator of section 5.2.
Finally, figure 5.15 shows, in a logarithmic scale, the difference between the cost function in (3.2), computed in the Admm and Optimal estimated state.

As can be seen the distance from the optimal solution is characterized by 10^{-9} order of magnitude that can be considered computational error again. At the same time the difference between the cost functions are of the same order of magnitude being expressed in a logarithmic scale.

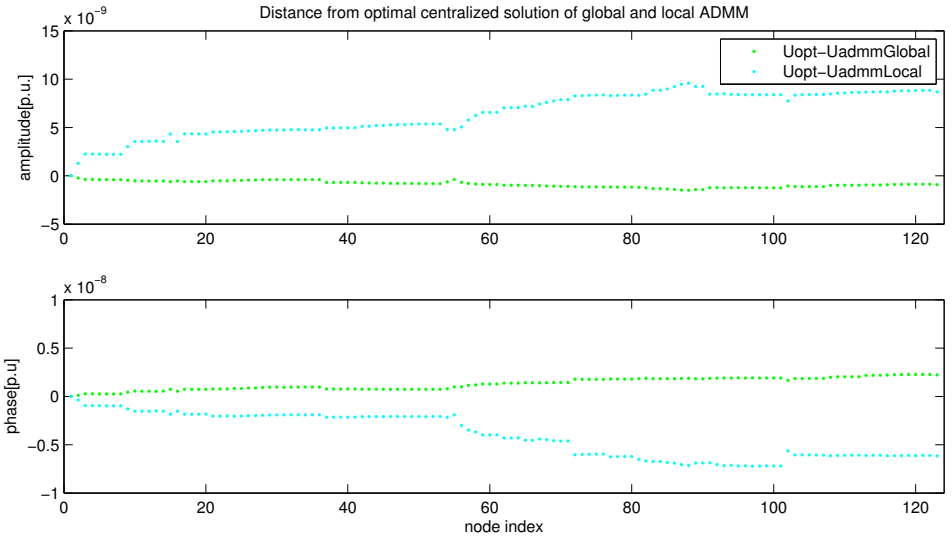


Figure 5.14: Distances of global and local Admm estimate state from the optimal solution. ($\sigma_V = 10^{-2} PCC_{VoltageAmplitude}[Volt]$; $\sigma_I = 10^{-2} i_{max}$; $\sigma_\theta = \sigma_\phi = 10^{-2}[rad]$; $\sigma_{sync} = 0[rad]$)

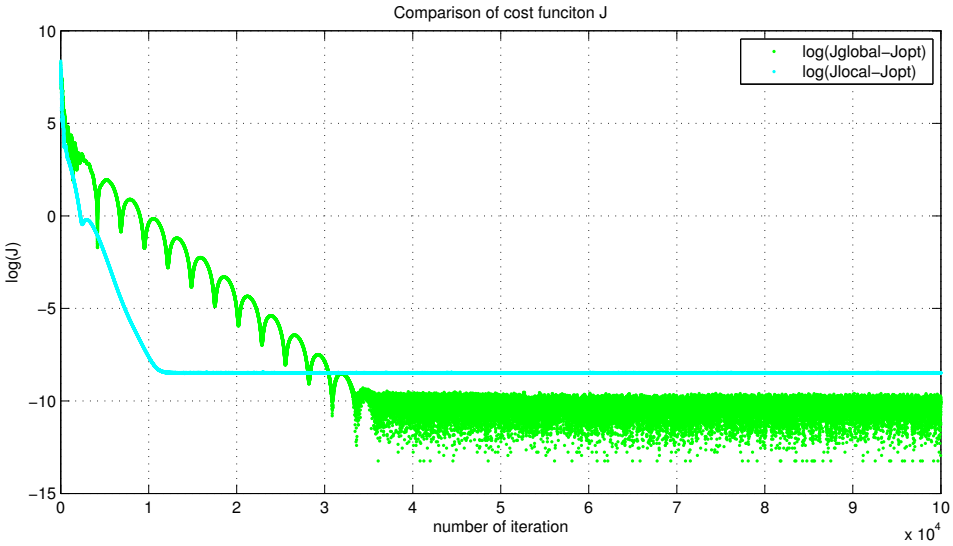


Figure 5.15: Comparison of cost function computed in Admm and Optimal estimated state. ($\sigma_V = 10^{-2} PCC_{VoltageAmplitude}[Volt]$; $\sigma_I = 10^{-2} i_{max}$; $\sigma_\theta = \sigma_\phi = 10^{-2}[rad]$; $\sigma_{sync} = 0[rad]$)

	Global - Local	Centralize - Local
min Amplitude Distance	$1.505 \cdot 10^{-9}$	$1.280 \cdot 10^{-9}$
Max Amplitude Distance	$1.107 \cdot 10^{-8}$	$9.584 \cdot 10^{-9}$
min Phase Distance	$4.845 \cdot 10^{-10}$	$3.724 \cdot 10^{-10}$
Max Phase Distance	$9.163 \cdot 10^{-9}$	$7.242 \cdot 10^{-9}$

Table 5.8: Maximum and minimum (p.u.) distance between estimations.

For the sake of clarity table 5.8 collects the amplitude and phase maximum and minimum distance between the global and local and between the centralize and the local solutions.

5.4.2 Performance of Admm scalable (local) algorithm

Once understood the behavior of the Admm algorithm with respect to the optimal centralize solution, is tested no more the performance of the algorithm over a variation of all noise standard deviation values, but only over a variation of the synchronization noise standard deviation value. This is because, as seen in section 5.3, the distributed estimator is not robust in presence of such kind of noise and requires at least GPS synchronization units to work well.

The tests in the following want to highlight the robustness of the scalable Admm algorithm in presence of synchronization noise and to analyze its convergence number of iterations and time.

Performance for $\sigma_{sync} = 10^{-2}[rad]$

Fixing the value of $\sigma_{sync} = 10^{-2}[rad]$, corresponding to a maximum error of $\simeq 100\mu s$, the scalable algorithm shows a good behavior. Figure 5.16 shows an appreciable improvement with respect to the measures.

Figure 5.17 shows the comparison between the distance of the Admm and of the centralize optimal solutions from the exact state. It easy to note that the two distance cannot be distinguish because they almost coincide.

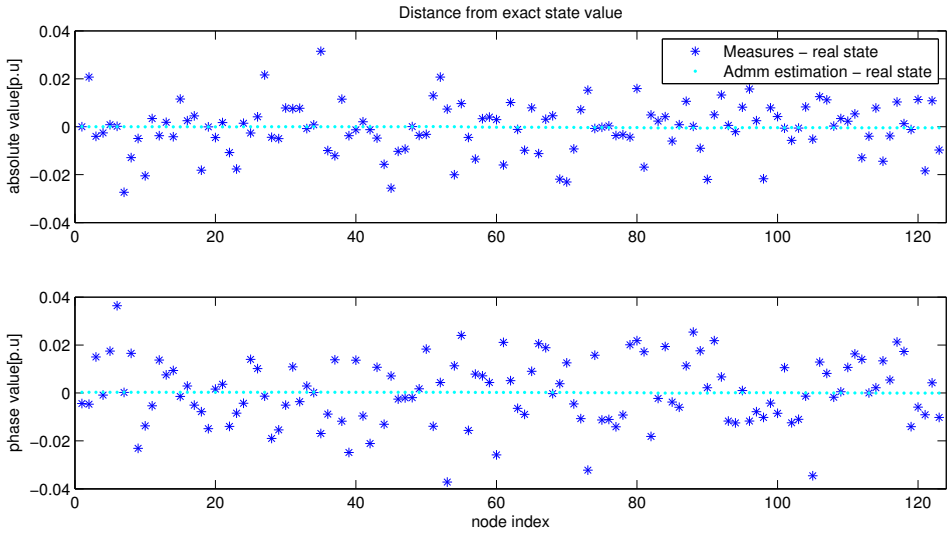


Figure 5.16: Distance of measures and local Admm estimate state from the exact state. ($\sigma_V = 10^{-2} PCC_{VoltageAmplitude}[Volt]$; $\sigma_I = 10^{-2} i_{max}$; $\sigma_\theta = \sigma_\phi = 10^{-2}[rad]$; $\sigma_{sync} = 10^{-2}[rad]$)

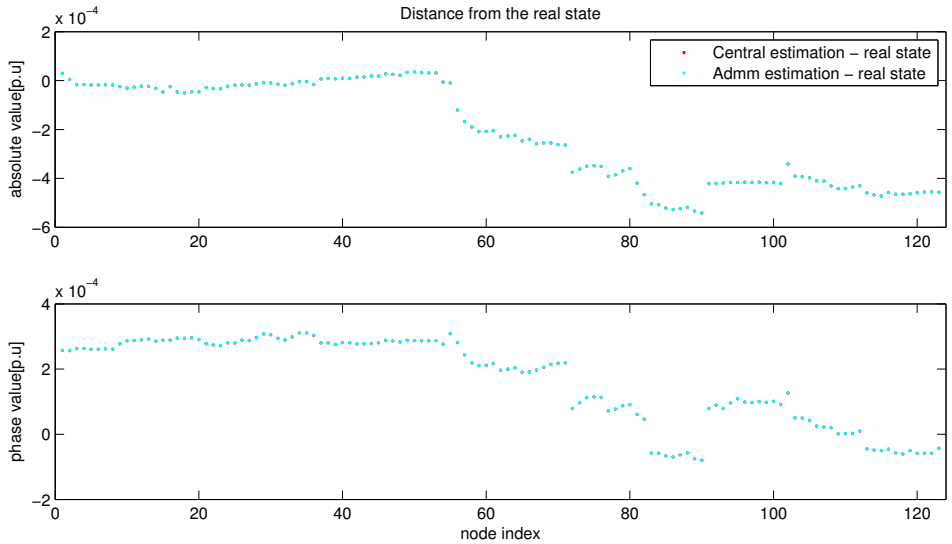


Figure 5.17: Comparison between the distance of local Admm and Optimal estimated state from the exact state. ($\sigma_V = 10^{-2} PCC_{VoltageAmplitude}[Volt]$; $\sigma_I = 10^{-2} i_{max}$; $\sigma_\theta = \sigma_\phi = 10^{-2}[rad]$; $\sigma_{sync} = 10^{-2}[rad]$)

Performance for $\sigma_{sync} = 10^{-1}[rad]$

The value of $\sigma_{sync} = 10^{-1}[rad]$, which corresponds to a maximum error of $\simeq 1ms$, is the maximum value for the synchronization noise standard deviation that leads to a useful central estimation as told in section 5.2.

The behavior of the Admm algorithm, once again, is to tend to the optimal solution. However, because of the huge synchronization error, the rate of convergence is slower than before. This simply means the algorithm requires more iterations to converge.

Figure 5.18 shows that, with respect to the measures, the improvement in the state knowledge is still good. Anyway the distance from the optimal solution, after 10^4 iterations, is still not negligible.

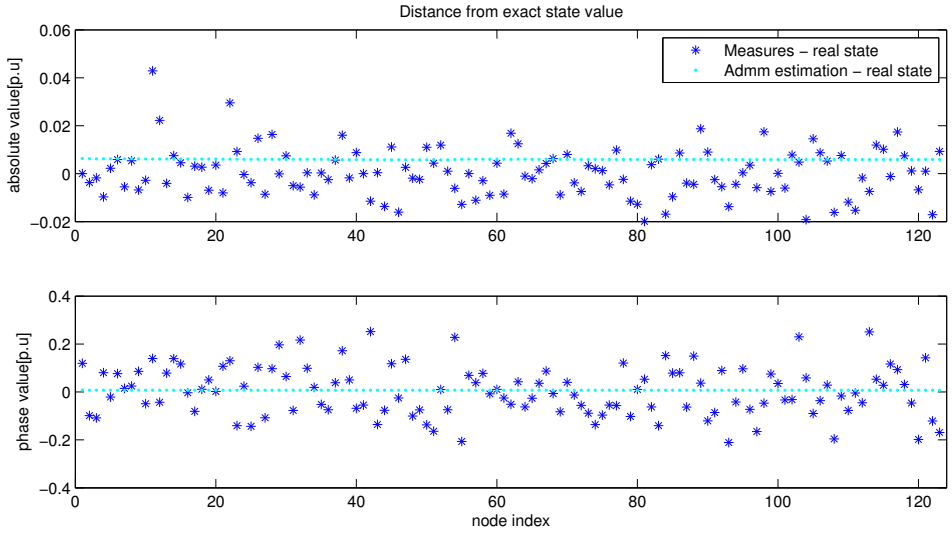


Figure 5.18: Distance of measures and local Admm estimated state from the exact state after 10^4 iterations. ($\sigma_V = 10^{-2} PCC_{VoltageAmplitude}[Volt]$; $\sigma_I = 10^{-2} i_{max}$; $\sigma_\theta = \sigma_\phi = 10^{-2}[rad]$; $\sigma_{sync} = 10^{-1}[rad]$)

To better understand it, the performances of the algorithm after 10^4 and after $2 \cdot 10^4$ iterations are compared in the following.

Figure 5.19 and figure 5.20 show the distance of the admm state and of the centralize state from the exact one. Consecutively table 5.9 reports the maximum and minimum values of that distances and the running time required.

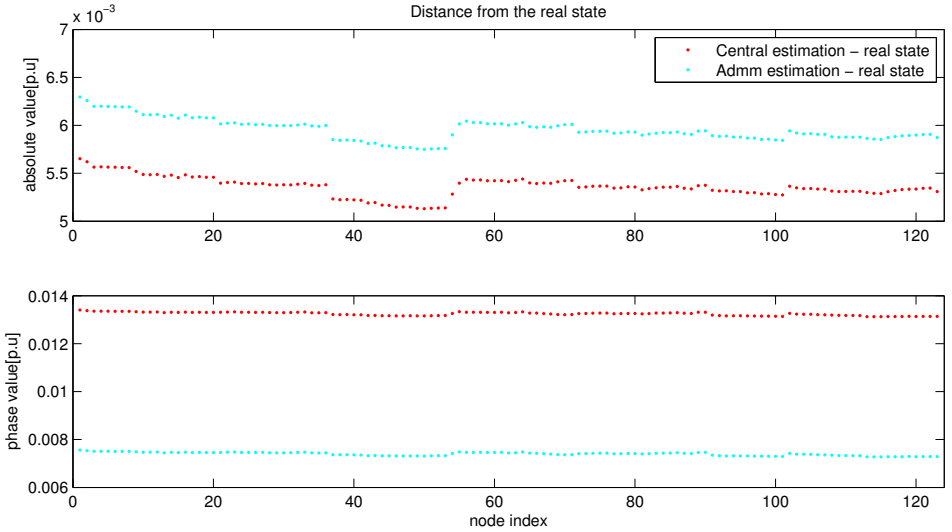


Figure 5.19: Distance of the local Admm and the centralize estimated state from the exact state after 10^4 iterations. ($\sigma_V = 10^{-2} PCC_{VoltageAmplitude}[Volt]$; $\sigma_I = 10^{-2} i_{max}$; $\sigma_\theta = \sigma_\phi = 10^{-2}[rad]$; $\sigma_{sync} = 10^{-1}[rad]$)

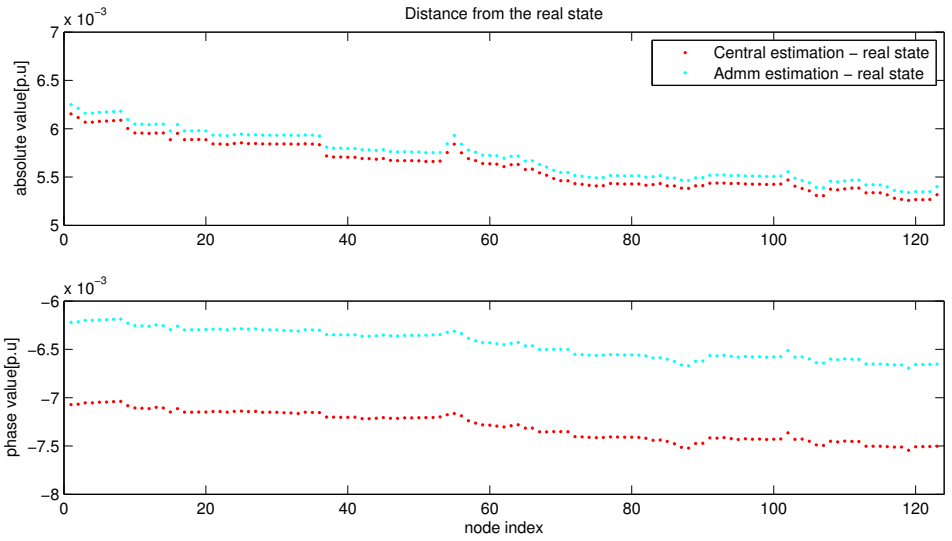


Figure 5.20: Distance of the local Admm and centralize estimated state from the exact state after $2{\cdot}10^4$ iterations. ($\sigma_V = 10^{-2} PCC_{VoltageAmplitude}[Volt]$; $\sigma_I = 10^{-2} i_{max}$; $\sigma_\theta = \sigma_\phi = 10^{-2}[rad]$; $\sigma_{sync} = 10^{-1}[rad]$)

	Central - Local	Central - Local
# iterations	10^4	$2 \cdot 10^4$
min Amplitude Distance	$5.612 \cdot 10^{-4}$	$8.186 \cdot 10^{-5}$
Max Amplitude Distance	$6.388 \cdot 10^{-5}$	$9.461 \cdot 10^{-5}$
min Phase Distance	$5.845 \cdot 10^{-3}$	$8.504 \cdot 10^{-4}$
Max Phase Distance	$5.853 \cdot 10^{-3}$	$8.534 \cdot 10^{-4}$
Running Time[s]	$1.074 \cdot 10^2$	$2.168 \cdot 10^2$

Table 5.9: Comparison of performance for different number of iterations: Maximum and minimum (p.u.) distance between admm and central estimations.

Note that, comparing the images, the relative distance between the admm and the central state gets smaller increasing the number of iterations.

From table 5.9 the order of magnitude decrease of one unit doubling the number of iterations. Anyway comparing table 5.9 with table 5.8 the distance between the two solution are still not negligible meaning that it should wait more iteration to reach a complete convergence.

The same behavior is pointed out by the cost functions progress as can be seen comparing figure 5.21 and figure 5.22.

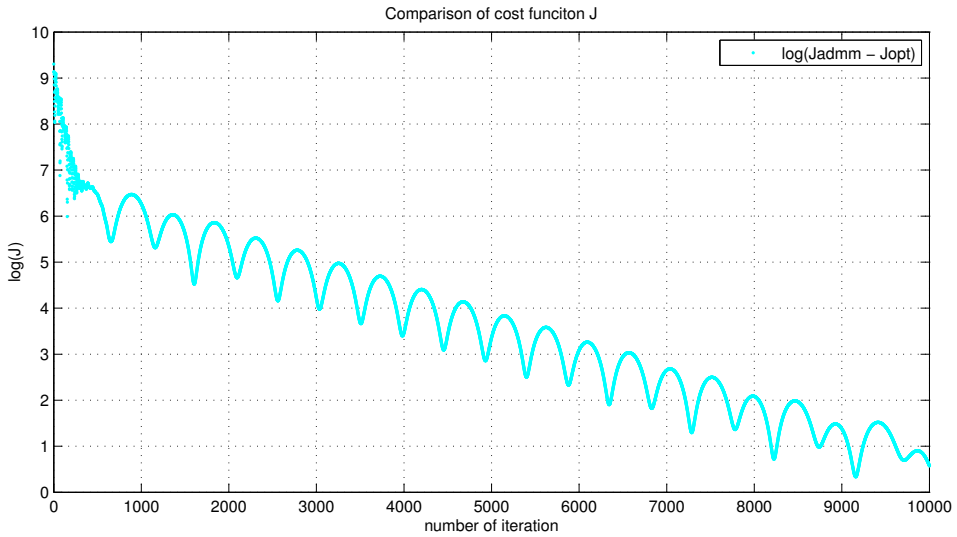


Figure 5.21: Comparison of cost function computed in Admm and Optimal estimated state after 10^4 iterations. ($\sigma_V = 10^{-2} PCC_{VoltageAmplitude}[Volt]$; $\sigma_I = 10^{-2} i_{max}$; $\sigma_\theta = \sigma_\phi = 10^{-2}[rad]$; $\sigma_{sync} = 0[rad]$)

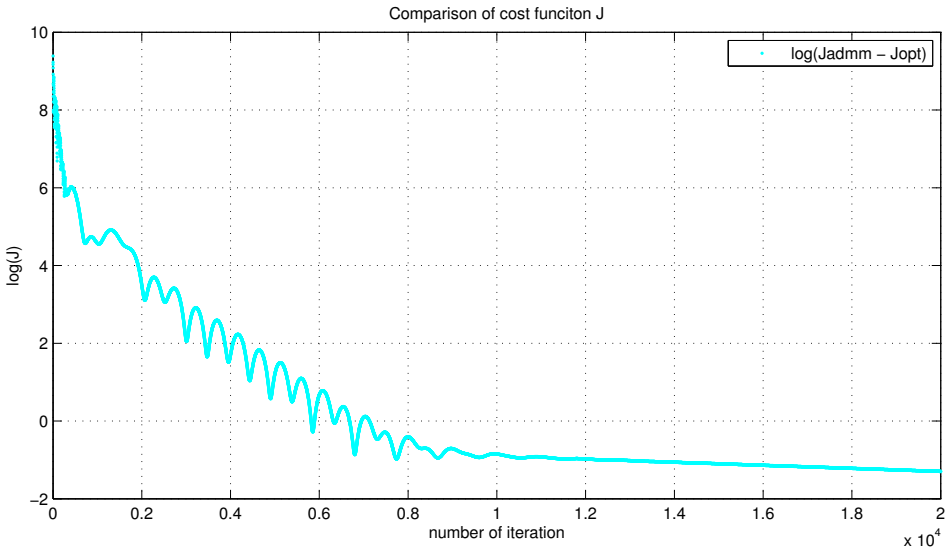


Figure 5.22: Comparison of cost function computed in Admm and Optimal estimated state after $2 \cdot 10^4$ iterations. ($\sigma_V = 10^{-2} PCC_{VoltageAmplitude}[Volt]$; $\sigma_I = 10^{-2} i_{max}$; $\sigma_\theta = \sigma_\phi = 10^{-2}[rad]$; $\sigma_{sync} = 0[rad]$)

5.4.3 Computational Effort

The scalable Admm algorithm is mainly characterized by:

- the capacity to converge to the optimal central solution after an adequate number of iteration;
- appreciable performance with a low number of iteration as well;
- convergence in presence of non trivial synchronization noise;
- being completely distributed and scalable, is robust to failures and to an increase of the grid topology.

However, it has to be underlined that the running time required is much greater than both of the central and the distributed solution proposed. Table 5.10 sum up it for different number of iteration. Note how it can influence the usability of the algorithm.

# iteration	10^3	10^4	$2 \cdot 10^4$	10^5
time[s]	9.296	$1.074 \cdot 10^2$	$2.168 \cdot 10^2$	$9.483 \cdot 10^2$

Table 5.10: Scalable Admm algorithm running time (s) for different number of iteration values.

5.5 Comparison between Distributed Estimator and Admm (scalable) Estimator

Once described the performances of both the algorithm proposed with respect to the central optimal estimation, it is useful to compare both solutions between each other.
To carry out these tests, the following values for the noise standard deviation are taken:

- $\sigma_V = 10^{-2} PCC_{VoltageAmplitue}[Volt]$;
- $\sigma_I = 10^{-2} i_{max}[Ampere]$;
- $\sigma_\theta = \sigma_\phi = 10^{-2}[rad]$;
- $\sigma_{sync} = 0[rad]$

No synchronization noise is assumed since the distributed estimator works only in presence of trivial values.
In particular is make a comparison for two different values of iteration for the Admm algorithm, which represents the bottleneck to its usability.

5.5.1 Comparison for 10^3 iterations

After 10^3 iterations the admm estimator is still not in convergence but its performance is appreciable with respect to the measurements. Figure 5.23 shows the distance of the two local estimator from the central optimal solution. Note that the algorithms are still comparable even if the admm is already lightly more accurate.

In addition to this, figure 5.24 shows the performance of the all the algorithm (the central estimator too) in comparison with the measurements value. Note how, with respect to the measures, all the algorithms works well.

Table 5.11 and table 5.12 sum up, respectively, the maximum and minimum distance from the real state and from the central optimal solution of the different quatities. In particular from table 5.12 is clear how both solutions are still comparable.

5.5.2 Comparison for 10^4 iterations

After 10^4 iterations the performance can only improve. The main difference between the first case shown is that the Admm is now almost in convergence how it can be seen in figure 5.25.

The figure shows how the Admm estimated state is much closer to the optimal central one with respect to the Distributed estimated state.

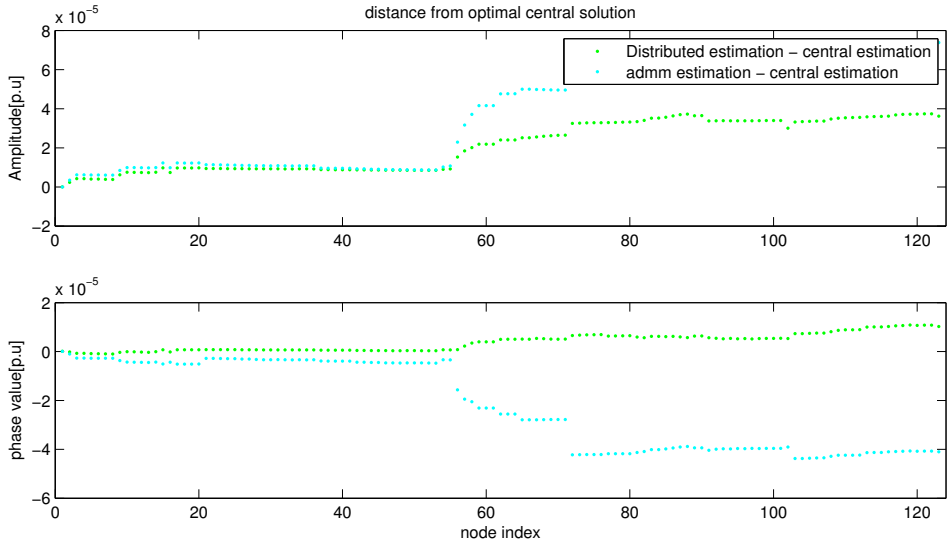


Figure 5.23: Distance of Distributed and Admm estimation from Optimal value after 10^3 iterations. ($\sigma_V = 10^{-2} PCC_{VoltageAmplitude}[Volt]$; $\sigma_I = 10^{-2} i_{max}$; $\sigma_\theta = \sigma_\phi = 10^{-2}[rad]$; $\sigma_{sync} = 0[rad]$)

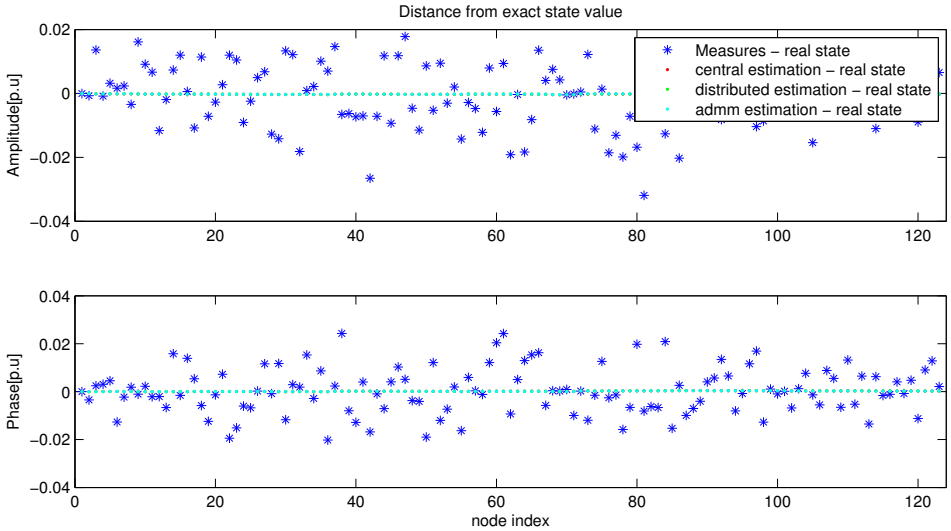


Figure 5.24: Distance of Measures, Central, Distributed and Admm estimators from exact state value after 10^3 iterations. ($\sigma_V = 10^{-2} PCC_{VoltageAmplitude}[Volt]$; $\sigma_I = 10^{-2} i_{max}$; $\sigma_\theta = \sigma_\phi = 10^{-2}[rad]$; $\sigma_{sync} = 0[rad]$)

	Measures	Central	Distributed	Admm
min Amplitude distance	$4.567 \cdot 10^{-5}$	$7.477 \cdot 10^{-5}$	$7.244 \cdot 10^{-5}$	$7.127 \cdot 10^{-5}$
Max Amplitude distance	$3.191 \cdot 10^{-2}$	$4.427 \cdot 10^{-4}$	$4.162 \cdot 10^{-4}$	$3.931 \cdot 10^{-4}$
min Phase distance	$1.588 \cdot 10^{-4}$	$8.697 \cdot 10^{-7}$	$2.568 \cdot 10^{-7}$	$3.611 \cdot 10^{-6}$
Max Phase distance	$2.430 \cdot 10^{-2}$	$4.468 \cdot 10^{-4}$	$4.532 \cdot 10^{-4}$	$4.074 \cdot 10^{-4}$

Table 5.11: Distance (p.u.) from real state value after 10^3 iterations.

	Distributed	Admm
min Amplitude distance	$2.325 \cdot 10^{-4}$	$3.493 \cdot 10^{-6}$
Max Amplitude distance	$3.740 \cdot 10^{-5}$	$7.422 \cdot 10^{-5}$
min Phase distance	$4.873 \cdot 10^{-8}$	$9.925 \cdot 10^{-7}$
Max Phase distance	$1.086 \cdot 10^{-5}$	$4.374 \cdot 10^{-5}$

Table 5.12: Distance (p.u.) from optimal central state after 10^3 iterations.

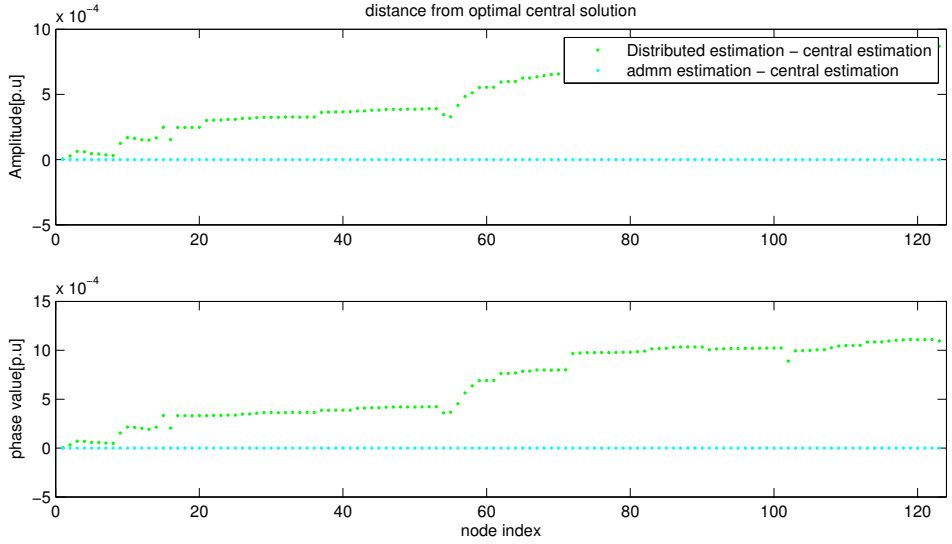


Figure 5.25: Distance of Distributed and Admm estimation from Optimal value after 10^4 iterations. ($\sigma_V = 10^{-2} PCC_{VoltageAmplitude}[Volt]$; $\sigma_I = 10^{-2} i_{max}$; $\sigma_\theta = \sigma_\phi = 10^{-2}[rad]$; $\sigma_{sync} = 0[rad]$)

Again table 5.13 and table 5.14 report the maximum and minimum distance of the quantities from the real state and from the optimal estimated state respectively.
Note from table 5.14, that the admm algorithm is almost in convergence.

	Measures	Central	Distributed	Admm
min Amplitude distance	$1.613 \cdot 10^{-5}$	$1.129 \cdot 10^{-6}$	$2.296 \cdot 10^{-6}$	$1.129 \cdot 10^{-6}$
Max Amplitude distance	$3.482 \cdot 10^{-2}$	$1.220 \cdot 10^{-4}$	$8.162 \cdot 10^{-4}$	$1.219 \cdot 10^{-4}$
min Phase distance	$1.580 \cdot 10^{-4}$	$3.303 \cdot 10^{-5}$	$6.608 \cdot 10^{-5}$	$3.303 \cdot 10^{-5}$
Max Phase distance	$3.022 \cdot 10^{-2}$	$5.562 \cdot 10^{-4}$	$1.574 \cdot 10^{-3}$	$5.563 \cdot 10^{-4}$

Table 5.13: Distance (p.u.) from real state value after 10^4 iterations.

	Distributed	Admm
min Amplitude distance	$2.792 \cdot 10^{-5}$	$7.459 \cdot 10^{-11}$
Max Amplitude distance	$8.924 \cdot 10^{-4}$	$1.724 \cdot 10^{-8}$
min Phase distance	$3.305 \cdot 10^{-5}$	$1.863 \cdot 10^{-11}$
Max Phase distance	$1.111 \cdot 10^{-3}$	$1.568 \cdot 10^{-8}$

Table 5.14: Distance (p.u.) from optimal central state after 10^4 iterations.

5.6 Power Losses Reduction via Reactive Power Control

In this section we will show the importance of estimation in a specific control algorithm. Specifically, the algorithm deals with reactive power control to reduce power losses through the network. Our purpose is to show how the estimation improves the algorithm efficiency with respect to the use of row data.

Let us introduce the following setup (see [14]): consider a smart low voltage microgrid in which a certain number nodes are equipped by microgenerators. The microgenerators are equipped with some sort of intelligence and they are divided into overlapping groups, namely clusters. In each cluster, one microgenerator is considered the header, that is, it can communicate and control the remaining microgenerators in the cluster and can communicate with other cluster headers as well. Figure 5.26 shows a schematic view of the setup.

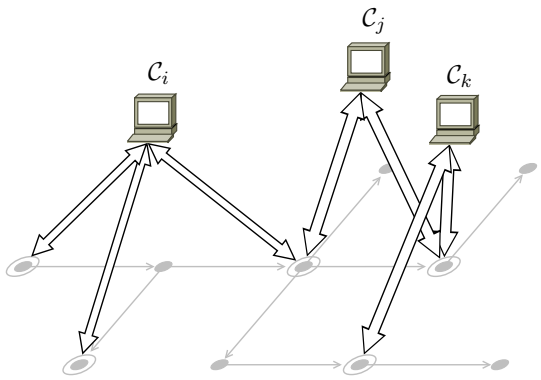


Figure 5.26: Schematic with of the microgrid setup. The circled nodes represent the microgenerators. The computer represent the cluster headers the overstand a certain number of microgenerators: they can communicate between each other.

The algorithm exploited to control the microgenerators is assumed to be a black box algorithm. It receives in input the state of the network and computes, as output, the amount of reactive power to correctly drive the microgenerators. Note that, to its purpose, any kind of network state can be considered, both measured or estimated. We want to show that the use of filtered data, instead of simple raw data, considerably improves its performance.

Let us consider the following default values for the noise standard deviation:

- $\sigma_V = 10^{-4} PCC_{VoltageAmplitude}[Volt]$;

- $\sigma_I = 10^{-4} i_{max}[Ampere]$;
- $\sigma_\theta = \sigma_\phi = 10^{-4}[rad]$;
- $\sigma_{sync} = 10^{-4}[rad]$

These values of standard deviation are considerably smaller with respect to those considered in the previous section tests. This is due to the fact that the measurements corresponding to these values are already bad enough to abruptly corrupt the algorithm functioning.

In addition to this two more assumption have been taken into account:

1. because the algorithm is more affected by synchronization errors, the following tests carry out an analysis only above different synchronization noise standard deviation values;

2. since the previous section had shown the comparable performance of the Distributed Estimator and the Admm Estimator[4], the control algorithm will be tested only using the Admm Estimator.

5.6.1 Performance for default values of standard deviation

The analysis shows that the measurements corresponding to default standard deviation values are still good enough to let the control algorithm work. Indeed figure 5.27 shows[5] that the performances of the algorithm using noisy measurements, filter data (estimated state) or the real state do not considerably differ.

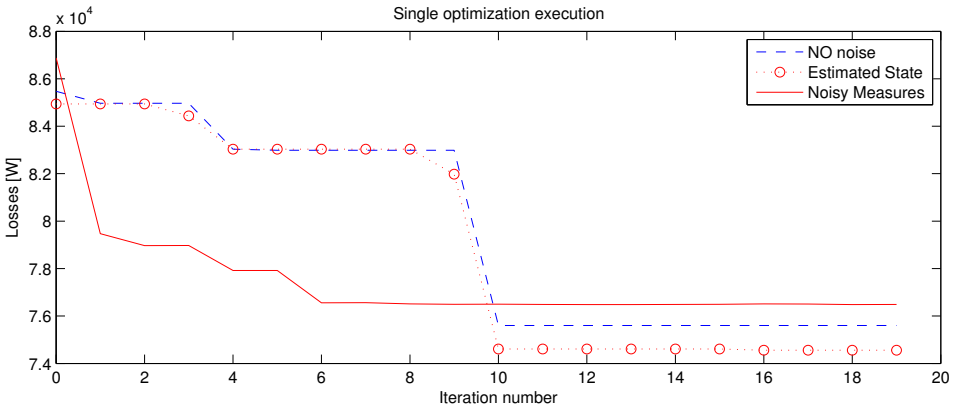


Figure 5.27: Performance of Reactive Power Control Algorithm for default values of standard deviation error.

[4]The performance corresponding to 10^3 iterations.

[5]The image reports the Average trend of the optimization algorithm over 500 runs.

Table 5.15 reports the losses before and after the algorithm optimization corresponding to the different set of data used.

	Real State	Estimated State	Measurements
Losses before optimization[KW]	86.878	86.878	86.878
Losses after optimization[KW]	74.560	74.559	76.486
Losses reduction after opt. [%]	14.17	14.18	11.96

Table 5.15: Losses before and after optimization for different data set.

5.6.2 Performance for $\sigma_{sync} = 10^{-3}[rad]$

The performances of the algorithm highlights the improvement in using filtered data instead of raw data. Figure 5.28 shows that if the algorithm runs over the raw data it can still achieve some sort of optimization but the amount of losses is appreciably higher.
Table 5.16 shows that the losses reduction due to optimization over the noisy measurements is still approximately of 14%. Meanwhile the amount of losses is $2KW$ greater.

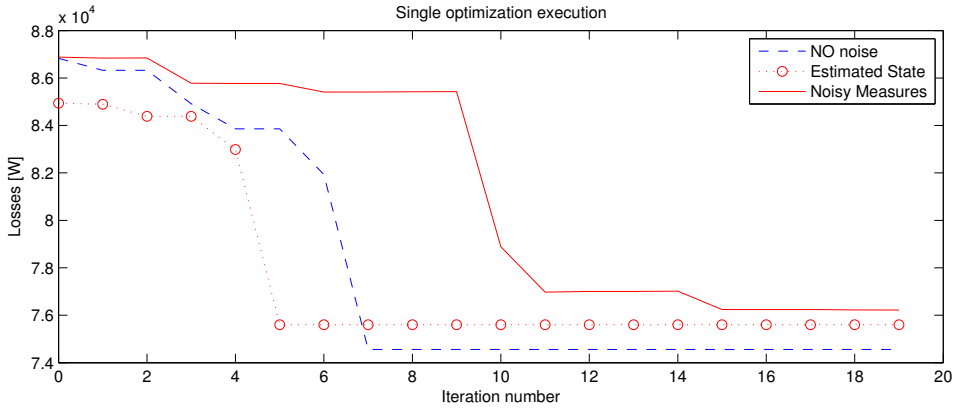


Figure 5.28: Performance of Reactive Power Control Algorithm for $\sigma_{sync} = 10^{-3}[rad]$.

	Real State	Estimated State	Measurements
Losses before optimization[KW]	86.878	86.878	86.878
Losses after optimization[KW]	74.560	75.603	76.221
Losses reduction after opt. [%]	14.18	12.97	12.26

Table 5.16: Losses before and after optimization for different data set.

5.6.3 Performance for $\sigma_{sync} = 10^{-2}[rad]$

The performances of the algorithm, increasing the synchronization noise, worsen with respect to the raw data but do not with respect to the filtered data as can be seen in figure 5.29.

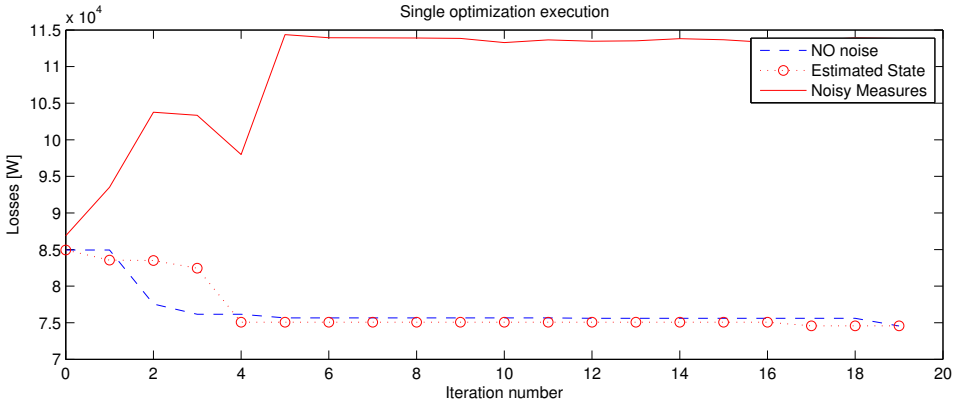


Figure 5.29: Performance of Reactive Power Control Algorithm for $\sigma_{sync} = 10^{-2}[rad]$.

This time the losses reductions drastically change their values as reported in table 5.17.
Note that not only the losses corresponding to the noisy measurements increase but the reduction decrease as well.

	Real State	Estimated State	Measurements
Losses before optimization[KW]	86.878	86.878	86.878
Losses after optimization[KW]	74.560	74.578	113.831
Losses reduction after opt. [%]	14.18	14.15	-31.02

Table 5.17: Losses before and after optimization for different data set.

5.6.4 Performance in using the Estimated State for greater value of standard deviation values

Finally is reported the algorithm performance using the estimated state for:

- $\sigma_V = 10^{-2} PCC_{VoltageAmplitude}[Volt]$;
- $\sigma_I = 10^{-2} i_{max}[Ampere]$;
- $\sigma_\theta = \sigma_\phi = 10^{-2}[rad]$;
- $\sigma_{sync} = 10^{-2}[rad]$

This values, corresponding to those seen in the previous test section, simulate the presence of relevant measurements error. Figure 5.30 shows the performances.

Figure 5.30: Performance of Reactive Power Control Algorithm for $\sigma_V = 10^{-2} PCC_{VoltageAmplitude}$, $\sigma_I = 10^{-2} i_{max}$, $\sigma_\theta = \sigma_\phi = \sigma_{sync} = 10^{-2}[rad]$.

Table 5.18 shows the losses and the losses reduction for the measurements as well. Note that the algorithm is completely unusable with raw data set. On the contrary the performance with the filtered data are still desirable.

	Real State	Estimated State	Measurements
Losses before optimization[KW]	86.878	86.878	86.878
Losses after optimization[KW]	74.560	74.565	259.551
Losses reduction after opt. [%]	14.18	14.17	-198

Table 5.18: Losses before and after optimization for different data set.

Chapter 6

Conclusion

We deal with the problem of estimation in low voltage power networks. This is well known to be the starting point for controlling a network since raw measurements are too inaccurate to work with. Specifically, we proposed two separate distributed and scalable solutions. Both of them are based on a suitable linearized model, of the measurements errors at every node. PMUs only measurements of voltage and current at every node of the grid are exploited.

After providing the maximum likelihood solution to the estimation problem we exploit it as element of comparison for the algorithms' solutions. We note that both the algorithms lead to appreciable performance improving greatly the knowledge of the grid state. However, the two solutions differ from each other substantially since:

- the first one:

  Pros:

  1. requires a very low computational effort and time to converge;
  2. is completely scalable and distributed;

  Cons:

  1. is based on a Jacobi like approximate procedure. This leads only to an approximate solution of the estimation problem that does not converge to the maximum likelihood solution;
  2. is able to manage wide errors ranges but do not deal with synchronization noise. This forces the use of at least a basic GPS synchronization unit.

- the second one:

  Pros:

  1. is based on a ADMM procedure that converges to the maximum likelihood solution;

2. deals with synchronization noise as well, requiring to be equipped with only simple PMUs;

Cons:

1. to reach the convergence needs considerable number of iterations and computational effort.

We provide a wide set of simulations, carried out on the `Ieee 37 and 123 nodes test feeder` ([15], [16]), to ensure the behavior and performances of the algorithms over a remarkable range of different noise standard deviations values. The results point out the algorithms limits as well.

In addition to this, a specific power network control algorithm has been tested to underline the importance of estimation. Specifically, a power losses reduction through reactive power compensation algorithm, [14], has been chosen as prototype. The simulations show how using the estimated state with respect to the raw measurements improve much more the performances. In particular, the estimated state leads to performances comparable with those obtained using the exact state value. The measurements on the contrary leads to an undesirable behavior.

Appendix A

Proofs about ADMM

A.1 Proof of equation (4.14)

Starting from (4.13), that is step 3. of the updating law, it can be written

$$\begin{aligned} z_{ij}(t+1) &= \underset{z_{ij}}{argmin}\ \mathcal{L}(X(t), z, \lambda(t), \mu(t)) \\ &= \underset{z_{ij}}{argmin}\ \Big\{-(\lambda_{ij}^T(t) + \mu_{ji}^T(t))z_{ij} + \frac{c}{2}\big[||X^{(i)}(t) - z_{ij}||^2 + ||X^{(j)}(t) - z_{ij}||^2\big]\Big\} \end{aligned}$$

considering the minimize only the i^{th} and j^{th} indexes and so, discarding what does not depend on them.

Through the first order optimality condition we have that

$$-\lambda_{ij}(t) - \mu_{ji}(t) - c\big(X^{(i)}(t) - z_{ij}\big) - c\big(X^{(j)}(t) - z_{ij}\big) = 0$$

and so

$$z_{ij}(t+1) = \frac{1}{2c}\Big(\lambda_{ij}(t) + \mu_{ji}(t) + c\big(X^{(i)}(t) + X^{(j)}(t)\big)\Big) \tag{A.1}$$

Note from (A.1) and (4.11) that, by a simple inductive proof, if

$$\lambda_{ij}(-1) = -\mu_{ji}(-1) \Longrightarrow \lambda_{ij}(t) = -\mu_{ji}(t)\ \forall t$$

This lets to discard one of the Lagrangian multiplier introduced.

In addition to this substituting (A.1) in (4.11) we lead, for λ_{ij}, to

$$\lambda_{ij}(t) = \lambda_{ij}(t-1) + \frac{c}{2}\big(X^{(i)}(t) - X^{(j)}(t)\big)$$

that is just equation (4.14). In this way it is possible to eliminate both z_{ij} and μ_{ij}, concluding the proof.

A.2 Proof of equation (4.16)

To write equation (4.16) starting from equation (4.15) is necessary to explicitly solve the first order optimality condition.

Rewriting (4.15), we get

$$\begin{aligned} X^{(i)}(t+1) &= \underset{X^{(i)}}{argmin}\Big\{ \sum_{i=1}^{N}\big(y_i(t)-A^{(i)}X^{(i)}\big)^T R_i^{-1}\big(y_i(t)-A^{(i)}X^{(i)}\big) + \\ &\quad + \sum_{i=1}^{N}\sum_{j\in\mathcal{N}_i} \lambda_{ij}^T(t)\big(X^{(i)}-z_{ij}(t)\big) + \mu_{ij}^T(t)\big(X^{(i)}-z_{ji}(t)\big) + \\ &\quad + \frac{c}{2}\sum_{i=1}^{N}\sum_{j\in\mathcal{N}_i} ||X^{(i)}-z_{ij}(t)||^2 + ||X^{(i)}-z_{ji}(t)||^2\Big\} \end{aligned}$$

Considering only the terms depending on $X^{(i)}$ (the other ones being constant are irrelevant to minimization so, can be discarded) we get

$$\begin{aligned} X^{(i)}(t+1) &= \underset{X^{(i)}}{argmin}\Big\{ \big(y_i(t)-A^{(i)}X^{(i)}\big)^T R_i^{-1}\big(y_i(t)-A^{(i)}X^{(i)}\big) + \\ &\quad + \sum_{j\in\mathcal{N}_i} \lambda_{ij}^T(t)\big(X^{(i)}-z_{ij}(t)\big) + \mu_{ij}^T(t)\big(X^{(i)}-z_{ji}(t)\big) + \\ &\quad + \frac{c}{2}\sum_{j\in\mathcal{N}_i} ||X^{(i)}-z_{ij}(t)||^2 + ||X^{(i)}-z_{ji}(t)||^2\Big\} \end{aligned}$$

The first order optimality condition is then equal to

$$\begin{aligned} 0 &= -2A^{(i)T}R_i^{-1}\big(y_i(t)-A^{(i)}X^{(i)}\big) + \sum_{j\in\mathcal{N}_i}\big(\lambda_{ij}(t)-\lambda_{ji}(t)\big) + \\ &\quad + 2c\sum_{j\in\mathcal{N}_i}\big(X^{(i)}-\frac{c}{2}\big(X^{(i)}(t)-X^{(j)}(t)\big)\big) \end{aligned}$$

Solving with respect to $X^{(i)}$ is easy to get exactly equation (4.16).

N.B. Note in the previous equations that quantities y, λ and X depending on time instants are considered as fixed values and when deriving, only quantities independent on time are to be considered as variables.

A.3 Proof of equation (4.19)

The augmented Lagrangian associated to problem (4.18) is equal to

$$\begin{aligned}\mathcal{L}(X,z,\lambda,\mu) &= \sum_i f_i\big(X^{(i)}\big) + \sum_i \sum_{j\in\mathcal{N}_i} \lambda_{ij}^T P_{ij}\big(X^{(i)} - z_{ij}\big) + \mu_{ij}^T P_{ij}\big(X^{(i)} - z_{ji} + \\ &\quad \frac{c}{2}\sum_i \sum_{j\in\mathcal{N}_i} (X^{(i)} - z_{ij})P_{ij}(X^{(i)} - z_{ij}) + (X^{(i)} - z_{ji})P_{ij}(X^{(i)} - z_{ji})\end{aligned}$$

Thanks to the proof of appendix A.1 is known that $\lambda_{ij}(t) = -\mu_{ji}(t)$ and, initializing $\lambda_{ij}(-1) = \lambda_{ji}(-1) = 0$ is easy to see that

$$\lambda_{ij} = -\lambda ji$$

In addition to this is known to be $z_{ij} = z_{ji}$ and so is possible to rewrite the Lagrangian as

$$\mathcal{L}(X,z,\lambda) = \sum_i f_i\big(X^{(i)}\big) + \sum_i \sum_{j\in\mathcal{N}_i} 2\lambda_{ij}^T P_{ij}\big(X^{(i)} - z_{ij}\big) + c(X^{(i)} - z_{ij})P_{ij}(X^{(i)} - z_{ij})$$

The ADMM procedure provides the sequential updates

$$\begin{aligned}X^{(i)}(t+1) &= \underset{X^{(i)}}{argmin}\ \mathcal{L}(X^{(i)}, z(t), \lambda(t)) \qquad (A.2)\\ z_{ij}(t+1) &= \underset{z_{ij}:\ P_{ij}z_{ij} = P_{ij}z_{ji}}{argmin}\ \mathcal{L}(X^{(i)}(t+1), z, \lambda(t)) \\ \lambda_{ij}(t+1) &= \lambda_{ij}(t) + c\big(X^{(i)}(t+1) - z_{ij}(t+1)\big)\end{aligned}$$

In a similar way of appendix A.1 is possible to get, for z_{ij},

$$P_{ij}z_{ij}(t+1) = P_{ij}\frac{X^{(i)}(t+1) + X^{(j)}(t+1)}{2} \qquad (A.3)$$

Exploiting (A.3) in the first of (A.2) and eliminating the constant term, unused in the first order optimality condition, we get

$$X^{(i)}(t+1) = \underset{X^{(i)}}{argmin}\ f_i(X^{(i)}) + \sum_{j\in\mathcal{N}_i} X^{(i)T} P_{ij}\big(2\lambda_{ij}(t) + cX^{(i)} - 2cz_{ij}(t)\big) \qquad (A.4)$$

Finally, defining

$$\begin{aligned}M_i &= c\sum_{j\in\mathcal{N}_i} P_{ij} \qquad (A.5)\\ \Lambda_i(t) &= -2\sum_{j\in\mathcal{N}_i} P_{ij}\lambda_{ij}(t) \\ U_i(t) &= 2c\sum_{j\in\mathcal{N}_i} P_{ij}z_{ij}(t) - 2M_iX^{(i)}(t)\end{aligned}$$

is easy, after substituting equation (A.3) in the third of (A.5) and the third of (A.2) in the expression of $\Lambda_i(t)$ in (A.5), to get equations (4.19)$\div$(4.23)

Bibliography

[1] A. Abur and A. G. Exposito, *Power System State Estimation, theory and implementation*. Marcel Dekker, 2004.

[2] F. C. Schweppe and D. B. Rom, "Power system static-state estimation, part i: Exact model," *Transaction on Power Apparatus and Systems*, vol. Pas-89, no. 1, pp. 120–125, January 1970.

[3] ——, "Power system static-state estimation, part ii: Approxiamte model," *Transaction on Power Apparatus and Systems*, vol. Pas-89, no. 1, pp. 125 – 130, January 1970.

[4] ——, "Power system static-state estimation, part iii: Implementation," *Transaction on Power Apparatus and Systems*, vol. Pas-89, no. 1, pp. 130 – 135, January 1970.

[5] G. Valverde and V. Terzja, "Pmu-based multi-area state estimation with low data exchange," *Innovative Smart Grid Technologies Conference Europe (ISGT Europe), 2010 IEEE PES*, pp. 1–7, October 2010.

[6] L. Zhao and A. Abur, "Multiarea state estimation using synchronized phasor measurements," *Ieee Transaction on Power System*, vol. 20, no. 2, pp. 611–617, May 2005.

[7] W. Jiang, V. Vittal, and G. T. Heydt, "A distributed state estimator utilizing synchronized phasor measurements," *Ieee Transaction on Power System*, vol. 22, no. 2, pp. 563–571, May 2007.

[8] C. Rakpenthai, S. Premrudeepreechacharn, S. Uatrongjit, and N. R. R. Watson, "Measurement placement for power system state estimation using decomposition technique," *Electric Power System Research*, vol. 75, pp. 41–49, 2005.

[9] G. N. Korres, "A distributed multiarea state estimation," *Ieee Transaction on Power System*, vol. 26, no. 1, pp. 73–84, February 2011.

[10] A. J. Conejo, S. de la Torre, and M. Canas, "An optimization approach to multiarea state estimation," *Ieee Transaction on Power System*, vol. 22, no. 1, pp. 213–221, February 2007.

[11] G. Mateos, I. D. Schizas, and G. B. Giannaakis, "Performance analysis of the consensus-based distributed lms algorithm," *EURASIP Journal on Advances in Signal Processing*, October 2009.

[12] B. H. Kim and R. Baldick, "A comparison of distributed optimal power flow algorithm," *Ieee Transaction on Power System*, vol. 15, no. 2, pp. 599–604, May 2000.

[13] T. Erseghe, "A distributed and scalable processing method based upon admm," *Ieee Signal Processing Letters*, vol. 19, pp. 563–566, 2012.

[14] S. Bolognani and S. Zampieri, "A distributed control strategy for reactive power compensation in smart microgrids," *arXiv*, vol. arXiv:1106.5626v2 [math.OC], Oct. 2011. [Online]. Available: http://arxiv.org/abs/1106.5626

[15] Radial distribution test feeders. [Online]. Available: http://ewh.ieee.org/soc/pes/dsacom/testfeeders/index.html

[16] W. H. Kersting, "Radial distribution test feeders," *Transaction on Power Systems*, vol. 6, no. 3, pp. 975–985, August 1991.

[17] F. Nogales, F. Prieto, and A. J. Conejo, "A decomposition methodology applied to the multi-area optimal power flow problem," *Annals od Operations Research*, vol. 120, no. 1, pp. 99–116, April 2003.

[18] A. J. Conejo, F. Nogales, and F. Prieto, "A decomposition procedure based on approximate newton directions," *Mathematical Programming*, pp. 495–515, April 2002.

[19] D. Bertsekas and J. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*. Old Tappan, (NJ) USA; Prentice Hall Inc., January 1989.

[20] B. H. Kim and R. Baldick, "Coarse-grained distributed optimal power flow," *Ieee Transaction on Power System*, vol. 12, no. 2, pp. 932–939, May 1997.