



UNIVERSITÀ DEGLI STUDI DI PADOVA

Dipartimento di Ingegneria Industriale DII

Corso di Laurea in Ingegneria dell'Energia Elettrica

Tesi di laurea magistrale

## **Storage management and real-time simulation of microgrids using a realistic testbed**

Simulazione in tempo reale e validazione sperimentale di gestione dello storage in una  
microgrid

### **Relatori**

Prof. Roberto Turri

Prof.ssa Zita Vale

Nicola Tomasone  
matricola: 1079418

Anno Accademico 2015-2016

Copyright © Nicola Tomasone, 2016

The present work, with the exception of where sources are reported, is released with Attribution-NonCommercial-ShareAlike 4.0 International Creative Commons license which can be viewed at <https://creativecommons.org/licenses/by-nc-sa/4.0/>.

# Summary

## English

The objective of the present work is the realization of a centralized secondary control system for an islanded microgrid with focus on the storage, RES and demand management. The control system, implemented to be used in a general islanded microgrid, was adapted to the existing laboratory of Renewable Sources of the research group Grupo de Investigação em Engenharia do Conhecimento e Apoio a Decisão (GECAD) and it was tested and validated on a typical scenario to evaluate its performances.

The thesis work had three main parts: a theoretical background study, a software implementation and the validation tests preceded by physical preparation of the test-bed. The manuscript is mainly focused on the first aspect. The second part is present in the attached code while the third part is only partially covered in the description of the test-bed and in the analysis of test results.

The thesis starts with a general presentation of the principal aspects of a microgrid system: first, a definition of microgrid and distributed generation, second, the reasons for which the microgrid should be a key element in the improvement of the existing power systems and a new reference model for the future ones. In the third Section the general structure of a microgrid is examined.

In the second Chapter, the attention is focused on the secondary control of a microgrid, mainly the Energy Management System (EMS), implemented in the following parts of the work. The structure of the control system is discussed by giving a global overview of all its components and their connections.

Since it is the core of the system, the optimization component is the first element of the Energy Management System (EMS) to be presented. The optimization problem is formulated using a linear programming approach. The inputs of the optimization component are the storage status, the forecasted production of generators, the forecasted load demand and the load status, while the output is the optimal scheduling of the controllable resources which are the storage systems and the load demand management. The algorithm has two main goals: to minimize the generation costs and to allow the respect of the technical constraints of the electrical system.

In the following parts, the inputs of the optimization component are analyzed.

The first input is the storage: in this part it is developed a model for a electrochemical storage system which allows the evaluation of the State Of Charge of storage. The model, non-linear because of the non linearity of the physical processes involved, is chosen after a brief presentation of the most common ones present in literature.

The second input is the forecasted production of the renewable generators. The models of a small wind generator and a photovoltaic generator are developed and coupled with forecast methods

for wind speed and solar irradiance because the wind and solar energy sources are aleatory.

Concluded the theoretical formulation of the EMS, the Grupo de Investigação em Engenharia do Conhecimento e Apoio a Decisão (GECAD) laboratory is presented. This Section describes all the components of the system including the wiring scheme, the implementation of the communication system between the measurement instruments and the data logging database, and the control system of wind generator and controllable loads.

The last part of the work shows the tests of the EMS and the discussion of their results.

Concerning the developed software, the theoretical formulation of the control system was implemented using the Python language and some of the free libraries available for that language. The software program was written using a modular concept. Thus, while some parts of the code are strictly related to the GECAD test-bed utilization, the main part is independent of the real system; moreover the modular approach simplifies a possible future expansion of the present work.

## Italiano

Obiettivo del presente lavoro di tesi è la realizzazione di un sistema di controllo secondario per una micro-rete con particolare attenzione al sistema di accumulo dell'energia, alle fonti rinnovabili e alla gestione dei carichi. Il sistema di controllo, sviluppato per essere utilizzato in una generica micro-rete, è adattato al laboratorio di Risorse Rinnovabili del gruppo di ricerca Grupo de Investigação em Engenharia do Conhecimento e Apoio a Decisão (GECAD) al fine di valutarne e validarne le prestazioni in uno scenario tipico del funzionamento della micro-rete.

Il lavoro di tesi presenta tre componenti principali: un lavoro teorico, una implementazione di un programma tramite scrittura di codice, la preparazione fisica del laboratorio e il suo successivo utilizzo per la verifica sperimentale del programma realizzato. L'elaborato scritto presenta principalmente la prima componente, la seconda è visionabile nel codice sorgente allegato, mentre la terza è, per ovvie ragioni, visibile solo in parte nelle descrizioni del laboratorio e del lavoro di preparazione così come nell'analisi dei risultati dell'esperimento.

Il lavoro inizia con una presentazione generale dei principali aspetti di una micro-rete: la definizione di micro-rete e di generazione distribuita, le ragioni per cui il concetto di micro-rete dovrebbe essere un elemento chiave nel miglioramento dell'attuale sistema elettrico e un modello di riferimento per quelli futuri. In un secondo momento vengono osservati i problemi collegati con la stabilità dei sistemi di potenza e il controllo di una micro-rete.

Successivamente all'introduzione, è trattato con maggior dettaglio il controllo secondario di micro-rete, nello specifico lo Energy Management System (EMS), implementato nelle parti seguenti del lavoro. Inoltre sono presentate la struttura del sistema di controllo e il perché della sua necessità, dando una visione globale dei suoi componenti e di come questi siano tra loro connessi.

Essendo il cuore del sistema, la componente di ottimizzazione è il primo elemento del EMS ad essere presentato. Il problema di ottimizzazione è formulato usando un approccio di programmazione lineare. Questo, avendo come dati di ingresso lo stato del sistema di accumulo d'energia, le previsioni di produzione dei generatori, la domanda prevista e la domanda attuale del carico, fornisce la pianificazione ottimale delle risorse controllabili che sono il sistema di accumulo e la gestione dei carichi controllabili. L'algoritmo ha due obiettivi principali: minimizzare il costo di generazione e rispettare i vincoli tecnici del sistema elettrico.

Nei successivi capitoli sono analizzati i dati di ingresso dell'algoritmo di ottimizzazione e ne sono mostrate le rispettive implementazioni.

Il primo dato di ingresso è lo stato del sistema di accumulo: in questa parte è sviluppato il modello di un sistema di accumulo elettrochimico che permetta la valutazione dello stato di carica dell'accumulo. Il modello, non lineare a causa della non linearità dei processi fisici coinvolti, è scelto dopo una breve presentazione dei più comuni modelli presenti in letteratura.

Il secondo dato di ingresso è la previsione di produzione dei generatori da fonti rinnovabili, per tale motivo è esposto lo sviluppo sia dei modelli di un generatore mini-eolico ed un generatore fotovoltaico, sia di un sistema di previsione della velocità del vento e dell'irradianza essendo queste ultime due fonti energetiche non programmabili e con carattere aleatorio.

Conclusa la formulazione teorica del EMS vi è una descrizione del laboratorio di Risorse Rinnovabili del GECAD. Il capitolo include una descrizione di tutti i componenti del laboratorio comprensiva di uno schema elettrico, una descrizione del sistema di comunicazione tra gli strumenti di

misura e la base dati che funge da archivio delle misure, una descrizione del sistema di controllo del generatore mini-eolico e dei carichi controllabili.

L'ultima parte consiste nella descrizione della prova e verifica del EMS e nella discussione dei risultati.

In merito alla parte del lavoro di tesi inerente alla programmazione, il sistema di controllo è sviluppato tramite il linguaggio di programmazione Python ed alcune delle sue librerie libere. Il programma è scritto utilizzando una struttura modulare. In questo modo, se alcune porzioni di codice sono strettamente legate all'utilizzo del laboratorio del GECAD, la parte principale di codice è indipendente dal laboratorio; in aggiunta un approccio modulare semplifica le possibili future espansioni di questo lavoro.

# Contents

<b>Summary</b>	<b>III</b>
<b>Glossary</b>	<b>XI</b>
<b>Acronyms</b>	<b>XIII</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Microgrid definition . . . . .	1
1.2 Reasons for microgrids development . . . . .	2
1.3 Structure of a microgrid . . . . .	2
1.4 Objective of the thesis . . . . .	3
<b>2 Energy Management System</b>	<b>5</b>
2.1 Typology selection . . . . .	5
2.2 EMS structure . . . . .	7
2.3 EMS algorithm . . . . .	8
<b>3 Unit Commitment</b>	<b>11</b>
3.1 Source constrains . . . . .	12
3.2 Load constrains . . . . .	12
3.2.1 Models of load based on the Demand Side Management . . . . .	14
3.2.2 Power components of Demande Side Management (DSM) . . . . .	14
3.3 Storage constrains . . . . .	16
3.4 Power balance . . . . .	17
3.5 Software implementation . . . . .	17
<b>4 Battery Model</b>	<b>19</b>
4.1 Introduction . . . . .	19
4.2 Basic chemistry of a closed secondary battery . . . . .	19
4.3 Model review . . . . .	21
4.3.1 Electrochemical . . . . .	22
4.3.2 Mathematical . . . . .	22
4.3.3 Equivalent Electrical Circuit . . . . .	22
4.4 Chosen Model Description . . . . .	23
4.4.1 Equivalent electrical circuit . . . . .	24

4.4.2	SOC counter . . . . .	26
4.4.3	Multi-cell model . . . . .	27
4.5	Algorithms . . . . .	28
4.5.1	Cell parameters estimation algorithm . . . . .	28
4.5.2	SOC estimation algorithm . . . . .	29
<b>5</b>	<b>Forecasting system</b>	<b>31</b>
5.1	Photovoltaic generator . . . . .	31
5.1.1	Basic photovoltaic concepts . . . . .	32
5.1.2	Irradiance estimation . . . . .	32
	DISC Model . . . . .	34
	Global Module Irradiance composition . . . . .	36
5.1.3	Model of the photovoltaic module . . . . .	38
	PVGIS Model . . . . .	39
	Thermal component of the model . . . . .	41
5.1.4	Complete system model . . . . .	42
5.2	Wind generator . . . . .	42
5.2.1	Wind Energy Conversion System (WECS) power output . . . . .	43
5.2.2	WECS model . . . . .	45
5.3	Forecasting methods summary . . . . .	47
5.3.1	Statistical linear methods . . . . .	47
5.3.2	Statistical non-linear methods . . . . .	48
5.4	Forecasting system implemented . . . . .	48
5.4.1	Photovoltaic . . . . .	49
5.4.2	Wind . . . . .	50
5.4.3	Load . . . . .	50
<b>6</b>	<b>Laboratory of Renewable Energy</b>	<b>53</b>
6.1	Laboratory description . . . . .	53
6.1.1	Power layer . . . . .	54
	Photovoltaic generation . . . . .	54
	Wind generation . . . . .	56
	DC buses and storage system . . . . .	58
	Loads inverters . . . . .	59
	Loads . . . . .	61
6.1.2	Data layer . . . . .	61
6.1.3	Control layer . . . . .	63
6.2	Laboratory set-up . . . . .	64
6.3	Developed Software . . . . .	66
<b>7</b>	<b>Experiment test</b>	<b>71</b>
7.1	Experiment set up . . . . .	71
7.2	Weather data . . . . .	72
7.3	Electrical data . . . . .	74
7.4	EMS output . . . . .	77

7.4.1	Consecutive constant loads . . . . .	77
7.4.2	Abrupt variation of load . . . . .	77
7.4.3	Soc estimator . . . . .	78
7.4.4	Variation of the forecasted wind generation . . . . .	78
<b>8</b>	<b>Conclusions</b>	<b>83</b>
	<b>Bibliography</b>	<b>85</b>
	<b>Annexes</b>	<b>93</b>



# Glossary

$AM$  Air Mass. 34, 35

$B$  Direct Module Irradiance. 32, 36, 37

$B_{0,h}$  Extraterrestrial Horizontal Irradiance. 35

$B_0$  Extraterrestrial Irradiance. 34–36

$B_h$  Direct Horizontal Irradiance. 34, 35, 37

$B_n$  Direct Normal Irradiance. 34–37

$C$  Charge Capacity in A h. 26–28

$D$  Diffuse Module Irradiance. 32, 37, 38

$D_h$  Diffuse Horizontal Irradiance. 36–38

$G$  Global Module Irradiance. 32, 34, 39–42

$G_h$  Global Horizontal Irradiance. 8, 34–38, 49, 50, 72, 73, 75

$G_{stc}$  Global Module Irradiance at Standard Condition. 39–41

$K_{n,clear}$  Direct Transmittance for Clear Sky. 35

$K_n$  Direct Transmittance. 34–36

$K_t$  Global Horizontal Transmittance. 35, 36

$R$  Reflected Module Irradiance. 32, 38

$T_{amb}$  Ambient Temperature. 39, 41

$T_{mod}$  Module Temperature. 40–42

$U_{oc}$  Open Circuit Voltage. 20, 22, 24–28

$\alpha$  Surface Azimuth. 37

$\beta$  Tilt Angle. 37, 38

$\delta$  Time Interval in seconds. 9, 11, 12, 14–17

$\psi_S$  Sun Azimuth. 37

$\theta_{ZS}$  Sun Zenith. 35–37

$p_{atm}$  Atmospheric Pressure. 35, 36

# Acronyms

**$U_{oc}$**  Open Circuit Voltage. 20, 23

**AC** Alternate Current. 42, 54

**ANN** Artificial Neural Network. 48

**ARMA** Auto-Regressive Moving Average. 47

**CBC** COIN Branch and Cut Solver. 18

**CHP** Combined Heat and Power. 2

**CPE** Constant Phase Element. 23

**DC** Direct Current. 31, 39, 42, 50, 65, 66, 71, 73, 75, 76, 78

**DG** Distributed Generation. 1

**DISC** Direct Insolation Simulation Code. 34, 36

**DSM** Demande Side Management. vii, 2, 8, 11, 14, 16

**ED** Economic Dispatch. 11

**EIS** Electrochemical Impedance Spectroscopy. 23

**EMS** Energy Management System. iii–vi, 3, 5–10, 15, 31, 50, 71–74, 77–79, 83, 84

**GECAD** Grupo de Investigação em Engenharia do Conhecimento e Apoio a Decisão. iii–vi, 6, 49, 50, 53, 63–66, 71, 72

**GUI** Graphical User Interface. 65, 66

**ISEP** Instituto Superior de Engenharia do Porto. 49, 53, 74

**LP** Linear Programming. 17, 18

**MAS** Multi Agent System. 5

**MILP** Mixed Integer Linear Programming. 7, 8, 17, 18

**MPPT** Maximum Power Point Tracker. 42

**NRFM** New Reference Forecast Model. 50, 51

**NWP** Numerical Weather Prediction. 47, 48

**OPF** Optimal Power Flow. 12

**PLC** Programmable Logic Controller. 61, 64, 65

**PVGIS** Photovoltaic Geographical Information System. 39–41

**RES** Renewable Energy Source. 2, 3, 6–8, 53, 72, 75, 76

**RMS** Root Mean Square. 50, 51

**SOC** State Of Charge. 5, 7, 8, 17, 23–29, 65, 78, 83

**SOH** State Of Helth. 22

**STC** Standard Test Condition. 20, 39, 40, 54

**UC** Unit Commitment. 7, 8, 11, 12, 17

**UTC** Coordinated Universal Time. 74, 77–82

**VPN** Virtual Private Network. 63

**VPP** Virtual Power Plant. 1

**WECS** Wind Energy Conversion System. viii, 2, 42–46, 83

# Chapter 1

## Introduction

### 1.1 Microgrid definition

A strict and general accepted definition of microgrid in power systems filed is not available because it is a relative new and still evolving concept so that it is more often defined by means of the changes it brings to existing system rather than what it is.

A quite general definition can be done highlighting three principal characteristics commonly shared by all its descriptions so that a microgrid can be defines as a energy system in which:

- given a small local geographical area,
- a set of electrical and thermal demands in the area is met by
- a set of small Distributed Generation (DG) energy sources located within the same geographical bounds through processes controlled by
- an completely automated elaborating system and
- the energy system can operates both isolated or connected to system of other area.

The above definition could be used also for a nationwide energetic system except that the geographical area is "small and local", the sources are "small" and the processes are completely automated, in addition the definition excludes the Virtual Power Plant (VPP) concept because the energy exchanges must be within the same area. The definition inherits the weakness of a definition of Distributed Generation (DG) where the size of geographical size and sources can have a wide range (see [2]) but distinguishes itself from DG because of the automated control [33].

Beside the difficulty of finding a standard definition the main concept of a microgrid is that the distance between the energy demand and the energy supply should be minimized in order to reduce the losses given by the transmission of energy and maximize the utilization of the local resources. In this vision the best design for a microgrid is a system in which a trigeneration plant (heat producer, cooler producer and electric producer) supplies the local demand [14].

The microgrid is also considered the basic element of the new conception of electrical distribution system, the Smart Grid, in which local sources and loads aggregated in a microgrid interact with other sources and loads in the distribution system as a single unit thanks to the coordination achieved with communication and control system.

## **1.2 Reasons for microgrids development**

The principal reasons normally given to promote the development of the microgrid are principally the reduction of the transmission losses because of the better use of local energy sources, especially Renewable Energy Source (RES), and heat of the power unit which otherwise will be wasted [15].

The advantages of a distributed system based on the microgrid concept over a centralized generation system are still not well outline for geographical area with a high density of population as shown in [38], differently for area with a low density or island the substitution of the expensive system based on oil thermoelectric generator with microgrid system with integration of RES and Demande Side Management (DSM) can lead to a best economical solution [59, 91, 92]. The microgrid grid concept is certainly the best solution if used to build a power system in region with low density of population in developing country given its low initial investment cost if compared to a central generation; in this last case the benefit of a community based electrification are also relevant [5, 41].

## **1.3 Structure of a microgrid**

The principals elements forming a typical microgrid are: the loads, the generators, the storage systems, the infrastructure of communication and the control systems.

The loads of a microgrid have an active part in the interaction with the rest of the system in fact they present a DSM element that, based on the load priority, curtails or shift in time the load demand. The curtailment of selected loads during an peck of power demand or shifting of loads demand in time instant with cheaper generation can in part brings the same benefits of a storage system for the power balance and the best economic scheduling of the system.

In the current implemented microgrid pilot test, the generators are mainly Combined Heat and Power (CHP) usually micro turbines capable of produce both electrical and thermal power that can be used giving a priority to the thermal load or the electrical load using both conventional (natural gas) or non-conventional (bio gas) sources, classic diesel generators which only feed an electrical demand and RES generator as the photovoltaic system, the WECS and the small or micro hydro.

The storage systems in microgrid have the fundamental role of balancing the load demand and generation because the limited inertia power of the small thermal generators (even absences in the RES generators) cannot be used in power balancing as with the generators of big power plants int the traditional main grid. The storage systems are dominated by the electrochemical technologies with lead-acid as first option given its industrial maturity and lower price. Other considered option are the other electrochemical technologies of lithium-ion and sodium-sulfur or the mechanical technology of flywheels [84].

The communication infrastructure is a key component of the microgrid (more in general of the smart grid) because allows the control units to monitor and manage the elements of the system. Both wireless and wire technologies use in the industrial control industries are used as the ZigBee or the RS485 protocol.

The control system has three components: a metering interface with the elements of the system to monitoring their current state and how they interact, a processing element to elaborate the information of the metering interface, an actuator interface that changes the state of the microgrid

elements based on the results of the processing unit. The control system is divided into hierarchical levels:

1. Primary control, usually located in the element of which is controller, it bases its actions only on local measurements to keep the frequency and voltage output in the imposed values and eliminate the mismatch between generators and loads active and reactive power,
2. Secondary control (Energy Management System), can be centralized (isolated microgrids with critical loads and fixed structures) or distributed (grid connected and highly variable microgrids) and it finds the optimal Local Commitment for each node that achieves the desired objectives, usually the reliable, secure and economical operation of microgrid, eliminate the steady state deviations of voltages and frequency after the transitory that cannot be handled by primary control

## 1.4 Objective of the thesis

In the following chapters it is developed a secondary control for an isolated microgrid that can be characteristic of a domestic power system in remote areas with only Renewable Energy Source generators (wind and photovoltaic). The developed EMS do not consider the thermal aspect of the microgrid.

The work starts with a general description of EMS in chapter 2 and in which its components and the used logic are presented. After the general description of the system each component is presented: the storage system in chapter 4, the generator and loads forecasting in chapter 5. In chapter 6 the used test-bed is presented while in the chapter 7 there are the results of the simulation. Finally in chapter 8, the possible expansions of the present work are pointed out.



## Chapter 2

# Energy Management System

An Energy Management System (EMS) is an element of the power system that, based on the current state of the system and the forecasted values of generation and consumption, finds the resources scheduling that optimize the overall use of the system minimizing the operational costs. The EMS is a form of secondary control, its duty is only to give dispatch signals to a single element or multiple elements, that is, to assign a power references to the them for each time interval of the ahead scheduling. The power references are obtained through signals of start up or shutdown for generators, shifting and curtailment for the loads, charging and discharging scheduling for storage systems. The control and protection in real-time of the power system (mainly voltage and current control) is demanded to the primary control system [44].

Two main different typologies of EMS can be found in literature: the decentralized and the centralized typologies. Decentralized EMS are based on the use of Multi Agent System (MAS) concepts for which every element in the power system is an entity that responds to the inputs coming from the surrounding environment [30, 89], in this approach the secondary and primary controls are located in the same element so that the system is flexible to the insertion or removal of an element. Differently in the centralized EMS, a single dedicated element is charged to elaborate all the environment signals and to broadcast the resulting optimizing response to each element in the system; the advantages of the centralized approach are the possibility of use simpler optimization algorithm if compared to the Multi Agent System (MAS) approach and a better response for system in which a strong cooperation between elements is mandatory [57].

### 2.1 Typology selection

In an isolated microgrid power system three main problems must be taken in consideration to select the EMS typology:

1. in an islanded system, a correct scheduling of the resources is fundamental to ensure the instant power balance,
2. the presence of an electrochemical storage system imposes the resolution of a highly non-linear problem in order to obtain a correct estimation of the State Of Charge (SOC) of the

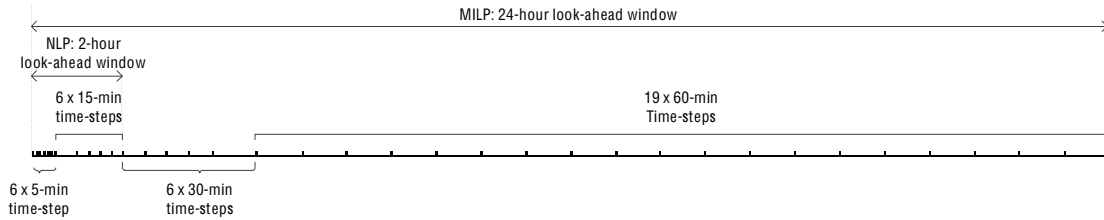
storage <sup>1</sup>,

3. the presence of the RES generator (namely wind and photovoltaic) with their non-programmable and intermittent production, introduces an high uncertainty in the input values of an optimization problem,
4. the limited number of elements in the system (both generators and loads), reduces the aggregation and smoothing effects typical of the distribution systems, thus the uncertainties of RES production and loads are increased.

The first point implies that the elements required a strong coordination and the potential and current capability of the resources must be correctly quantified, in particular the storage system resource can not be approximated without serious implication, that implies also that the second point is mandatory. A non-linear optimization problem must be solve.

The third and forth points imply that a robust prediction method must be used to overcome the uncertainty but it is intrinsic in any forecasting model to decrease the output accuracy when the forecasting time window is increased, in addition simpler model are able to predict accurately short-time forecast.

Given the aforementioned reasons the EMS was chosen with the following characteristic: centralized approach seems to be preferable given both the small size of the reference microgrid with a limited number of elements (so that the flexibility of the system is not compromised by the central approach) and the necessity of a strong coordination between them. The problems of aleatory RES generation and high variability of the loads are reduced using a rolling horizon strategy. The rolling horizon strategy differs from the standard methodology because instead of forecasting only once the optimal scheduling for all the future time intervals, it optimizes the scheduling for all the future time intervals but transmits only the next time step scheduling then updates the inputs for the new state of the system and the new forecasted values and reruns the optimization problem [49, 61].



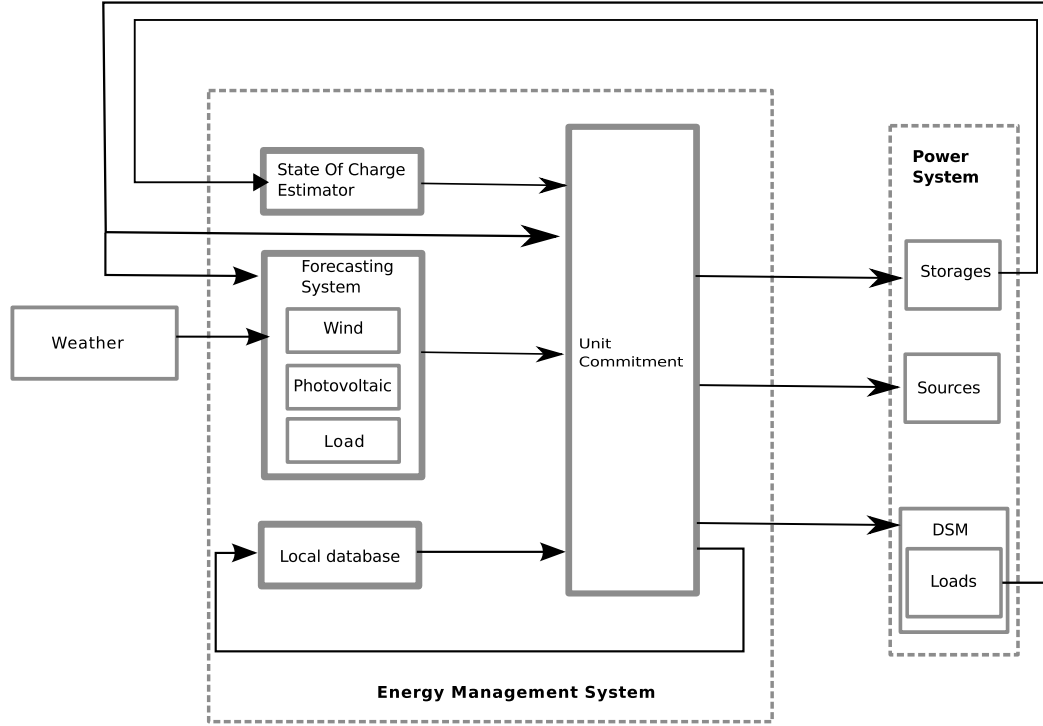
**Figure 2.1:** Forecasting window of the rolling horizon method with variable time resolution: the immediately following time intervals are shorter than the more distant intervals. Source [58]

In order to reduce the computational cost of the rolling horizon strategy the method used in [58] is implemented: the time length of the future time intervals is not constant, instead the immediately following time intervals are shorter than the time intervals more in distant in the future, see figure 2.1 as example. The variable time step size method can achieve a sensible reduction of total

<sup>1</sup> For the GECAD laboratory F microgrid it is also important to know the voltage at the storage clumps because they are the DC bus of the system on which all the converters base the operation state.

number of time interval for which the optimum scheduling must be found, to that the computational time of each optimization is reduce but at the same time the optimization is not reduced because the relaxation is done in time intervals where the forecasting uncertainty is higher.

## 2.2 EMS structure



**Figure 2.2:** Diagram of the implemented EMS: the input of the optimization process are on the left, the Unit Commitment (UC) unit is in the middle and on the right there is the power elements of system. The arrow indicate the signals directions.

In figure 2.2 it is shown the scheme of the implemented EMS and its interactions (arrows) with the power system layer of the microgrid and the atmospheric weather. The EMS is composed by four principal elements: the Unit Commitment element, the estimator of the current storage system State Of Charge (SOC), the forecasting element and the local database.

The UC element is the core of the EMS, the elaborating unit that has to compute the optimal scheduling given the forecasted values as inputs; it is implemented as a solver of a Mixed Integer Linear Programming (MILP) as illustrated in chapter 3.

The task of SOC estimator is to find the current energy level in the storage system. The correct knowledge of the remaining stored energy is fundamental for an stand-alone power system with high RES penetration because it acts like time buffer between generation and load demand. In

chapter 4 a SOC estimator for an electrochemical storage system based on lead-acid technology is implemented with a methodology that allows to use only a limited number of electrical quantities: the last voltage and tension values at storage clamps.

The third element, the forecasting unit, is required because the used RES wind power and solar power are not programmable and, especially wind power, highly variable. This unit needs as inputs both the weather quantities (historical and current) of Global Horizontal Irradiance, atmospheric pressure, air temperature, wind speed and current state of the loads, mainly the curtailment given by the DSM element and the power; it is explained in chapter 5.

The local database is an internal element which is interfaced only with the UC unit, because it is used by the latter to overcome some problems that may arise during the optimization process as it is explain in section 2.3. A second use of the database not functional to the EMS is it to analyze and debug the EMS both on-line and off-line.

The output generated by the EMS and send to the power layer of the microgrid are the activation statuses and committed powers for the storage and sources while for the loads they are the curtailment status, the shift status and the committed load power.

## 2.3 EMS algorithm

In figure 2.3 the flowchart of the implemented EMS for each optimization run is shown.

At the beginning of each run the current time is compared with the last time instant of the scheduling window which obtain summing the initial time with the total length of the forecasting window; if the current time is greater than the computed final time the process stops, otherwise the EMS starts retrieving the inputs data.

A set number of storage current and voltage values are retrieved and used in the SOC estimator. The weather data of the time interval ending with the current time and starting with the time instant which in the past a delta time equal to the forecasting window, are fetched and passed to the two model of wind generator and photovoltaic generator in order to forecast the maximum usable power of the two sources. Thirdly the status of the loads is asked to the DSM: if a load is curtailed, its maximum forecasted power is set to zero while, if the load is connected to the system, its current power is fetched from the database and pass to the load forecast system that then sets the maximum forecasted load power.

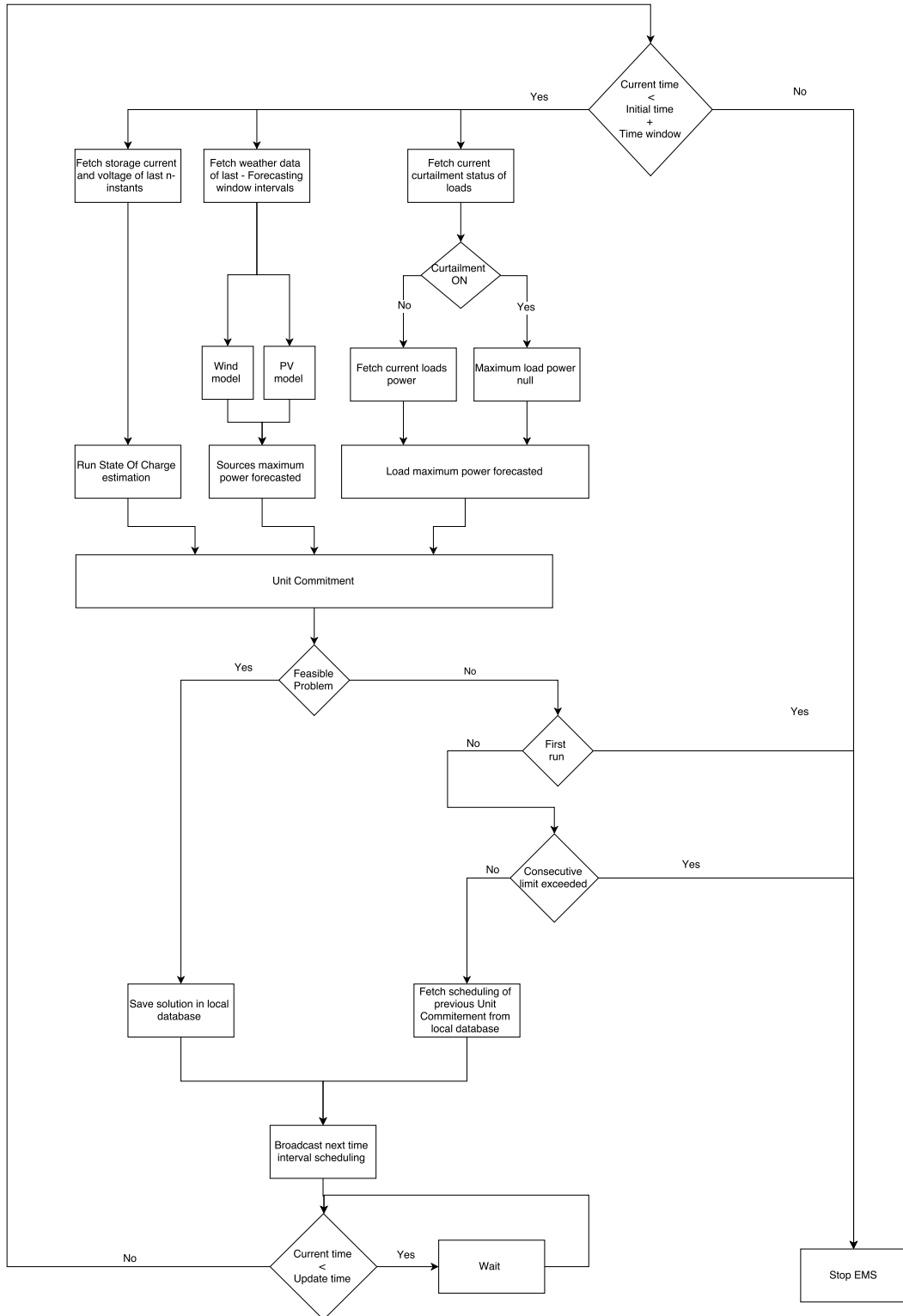
When the maximum power of all the elements in the microgrid are defined, the Unit Commitment is started in order to find the optimum scheduling for all the time intervals of the forecasting window as explained in chapter 3. The possible result of the UC for every source, storage and load element is saved in the local database. After storing the result, the EMS send only the optimal scheduling for the first time interval of the forecasting window to primary controllers of the microgrid. A Mixed Integer Linear Programming (MILP) as the developed UC problem has always a solution but only if it is well-formulated [25], if not it can be infeasible with no solution and even if well-formulated the resolution time can be longer than the update time, thus the EMS must be able to handle this case. The latter case is managed with a time limit in the Mixed Integer Linear Programming (MILP) resolver so that a too long time resolution is stopped and considered as infeasible problem. In case of an infeasible problem, it is not possible to obtain an update optimal scheduling but a sub-optimal solution is given by the solution found in the previous run for the time

interval following the already broadcast one as shown in equations 2.1.

$$\begin{aligned}
 opt_1 &= [\delta_0, \delta_1, \delta_2, \dots, \delta_{T-1}] \\
 opt_{1,signal} &= \delta_0 \\
 opt_2 &= [x, x, \dots, x] \\
 opt_{2,signal} &= \delta_1
 \end{aligned} \tag{2.1}$$

This method basically switch the time window from a rolling to a static one, but if used for an elevated number of consecutive runs, it leads to an increase of the uncertainty errors even bigger than the classic static time window because of the growing variable step size, thus a limit must be imposed. The scheme used for an infeasible problem fails if the problem is the first of the EMS runs, if this case the precautionary principle suggests to stop the EMS to find the reason of the failure rather than skip the first run and try again in the following run.

The EMS finally computes the total time used to execute the previous steps and compare it to a set updated time: if total time is bigger, it immediately starts again the optimization problem while, if is smaller, it waits the time required to reach the update time and then starts a new optimization process.



**Figure 2.3:** Flowchart of the implemented EMS.

## Chapter 3

# Unit Commitment

The Unit Commitment (UC) is defined as an optimization problem in which the resources of the power system (generators, storage and DSM) are allocated so that the total load demand is satisfied with the lowest cost; it is an extension of the Economic Dispatch (ED) where time constrain of the resources are considered [25]. The UC system has to send the activation and deactivation signals to the resources of the power system. The objective function of the optimization problem to be minimized is usually a function of costs; costs that are both of investment and operational. In the proposed implementation of the UC, based on the models described in [11, 53, 54, 73, 74], the function cost is a linear function of the power consumption and the Load Management options, that is it considers only the operational costs, so that can be defined as in equation 3.1,

$$f = \sum_{t=0}^{T-1} \gamma \delta \left( \sum_{s=0}^{S-1} p_{s,t} c_{s,t} + \sum_{l=0}^{L-1} (p_{l,t}^c c_{l,t}^c + C_{l,t}^s) + \sum_{b=0}^{B-1} p_{b,t}^{dsc} c_{b,t}^{dsc} + p_{b,t}^{ch} c_{b,t}^{ch} \right) \quad (3.1)$$

where

- $T$  is the total number of time interval  $\delta$  (s) that form the time window of scheduling,
- $S$ ,  $L$  and  $B$  are the total number of sources, loads and storage systems electrically interconnected (the corresponding lowercase indicates the nth element),
- $p_{s,t}$  is the power output of the nth source,  $p_{l,t}^c$  is the saved power due to curtailment of the nth load,  $p_{l,t}^{s,out}$  is the power shifted from one time interval to an other time interval for the same nth load,  $p_{b,t}^{dsc}$  and  $p_{b,t}^{ch}$  are the discharging and charging power of the nth storage system,
- $c_{s,t}$ ,  $c_{l,t}^c$ ,  $c_{l,t}^s$ ,  $c_{b,t}^{dsc}$ ,  $c_{b,t}^{ch}$  are the cost per energy of the related power (€/Wh). The coefficients are input values of the optimization problem which are dependent of the element ( $s, l, b$ ) and the time interval ( $t$ ), the latter relation allows the possible use of an model for prices forecast; otherwise they can be used as order of priority coefficients where the element with highest priority has the lowest coefficient.
- $C_{l,t}^s$  is the cost function for the load shifting defined in subsection 3.2.2,
- $\gamma = \frac{1}{3600s}$  is a coefficient that is necessary to make the cost (having the hour as time unit) and the energy ( $\delta p_t$  having the seconds as time unit) comparable.

### 3.1 Source constrains

The source of the system can be of different typologies: programmable as micro-turbine or a combined heat and power generator, non-programmable and intermittent as wind and photovoltaic generator or even a connection to a distribution system. The model in the following section doesn't consider the specificity of the generator, it imposes only constrains on minimum and maximum power delivered by the generator and an on-off control.

Equation 3.2 express the minimum  $p_{s,t}^{min}$  and maximum  $p_{s,t}^{max}$  constrains at each time interval  $t \in [0, 1, \dots, T-1]$  for all the generators of the system  $s \in [0, 1, \dots, S-1]$  and they are annulled by the binary source status variable  $\sigma_{s,t} \in \mathbb{B} = [0, 1] \subset \mathbb{Z}$  when it is set to zero by the linear problem resolver.

$$\sigma_{s,t} p_{s,t}^{min} \leq p_{s,t} \leq \sigma_{s,t} p_{s,t}^{max} \quad (3.2)$$

As formulated in equation 3.2, it is assumed that the generic generator has a response to a shutdown or switch on command as fast as the minimum time interval  $\delta$ . If the rump time of generator is required, the models suggested in [61, 58] can be considered.

The constrain  $p_{s,t}^{max}$  is an input value of the algorithm that is set equal to the forecast engine output when the source is an non-programmable intermittent generator. The constrain  $p_{s,t}^{min}$  is set to zero by default but it can be changed if a second stage optimization (like a Optimal Power Flow (OPF)), finds that the scheduled Unit Commitment doesn't provide enough power for the time interval  $t$ .

### 3.2 Load constrains

Given the prediction window  $\mathbb{T}$  and the set of loads  $\mathbb{L}$ , the forecast engines provides a forecast load  $p_{l,t}^f$  for each time interval  $\delta$  and for each load  $l \in \mathbb{L}$ . A general load  $l$  can be shiftable, that is the energy unit  $p_{l,j}^f \delta_j$ , can be relocated from time interval  $j$  to time interval  $t$ , curtailable that is the energy unit can be disconnected by the EMS without be supplied in the other time intervals, both or nether of them.

The processes of curtail and shift which are reallocation of energy, can be implemented defining a 3-dimension table  $M$  of size  $\mathbb{L} \times \mathbb{T} \times \mathbb{T}$ <sup>1</sup> whose generic binary element  $m_{l,j,t} \in \mathbb{B} = [0, 1] \subset \mathbb{Z}$  is equal to 1 when the forecasted load energy of load  $l$  is shifted from time interval  $j$  for which it was forecasted or planned before UC to the new time interval  $t$ ; it is worth to highlight that the element shifted is the energy (Wh) and not the power (W) because the time intervals  $\delta$  have different length so that the same energy relocated into an other time interval can have a different power value given by  $p_{relocated} = p_{forecasted} \frac{\delta_{forecasted}}{\delta_{relocated}}$ .

<sup>1</sup>The principal weak point of this implementation of the Load Management is the number of variables  $m_{l,j,t}$  that the solver has to compute which are  $L \times T \times T = L \times T^2$ .

$$M_l = \begin{bmatrix} m_{l,0,0} & m_{l,0,1} & \dots & m_{l,0,t} & \dots & m_{l,0,T-1} \\ m_{l,1,0} & m_{l,1,1} & \dots & m_{l,1,t} & \dots & m_{l,1,T-1} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ m_{l,j,0} & m_{l,j,1} & \dots & m_{l,j,t} & \dots & m_{l,j,T-1} \\ \vdots & \vdots & & \vdots & \ddots & \vdots \\ m_{l,T-1,0} & m_{l,T-1,1} & \dots & m_{l,T-1,t} & \dots & m_{l,T-1,T-1} \end{bmatrix} \quad (3.3)$$

To explain the better the function of the table  $M$  let analysis a simple case (see 3.4) for a single load and a window of five time intervals:

$$M = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.4)$$

the 2-dimension table is a record of the following load allocations looking at the rows:

- the energy slot of time  $j = 0$  (first row) is curtailed which is interpretable as a shift out of the original time without reallocation into an other slot (all the elements of the first row are null),
- the load is not modified from the original forecasted value in time interval  $j = 1$  so that the unity element is on the diagonal,
- the energy slot of time  $j = 2$  is shifted forward into the time interval  $t = 4$ ,
- the energy slot of time  $j = 3$  is shifted backward into time slot  $t = 2$ ,
- the energy slot of time  $j = 4$  is not modified.

The final power used in the post optimization time scheduling  $t \in [0, 4]$  is computed summing the elements of each column so that:

- in time instant  $t = 0$  there is no load,
- in  $t = 1$  the load is the original  $j = 1$  load,
- in  $t = 2$  the load is given by the load which was forecasted for  $j = 3$ ,
- in  $t = 3$  there is no load,
- in  $t = 4$  the load is given by the sum of the original load in  $j = 4$  plus the load shifted from the pre optimization  $j = 2$ .

The table as define in 3.4 allows a not physical case shown in table 3.5

$$M = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.5)$$

where the original forecasted load in  $j = 1$  is both keep in time  $t = 1$  and shifted into time instant  $t = 3$ , but the forecasted energy load of a time interval is unique, it can be curtailed or shifted but not duplicated, thus only one element per row can be not null; this constrain is set by equation 3.6 (the constrain must be an inequality to allow the load to be curtailed as in  $j = 0$ )

$$\sum_{t=0}^T m_{l,j,t} \leq 1 \quad \forall j \in \mathbb{T}. \quad (3.6)$$

### 3.2.1 Models of load based on the Demand Side Management

Considering the DSM of a microgrid, the load can be classified into four types: completely manageable (both shifting and curtailment), only shiftable, only curtailable and not controllable.

The DSM of a completely controllable load is fully described by table 3.3 and equation 3.6.

An only shiftable load in addition to the table 3.3 and equation 3.6 requires the constrain of equation 3.7

$$\sum_{t=0}^T m_{l,j,t} = 1 \quad \forall j \in \mathbb{T} \quad (3.7)$$

that does not allow a load to be curtailed as in table 3.4 for initial time intervals  $j = 0$  and  $j = 2$  where

$$\sum_{t=0}^T m_{j,t} = 0 \quad j = 0, 2.$$

A load which is only curtailable requires table 3.3 and the constrain of equation 3.8 (which is a restriction of equation 3.6)

$$m_{l,j,t} = 0 \quad \forall t \in \mathbb{T}, \forall j \in \mathbb{T}, j \neq t \quad (3.8)$$

that allows not null elements only in the diagonal position of the table, e.g. in table 3.4 the elements  $m_{2,4}$  and  $m_{3,2}$  were not permitted.

A load which is neither shiftable nor curtailable is describe by the table 3.3 and constrain of equation 3.9

$$m_{l,j,t} = 1 \quad \forall j \in \mathbb{T}, j = t. \quad (3.9)$$

### 3.2.2 Power components of DSM

The objective function 3.1 needs knowledge of the power exchanged by loads with the rest of the power system split into the shift component and the curtail component while the equation 3.24 requires the total load power. Starting from the time table 3.3 and the forecasted (or planned) load time series  $P_l^f = (p_{l,0}^f, p_{l,1}^f, \dots, p_{l,t}^f, \dots, p_{l,T-1}^f)$  is possible to obtain all the components.

Considering the most general case of load which is both shiftable and curtailable, the load energy  $p_{l,t} \delta_t$  of load  $l$  in time interval  $t$  is defined by equation 3.10

$$p_{l,t} = \frac{1}{\delta_t} \sum_{j=0}^J m_{l,j,t} p_{l,j}^f \delta_j \quad (3.10)$$

that is, the load energy of time interval  $t$  is the sum of all the forecasted load energies which the EMS allocates in that time instant.

$$M = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} \quad (3.11)$$

As example, for the load described in table 3.11, the load power at time  $t = 3$  is equal to

$$p_{t=3} = \frac{1}{\delta_{t=3}} \left( p_{j=1}^f \delta_{j=1} + p_{j=3}^f \delta_{j=3} + p_{j=4}^f \delta_{j=4} \right).$$

The curtail load power component  $p_{l,j}^c$  is evaluated by means of equation 3.12

$$p_{l,j}^c = \left( 1 - \sum_{t=0}^T m_{l,j,t} \right) p_{l,j}^f \quad (3.12)$$

in fact the sum term in brackets is null only if the  $j - th$  row of table 3.3 has no unit element excluding the shifted and not modifiable loads, while equation 3.6 ensures that the bracket element is included in  $[0, 1] \subset \mathbb{R}$ ; moreover the  $m_{l,j,j}$  is not present in the sum thus the problem is still linear (otherwise it could became quadratic).

The shifted load power component must be split into two different components to properly count the shift cost in equation 3.1: the power shift out from a time interval  $j$  called shift-out power  $p_{l,j}^{s,out}$  and the shift-in power  $p_{l,t}^{s,in}$  shift into the time interval  $t$  from all the possible  $j$  time intervals.

The shift-out power is computed by equation 3.13

$$p_{l,j}^{s,out} = \left( (1 - m_{l,j,j}) \sum_{\substack{t=0, \\ t \neq j}}^T m_{l,j,t} \right) p_{l,j}^f \quad (3.13)$$

for which the

- $(1 - m_{l,j,j})$  is not null when the load of time interval  $\delta_j$  is moved away because of a shift or a curtailment,
- $\sum_{t=0, t \neq j}^T m_{l,j,t}$  is null only if the load is curtailed.

As an example if the table 3.4 is analyzed the following results are given:

$$p_{j=0}^{s,out} = ((1 - 0) \quad 0) p_{j=0} = 0 \quad (3.14)$$

$$p_{j=1}^{s,out} = ((1 - 1) \quad 0) p_{j=1} = 0 \quad (3.15)$$

$$p_{j=2}^{s,out} = ((1 - 0) \quad 1) p_{j=2} = p_{j=2} \quad (3.16)$$

in 3.14 the load is curtailed, in 3.15 the load has not DSM, in 3.16 the load is shifted out from the original time interval  $\delta_{j=2}$  to  $\delta_{j=4}$ .

The shift-in power is computable using equation 3.17

$$p_{l,t}^{s,in} = \frac{1}{\delta_t} \sum_{\substack{j=0, \\ j \neq t}}^T m_{l,j,t} p_{l,j}^f \delta_j \quad (3.17)$$

which is equal to equation 3.12 except for the exclusion of the index  $j = t$ , so that the eventually not moved load of time interval  $\delta_j$  is not accounted. As example, considering the table 3.11, the shift in power for each time interval  $\delta_t$  is:

$$p_{t=3} = \frac{1}{\delta_{t=3}} \left( p_{j=2}^f \delta_{j=2} + p_{j=4}^f \delta_{j=4} \right).$$

The shift cost is defined as the price of moving an energy slot from the original time interval  $j$  to the different time interval  $t$ , thus from equation 3.17 it is possible to define the shift cost function  $C_{l,t}^s$  present in the objective function 3.18. The cost function is

$$C_{l,t}^s = \frac{1}{\delta} \sum_{\substack{j=0, \\ j \neq t}}^{T-1} m_{l,j,t} p_{l,j}^f \delta_j c_{l,j}^s \quad (3.18)$$

with  $c_{l,j}^s$  cost of shift away the load  $l$  at time interval  $j$  per energy unit.

Other model of load shift which are under consideration are those proposed in [85], where the load is only shifted preserving the curve profile over the total time length, and in [46].

### 3.3 Storage constrains

Following the model of source presented in section 3.1, equation 3.20 for the charging power and equation 3.19 for the discharging power, set minimum and maximum limits of power given as input to the solver, while the binary variables  $\beta_{b,t}^{ch}, \beta_{b,t}^{dsc} \in \mathbb{B}$  are used by the solver to control the charging and discharging processes. The constrain on the binary variables of equation 3.21 forbids that in the same instant the storage system is object of a charging and discharging processes.

$$p_{b,t}^{ch,min} \beta_{b,t}^{ch} \leq p_{b,t}^{ch} \leq \beta_{b,t}^{ch} p_{b,t}^{ch,max} \quad (3.19)$$

$$p_{b,t}^{dsc,min} \beta_{b,t}^{dsc} \leq p_{b,t}^{dsc} \leq \beta_{b,t}^{dsc} p_{b,t}^{dsc,max} \quad (3.20)$$

$$\beta_{b,t}^{ch} + \beta_{b,t}^{dsc} \leq 1 \quad (3.21)$$

The energy of the storage system for each time interval is computed with the linear expression of equation 3.22

$$E_{b,t+1} = E_{b,t} + \left( \eta_b^{ch} p_{b,t}^{ch} - \frac{1}{\eta_b^{dsc}} p_{b,t}^{dsc} \right) \delta, \quad t \in [1, 2, \dots, T-1] \quad (3.22)$$

where  $\eta_b^{ch}, \eta_b^{dsc} \in [0, 1] \subset \mathbb{R}$  are the efficiency of the two processes for the storage  $b$ , while the initial energy  $E_{b,0}$  is a required input which can be provided by a direct measure or using a more complex model as is proposed in the present work where the detailed non-linear model of section 4.4 allows a state estimation of the SOC which is used as input value for the linear model in

$$E_{b,0} = SOCE_b^{max}$$

The energy stocked in the storage at each time interval has its boundaries set by equation 3.23.

$$E_b^{min} \leq E_{b,t} \leq E_b^{max}, \quad \forall b \in \mathbb{B}, \forall t \in \mathbb{T} \quad (3.23)$$

An optional constrain can be set to control the final state of the storage system at the end of the commitment window

$$E_{b,T} = E_b^{end}.$$

### 3.4 Power balance

The UC algorithm has to ensure that loads demand is satisfied by the sources in each time interval, that is the power balance between the elements of the system must be respected. In the present implementation the problem is simplified because the system elements connected between each other are assume linked through single node. The simplifying assumption ignore the power losses and physical constrains given by the electrical network that in reality connects the elements. An other simplification is given by the fact that only the active power flow is considered. With the previous assumptions the active power balance is given by equation 3.24

$$\sum_{s=0}^{S-1} p_{s,t} + \sum_{b=0}^{B-1} (p_{b,t}^{dsc} - p_{b,t}^{ch}) - \sum_{l=0}^{L-1} p_{l,t} = 0, \quad \forall t \in \mathbb{T} \quad (3.24)$$

The equation is based on the flow convention that the power entering a generic node has a positive sign (sources and storage systems during the discharge process) while the power exiting the node has a negative sign (loads and storage systems during the charging process).

### 3.5 Software implementation

The optimization problem formulated in the previous sections falls under the group of Mixed Integer Linear Programming (MILP): it is a linear problem because both the objective function and the constrains are linear combinations of variables, it is mixed integer because, in addition to continuous variables, presents binary variables which are considered as integers in the set  $b \in [0, 1] \subset \mathbb{Z}$ .

The Mixed Integer Linear Programming is resolved by means of the `unit_commitment` function of the `scheduling` module. The function has two required inputs: a time window which is a set of time intervals  $(\delta_0, \delta_1, \dots, \delta_{T-1})$  and a node object that contains all the elements (sources, storage systems and loads) of the system for which the UC is wanted. The function resolves the optimization problem using the library PuLP [80], an Linear Programming (LP) modeler written

in Python [27] that easily allows to convert the mathematical problem presented in the previous sections into a LP or, as in the case of interest into a MILP written in a low level language. The generated problem is then passed to a selected solver by PuLP and the output of the solver is reconverted in Python object. The chosen solver is the COIN Branch and Cut Solver (CBC) [26] that, as its name suggests, uses the branch-and-cut method to solve the MILP. The method [60] starts the resolution with a relaxation of the integer variables (e.g. 0, 1 if binary) which are considered continuous with a low and an upper bounds, then the obtained LP is solve; if the results for the integer variables are integers, the algorithm stops otherwise for each integer variable with real solution (e.g. 0,5) two new LP problems are set up one with the ceiling of the found real solution (e.g. 1) as lower bound and the other with the floor of the found real solution (e.g. 0) as upper bound, then the two problem are solved using the relaxed approximation. If the solutions of the two problems are not feasible or the are bigger than the actual upper bound, they are prune; if the solution is lower than the upper bound, its upper bound becomes the new global upper bound while the solution becomes the best new solution.

In **Annex Unit Commitment** it is shown the use of the `unit_commitment` function to evaluate some basic scenarios.

## Chapter 4

# Battery Model

### 4.1 Introduction

The definition of a battery is: a system composed of one or more electrochemical cell in which the chemical bound energy is directly converted into electrical power by means of oxidation-reduction (redox) reactions. Because of the generality of the definitions, a large number of different technologies can be call batteries, but a first division is between batteries in which the energy conversion can be reverse (secondary) and batteries in which the conversion is only from chemical energy to electric power (primary). A second categorization is based on the location of the reactants and of the conversion processes: the reactants can be inside the two electrodes that are both the energy storage and the conversion elements, thus these batteries are closed systems with a fixed quantity of reactants; an other option is when the reactants are stored outside the electrochemical cell (thus they are open systems) and the electrodes are just the charge-transfer media where the redox reaction occurs like the redox flow batteries [94]. In power system applications of interest for the present work, only rechargeable technologies are used, thus the following sections focus solely on secondary batteries with fixed supply of reactants. Because of its maturity and its wide use in stationary energy storage [86], the batteries based on lead are the main object of the following sections.

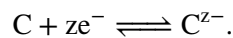
### 4.2 Basic chemistry of a closed secondary battery

The basic unit of battery, the electrochemical cell, is composed of three elements: a negative electrode and a positive electrode separated by an electrolyte. The first two elements are electron conductors with different electron affinity while the last is an ionic conductor which electronically isolates the electrodes [9].

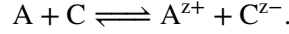
When the two electrodes are connected by an external electronic conductor, a oxidation reaction happens at the negative electrode



and the charge  $ze^{-}$  goes from the negative electrode into the positive electrode (the electrochemical cell does work) where the reduction reaction happens

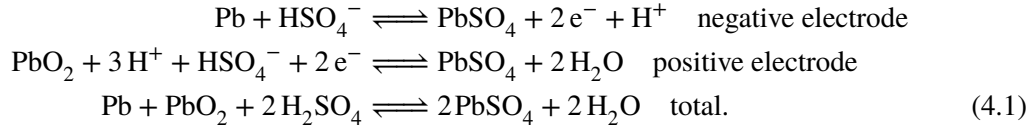


The process would stop as soon as the voltage difference between the two electrodes was null because of the concentration of negative charges into the positive electrodes and positive charges in the negative electrode, if not that inside the electrolyte the positive ions (cations)  $A^{z+}$  move from the negative electrode to the positive electrode while the negative ions (anions) move from the positive electrode to the negative electrode, so that the charges flow is closed and the process continues and the complete redox reaction is



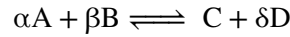
When the two electrodes are connected by an external power source that forces the electrical charge  $ze^-$  to move from the positive electrode into the negative electrode (so that the power source does work), the redox reaction is reversed.

In a lead acid cell the negative electrode is made of lead Pb, the positive electrode of lead dioxide  $PbO_2$  and the electrolyte is sulfuric acid  $H_2SO_4$ . The redox reaction for the discharge process is given in equation 4.1



The lead is converted into the low conducting lead sulfate  $PbSO_4$  at both electrodes while the electrolyte concentration of sulfuric acid into water  $H_2O$  decreases. It is worth to not that the electrolyte of the lead acid cell is not only an ionic conductor but also an active reactant, consequently it is possible to determinate the State of Charge of the electrochemical cell measuring the concentration of the sulfuric acid.

The open circuit voltage  $U_{oc}$ , which is the voltage different between the two electrodes of an electrochemical cell site of the generic redox reaction



in equilibrium state, is given by the Nernst's equation

$$U_{oc} = \Delta(U_{oc})_{stc} + \frac{RT}{nF} \log \left( \frac{a_C^\gamma a_D^\delta}{a_A^\alpha a_B^\beta} \right)$$

where  $(U_{oc})_{stc}$  is the voltage difference at Standard Test Condition <sup>1</sup>,  $R$  is the gas constant,  $T$  the temperature of the cell,  $n$  the number of electrons involved in the redox reaction and  $a_j$  the activity of the j-th element.

During the discharging or recharging process, the electrochemical cell is not in equilibrium state and the processes are not reversible, thus the Nernst's equation, which is based on thermodynamic theory, is not sufficient to describe the voltage difference between the two electrodes  $U$ . The difference between the two voltages is called polarization  $\eta_p = U - \text{OpenCircuitVoltage}(U_{oc})$  and it is due to three main kinetic

<sup>1</sup>Reversible voltage difference at standard pressure  $p_{stc} = 101\,325\, \text{kg m}^{-1} \text{s}^{-2}$  and standard temperature  $T_{stc} = 298,15\, \text{K}$

activation polarization, linked to the charge-transfer process at the electrode-electrolyte interface

$$\eta_{p,a} = \left( \frac{RT}{\alpha_{p,a} z F} \right) \log \left( \frac{i}{i_0} \right)$$

with  $i_0$  exchange current of the redox reaction and  $\alpha_{p,a}$  transfer coefficient,

ohmic polarization, linked to the internal ohmic resistivity of the cell components

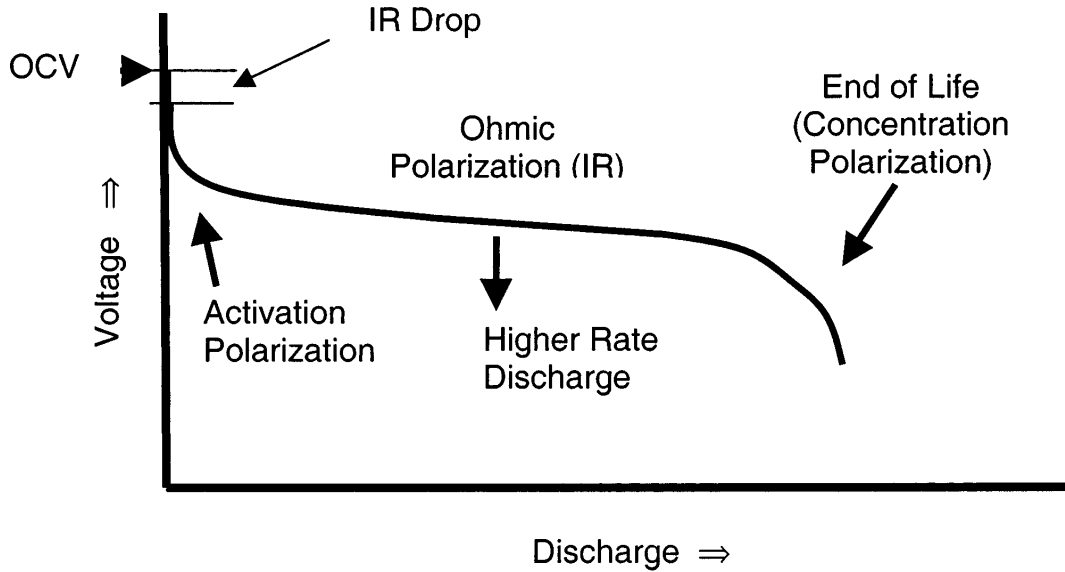
$$\eta_{p,\Omega} = R_{int} i,$$

concentration polarization, linked to the mass-transfer process in the electrolyte

$$\eta_{p,c} = \left( \frac{RT}{n} \right) \log \left( \frac{C}{C_0} \right)$$

with  $C$  and  $C_0$  reactant concentrations at electrode surface and in the bulk solution.

The polarization is given by the sum of the three phenomena, each of one has a different weight in different cell chemistry, so that the polarization  $\eta_p = \eta_{p,a} + \eta_{p,\Omega} + \eta_{p,c}$  is a non-linear phenomenon. In figure 4.1 is shown a typical discharge curve in which the results of the three polarization are highlighted [66, 94, 9].



**Figure 4.1:** Typical discharge profile of an electrochemical cell in which the three polarization components activation, ohmic and concentration are highlighted. Source [94, p. 4250]

### 4.3 Model review

The electrochemical storage systems, based on the redox reaction, have differences in the chemical components, in the secondary reaction process, in the shapes and sizes of the cells

but, as shown in section 4.2, they have common non-linear behavior during the operational use. Because of the different needs of analysis which cover from the optimization of the single cell to the integration of a storage technology into a power systems, different models were developed based on totally different methodology. The principals models can be divided into three main groups: electrochemical, mathematical, equivalent electrical circuit [13, 72].

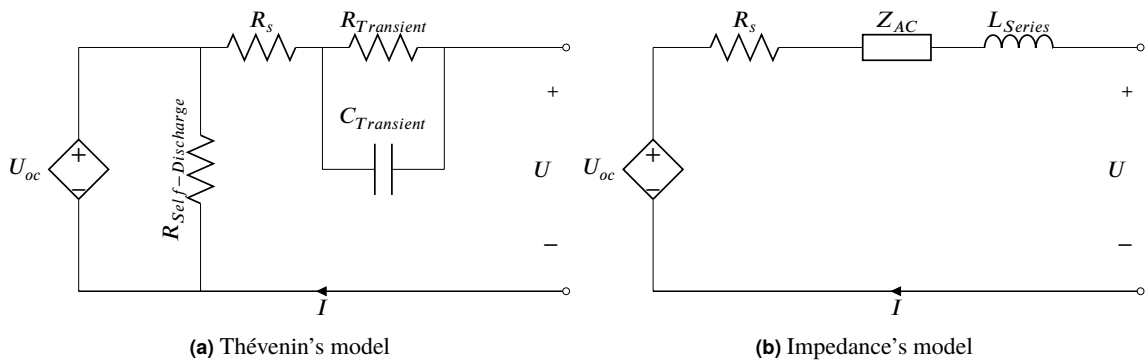
### 4.3.1 Electrochemical

The electrochemical models describe analytically the physical and chemical phenomena in the elementary cell of the storage system, linking together cell parameters as chemistry and shape of the cell plates, macroscopic quantities as voltage and current at cell clamps, microscopic quantities as charge concentration. These models require to solve a system of coupled Partial Differential Equations (PDE), with their initial and boundary conditions, which are non-linear time and space variant; thus the resolutions have a high computational cost and long time of simulation up to hours for a single discharging-charging cycle, leaving them only for design and optimization applications.

### 4.3.2 Mathematical

The mathematical models, using empirical equations (as the Peukert's law) or stochastic methods (as fuzzy logic, neural network or Kalman filters), describe only macroscopic quantities as the soc or the State Of Health (SOH). They have low computational cost but with low accuracy of the results (errors with an order of magnitude of 5 % to 20 %) without giving information of the electrical quantities as cell voltage or exchanged current. These models are usually used for real time simulations in the battery manager of electrical motor vehicles.

### 4.3.3 Equivalent Electrical Circuit



**Figure 4.2:** Equivalent Electrical Circuits. Adaptation from [13].

The electrical circuit models, neglecting the real physical and chemical inner processes, simulate the electrical behavior of a electrochemical cell by means of an equivalent electrical

circuits which have the same responses of the cell for a given input signal (usually a current signal). The complexity and computational cost of these models are lower than the electrochemical models but they have an error on simulation of the electrical quantities which of the order of 1 % to 5 %.

The values of the equivalent circuit components are linked to the single cell model under investigation and they are evaluated by experimental measures of the electrical quantities changing other physical quantities of cell as the state of charge, its temperature or the completed cycles. The data acquisition is a long process because of the time required by the cell to pass from one equilibrium state to an other one, e.g. the estimation of the relationship between the SOC and the  $U_{oc}$  demands up to more than twenty-four hours between two data records [1, p. 267]; furthermore, differently from electrochemical models, it is not possible to change the value of one single physical element of the cell without re-evaluating all the circuit component values, in fact there are not direct and unambiguous relations between the physical quantities and the circuit parameters.

The equivalent circuit models are usually divided into two main groups: the Thévenin models and the impedance models.

The Thévenin models (figure 4.2a) consist in a linear electrical circuit with generators, resistors and capacities (a RC circuit): the ideal voltage generator models the open circuit voltage, a series resistor permits to simulate the voltage difference between open and close circuit in a steady state while the transient behavior is given by one or more series impedance<sup>2</sup> made by a RC parallel. A more accurate model has to describe the dependence of cell electrical quantities from non electrical variables as SOC or temperature, thus the values of the circuit components become functions of the aforementioned quantities, functions which can be non linear.

The impedance models (figure 4.2b) substitute the RC parallel with a Constant Phase Element (CPE)  $Z_{AC}$  which has the same frequency response of the electrochemical cell. The evaluation of the constant impedance requires the use of the Electrochemical Impedance Spectroscopy (EIS), which is a frequency response of the cell to small signals, that is, alternate current signals of a few millivolts (1 mV to 10 mV) so that the voltage variation is considered a linear function of the current.

## 4.4 Chosen Model Description

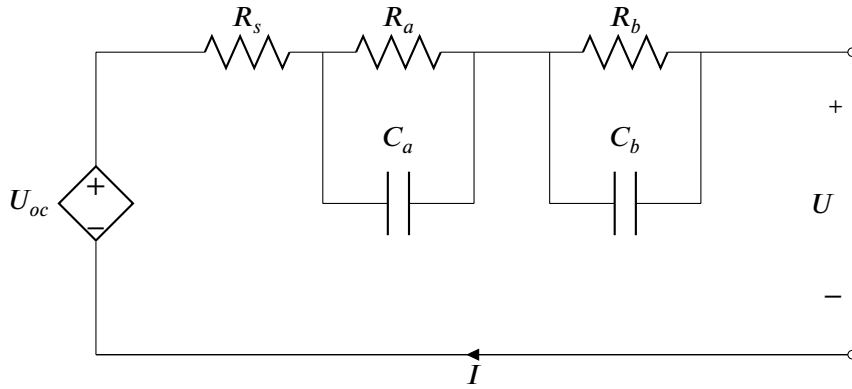
The model of the electrochemical storage system used in the EMS is chosen by the its use in the following, that is, the high frequency SOC estimation during the normal operation of the power system. Considering the final goal of the model, the following characteristics are required:

---

<sup>2</sup>The authors of [97, p. 4], minimizing the computational cost and the error of different models, found the optimum number of series impedances is two.

- a fast resolution time because the SOC is estimated in real time at the beginning of each control cycle,
- using only operational current and voltage data because the run time nature of the EMS does not allow a rest time to execute SOC evaluation dedicated tests,
- be usable in different regimes as charging and discharging processes with a width power range,
- a short-time time series of recorded data of inputs during the online estimation to do not rely on large data storage system,
- a minimum set of parameters which can be found in the cell datasheet in order to eliminate the necessity of dedicated cell tests.

Given the previous requirements, the most appropriate model is found to be an hybrid model that combines Thévenin model [13, 21], having circuit parameters function of the SOC, and a SOC counter based on the current integration. The defined model is then used in a minimization problem which finds the actual SOC minimizing the difference between the recorded voltage and the voltage of the model for the recorded current [21].



**Figure 4.3:** Thévenin component of the electrochemical cell model.

#### 4.4.1 Equivalent electrical circuit

The equivalent circuit (see figure 4.3) is composed by an ideal voltage source  $U_{oc}$ , which represents the open source voltage, a series resistors  $R_s$  and two impedances each one composed by a parallel of resistor  $R$  and capacitor  $C$ . The resistors and the capacitors are modeled, as shown in equation 4.2, with a constant part and an exponential part function of the SOC.

$$\begin{aligned}
 R_s &= \theta_1 \exp(\theta_2 SOC) + \theta_3, & R_s &\in [0, \infty) \subset \mathbb{R}, & \theta_2 &\in (-\infty, 0] \subset \mathbb{R} \\
 R_a &= \theta_4 \exp(\theta_5 SOC) + \theta_6, & R_a &\in [0, \infty) \subset \mathbb{R}, & \theta_5 &\in (-\infty, 0] \subset \mathbb{R} \\
 R_b &= \theta_7 \exp(\theta_8 SOC) + \theta_9, & R_b &\in [0, \infty) \subset \mathbb{R}, & \theta_8 &\in (-\infty, 0] \subset \mathbb{R} \\
 C_a &= \theta_{10} \exp(\theta_{11} SOC) + \theta_{12}, & C_a &\in [0, \infty) \subset \mathbb{R}, & \theta_{11} &\in (-\infty, 0] \subset \mathbb{R} \\
 C_b &= \theta_{13} \exp(\theta_{14} SOC) + \theta_{15}, & C_b &\in [0, \infty) \subset \mathbb{R}, & \theta_{14} &\in (-\infty, 0] \subset \mathbb{R}
 \end{aligned} \quad (4.2)$$

Experimental extrapolations of the above parameters show the exponential part is an inverse function of the SOC and the parameters are almost constant but not for low SOC [13], thus the constraints on the theta coefficients.

Following the generator convention for the current sign <sup>3</sup>, the voltage at the cell clamps  $U$  is given by the open circuit voltage  $U_{oc}$  minus the voltages  $U_{R_s}$ ,  $U_a$  and  $U_b$  as shown in equation 4.3

$$U(t) = U_{oc}(t) - (U_{R_s}(t) + U_a(t) + U_b(t)) \quad (4.3)$$

The open circuit voltage  $U_{oc}$  is a function of the SOC and has to be evaluated by experimental tests that measure the voltage at the electrochemical cell clamps when no load or generator is connected to the cell which is in equilibrium state; the relation can be non linear, as for the lithium ion cells [13], or quasi-linear as for the lead acid cells [12, p. 2]. The quasi-linear relation of  $U_{oc}$  and SOC for a lead acid battery is shown in figure 4.4 and can be written as in equation 4.4 where the minimum voltage is usually 1,75 V and the maximum range between 2,125 V and 2,05 V [45, p. 23.2.1].

$$U_{oc}(t) = (U_{oc}^{max} - U_{oc}^{min}) SOC(t) + U_{oc}^{min} \quad (4.4)$$

The voltage of the series resistor  $R_s$  is simply given by the Ohm law equation 4.5

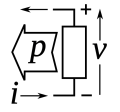
$$U_{R_s}(t) = R_s(SOC(t))i(t) \quad (4.5)$$

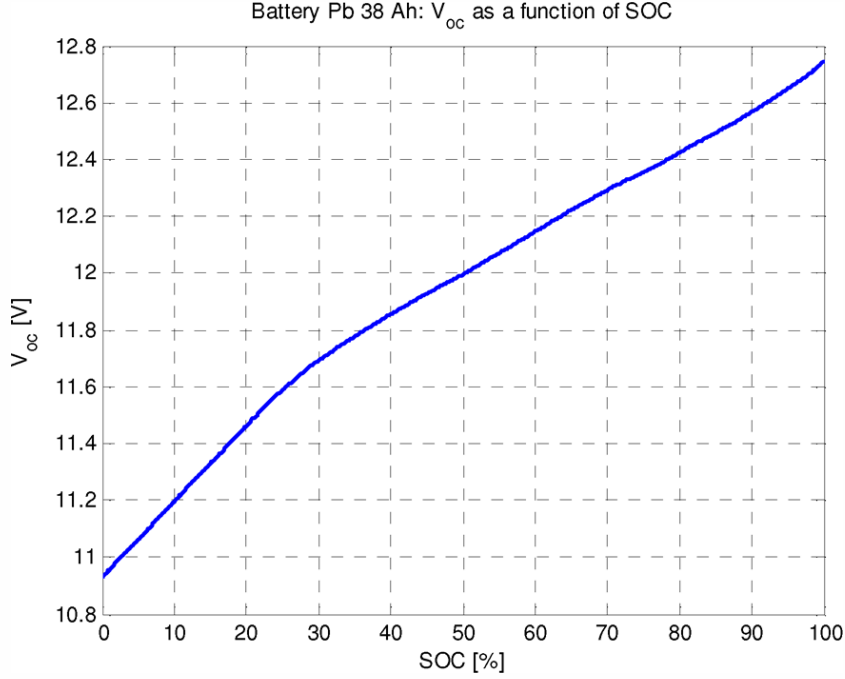
The voltage of each linear parallel RC for the current  $i(t)$  and the time constant  $\tau_j = R_j C_j$ , is given by equation 4.6 [17, p. 126].

$$U_j(t) = U_j(t_0) \exp\left(\frac{-t}{\tau_j}\right) + \int_{t_0}^t \frac{1}{C_j} \exp\left(-\frac{(t-t')}{\tau_j}\right) i(t') dt', \quad j = a, b. \quad (4.6)$$

If the time interval of integration  $\Delta t = [t_0, t]$  is small enough that the current  $i(t)$  and the SOC can be considered constant (thus all the SOC related parameters of the circuit), equation 4.3, equation 4.4, equation 4.6 and equation 4.5 are simplified in the discrete form of equation 4.7.

<sup>3</sup>Voltage positive when current goes out of the electrical element





**Figure 4.4:** Quasi-linear relation between the Open Circuit Voltage  $U_{oc}$  and the SOC for a generic lead acid battery. Source [6]

$$\begin{aligned}
 U_t &= U_{oc,t} - (U_{R_s,t} + U_{at} + U_{bt}) \\
 U_{oc,t} &= (U_{oc}^{max} - U_{oc}^{min}) SOC_t + U_{oc}^{min} \\
 U_{R_s,t} &= R_s(SOC_t) I_t \\
 \tau_j(SOC_t) &= R_j(SOC_t) C_j(SOC_t), \quad j = a, b \\
 U_{jt} &= R_j(SOC_t) I_t + (U_{jt_0} - R_j(SOC_t) I_t) \exp\left(\frac{-\Delta t}{\tau_j(SOC_t)}\right), \quad j = a, b.
 \end{aligned} \tag{4.7}$$

#### 4.4.2 SOC counter

The soc counter computes the SOC through a coulomb counting method, that is, an integration over time of the net current.

The charge exchange process is modeled as shown in equation 4.8 where  $\gamma$  is a conversion factor which is equal to 3600 if the capacity  $C$  is in A h.

$$\Delta SOC(t) = \int_{t_0}^t \frac{-i(t')}{\gamma C} dt' \tag{4.8}$$

Considering a discrete time implementation in which at each time interval  $\Delta t = t - t_0$  the current  $i(t)$  has a constant value  $I_t$ , the equation 4.8, becomes equation 4.9

$$\Delta SOC_t = \frac{-I_t \Delta t}{\gamma C} \quad (4.9)$$

The discrete SOC is finally given by equation 4.10

$$SOC_t = SOC_0 + \sum_{j=0}^t \Delta SOC_j. \quad (4.10)$$

#### 4.4.3 Multi-cell model

The model developed in the previous subsections is related to a single electrochemical cell, while the battery systems are composed of more cells, thus the model must be scaled to a multi-cell model.

The standard way to describe the internal connections between the single cells is using the expression

$$sSpP$$

where  $p$  is the integer number of parallel branches and  $s$  is the number of cells connected in series for each parallel branch. The battery voltage is given by the sum of the  $s$  series cells of each branch, on the contrary the total capacity of the battery is given by the product of the single cell capacity and the number of parallel branches  $p$ .

Considering a generic  $sSpP$  multi-cell battery, its equivalent electrical circuit is the result of the connection in series and parallel of the single cell model. The circuit has  $sp$  parameters  $\theta$  plus  $sp$  initial condition  $Q_0$  and  $sp$  open circuit voltage function  $U_{oc}$ ; the computational cost of model is too high if compared to its limited benefits, in fact the model is not able to simulate all the inter-cell losses and the limitation to the battery capacity given by the capacity mismatch between cells. In order to obtain a fast simulation, it is assumed that the cells have the same capacity and no mismatch is present, thus the current at the battery clamps is equally split between the parallel branches in which the branch current passing through cells is equal. The previous assumptions allow to define an equivalent circuit for the entire battery so that the equations 4.7 become equation 4.11

$$\begin{aligned}
 U_t^b &= U_{oc}^b - (U_{R_s}^b + U_{a,t}^b + U_{b,t}^b) \\
 U_{oc}^b &= s (U_{oc}^{max} - U_{oc}^{min}) SOC_t + U_{oc}^{min} \\
 U_{R_s}^b &= \frac{s}{p} R_s(SOC_t) I_t^b \\
 \tau_j^b(SOC_t) &= R_j^b(SOC_t) C_j^b(SOC_t) = \frac{s}{p} R_j(SOC_t) \frac{p}{s} C_j(SOC_t) = \tau_j(SOC_t), \quad j = a, b \\
 U_{j,t}^b &= \frac{s}{p} R_j(SOC_t) I_t^b + \left( s U_{j,t_0} - \frac{s}{p} R_j(SOC_t) I_t^b \right) \exp \left( \frac{-\Delta t}{\tau_j^b(SOC_t)} \right), \quad j = a, b. \quad (4.11)
 \end{aligned}$$

Concerning the SOC counter, the modification are related to the capacity and current used in the equation 4.9, which are not the single cell quantities but the battery capacity and current  $C_b = pC$  and  $I_t = \frac{I_{b,t}}{p}$ , so that the resulting equation is 4.12

$$\Delta SOC_t = \frac{I_{b,t} \Delta t}{\gamma C p^2}. \quad (4.12)$$

## 4.5 Algorithms

The developed model needs to be fitted to the actual storage system and in order to do that is necessary to set the values of the following parameters:

- the parameters of the circuit elements  $\theta_1, \dots, \theta_{15}$ ,
- the maximum and minimum cell voltages  $U_{max}$  and  $U_{min}$ ,
- the single cell charge capacity  $C$ ,
- the parameters  $p$  and  $s$  of cells connections inside the battery.

The algorithms used to tune the model parameters and to estimate the SOC are implemented starting from the work [21].

### 4.5.1 Cell parameters estimation algorithm

While the parameters  $U_{max}$ ,  $U_{min}$  and  $C$  can be found in the datasheet of the electrochemical cell, the fifteen parameters  $\theta = \{\theta_1, \dots, \theta_{15}\}$ , must be evaluated through experimental data or data taken during the normal operation on the system.

The fitting algorithm used to estimate the cell parameters consists in an optimization problem which minimizes the difference between the terminal voltage  $U_t$ , defined in equation 4.3, computed by the model for a recorded current  $I_t^{rec}$  and the recorded voltage  $U_t^{rec}$ , having as variables the  $\theta$  and the initial state  $\mathbb{Q}_0 = \{SOC_0, U_{a,0}, U_{b,0}\}$ . For the time series  $\mathbb{D} =$

$\{(U_t^{rec}, I_t^{rec}), t = 0, 1, \dots, T\}$  and the variables  $Q_0, \theta$ , the error function between the two voltages is defined as in equation 4.13

$$\epsilon(\mathbb{D}, \theta, Q_0) = \frac{1}{T} \sum_{t=0}^T (U_t(\theta, Q_0, I_t^{rec}) - U_t^{rec})^2. \quad (4.13)$$

Given the set of variables  $\{\theta, Q_{00}, \dots, Q_{0K}\}$  and the set of time series  $\{\mathbb{D}_k, k = 1, 2, \dots, K\}$  obtained from historical operational data, the objective function of the minimization problem is defined in equation 4.14 while the constraints function are expressed in equation 4.15.

$$f(\theta, Q_{00}, \dots, Q_{0K}, \mathbb{D}_0, \dots, \mathbb{D}_K) = \min_{\theta, Q_{00}, \dots, Q_{0K}} \left\{ \frac{1}{K} \sum_{k=0}^K \epsilon(\mathbb{D}_k, \theta, Q_0) \right\} \quad (4.14)$$

$$\begin{cases} \theta_1 \exp(\theta_2 SOC) + \theta_3 \geq 0 \\ \theta_4 \exp(\theta_5 SOC) + \theta_6 \geq 0 \\ \vdots \\ \theta_{13} \exp(\theta_{14} SOC) + \theta_{15} \geq 0 \end{cases} \quad 0 \leq SOC \leq 1 \quad (4.15)$$

#### 4.5.2 SOC estimation algorithm

When all the parameters of the cell model are known, is possible to estimate the SOC of the storage for a time instant  $t'$  just using a time series of voltage and current at the battery clamps.

At time instant  $t'$  using the last  $n - th$  time instant recorded values, the time series  $\mathbb{D}' = \{(U_t^{rec}, I_t^{rec}), t = t' - n, t' - n + 1, \dots, t'\}$  is built; the number  $n$  should be small in order to have a small data-storage requirement and to reduce the drift error due to the integration of current in the SOC counter. The optimization problem defined in equation 4.16, which minimizes the error function 4.13 for the time series  $\mathbb{D}'$  having only  $Q_0' = SOC_0, U_{a,0}, U_{b,0}$  as unknown variables, finds the initial values  $SOC_0$  of the storage system.

$$\min_{Q_0'} \{ \epsilon(Q_0', \mathbb{D}', \theta) \} \quad (4.16)$$

Using the  $SOC_0$  and the current values  $I_t$  of the time series  $\mathbb{D}'$  in equation 4.8, it is possible to compute the value of  $SOC_{t'}$ .



## Chapter 5

# Forecasting system

The forecasting of the inputs of a EMS in a microgrid, mainly not programmable sources power output and loads power demand, are of fundamental importance to optimize the management of the power system and it becomes of primary relevance in islanded microgrids where the instant power balance cannot be demanded to the main grid but, how the authors express in [33], differently from the classical interconnected systems, « it is recognized, however, that it is still premature to propose a particular operational forecasting tool for microgrids».

In this chapter, the methods used in the developed EMS to forecast two not programmable renewable sources (photovoltaic generator and wind generator) load, are presented. The structure of the chapter is the following: firstly the models used for the photovoltaic generator and the wind generator are described, then the more used forecasting methods found in literature are summarized, finally the chosen methods, the reasons for the choice and their implementation close the chapter.

### 5.1 Photovoltaic generator

A photovoltaic system is a device cable of converting the power of an electromagnetic radiation directly into electric power due to the photoelectric property of the constituent material.

Although the theoretical explanation of the photovoltaic generation is understood since 1904 [22] and starting from the 50's the photovoltaic technologies are used to power isolated electrical system in remote or extraterrestrial area, it is from the beginning of the XXI century that the photovoltaic technology entered the power system domain.

In the following a basic summary of the physic which allowed the direct conversion of electromagnetic power into electrical Direct Current (DC) power without any mechanical movement is exposed stressing the main phenomena for which a model must be accurate, then the implemented model of a photovoltaic system is presented.

### 5.1.1 Basic photovoltaic concepts

The conversion of the energy of an electromagnetic radiation into electric energy is based on the propriety of some materials, called semiconductors, of releasing electrons if they are excited by a photon carrying a specific amount of energy.

In the field of quantum theory, materials can be divided based on the discrete allowed energy states of their electrons, states which are called bands and are separated by not allowed energy states called gaps. The semiconductors, when at absolute zero temperature (0 K), have no electron in the most energetic band in which the electrons have enough energy to run away from the nuclei (thus it is called conduction band) and even at standard temperature (298,15 K) a percent as small as  $1 \times 10^{-10}$  of electron is free. If a photon that hits a semiconductor has an energy greater or equal the energy different between the conduction band and the just lower band, the photon is absorbed and one electron passed from the lower band into the conduction band leaving a hole in the lower band. The electron can recombine with an hole and the semiconductor emits a photon(see figure 5.1), but if an electrostatic field is present inside the material, the electron, which is a negative charge, and the hole, which is a positive charge, are separated allowing the generation of an electric current.

The internal electrostatic field is created by the union of two elements made of the same semiconductor material but each one with a distinct impurity: the impurity of one side makes the semiconductors prone to accept electrons (p-type) while the impurity of the other side makes the semiconductor prone to accept holes (n-type). When the two semiconductor parts are united, at the interface between them, electrons start going from the n-type to the p-type, while the holes start going in the opposite direction. The process stops when the concentration of electron inside the p-type side of interface and the concentration of holes in the n-type side of the interface, generate a electrostatic field which opposes the diffusion process.

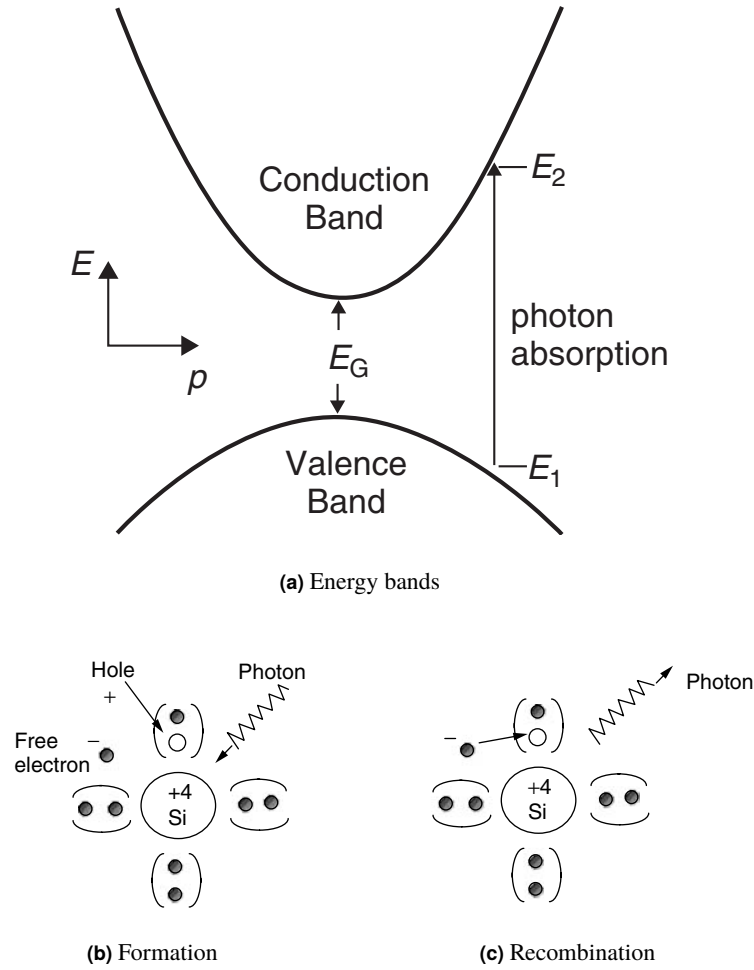
After the generation of a hole-electron pair, if one of them reach the proximity of the interface called depletion zone, it is pushed into the side opposite of that of the other one: the electrons are pushed into the n-type element while the holes into the p-type element. If the two elements are connected by an external conductor, electrons flow from the n-type element to the p-type element where they recombine with holes.

The two physical quantities which regulate the photovoltaic conversion are the photon energy (that is the irradiance over the semiconductor material) and the temperature of the material (the energy state of electron without photon interaction), thus a photovoltaic model required as main inputs the two physical quantities.

### 5.1.2 Irradiance estimation

Irradiance is the measure of the electromagnetic power of a radiant flux over a surface area orthogonal to the flux ( $\text{W m}^{-2}$ ).

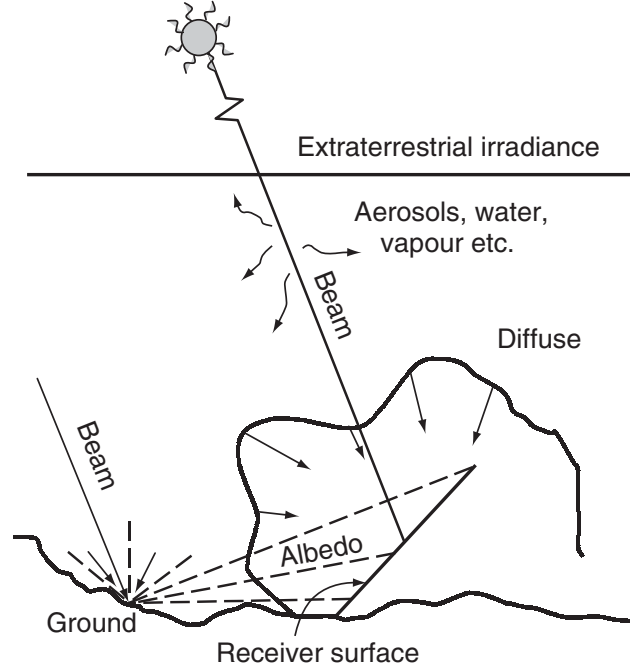
Considering a generic flat surface represented in figure 5.2, the irradiance of the solar flux perpendicularly to the surface  $G$  can be divided into three components: a direct component  $B$ , a diffuse component  $D$  and a reflected component  $R$ . The direct component is the flux



**Figure 5.1:** In the formation process, the absorption of a photon move an electron to the conduction band generating a free electron and an hole. In the recombination process the reunion of an electron and a photon generates a photon. Sources [48, 50]

that reaches the surface through a direct path between the latter and the source, the diffuse component is composed by all the indirect fluxes given by the scattering propriety of the medium between source and receiving surface, the reflected component is given by the all the fluxes which were reflected by other surfaces and directly reach the surface of interest without be affected by scattering processes. The three irradiance components of solar irradiance are measured using two instruments: the pyranometer that measures the total irradiance (or the diffuse by the means of a shading ring) and the pyrheliometer that looks at the sun through a tube in order to measure only the direct irradiance.

Technologies like solar concentrators use only the direct component while flat photovoltaic modules use the global flux. Being the latter technology of interest for the following development of the photovoltaic generator model and given the solar radiation measure available,



**Figure 5.2:** Components of the solar irradiance hitting a generic surface: the extraterrestrial irradiance is split by atmosphere into a direct (beam) component and a diffuse component while the third component albedo is given by the reflection on the ground of both the direct and diffuse components. Source [48].

it is investigated a method that, starting from the measure of the global horizontal irradiance  $G_h$  that is the global irradiance for surface horizontal to reference ground, evaluates the global irradiance on a surface  $G$  with a generic position and inclination on Earth. The process of evaluation starts inferring the direct normal irradiance  $B_n$  (the direct component of irradiance of a surface orthogonal to the flux) from the  $G_h$  by means of the Direct Insolation Simulation Code (DISC) model [51], then the direct, diffuse and reflected irradiance on the surface are evaluated and added together to obtain the global irradiance on the surface  $G$ .

### DISC Model

The main point of the *DISC* model is the empirical assumption that the  $B_h$  and  $G_h$  are mainly correlated by the air mass  $AM$  which is the relative length of the direct-beam path through the atmosphere compared with a vertical path directly to sea level. The quantification of this correlation involves the definition of two parameters: the direct normal transmittance  $K_n$

$$K_n = \frac{B_n}{B_0}$$

ratio of the  $B_n$  over the extraterrestrial irradiance  $B_0$ , and the global horizontal transmittance  $K_t$

$$K_t = \frac{G_h}{B_{0,h}}$$

where  $B_{0,h}$  is the extraterrestrial irradiance over a surface parallel to the reference ground. An analysis the two aforementioned parameters revealed that they are linked by a function of the AM.

The model gives a simple algorithm to find the relationship between  $B_h$  and  $G_h$ ; the algorithm steps are:

1. calculate the  $B_{0,h}$  with

$$B_{0,h} = B_0 \epsilon_0 \cos(\theta_{ZS})$$

where  $\epsilon_0$  is reciprocal of the square of the earth radius vector given by

$$\epsilon_0 = 1,00011 + 0,034221 \cos(\xi) + 0,00128 \sin(\xi) + 0,000719 \cos(2\xi) + 0,000077 \sin(2\xi)$$

with  $\xi$  eccentric anomaly of earth in its orbit around the sun given by

$$\xi = \left( \frac{2\pi}{365} \right) (d_n - 1)$$

$d_n$  the consecutive day number of the year (1 is the 1st of January, 365 is the 31st of December),

2. calculate  $K_t$  using the measure of  $G_h$  as

$$G_h = \frac{G_h}{B_{0,h}}$$

3. calculate the AM as

$$AM = \left( \frac{1}{\cos(\theta_{ZS}) + 0,15(93,885 - \theta_{ZS})^{-1,253}} \right) \left( \frac{p_{atm}}{p_{atm,std}} \right)$$

with  $p_{atm,std}$  atmospheric pressure at standard condition and the actual atmospheric pressure  $p_{atm}$ .

4. calculate the direct normal transmittance for a clear-sky condition  $K_{n,clear}$  as

$$K_{n,clear} = 0,866 - 0,122AM + 0,0121AM^2 - 0,000653AM^3 + 0,000014AM^4$$

5. calculate the  $K_n$  as

$$K_n = K_{n,clear} + \Delta K_n$$

where the variation of the  $K_n$  from the  $K_{n,clear}$  is due to the current atmospheric state that can be described by proprieties which are air mass, cloud cover, water vapor and are correlated with  $K_t$  and is extract fitting experimental data so that it is expressed by

$$\Delta K_n = a(K_t) + b(K_t) \exp(c(K_t) - AM)$$

where the three parameters  $a(K_t)$ ,  $b(K_t)$ ,  $c(K_t)$  are selected in function of the value of  $K_t$  as

$$\begin{aligned} a &= 0,512 - 1,56K_t + 2,286K_t^2 - 2,222K_t^3 \\ b &= 0,370 + 0,962K_t \\ c &= -0,280 + 0,932K_t - 2,048K_t^2 \end{aligned}$$

if  $K_t \leq 0,60$ , or

$$\begin{aligned} a &= -5,743 + 21,77K_t - 27,49K_t^2 + 11,56K_t^3 \\ b &= 41,40 - 118,5K_t + 66,05K_t^2 + 31,90K_t^3 \\ c &= -47,01 + 184,2K_t - 222,0K_t^2 + 73,81K_t^3 \end{aligned}$$

if  $K_t > 0,60$ .

6. finally from the definition of  $K_n$ , compute the  $B_n$  as

$$B_n = K_n B_0.$$

The simplicity of the DISC model limits the precision of the computed if compared to more complex model as the modified DISC (DIRINT) [62] (which is an improvement of the DISC) or the DIRINDEX [63] as described in [90], but it requires only four inputs:  $G_h$ ,  $\theta_{ZS}$  the day of the year  $n$  and the pressure input  $p_{atm}$ .

### Global Module Irradiance composition

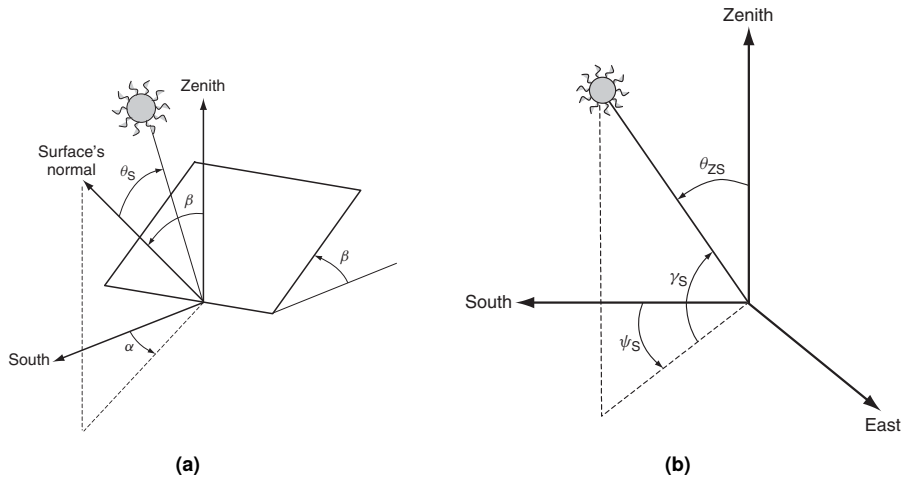


Figure 5.3: Source [48]

If the direct normal irradiance  $B_n$  is known, the direct irradiance on the surface  $B$  and the direct horizontal irradiance  $D_h$  can be obtain with geometrical consideration of the relative

positions between irradiance source and the ground and between source and surface. Direct irradiance on surface  $B$  is given by

$$B = B_n (\cos(\beta) \cos(\theta_{ZS}) + \sin(\beta) \sin(\theta_{ZS}) \cos(\psi_S - \alpha))$$

and the direct horizontal irradiance  $D_h$

$$B_h = B_n \cos(\theta_{ZS})$$

with (see figure 5.3):

- tilt angle  $\beta$ , inclination of the surface respect the reference ground measured starting from the side not directly exposed to the solar flux,
- surface azimuth  $\alpha$  angle between the surface normal to the exposed side of the surface and the true geographical South point (Northern Hemisphere) or the true North point (Southern Hemisphere); by convention the azimuth is positive in East-South sector and negative in South-West sector in the North hemisphere while it has opposite signs in the South hemisphere,
- sun azimuth  $\psi_S$ , angle between the true South or true North cardinal points (based on the hemisphere, as for  $\alpha$ ) and the segment that links the Sun with the surface.

The diffuse irradiance that reaches the surface  $D$  is due to multiple and different scattering and reflecting processes related to the composition of the air, the presence of clouds and even the reflective property of the ground that can reflect the solar flux back into the atmosphere where it is scattered albedo radiation).

Different models were proposed to estimate the diffuse flux, but because of the complexity of the problem, they are all empirical or semi-empirical models. Usually the diffuse radiation is split into three parts: isotropic which estimates the uniform flux from the sky dome, circumsolar which considers the scattered flux near the sun disc, horizon brightening which estimates the increase of irradiance near the horizon given by the albedo radiation of the ground. The simplest models consider only the isotropic radiation [88], while more complex models add the remaining second component as the Hay-Davies model [34], and the third one as in Reindl [69]. Considering the input data in posses, the simplest model is chosen even if it underestimate the diffuse irradiance on equator tiled surfaces.

The isotropic model considers that the irradiance emitted by a solid angle of sky is constant for all the possible orientation of the solid angle. An horizontal surface sees all the sky dome, thus all the diffuse irradiance of the sky which dome is called horizontal diffuse irradiance  $D_h$  and is evaluated as

$$D_h = G_h - B_h.$$

Always for the isotropic assumption, the diffuse irradiance reaching the surface  $D$  is only function of how much sky dome the surface sees, that is a vertical surface sees half sky dome, a horizontal surface sees the full dome, a upside-down surface does not see any part

of the sky. The diffuse surface irradiance can be expressed as function of the total diffuse irradiance and the portion of sky that it sees using the  $\beta$  is equation

$$D = D_h \frac{1 + \cos(\beta)}{2}. \quad (5.1)$$

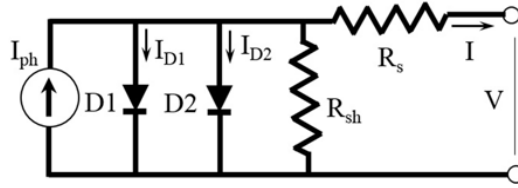
The reflected irradiance  $R$  is heavily dependent on geometry and reflecting propriety of the material on which the direct and diffuse irradiance are reflected, thus an high amount of assumption are required indeed the share is always limited if compared to the direct and diffuse irradiance so that a complex model is not required.

The model used assumes that the reflecting element is flat and horizontal respect to the surface of interest, has infinite size and reflects isotropically. The reflected irradiance can be estimated as

$$R = \rho G_h \frac{1 - \cos(\beta)}{2} \quad (5.2)$$

where  $\rho$  is the dimensionless reflective coefficient of the material which varies from 0,8 for fresh snow to 0,1 for bituminous materials; if unknown, it is set at 0,2.

### 5.1.3 Model of the photovoltaic module



**Figure 5.4:** Equivalent electrical circuit for a photovoltaic cell. Source [40]

All the models of a photovoltaic generator system start modeling the elementary unit of the generator, the photovoltaic cell then, from it the entire system is modeled. The cell models present in literature can be divided into two main groups: analytic models and empirical models.

The analytic models are based upon the description of the electrical characteristic (U-I) of the cell for each solar irradiation and working temperature through a set of analytic equations. With the assumptions that the cell material have constant proprieties respect the variation of irradiance [40], the equations are generally obtained from an equivalent non-linear lumped circuit formed by an ideal current generator in parallel with one or two real diode (see figure 5.4 ) so that the generated current is given by equation 5.3 [7]

$$\begin{aligned} I &= I_{ph} - I_{D1} - I_{D2} - \left( \frac{V + I R_s}{R_{sh}} \right) \\ I_{D1} &= I_{01} \left( \exp \left( \frac{V + I R_s}{a_1 V_{T1}} \right) - 1 \right) \\ I_{D2} &= I_{02} \left( \exp \left( \frac{V + I R_s}{a_2 V_{T2}} \right) - 1 \right) \end{aligned} \quad (5.3)$$

where  $I_{01}$  and  $I_{02}$  are reverse saturation current of the diodes,  $V_{T1}$  and  $V_{T2}$  are their thermal voltage,  $a_1$  and  $a_2$  represent the diode ideality constants while parallel and series resistors model the dissipative and parasitic currents inside the cell. The drawback of the analytic models are that they required the solution of an implicit transcendental equation usually obtained with by approximations. A classical analytic model is the five-parameters model presented in [78] and its evolution [8].

The empirical models, instead of using equations based on the physical processes happening in the cell, try to find a correlation between the inputs (typically irradiance and temperature) and the output (electrical DC power) through fitting experimental data. Even if these models have a lower accuracy respect the analytical models and they required an elevate number of experimental data, after the evaluation of the model parameter which can be time consuming, the computational cost of the empirical models is inferior than the analytical models. Examples of empirical models are Sandia PV Array Performance Model [19], PVWatts [20] and Photovoltaic Geographical Information System (PVGIS) [23, 36].

Considering that the final goal of the photovoltaic generator model under development is to forecast the power output of a photovoltaic system for a real time optimal scheduling of resources in a microgrid, the priority is given to the speed feature, thus the low computational cost, of the model that has to evaluate only the DC power output. The power output of a photovoltaic module has a strong non-linear relation with the irradiance and the module temperature which is itself a function of the irradiance and other weather and structural variables mainly related to the wind exposure; for that a model with only irradiance  $G$ , ambient temperature  $T_{amb}$  and wind exposure is asked. The power performance model used by the PVGIS of European Commission's Joint Research Centre (JRC) project is chosen.

### PVGIS Model

The PVGIS model evaluates the variation of output power when irradiance  $G$  on the module and ambient temperature  $T_{amb}$  diverge from the standard condition by means of an empirical relative efficiency function which is function of the  $G$  and the module temperature. The power output of a photovoltaic module is

$$P = GA\eta$$

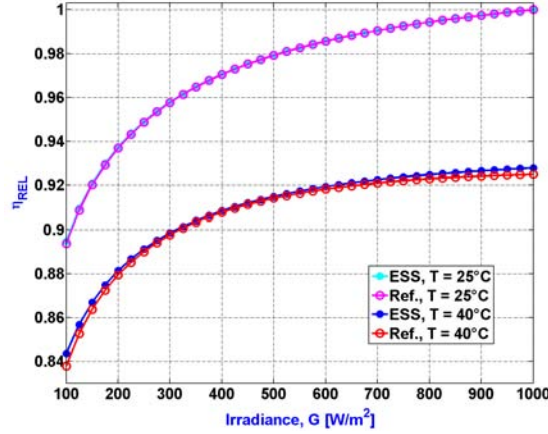
where  $A$  is the module surface area hit by the solar flux and  $\eta$  is the module efficiency. The power at Standard Test Condition (STC), that is with  $G_{stc}$  and efficiency  $\eta_{stc}$ , is

$$P_{stc} = G_{stc}A\eta_{stc}$$

so that the generic power output can be expressed as function of the values in standard condition as

$$P_{module} = P_{stc} \frac{G}{G_{stc}} \eta_r. \quad (5.4)$$

The relative efficiency  $\eta_r = \frac{\eta}{\eta_{stc}}$  has to address the strong variation of power output when the module temperature and the solar irradiance divert from the standard condition. In figure 5.5



**Figure 5.5:** Relative efficiency characteristics as function of  $G$  at module temperatures  $T_{mod}$  of 298,15 K and 313,15 K for two mono-crystalline silicon modules. The characteristics have logarithmic shapes for the  $G$  variable and show a degradation of performance at higher module temperature. Source [96, p. 252]

it is possible to see that the relative efficiency  $\eta_r$  has a logarithmic shape for the irradiance  $G$  variable, thus in PVGIS model, it is expressed as

$$\eta_r = 1 + k_1 \log(G') + k_2 \log(G')^2 + T' (k_3 + k_4 \log(G') + k_5 \log(G')^2) + k_6 T'^2 \quad (5.5)$$

where the  $G' = \frac{G}{G_{stc}}$  and  $T' = T_{mod} - T_{module, stc}$  are the irradiance and temperature <sup>1</sup> normalized to STC, while the six coefficients  $k_1, \dots, k_6$  are empirically evaluated for a given photovoltaic technology based on the power characteristic (in [96] the author tries to relate them to the electrical parameters of the Voltage-Current characteristic).

The method used to found the coefficients, as described in [36, p. 329], has the following steps:

1. given a set of modules, for each module the maximum output power is measured for seven irradiances  $G$  between  $200 \text{ W m}^{-2}$  and  $1000 \text{ W m}^{-2}$  and for four module temperature  $T$  between 298,15 K and 333,15 K for a total of twenty-eight measures in a similar way of what is prescribed in [37],
2. for each module, a least-square algorithm is used to fit equation 5.4 to the experimental measure of the previous step with  $k_1, \dots, k_6$  and  $P_{stc}$  as variables,
3. for each module, the measure power is normalized to the evaluated  $P_{stc}$  so that the relative efficiency of modules with different power at standard condition are comparable ( $\frac{P}{P_{stc}} = G' \eta_r$ ),

<sup>1</sup>In the source temperatures are in °C but the model can be used also with temperatures in K because variation in °C and K are equal.

4. the normalized values related to a photovoltaic technology are group together a fitting algorithm is used on equation 5.4 with  $k_1, \dots, k_6$  as variables and  $P_{stc} = 1$  W due to the normalization process.

The result of the described method for a generic crystalline silicon module present in [36] are shown in table 5.1

Technology	$k_1$	$k_2$	$k_3$ (K <sup>-1</sup> )	$k_4$ (K <sup>-1</sup> )	$k_5$ (K <sup>-1</sup> )	$k_6$ (K <sup>-2</sup> )
c-Si	-0,017 162	-0,040 289	-0,004 681	0,000 148	0,000 169	0,000 005

**Table 5.1:** Coefficients of the relative efficiency  $\eta_r$  of the PVGIS model for a generic crystalline silicon photovoltaic module. Source [36]

### Thermal component of the model

The model requires the working temperature of the module, quantity that can be directly measured with a sensor attach to the back of the module or, by means of a thermal model, indirectly obtained from the ambient temperature and other weather quantities which can be measured or inferred by a atmospheric model. In the following the empirical thermal model developed in [19] is exposed.

The model assumptions are that the thermal system is in a steady state and the working temperature of the module  $T_{mod}$  depends mainly on the ambient temperature  $T_{amb}$ , the irradiance  $G$ , and the convective heat transfer between module surface and air due to wind. The temperature at the back of the module  $T_{mod,back}$  is then given by equations 5.6

$$T_{mod,back} = T_{amb} + G \exp(a + b v) \quad (5.6)$$

where  $v$  is the wind speed in  $\text{m s}^{-1}$  at the reference height of 10 m,  $a$  and  $b$  are two negative coefficients that respectively determinate the temperature variation due to solar irradiance gmi for low and high wind speed  $v$ . In order to evaluate  $a$  and  $b$ , records of module temperature, ambient temperature and the speed for different module configurations are used to fit equation 5.4 using  $a$  and  $b$  as variables. Different module configurations are required because the convective transfer due to wind depends on the exposition of the module to the air. The working temperature inside the module  $T_{mod}$  is then given by equation 5.7

$$T_{mod} = T_{mod,back} + \frac{G}{G_{stc}} \Delta T_{mod} \quad (5.7)$$

where  $\Delta T_{mod}$  is the difference between inside and back of the module when it is exposed to  $G_{stc}$  in different module configurations and it is measured with two thermal sensors.

The results of the fitting process for coefficients  $a$ ,  $b$  and  $m$  are shown in table 5.2 <sup>2</sup>.

<sup>2</sup>In the source temperatures are in °C but they can be used also with temperatures in K because  $a$  and  $b$  are used to compute a temperature variation and  $\Delta T_{mod}$  is a temperature variation; variation in °C and K are equal.

Module technology	Configuration	a	b	$\Delta T_{mod}$ (K)
Glass/cell/glass	Open rack	−3,47	−0,0594	3
Glass/cell/glass	Close roof mount	−2,98	−0,0471	1
Glass/cell/polymer sheet	Open rack	−3,56	−0,0750	3
Glass/cell/polymer sheet	Insulated back	−2,81	−0,0455	0
Polymer/thin-film/steel	Open rack	−3,58	−0,113	3

**Table 5.2:** Coefficients  $a$ ,  $b$  and  $\Delta T_{mod}$  for different module configurations. Source [19]

### 5.1.4 Complete system model

A photovoltaic system is normally composed by one or more photovoltaic modules connected together, a system of power conversion which can be a DC-DC type or a DC-AC type with Maximum Power Point Tracker (MPPT) that, conditioning the current and voltage of the load seen by the modules, allows the modules to work with the maximum efficiency, protection systems, connection cables and loads. All the elements of the system between the modules and the loads add losses with a reduction of the overall efficiency. In small system typical of a microgrid, the principal loss is given by the power converter system because the cables and protections losses are minimum, due to their shortness and limited number, and because the mismatch losses given by not perfectly equal electric characteristic at same  $G$  and temperature between modules are ignorable because of the limited number of modules, . For the aforementioned reasons, the complete system model is considered as the power  $p_{module}$  of the single photovoltaic module multiplied the total number  $n_{modules}$  of modules connected to a single power converter minus the power loss of the latter which can be expressed as an efficiency  $\eta_{converter}$ , thus the complete power output of the photovoltaic generator system is given by equation 5.8

$$P_{sys} = \eta_{converter} n_{modules} P_{module}. \quad (5.8)$$

The model is present is split in the function **irradiance\_component** in module **forecasting** for the evaluation of the  $G$  and in **photovoltaic\_generator** of module **models**. For a description of the implemented code see **Annex Photovoltaic model**.

## 5.2 Wind generator

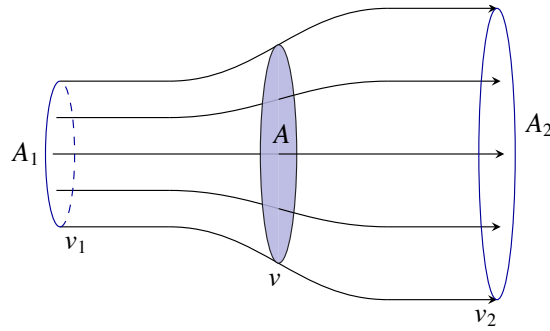
A wind generator or Wind Energy Conversion System (WECS) is a device capable of converting the mechanical power of a moving air mass into mechanical or more often electrical power usable to do work.

The WECS used for electrical power production are composed by three principal elements: the turbine, the electric generator and the power conversion device. The turbine converts the mechanical power of the wind flux into rotational mechanical power, the electrical generator converts the rotational motion into electrical power, finally the power conversion device

adapts the electrical values of voltage, current and frequency so that they submit the required standards. The differences in these three elements are used to classified the WECS into sub-groups: refereed to the orientation of the turbine rotation axis respect to the ground, there are Vertical an Horizontal axis typologies with the latter split into the upwind and downwind typologies, refereed to the electric generator there are induction and synchronous typologies.

The following sections focus on the small horizontal axis with permanent magnet synchronous generator. After a short derivation of the wind energy, the implemented model used for the forecasting system is presented.

### 5.2.1 WECS power output



**Figure 5.6:** Stream tube of an ideal air flow passing through a extracting actuator disc  $A$ : the reduction of kinetic energy ( $v_2 < v_1$ ) converted into mechanical power by the disc actuator, results in an increase of the stream section for the principle of mass conservation.

Analyzing the interaction of a wind turbine with a moving air mass, assuming that:

- the air mass interacting with the wind turbine does not interact with the remaining air so that a stream tube can be define (see figure 5.6),
- in each axial section of the air velocity is uniform in particular in the actuator disc section  $A$ ,
- in the far away sections of the stream tube before  $A_1$  and after  $A_2$  the air has the ambient pressure,
- the air fluid is incompressible

then the wind power  $P$  converted by the turbine is given by equation

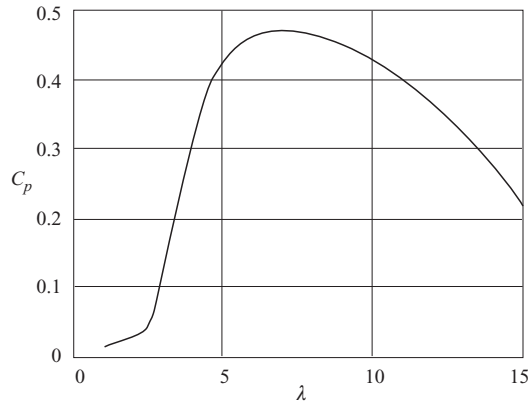
$$P = \frac{1}{2} \rho_{air} A v_1^3 C_p(\lambda, \theta) = \left( \frac{1}{2} \pi \rho_{air} R^2 v_1^3 \right) C_p(\lambda, \theta) \quad (5.9)$$

with  $\rho_{air}$  is the air density (at standard condition  $\rho_{air, std} = 1,225 \text{ kg m}^{-3}$ ),  $R$  is the radius of the wind turbine,  $v_1$  is the wind speed at the beginning of the stream flux and  $C_p(\lambda, \theta)$  is a

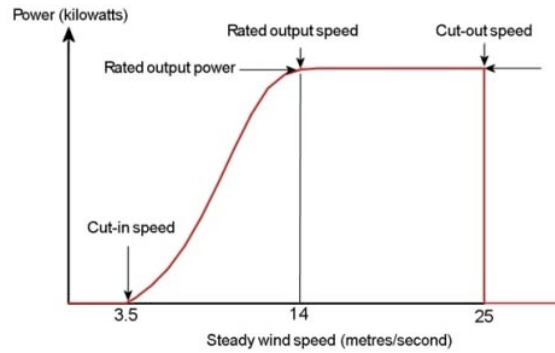
power coefficient which is a function of the pitch angle  $\theta$  of the turbine blades and the tip speed ratio

$$\lambda = \frac{\omega R}{v_1}$$

which is the ration between the tip velocity of blade and the wind speed. Therefore the converted power is given by the wind kinetic energy (terms between brackets) and the power coefficient  $C_p$  that accounts for the fluid dynamic between the air and the turbine blades. The power coefficient for a fixed pitch angle and a variable tip speed ratio is shown in figure 5.7: it is visible that the coefficient reaches its maximum with a high slope and then starts to decrease with high wind speed due to the predominance of drag forces. Equation 5.9 shows that the output power  $P$  has a quadratic relation with the blades length  $R$  that explains why there is tendency to increase the size of WECS: the cost of the blades almost linear with their length that is a half of the power output so that bigger turbines are more cost effective [50].



**Figure 5.7:** Characteristic of the power coefficient  $C_p$  as function of the tip speed ratio



**Figure 5.8:** Power characteristic of a WECS with the three points highlighted: the cut-in speed, the rated output speed and the cut-out speed.

The total power characteristic of a wind generator is given by the curve that links the input wind speed to the electrical power output of the electrical generator or the power converter

device. Figure 5.8 shows a typical power characteristic with three three points that define the state of the generator:

- the cut-in speed is the minimum wind speed required to wind the turbine inertia so that for lower speed no kinetic energy is converted into angular motion,
- the rated speed is the wind speed for which the power coefficient  $C_p$  has the maximum value so that, after a high sloped grow, the power output of the WECS reaches its maximum and holds that value until
- the cut-out speed that is the wind speed for which the turbine blades are projected maximize the wind drag force on them so that they automatically slow down for two reasons: to prevent the break due to stress and because the power coefficient  $C_p$  has low value for wind speed higher than the cut-out speed.

Equation 5.9 shows that the power output  $P$  of the WECS grows with the cubic of the wind speed  $v$  thus a small variation of the latter produces a big variation in the former, then the best way to maximize the power output of the system is to use higher tower because a moving fluid near a solid boundary (like the earth ground) has a speed gradient which grows as expressed in equation 5.10

$$v = \log \left( \frac{h}{h_0} \right) \quad (5.10)$$

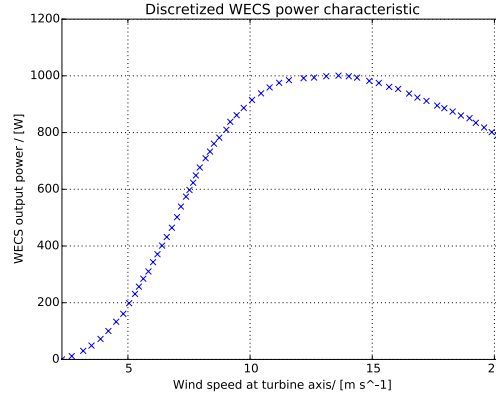
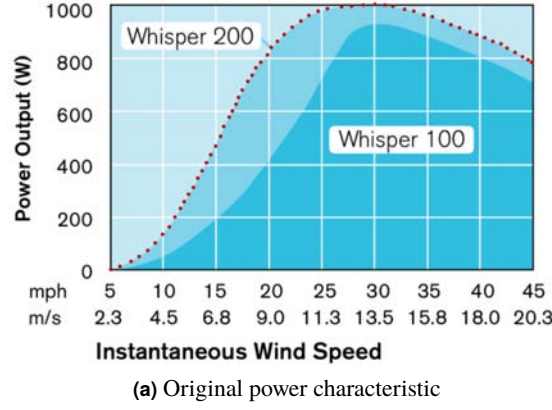
with  $h$ [m] distance between the boundary and the point of interest and  $h_0$ [m] roughness length is the distance at which the wind speed theoretically becomes null and it is called roughness because is related to the conformation of the boundary with a empirical values for different ground typologies given in table 5.3.

Type of Terrain	$h_0$ (m)
Cities, forests	0,7
Suburbs, wooded countryside	0,3
Villages, countryside with trees and hedges	0,1
Open farmland, few trees and buildings	0,03
Flat grassy plains	0,01
Flat desert, rough sea	0,001

**Table 5.3:** Empirical values of the roughness length  $h_0$  for different conformation of the earth ground: the presence of obstacles increases the roughness length. Source [10, p. 17]

### 5.2.2 WECS model

The implemented model, based on the power characteristic of WECS which is usually provided by the generator producer, gives the instant power for a given wind speed input without considering the dynamic time constant of the system.



**Figure 5.9:** Power characteristic of a Wisper200 WECS during the discretization process with the WebPlot-Digitizer software [70] and the discretized version used in the model.

If the power characteristic is given as a plotted continuous curve, it is discretized, as shown in figure 5.9, then for a generic wind speed input if it is outside the working interval delimited by cut-in and cut-out speeds, the related power output is set null while if it is inside the interval a linear interpolation of the sampled power characteristic is used to obtain the power output.

Due to the fact that the device recording the wind speed can be at a different highness of the wind turbine axis, the recorded wind speed values are preprocessed using the equation 5.11 given by the ration of equation 5.10 for the recording device highness  $h_{dev}$  and the turbine axis highness  $h_{axis}$

$$v_{axis} = v_{dev} \frac{\log\left(\frac{h_{axis}}{h_0}\right)}{\log\left(\frac{h_{dev}}{h_0}\right)}. \quad (5.11)$$

## 5.3 Forecasting methods summary

While load forecasting is mainly based only on statistical methods of user classification, the forecasting of not programmable generation is a multi fields problem that spans from statistic to natural science and there is a huge number of publications in literature but all the methods can be divided into two main groups: the physical approach and the statistical approach.

The physical approach uses detailed physical models of the atmosphere that, using inputs provided by the global weather stations network (as wind velocity, solar irradiance, air temperature, pressure) and GIS systems (as surface properties), try to forecast the change in the geographical area of the power system, then the forecasted weather conditions are down-scaled to the generators always using physical relations between global and local properties of the atmosphere. This approach heavily depends on the Numerical Weather Prediction (NWP) models, models used to forecast the weather state up to fifteen days ahead. The principals NWP models are the US National Oceanic and Atmosphere Administration (NOAA), the Global Forecasted System (GFS) and the European Centre for Medium-Range Weather Forecasts (ECMWF).

The statistical approach tries to find a relation between recorded weather data input and the recorded generators output data considering the generator systems as black-boxes, that is there is no mathematical model but only the discovery of patterns. The methods following a statistical approach are even more split into linear method and not-linear method.

### 5.3.1 Statistical linear methods

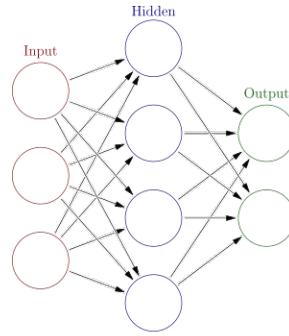
The statistical linear methods consist in an analysis of time-series define as training series, a set of successive recorded data made over a time interval, that gives a series of coefficients which summarize the patterns and statistics of time-series. Using a linear combination of the coefficients and the past values, it is possible to predict the future trend of series having the same quantities of the training ones but recorded in other time intervals.

Excluded less common methods as the Exponential Smoothing or the gray predictors, the principal methods are the Auto-Regressive Moving Average (ARMA) and its variants as Auto Regressive Integrated Moving Average (ARIMA) and Seasonal Auto Regressive Moving Average (s-ARMA).

The ARMA methods consider the forecasted quantity  $S(t)$  composed of two parts as shown in equation 5.12:

$$S(t) = \sum_{i=0}^{p-1} a_i S(t-i) + \sum_{j=0}^{q-1} b_j \epsilon(t-j) \quad (5.12)$$

an autoregressive part based on the known past values of the forecasted quantity  $S(t-i)$  and a moving average part of the error  $\epsilon(t-j)$  which is considered an uncorrelated random variable with zero mean and constant variance.



**Figure 5.10:** Basic configuration of an Artificial Neural Network (ANN): the input layer receives the historical data while the hidden layer and the output layer give the forecasted values of the input quantity. Source [29]

### 5.3.2 Statistical non-linear methods

The field of the statistical non-linear methods is dominated by the ANN: a statistical computational model, inspired by the structure and functions of biological neural networks, that «consists of simple processing units, the neurons, and directed, weighted connections between those neurons» [43]. Usually the most use configuration is the Multilayer Perceptron (MLP) in which the network has three layers (see figure 5.10) an input layer, a hidden layer and an output layer. The historical data of the quantity of interest and other exogenous inputs are given to the input layer (training phase), during this process the weighted connections between input hidden and output layers are modified so that when fed with the input quantities of other other time instant the models forecasts the future values of input of interest [65].

The ANN methods give better results if compared to the linear method [68] especially if a high number of exogenous inputs are provided.

The best results obtained are given by the hybrid approach which use the NWP output as input of a ANN but obviously it has an higher complexity and computational cost than the simpler approaches [77].

## 5.4 Forecasting system implemented

The forecasting methods briefly summarized in section 5.3, even if have a well usage in the power utilities, have a limit for the project under development: they require a complex organization and computational systems (NWP) or an high number of historical data (statistical methods). The present project wants to be as much as possible independent from external resources (its final goal is to manage small microgrids in island mode) and from historical data so that in can be used in new systems. In addition, as stated in [33], in very small application, where any aggregation and smoothing effect derived from multiple sources and loads are absent, the usually applied method of forecasting is the conceptually simple persistence method usually used as reference method to benchmark all the other more complex presented methods.

Persistence method is based on the assumption that the aleatory variable for which the forecasting is needed, conserves the same values from one time instant to the other. For the aleatory variable  $x$  with recorded value  $x(t)$  at time instant  $t$ , the future values of the variable at time instant  $t + k$ ,  $k = 1, 2, \dots, n$  ( $n$  final number of forecasted time intervals) is considered given by

$$x(t + k) = x(t) + \epsilon(t + k)$$

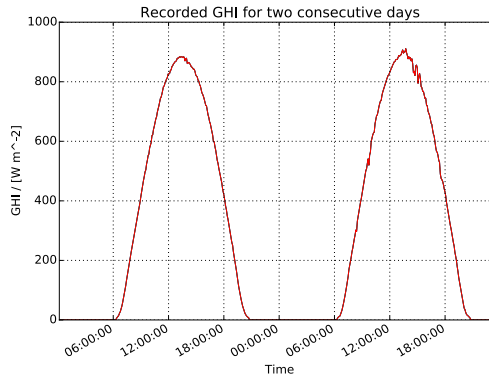
that is, the future value is the same of the actual value plus an aleatory error  $\epsilon(t + k)$ ; the forecasted value  $x'(t + k)$  is then assumed to be

$$x'(t + k) = x(t) \quad k = 1, 2, \dots, n.$$

The persistence method can be used in weather quantities (irradiance  $G_h$ , wind speed) short time forecasting up to 3 h because the global trend of atmosphere is quasi-stationary with a frequency of  $1e - 4$  [56].

The forecasting method used only needs four physical quantities  $G_h$ , wind speed, atmospheric temperature and atmospheric pressure which are locally recorded by the meteorological station <sup>3</sup> of the Instituto Superior de Engenharia do Porto (ISEP) institute and recorded in the GECAD database. The database has a sampling frequency of  $3,3 \times 10^{-3}$  Hz (300 s).

#### 5.4.1 Photovoltaic



**Figure 5.11:** Recorded  $G_h$  of two consecutive days: a strong daily pattern is visible; the presence of an aleatory component is also observable.

If the atmospheric phenomena were excludable, it could be possible to determinate the time trend of the  $G_h$  with a perfect forecast based on the deterministic equations that describe the relative motion between Earth and Sun. In reality the actual atmospheric state cannot

<sup>3</sup>The meteorological instant data and their time elaborations are available at the web page <http://meteo.isep.ipp.pt/>.

be neglected so that the time trend of the ghi has an aleatory behavior but, unlike the wind speed, the trend has a strong daily pattern (see figure 5.11). The presence of the strong daily pattern gives the theoretical foundation to the method used in very small power system to forecast the photovoltaic production: the diurnal persistence [33]. The diurnal persistence method asserts that the forecasted value of the quantity of interest is equal to the value of the same quantity at the same time instant but of twenty-hour behind.

The implemented method fetches recorded data of  $G_h$ , ambient temperature, pressure and wind speed from the GECAD database starting from the time instant which is twice the length of the forecasting window ahead the actual time; using this strategy the last time instant of the fetched data is twenty-hour ahead the last forecasted time instant. The data of the four quantities are used as inputs of the generator model exposed in section 5.1 that gives as output the forecasted photovoltaic DC power which is used by the scheduling algorithm presented in chapter 3.

### 5.4.2 Wind

The last recorded value of the wind speed is used as input of the wind generator model presented in section 5.2 which gives as output the DC power  $P_{wind}$  available at the wind generator rectifier. The power  $P_{wind}(t+k)$  of the following time instants is then considered constant and equal to  $P_{wind}(t)$ . The high frequency of sample upload of the EMS (300 s), ensures that the average forecasted values do not diverge from the average real values.

In the selection of the model, another model was taken into consideration: the New Reference Forecast Model (NRFM). The NRFM was proposed in [56] as a replacement of the persistence model for benchmark tests because it is supposed to have better performance in forecast over 3 and it is the union of the persistence ( $x(t)$ ) and the mean of recorded values until time instant  $t$

$$x'(t+k) = a_k x(t) + (1 - a_k) \bar{x}$$

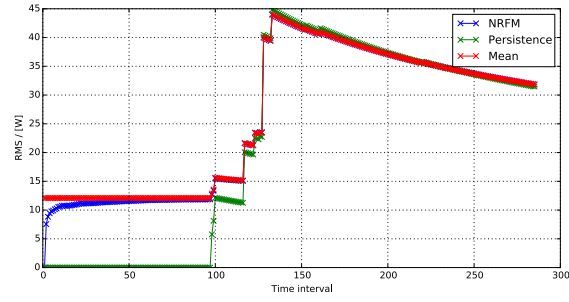
where  $a_k$  is correlation between  $x(t)$  and  $x(t+k)$

$$a_k = \frac{\frac{1}{N} \sum_{t=0}^{N-k-1} (x(t) - \bar{x})}{\frac{1}{N} \sum_{t=1}^{N-k-1} (x(t) - \bar{x})^2}.$$

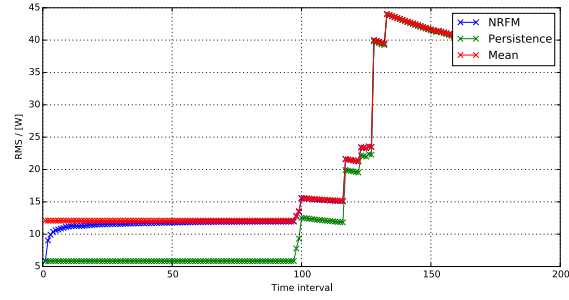
The RMS of the persistence, NRFM and mean methods are shown in figure 5.12 where the three methods are used to forecast a 24 h ahead wind power production. The persistence method has a lower RMS for the first time instants both for initial wind speed equal to the lowest of the time series and for initial wind speed equal to the mean of the time series. The results of the performed simulation are in accordance with the results in [28] (See Annex NRFM). Given the above results, the Persistence model was chosen.

### 5.4.3 Load

The absence of records for the available loads in the laboratory and the use of only load manually modified, allow only the use of a persistence model for which the future load is constant for each forecasted time instant and equal to the last recorded value.



(a) Initial wind speed  $v_{wind} = 0 \text{ m s}^{-1}$



(b) Initial wind speed  $v_{wind} = 4,8 \text{ m s}^{-1}$

**Figure 5.12:** Root Mean Square (RMS) of the persistence method, NRFM and mean method for two different initial speeds in a 24 h forecasting. The persistence method has a better for the initial time instants while, as expected, its performance degrades in the long run.



## Chapter 6

# Laboratory of Renewable Energy

The Laboratory of Renewable Energy of GECAD is a laboratory developed to study a micro-grid system based on local solar and wind Renewable Energy Source (RES) and technologies already commercialized.

The laboratory is located in the last floor of the F building (so that it is also called laboratory F) in the ISEP campus while the RES are situated in the rooftop of the building at  $41^{\circ}10'45''$  North and  $8^{\circ}36'28''$  West.

In the following sections the laboratory is described in its components, the practical work done is summarized and the implemented software related to the laboratory is presented.

### 6.1 Laboratory description



**Figure 6.1:** Panoramic photography of the GECAD Laboratory of Renewable Energy. On the left the inverters and batteries, in the middle the measures instruments, on the right the RES generators power converters and the loads power sockets.

The laboratory, shown in figure 6.1, is divisible into three layers: a power layer, a measure layer and a control layer. The first layer comprehends all the elements that exchange power as the RES generator, the power converters and the loads, the second layer is composed by the instruments that measure and store the electrical quantities of the power layer while the

last layer includes the devices that control the power layer, mainly the switchers both manual and remote controlled.

The electrical diagram of figure 6.2 is used as reference in order to have a clear vision of the laboratory.

### 6.1.1 Power layer

The power layer presents a structure in which the two photovoltaic generators (one fixed, one with a two-axis sun tracker) and a wind generator are connected to two electrochemical storage systems through power converters and then the storage systems are connected to the Alternate Current (AC) loads through different inverters.

#### Photovoltaic generation

The fixed photovoltaic generator, denominated in the electrical diagram (figure 6.2) as PV FIXED, is composed by the parallel of three multicrystalline silicon modules KC200GHT-2 produced by the German firm Kyocera. Each module has a rated power at STC of 200 W for a total rated power of 600 W.

The three panels, shown in figure 6.3, are mounted on a free standing rack with an azimuth angle of  $\alpha_{fixPV} = 8,0^\circ$  (West) and a tilt angle of  $\beta_{fixPV} = 40,0^\circ$ . The installation has also one Sunny SensorBox, linked with a PT100 module temperature sensor, which could be used to register the irradiation and the temperature of the modules.

The fixed photovoltaic generator converter (named `reg pv` in the electrical diagram) is a Steca TAROM 245 which is a hybrid charge regulator with a shunt linear PWM over voltage protection [79, p. 7] and a booster function control [79, p. 9].

The electric diagram of the charger in figure 6.4 shows the shunt configuration with the shunt MOSFET power switch (a) and the reverse shunt diode (b). When the batteries are reaching the full charge voltage, the controller of the charger starts changing the switching frequency of the MOSFET reducing the resistance of the shunt leg and thus creating a near short circuit at the clamp of photovoltaic panels. The current going to the battery is reduced gradually while the PV current reaches the short circuit value. The reverse diode prevents the batteries to discharge during the night when the photovoltaic modules act as resistors. This kind of configuration is it possible thanks to the fact that the PV modules have an intrinsic limit on current output, however the configuration is suitable only for system with photovoltaic current under 20 A [67, p. 728].

The controller is set with the **Voltage regulation mode** which uses the bulk voltage of the batteries as input value for all the controlling functions instead of the State Of Charge (SOC); that is necessary because there are other power converter linked to the battery bank.

In 6.13, the technical data of the Steca Tarom 245 are shown.

The second photovoltaic generator, denominated in the electrical diagram (figure 6.2) as PV TRACKER, is composed by two Kyocera KC200GHT-2 modules mounted on a two-axis

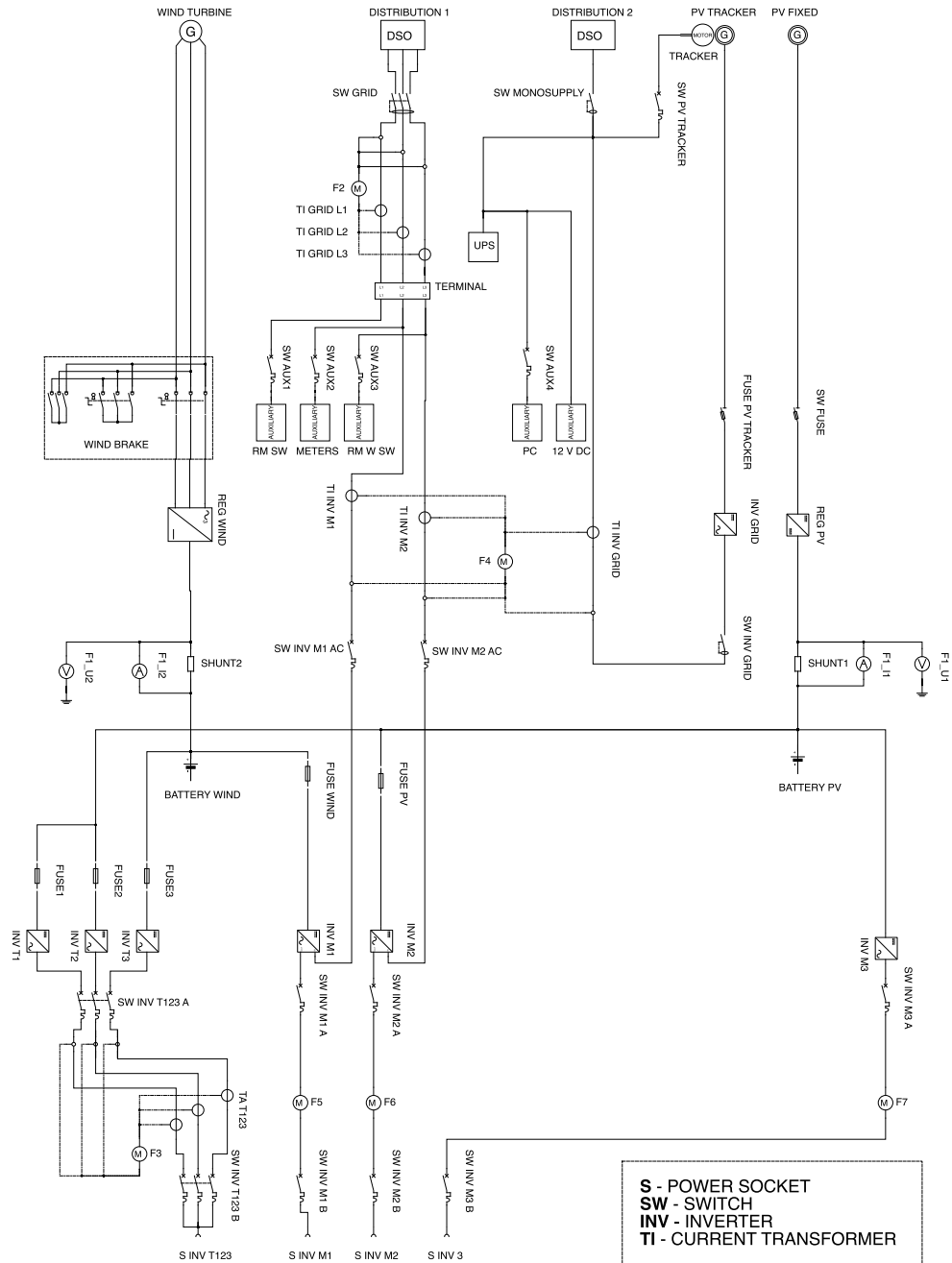
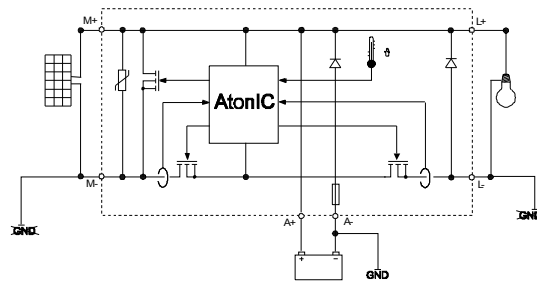


Figure 6.2: Electrical diagram of the GECAD laboratory.



**Figure 6.3:** Front (a) and back (b) side of the three fixed photovoltaic panels: free standing installation with no obstacles between panels and sun.



**Figure 6.4:** Electrical diagram of the Steca Tarom 245. Element a is the shunt MOSFET and b is the shunt reverse diode. Source [79, p. 21]

tracker system as it is possible to see in figure 6.5 and an other Sunny SensorBox (with PT100).

The converter of the tracking photovoltaic is a line-commutated [4, p. 3] inverter Sunny Boy SB 1100LV produced by SMA with a nominal power output of 1,00 kW.

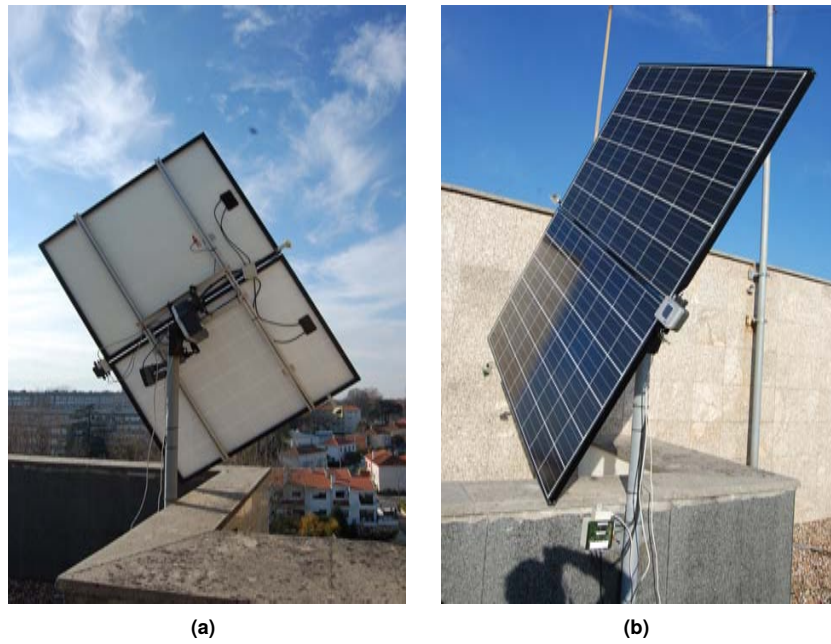
The inverter has an efficiency of  $\eta = 92\%$  with the curve shown in figure 6.6.

### Wind generation

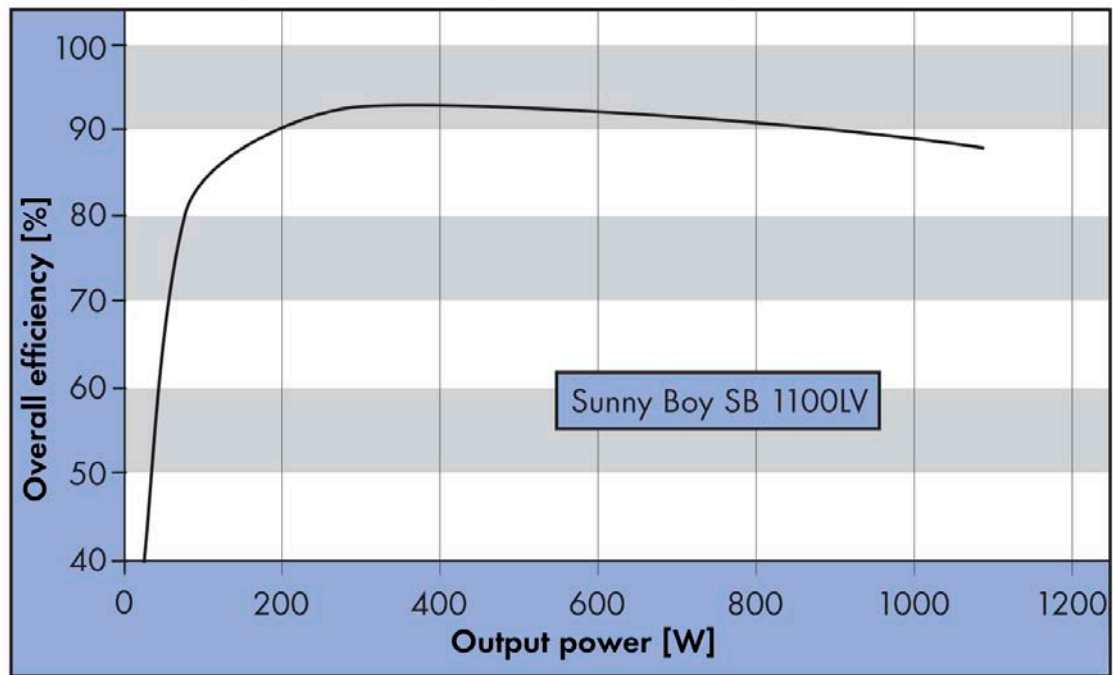
The wind generator, denominated in the electrical diagram (figure 6.2) as WIND TURBINE, is the Whisper 200 model of the Southwest Windpower, a small size up wind three blades turbine with a diameter of 2,7 m coupled with a three phase permanent magnetic alternator with a rated power of 1 kW at a wind speed of  $11,6 \text{ m s}^{-1}$ .

In figure 6.7 it is shown the GECAD wind generator which is the tallest one in the middle.

The generator has a wind speed operational range from  $3,1 \text{ m s}^{-1}$  of cut-in to  $18 \text{ m s}^{-1}$  of cut-off and a survival wind speed of  $55 \text{ m s}^{-1}$ . The wind generator converter *reg wind*, is composed by a three-phase full bridge rectifier followed by a step-down DC-DC converter



**Figure 6.5:** Photovoltaic generator PV TRACKER with two-axis tracker system.



**Figure 6.6:** Efficiency of tracking photovoltaic generator converter Sunny Boy 1100 LV as function of the AC power output. Source [3, p. 37].



**Figure 6.7:** Wind generator Whisper 200 location on the top of Building F, the GECAD generator is the tallest one in the middle.

with a controller which has a voltage and a current loop control on the load terminal [47]. The converter has a built-in dump load automatically used by the microprocessor by a PWM control.

The controller of the converter fully supplies the battery connected to the load side until the voltage at the battery clamps reaches the Dump load ON voltage when it starts supplies the dump load in a gradually way until the voltage reaches the Battery O/V when all the power is used by the dump load. The values of voltage set for the electrochemical battery bank are: Dump load ON 26,4 V and Battery O/V 28,4 V.

No data about the efficiency have been found but the power curve of the generator considers the power converter efficiency.

### DC buses and storage system

It seems reasonable to define the DC buses as the set of elements between the sources converters and the loads converters, thus it is formed by two battery banks and the cables (with their fuse protection) linking the power converters to them. Two different DC buses can be outlined: one related to the BATTERY WIND bank and the other to the BATTERY PV bank. The two buses have separate sources (BATTERY WIND fed by the WIND REG and the BATTERY PV fed by the REG PV) and they have different loads excluding the three phase load of S INV T123 which has two phases supplied by the PV bus and one by the WIND bus.

Each one of the two battery banks, shown in figure 6.8, are formed by twelve battery EXIDE



**Figure 6.8:** The two battery banks with the wind bank on the left and the pv bank on the right. A single bank is composed of 12 batteries with a capacity of 190 A h and a clamp voltage of 2 V. The red arrow points to the temperature sensor linked to the power converter INV T3.

Classic Solar 190 connected in series. The battery is a lead acid model with a nominal capacity of  $C_{120} = 190$  A h, clamp voltage of 2 V at 25 °C, a short circuit current  $I_{sc} = 1400$  A and a life of 2800 cycles at 60 % of DoD [24, p. 4]. The single bank has a nominal clamp voltage  $V_{bank, nom} = 24$  V and a capacity of  $C_{bank, 120} = 2280$  A h which is  $E_{bank, 120} = 54,72$  kW h for a constant voltage equal to the nominal value. The laboratory has two bidirectional power converters INV M1 and INV M2, one for each battery bank. They can operate as battery rechargers and as sinusoidal inverters and the two operation modes can be used contemporaneously allowing the recharging of the battery and the feeding of the AC loads from the DSO grid DISTRIBUTION 1.

The characteristics of the two inverters are described in 6.1.1.

### Loads inverters

The three power converters inv T1, inv T2 and inv T3, shown in figure 6.9, are of model Sunny Island 2224. They are bidirectional converter, thus they can operate as inverters from the battery banks to the AC bus and as battery but also as charger; the latter operation mode is not possible in the laboratory because the AC bus is composed of only loads, that is passive elements.



**Figure 6.9:** The three converters inv T1, inv T2 and inv T3 Sunny Island 2224 and the Sunny Remote Control on the top of inv T2.

The three converters are linked to both of the two DC buses, with INV T3 connected to battery bank *battery wind* while INV T1 and INV T2 are linked in to battery bank *battery pv* with a parallel connection in the central fuse box.

The technical characteristics of the Sunny Island 2224 are listed in 6.14.

The inverters are also connected between each other by CATe-FTP patch cables which establish a data connection necessary to operate the three inverters as a three-phase generator. The inverters are controlled by the Sunny Remote Control.

Inverter INV T3 has a temperature sensor in the battery bank *BATTERY PV* which is used to arrest the operation of the inverters if the temperature of the battery bank reach a set value.

The second group of power converters is composed by one inverter INV M3 and two inverter-chargers INV M1 and INV M2. As well shown in figure 6.2, while each power converter feeds one and only one single-phase AC bus, the INV M2 and INV M3 share the same DC bus formed by the battery bank *BATTERY PV*, instead the INV M1 is linked to the DC bus of *BATTERY WIND*. The two chargers are also connected to distribution grid *DISTRIBUTION 1*, INV M1 to phase L2 and INV M2 to phase L3 through two switches, thus they can be isolated from the grid operating only as inverters. The chargers, when connected to the grid, can simultaneously recharge the batteries and feeds the AC loads.

The inverter-charger INV M1 is a Studer Compact C-2600 of which characteristics are shown

in 6.15.

The inverter-charger INV M2 is a Studer XP-COMPACT 2200-24 of which characteristics are shown in 6.16.

The inverter INV M3 is a Studer SI 3324 of which characteristics are shown in 6.17.

## Loads

The electric load of the laboratory is a variable universal load unit of the Elettronica Veneta brand. The load, which can be used both with single-phase (230 V) and three-phase (400 V) configuration, has three sectors: a pure resistive sector with three resistors of 300 W for a total of 900 W, a pure capacitive sector with a total power load of 900 var divided between three equal capacitors, a pure inductive sector of three equal inductances for a total of 900 var. The three typologies of sectors can be connected to obtain the required load and each elements is adjustable with manual control knobs.

### 6.1.2 Data layer

The laboratory has a measure system which allows the analysis of the electrical quantities of the different subsets of the microgrid. The measures are both locally displayed by measure instruments and recorded in a central database allowing a remote visualization and post experiment data analysis.

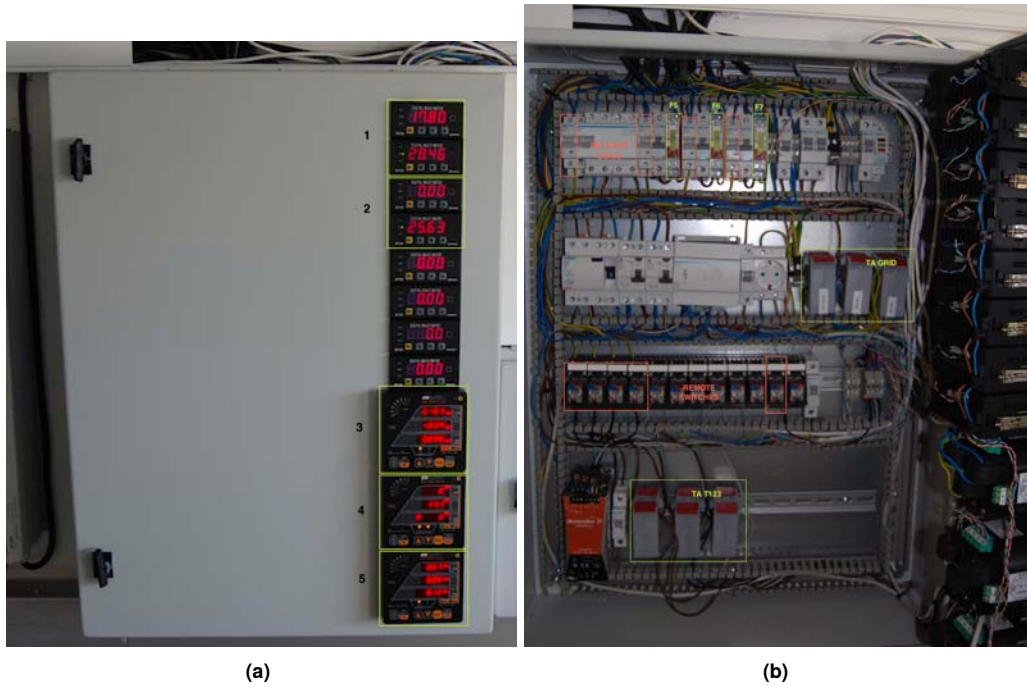
The measure system divides the electrical grid of the laboratory in four different parts and for each of one has a different set of quantities measured though the voltage and the current quantities are always measured. The four parts are: the two DC buses, the two links between the DSO and the microgrid, the three-phase AC bus and the three single-phase AC buses.

The two measured quantities of the DC buses are current and voltage. The measuring points are located in two junction boxes where the voltage probes are linked to the phase and ground while the current probes are connected to a shunt resistor placed in series with the phase cable. The probes are connected to the four Autonics MT4W voltage meters located in the front lid of the power panel as shown in figure 6.2: the group number 1 are the elements F1\_I1 and F1\_U1 in figure 6.2, group 2 are elements F1\_I2 and F1\_U2.

The meter F1\_I1 and F1\_I2 display only the voltage between the clamps of the two shunt resistors which have a linear characteristic for current values below 50 A with a slope factor of  $\frac{60 \text{ mV}}{50 \text{ A}}$ . The Programmable Logic Controller (PLC) does the math showing and recording the converted value of current.

The laboratory is connected to the main distribution grid of the department by two AC links: a three-phase one called DISTRIBUTION 1 and a single-phase one called DISTRIBUTION 2 in the electrical diagram 6.2.

The electrical quantities of the two links are measured by the two meters F2 (number 3 in figure 6.10) and F4 (number 4) which are two power meter SATEC PM 130 EH PLUS, respectively. While the meter F2 analyses the all electrical quantities of the grid below the



**Figure 6.10:** Measure instruments in the laboratory: outside and inside

three-phase general switch, thus includes the auxiliary loads as the remote switches RM SW, the F4 analyses only the links between the two DSOs and the three power converters INV M1, INV M2 and INV REDE. The voltage probes are directly linked to the two switches while the current probes are connected to the current transformers TI GRID and TI INV shown in figure 6.10.

The measured quantities for each phase are:

- Voltage V,
- Current A,
- Active power W,
- Reactive power var,
- Apparent power A,
- Power factor,
- Frequency Hz.

The SATEC PM 130 EH PLUS has a sampling rate of 128 sample/cycle and it communicates to the data logging system by a RS-485 port with has a Baud rate up to 115,2 kbps [71, pp. 135,136].

The three-phase AC bus is monitored by a SATEC PM 130 EH PLUS named F3 in the diagram 6.2 (number 4 in figure 6.10) which has the voltage probes linked to the output clamps of the switch SW INV T123 A while the current quantities are measured through three TI TI T123.

The three single-phase AC buses have a power analyzer CIRCUTOR CVM-1D each one labeled F5, F6 and F7 shown in figure 6.10.

The three power meters analyse the following quantities:

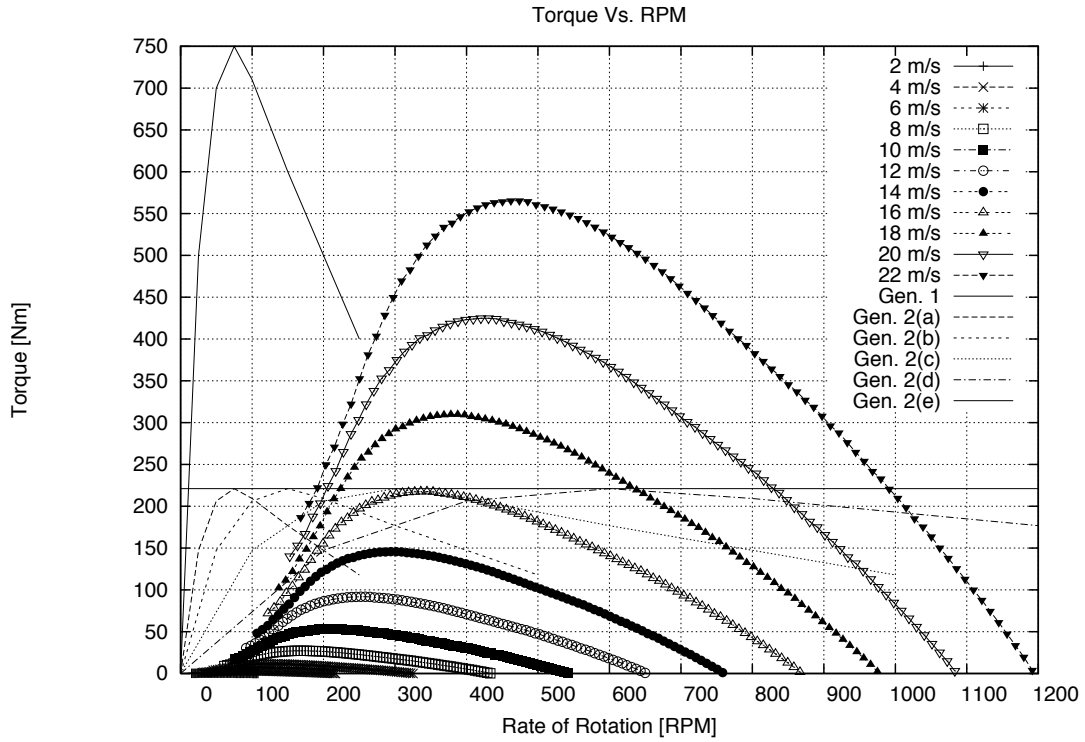
- Voltage V,
- Current A,
- Active power W,
- Reactive power (sum of capacitive and inductive) var,
- Inductive component of reactive power var,
- Capacitive component of reactive power var,
- Apparent power V A,
- Power factor,
- Active energy W h.

All the measuring instruments are connected to a communication board through a RS-485 via a Modbus/RTU protocol [16]. The logging system uses the software Movicon ®11 [64] to fetch the measured values from the instruments and save them on a Microsoft data base stored in the mainframe of GECAD Virtual Private Network (VPN).

### 6.1.3 Control layer

In the actual configuration the laboratory permits two kind of remote control of the components: the activation of the electro-dynamic wind brake wind brake and the curtailment of the loads.

The basis of the electro-dynamic wind brake is that short-circuiting the stator windings of the wind generator it is created a back-torque proportional to the rotational speed of the wind turbine. If the back-torque (negative value) is bigger than the wind turbine torque (positive value), the rotor decelerates. This technique has is not intrinsically secure so it is used only for small system; its limit is that while the back-torque pick for the electrical generator is unique and independent from the wind speed, the turbine torque pick is a function of the wind speed as shown in figure 6.11 thus, for wind speed higher than a specific value (16 in the figure) the turbine torque is higher than the short-circuit back-torque for every generator angular velocity thus the net torque is positive and the generator accelerates. The electro-dynamic brake is constitute of a manual switch and two remote-controlled switches (a normally open



**Figure 6.11:** Example of wind turbine torque and generator short-circuit back-torque as functions of the wind speed: it is clear that for high wind speed the back-torque is always smaller than the wind turbine torque so that the electro-dynamic brake is unusable. Source [55, p. 5].

and a normally closed), all the three switches are in parallel with the three-phase wiring between generator and converter; in order to deactivate the brake the manual switch must be open and a impulse signal must be given to the two remote controlled so that the normally closed switch opens and the normally open closed eliminating the windings short-circuit.

The loads curtailment is realized with release shunt coils that are opened by a current impulse and can be closed only manually.

Both the two operations are realized controlling through the PLC the remote switches (figure 6.10) but while for the former the switch activates two commutators of the wind brake to open the short circuit, for the latter the switches activate the release coils of the AC buses SW INV T123 A, SW INV M1 A, SW INV M2 A and SW INV M3 A.

## 6.2 Laboratory set-up

At the beginning of the present work, the Laboratory of Renewable Energy was not operative because since a semester before the GECAD group has been focus on the development of a new laboratory located in the N building of ISEP campus, leaving the laboratory of F building

not used. In order to the laboratory a preliminary work of fixing and implementation was required at each of the three layers afore mentioned.

The starting point was the realization of an electrical diagram 6.2 of the laboratory in order to understand firstly how to restore the connections between the elements of the laboratory and secondly which quantities where measured with the actual configuration of the measure layer. The drawing of the electrical diagram, done using the open source software QElectroTech [95], required multiple introspections of the power panel of figure 6.10 and a complete tear down of the laboratory cabling between the power panel and the other elements.

The following step was fixing of the power layer for which the connection of the wind generator and the wind power converter were restored using the already installed three-phase cable, the connection between the three-phase load and the S INV T123 socket was built through a new cable, finally the battery cells were refilled with distilled water, their acidity was inspected and a few discharge and recharge processes were done to check their functionality.

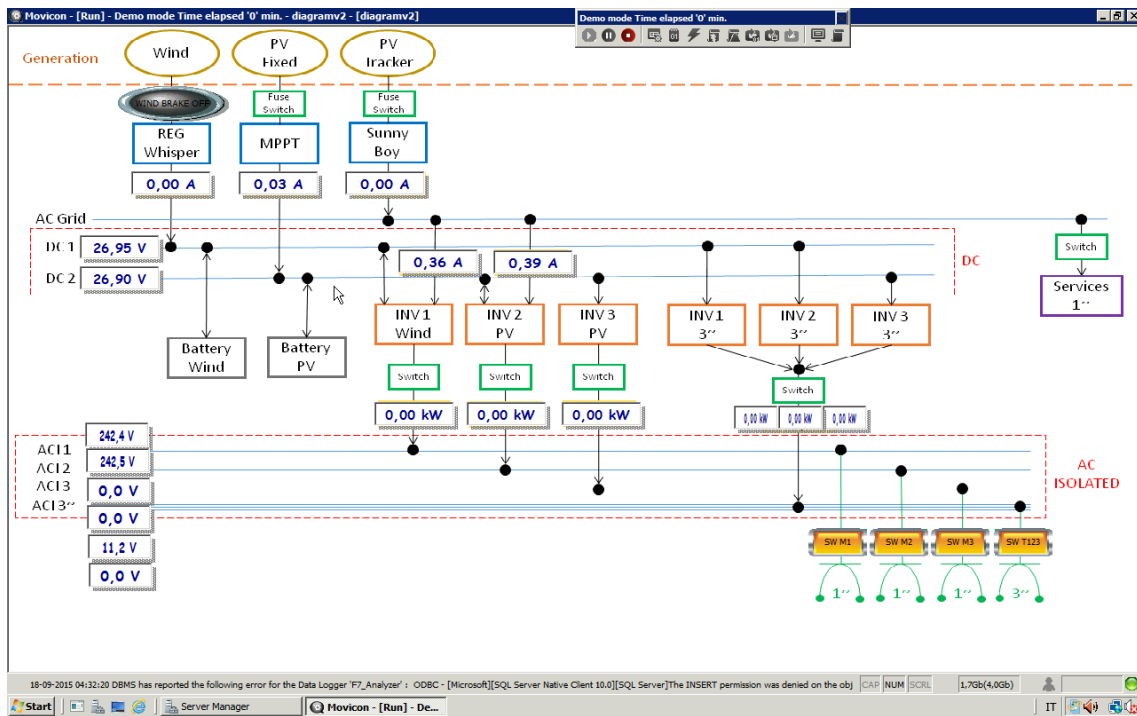
Completed the power layer set-up, the focus was shifted to the data layer which required, together with the wiring scheme drawing, the most of time dedicated to the set-up of the laboratory. Firstly two broken measure instruments, F1\_U2 and F1\_I2, were substituted with two new Autronics MT4W voltage meters, then the shunt resistor connected to the F1\_I1 was moved from the input clamps of the reg pv to its output clamps because in the old configuration the measured current was that one generated by the photovoltaic modules not the actual power in the DC bus so that when the reg pv stopped feeding the DC bus, the measure current had a value close to the short circuit current of the photovoltaic module as exposed in subsection 6.1.1. The connection of measure transformer (TI) TA T123 was redone because it the F3 was measuring a negative active power while the three-phase socket S INV T123 is connected to a load; it was discovered that the measure transformer clamps were inverted. Completed the local measures system, the remote control and data acquisition set-up started with a reconnection of the measure instruments data outputs to the PLC, then the PLC was programmed, with the XG5000 @software, allocating the memory registers for data inputs and writing the code use by PLC to convert the voltage measures of the F1\_I1 and F1\_I2 into current measures so that the currents values are directly stored. Finally the PLC was added to the GECAD VPN and a Microsoft @SQL Server database was created inside the GECAD data server located in the N building so that the data server fetches the measures data every ten seconds from the PLC and stores them in the database from where can be requested. The data layer was completed adding a local computer that can be used to program the PLC and to show the real time data acquired by means of the Movicon @Graphical User Interface (GUI) described in section 6.3, and a Uninterruptible Power Supply that feeds the measure instruments, the PLC and the computer in case of black-out of the main grid.

The set-up of the control layer primarily consisted in the implementation of the electro-dynamic wind brake described in subsection 6.1.3, programming the PLC to control both the wind brake and the already installed switches of load curtailment.

The las step of the laboratory preparation was the check of the configurations of all the power converters mainly the SOC memorized by the photovoltaic charger REG WIND and

the Sunny Remote Control, the end of charging voltage and the maximum exchanged currents between loads and DC buses.

### 6.3 Developed Software



**Figure 6.12:** GUI in Movicon ®11 to control the electrical values of the laboratory power layer and both to actuate the curtailment of the loads and deactivated the wind electrical brake.

The software related to the laboratory consists into two parts: a simple monitor and control GUI for the laboratory created with the Movicon ®11 software and a Python module (based on the Python packages NumPy [93], SciPy [39], Pandas [52] and pyodbc [42]) `gecad_tools` to both fetch electrical data from GECAD database and actuate the control layer in a programmable way.

The Movicon ®11 GUI, shown in figure 6.12, displays the principal electrical quantities of the three sections of the power layer: generators, DC bus and loads. For all the section the instant voltage is displayed, for the first two the current is displayed while for the loads section the active powers are shown. The GUI also has five buttons two use the control layer of the laboratory: the remote control of the electro-dynamic wind brake in the left-top corner and the four curtailment buttons for the loads in the right-low corner.

The Python module `gecad_tools` has the `electrical_data` function and the `Modbus` class. The function is used to get the data related to the electrical quantity of the laboratory which are post processed and elaborated in order to obtain meaningful data; its implementation is exposed in the **Annex electrical database function**. The `Modbus` class, after its initialization which consists in assigning a Modbus port number to a element of the system, can be used to open the remote switches and read their status.

Type of controller	235	245	440
System voltage	12/24V		48V
max. input voltage	48V		90V
Nominal module current at 20°C	35A	45A	40A
Nominal load current at 20°C	35A	45A	40A
max. current for 10s	45A	58A	52A
Surge current for 0,5s	56A	72A	64A
max. pulse current (10ms)	140A	180A	160A
Temp. range during operation	-10°C...60°C		
storage temperature	-25°C...80°C		
Connecting terminals	16/25mm <sup>2</sup>		
Weight	550g		
Dimensions	188x128x49mm		
Own consumption	14mA		
Type of protection	IP32		

**Figure 6.13:** Technical data of the fix pv converter Steca Tarom 245. Source [79, p. 21].

	SI 2012	SI 2224
<b>Output Values</b>		
Nominal AC voltage ( $U_{AC, nom}$ ) (adjustable)	230 V (202 to 253 V)	230 V (202 to 253 V)
Nominal frequency ( $f_{nom}$ )	50 Hz (45 to 65 Hz)	50 Hz (45 to 65 Hz)
Continuous AC output power ( $P_{nom}$ ) at 25 °C	2000 W	2200 W
Continuous AC output power ( $P_{nom}$ ) at 45 °C	1400 W (-30 %)	1600 W (-27 %)
AC output power for 30 min at 25 °C	2500 W	2900 W
AC output power for 5 min at 25 °C	3600 W	3800 W
AC output power for 1 min at 25 °C	3800 W	3800 W
Nominal AC current ( $I_{AC, nom}$ )	8.7 A	9.6 A
Max. stand-alone grid current (limitations based on hardware)	25 A <sub>peak</sub> (500 ms)	25 A <sub>peak</sub> (500 ms)
Max. stand-alone grid current (limitations based on software)	17 A <sub>eff</sub> (2.5 s)	17 A <sub>eff</sub> (2.5 s)
Harmonic distortion of output voltage ( $K_{VAC}$ )	< 4 %	< 4 %
Power factor ( $\cos \varphi$ )	-1 to +1	-1 to +1
<b>Input Values</b>		
Input voltage ( $U_{AC, ext}$ ) (adjustable)	230 V (172.5 to 264.5 V)	230 V (172.5 to 264.5 V)
Input frequency ( $f_{ext}$ ) (adjustable)	50 Hz (40 to 70 Hz)	50 Hz (40 to 70 Hz)
Max. AC input current ( $I_{AC, ext}$ ) (adjustable)	25 A	25 A
Max. input power ( $P_{AC, ext}$ )	5.75 kW	5.75 kW
<b>Battery Data</b>		
Battery voltage ( $U_{Bat, nom}$ ) (range)	12 V (8.4 V to 15.6 V)	24 V (16.8 to 31.5 V)

(a)

<b>Efficiency / Power absorbed</b>		
Max. efficiency	93.0 %	93.6 %
Internal consumption with no load (in standby mode)	6 W	6 W

(b)

**Figure 6.14:** Characteristics of the Sunny Island 2224. Source [4, p. 177].

Model	C 1600-12	C 2600-24	C 4000-48
<b>Inverter</b>			
Nominal battery voltage	12V	24V	48V
Input voltage range	9.5 - 17 V	19 - 34V	38 - 68V
Continuous power @ 25°C	1300VA	2300VA	3500VA
<b>Power 30 min. @ 25°C</b>	<b>1600VA</b>	<b>2600VA</b>	<b>4000VA</b>
Maximum power load 5 sec.	3 x Pnom		
Maximum load	up to short circuit		
Maximum asymmetric load	up to Pcont.		
Stand-by adjustment	1 to 25W		
Cos φ	0.1 - 1		
Maximum efficiency	94%	95%	
Consumption OFF/Stand-by/ON	0.5/0.6/6W	0.8/0.9/9W	1.2/1.4/12W
Output voltage	230Vac (- 10% / 0)		
Output frequency crystal controlled	50Hz +/- 0.05%		
Total harmonic distortion	< 2%		
Dynamic behaviour on load change 0 to 100%	0.5 ms		
Overload and short circuit protection	Automatic disconnection with 3 time restart attempt		
Overheat protection	Acoustic warning before shut-off - with automatic restart		
<b>Battery charger (4 STEP) I-U-Uo-Equalize (every 25 cycles)</b>			
Charging current adjustable	0 - 55A	0 - 55A	0 - 50A
Input current balance adjustment (Power Sharing)	1 - 16A		
Maximum input voltage	265Vac		
Minimum input voltage	Adjustable threshold from 150 to 230Vac		
Input frequency	45 - 65Hz		
Power Factor Correction (PFC)	EN 61000-3-2		

Figure 6.15: Technical characteristics of the inverter-charger INV M1 Studer Compact C-2600. Source[82, p. 29].

Model	XPC 1400-12	XPC 2200-24	XPC 2200-48
<b>Inverter</b>			
Nominal battery voltage	12V	24V	48V
Input voltage range	9.5 - 17V	19 - 34V	38 - 68V
Continuous power @ 25°C	1100VA	1600VA	1600VA
Power 30 min. @ 25°C	1400VA	2200VA	2200VA
Maximum power load 5 sec.	3 x Pnom		
Maximum load	up to short circuit		
Maximum asymmetric load	up to Pcont.		
Stand-by adjustment	1 to 25W		
Cos φ	0.1 - 1		
Maximum efficiency	94%	95%	
Consumption OFF/Stand-by/ON	0.5/0.6/4W	0.8/0.9/7W	1.2/1.3/7W
Output voltage	230Vac (- 10% / 0)		
Output frequency crystal controlled	50Hz +/- 0.05%		
Total harmonic distortion	< 4%	< 2%	
Dynamic behaviour on load change 0 to 100%	0.5 ms		
Overload and short circuit protection	Automatic disconnection with 3 time restart attempt		
Overheat protection	Acoustic warning before shut-off - with automatic restart		
<b>Battery charger (4 STEP) I-U-Uo-Equalize (every 25 cycles)</b>			
Charging current adjustable	0 - 45A	0 - 37A	0 - 20A
Input current balance adjustment (Power Sharing)	n.a.		
Maximum input voltage	265Vac		
Minimum input voltage	Adjustable threshold from 150 to 230Vac		
Input frequency	45 - 65Hz		
Power Factor Correction (PFC)	EN 61000-3-2		

Figure 6.16: Technical characteristics of the inverter-charger INV M2 Studer XP-COMPACT 2200-24. Source[83, p. 28].

<b>Model SI</b>	<b>612 624 648</b>	<b>812 824</b>	<b>1212 1224 1248</b>	<b>1624</b>	<b>2324 2348</b>	<b>3324</b>	<b>3548</b>
Input voltage (Unom) [V]	12/24/48	12/24	12/24/48	24	24V/48	24	48
Nominal power [W]	600	800	1200	1600	2300	3300	3500
« Standby » current [mA] Power « ON » no load [W]	25/21/10 2.6	25/21 2.8	25/21/12 4.8	21 5.8	25/17 9	25 13	30 17
Power « ON » no load [W] <i>TWINPOWER</i> system	-----	-----	< 0.5	< 0.5	< 0.6	< 0.7	< 0.8
Maximum efficiency [%]	91	92	93 - 95	93 - 95	95	95	95
Length L x 124 (H) x 215 (W) [mm]	276	276	391	391	591	636	791
Weight [kg]	6.9	10.4	13.2	15.2	27	30	38

**Figure 6.17:** Technical characteristics of the inverter INV M3 Studer SI 3324. Source[81, p. 28].

## Chapter 7

# Experiment test

The following scenario, performed with the testbed of the GECAD Laboratory of Renewable Energy, is used to test the final implementation of the developed EMS.

Given the experimental resources of the laboratory, the scenario considers only a curtailable load and consists in a few daily hours of EMS usage split between a morning period and a afternoon period. During the two periods of experiment, a variable load profile is manually set at different power levels to analyze the response of the system.

### 7.1 Experiment set up

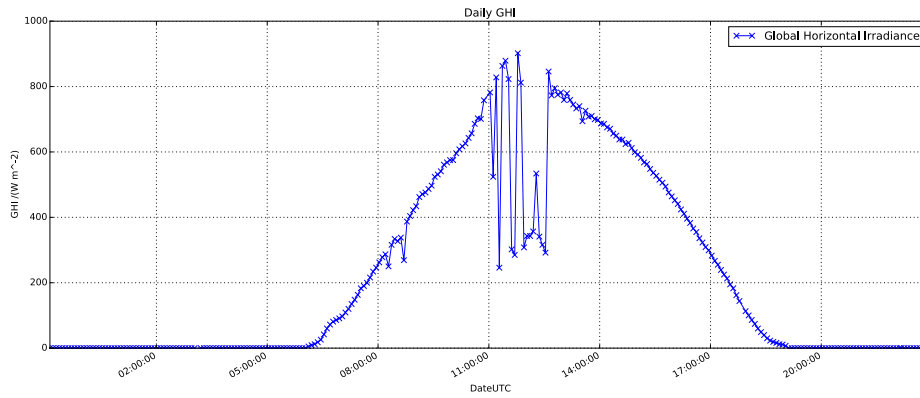
The laboratory is configured as describe in the following with reference to the electrical diagram of the laboratory in figure 6.2:

- The islanded state is obtained opening the switches SW INV M1 AC and SW INV M2 AC so that the two battery rechargers INV M1 and INV M2 cannot feed the DC buses,
- the two electrochemical storage BATTERY PV and BATTERY WIND,
- the wind generator is activated switching on the REG PV and switching off both the manual and remote part of the electrodynamic brake WIND BRAKE,
- the three-phase manual load described in 6.1.1 is connected to the power socket S INV 3 and both the aforementioned switch and the SW INV T123 A are closed,
- the three inverters Sunny Island 2224 INV T1, INV T2 and INV T3 are switched on and put in stand-by state,
- the forecasting window of the EMS has a total length of 24 h divided into six intervals of 5 min, six intervals of 15 min, six intervals of 30 min and nineteen intervals of 60 min,

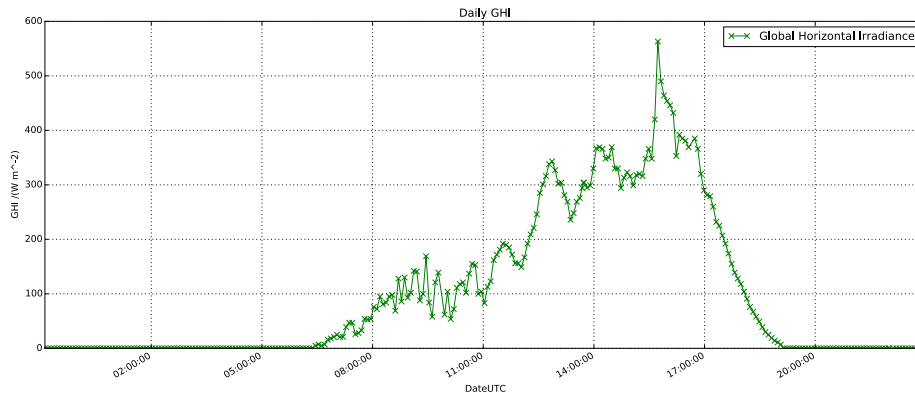
- the update time is set at 300 s,
- the developed software code is run on a Apple ®MacBook5,2 with a Intel Core 2 Duo processor and 4 GB of RAM, connected to the GECAD VPN.

## 7.2 Weather data

In order to have a complete view of the Renewable Energy Source available during the experiment the weather data are presented for the entire day even if only a part of them are actually used by the EMS.



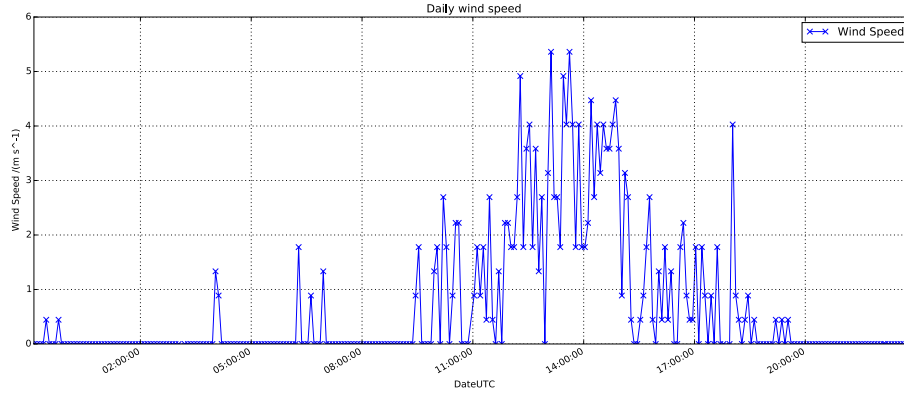
(a) Experiment day



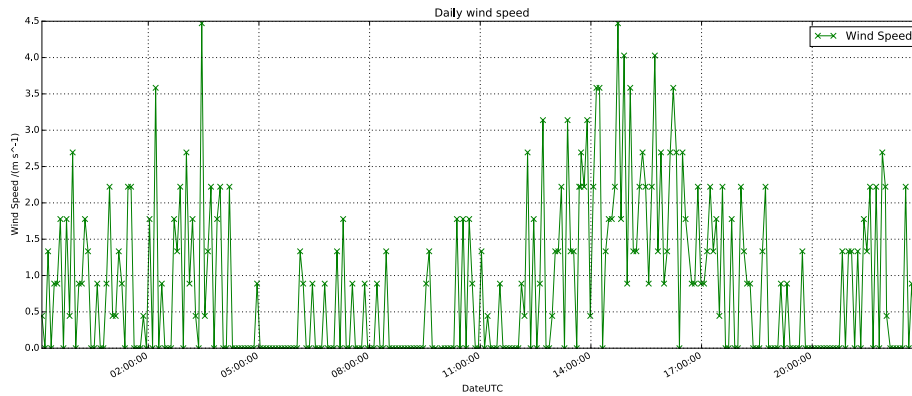
(b) Day before

**Figure 7.1:** The daily  $G_h$  for the day of the experiment (a) and the day before (b): the first one is what the photovoltaic generator can really use while the second is what the EMS uses as input of the forecasting unit.

The  $G_h$  of the day of interest and the day before are shown in figure 7.1, the first quantity is what the photovoltaic generator can really convert while the second is what the EMS uses as input of the forecasting unit.



(a) Experiment day

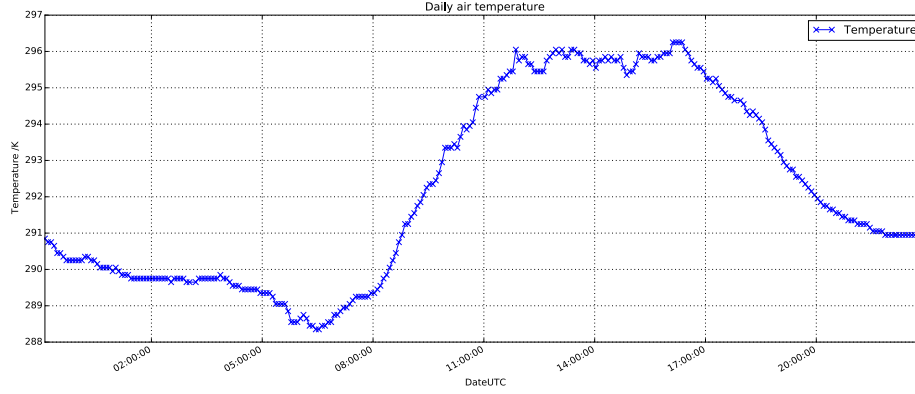


(b) Day before

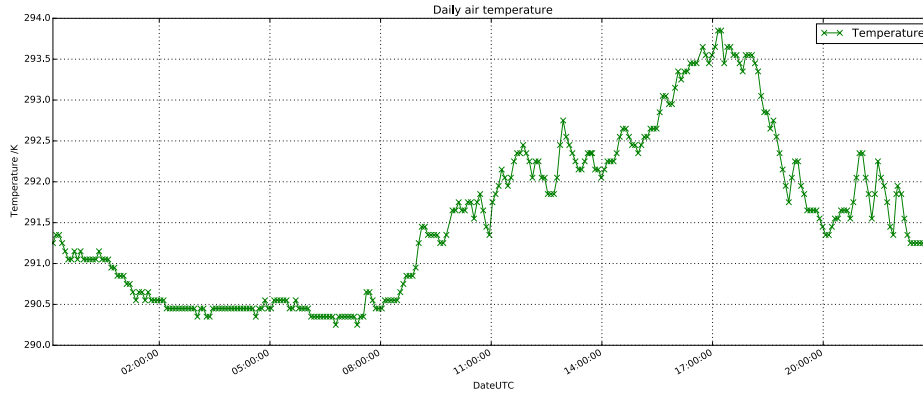
**Figure 7.2:** The wind speed in the day of the experiment (a) and in day before (b)

A lower value and a more variable trend found for the day before  $G_h$  can be seen also in the wind (figure 7.2) and temperature (figure 7.3) data suggesting that the day before the atmospheric state was more cloudy.

The lower values of the  $G_h$  data for the day before the experiment leads to an underestimation of the real photovoltaic potential for the day of experiment; in fact assuming that a perfected forecast of the weather values were possible and using both the day before data and the current day data as input of the photovoltaic model, a comparison of the two output DC power can be done as shown in figure 7.4.



(a) Experiment day



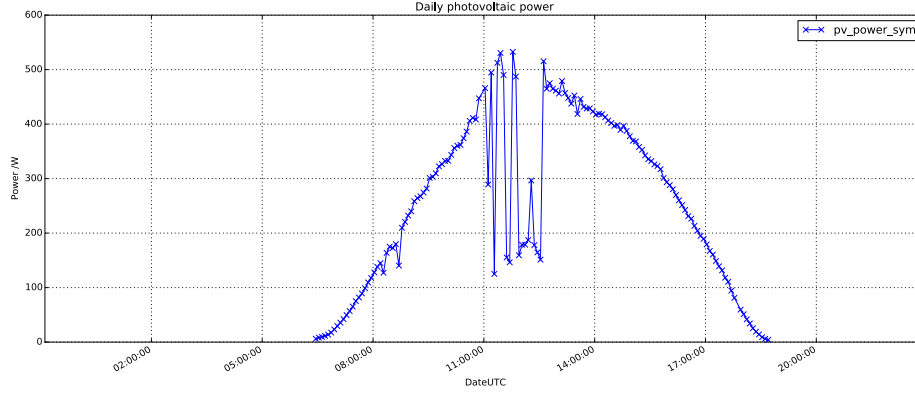
(b) Day before

**Figure 7.3:** Air temperature in the day of the experiment (a) and in day before (b)

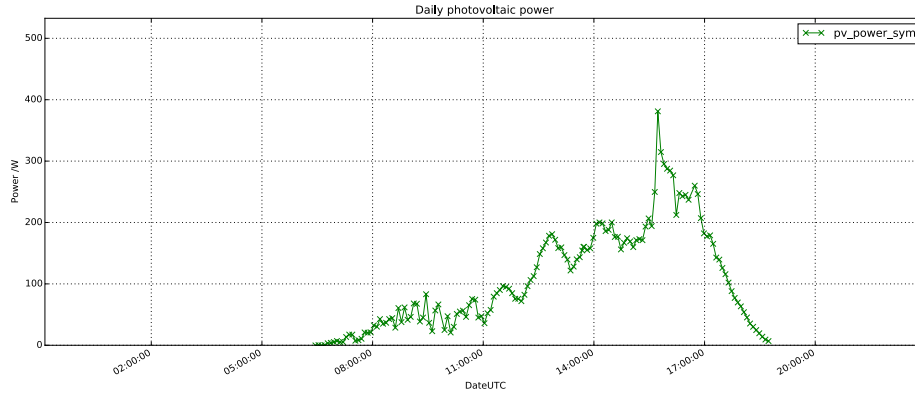
### 7.3 Electrical data

The analysis of the results of the EMS required a previous knowledge of the power exchange that actually happens in microgrid so that a comparison between actual data and the EMS data is meaningful. For this section it necessary to highlight that between the 10:30 Coordinated Universal Time (UTC) and the 13:30 UTC no data are recorded due to a problem in the ISEP LAN that cuts out the laboratory net from the outside net; the lack of data doesn't invalidate the experiment because the analysis is conduct on the morning section before the LAN black-out and the afternoon section after the net connectivity is restored.

The first quantities are the load demand power profile is shown in figure 7.5. The three phases of the manual load are changed in synchronous, left in the new configuration for a variable time interval and them quickly changed again to obtain a stepped profile. The plot shows the



(a) Perfect forecast



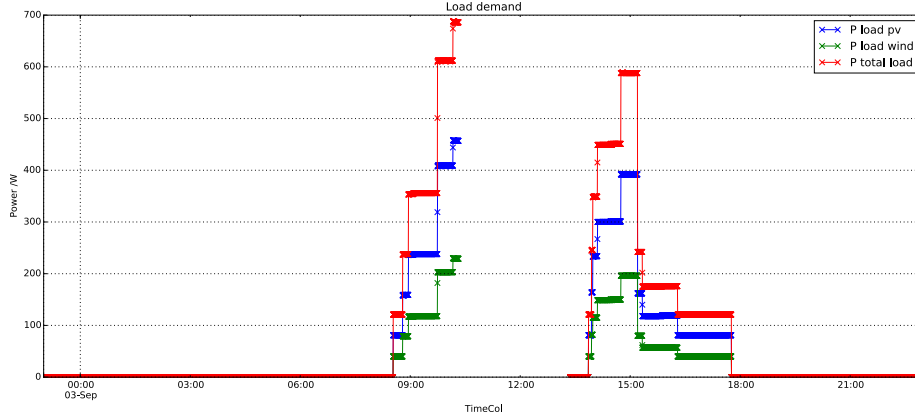
(b) Diurnal forecast

**Figure 7.4:** Comparison of the output of the photovoltaic model for a perfect  $G_h$  forecast and a diurnal forecast. In this case the diurnal forecasting method underestimates the DC output of the photovoltaic system.

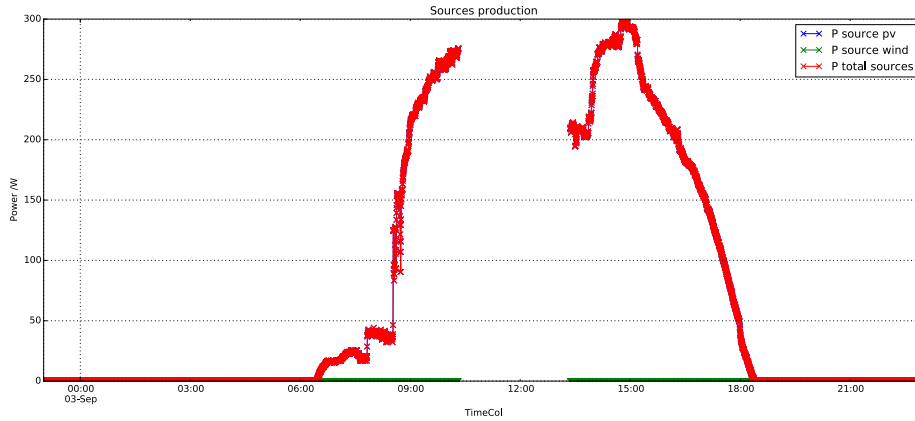
total load profile and its two components: the power extracted from the photovoltaic DC bus and the power demanded to the wind bus; the first bus feeds the two inverters INV T1 and INV T2 while the third inverter INV T3 is connected to the wind bus, thus the instant power on the photovoltaic bus is twice the power of the wind bus. The total load has maximum of constant 688 W.

The second power flux of interest is the RES sources production visible in figure 7.6. During the experiment day the only effective source is the photovoltaic generator because, although the wind has some gusts with a speed higher than the cut-in speed of the generator, they probably have a time duration shorter than the generator start up dynamic.

For last the power exchanged at the storage systems clamps is displayed in figure 7.7

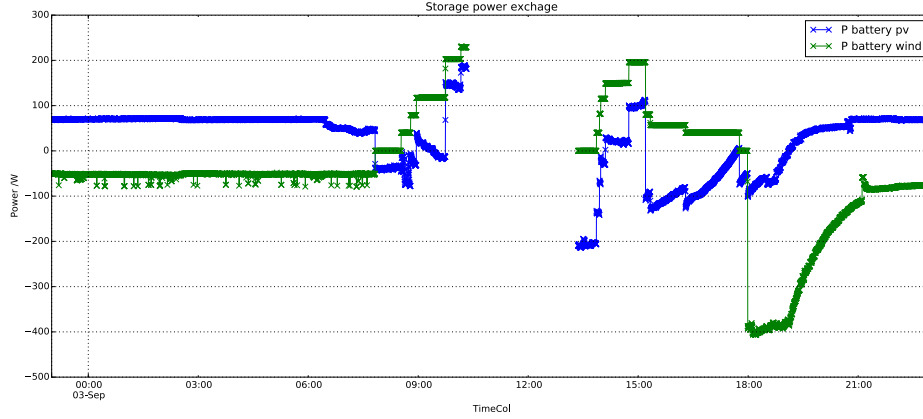


**Figure 7.5:** Load profile: photovoltaic DC bus component, wind DC bus component and their sum. The different distribution of the three-phase load in the two buses is clearly seen.



**Figure 7.6:** Daily generation of the two RES sources of the microgrid: during the experiment day no actual wind power was obtain because of low wind speed.

Considering only the value in the time interval of the experiment, the two powers show the different supplies of the respective generator: the wind plot follow the load demand on the wind DC bus because no power recharges the storage so that the plot has only positive values (generator convention); the photovoltaic plot clearly shows the photovoltaic source influence both for the negative values and for the not strictly stepped profile. The power exchange before and after the experiment time interval are due to the two battery rechargers: the wind storage system has an higher recharge power after the conclusion of the experiment because of the lack of wind generation during the load connection.



**Figure 7.7:** Power exchanged at the clamps of the photovoltaic and of the wind storage systems. A positive power goes out the storage system while a negative value indicates a charging process.

## 7.4 EMS output

The two sets of data for which the EMS optimization is performed are from 8:30 UTC to 10:30 UTC and from 13:55 UTC and 17:50 UTC. For each run of the EMS a plot of the forecasted scheduling is saved, in the following the most relevant plots are commented.

### 7.4.1 Consecutive constant loads

The first three consecutive EMS scheduling at 8:37 UTC, 8:42 UTC and 8:47 UTC are shown in figure 7.8. For each plot the initial time interval correspond to the time of execution of the scheduling, so that the second time interval of the first plot corresponds to the first interval of the second plot.

The total load has a constant value of 120 W as shown in figure 7.5 while the forecasted source power is of about 50 W which is the photovoltaic forecasted power. Because of the power deficit of the sources the storage system are used. The sequences of plots shows that for a constant load the scheduling for the ahead time interval remains almost constant and the forecasted photovoltaic generation shifts to the left. It is worth to note that the forecasted source power for the photovoltaic generation in the first time interval of scheduling is significantly lower than the real power (see figure 7.4) so that in reality the source is able to feed the load and even recharge the battery (see figure 7.7).

### 7.4.2 Abrupt variation of load

At time instant 9:03 UTC, the load abruptly changes reaching the value of about 350 W. As shown in figure 7.9, the EMS adapts the scheduling to the new state of the microgrid with an increment of the storage output. Because of the persistence method used in the load

forecasting, the abrupt change is propagated in the future time interval optimization with an overall increase of the storage component which is supposed to undergo a deep discharge so that in the last time intervals the EMS schedules a curtailment of the load.

At time interval 9:48 UTC the load has another abrupt increment reaching the 600 W, the EMS again increases the storage usage so that the storage system reaches the discharged status faster and thus the load curtailment is anticipated.

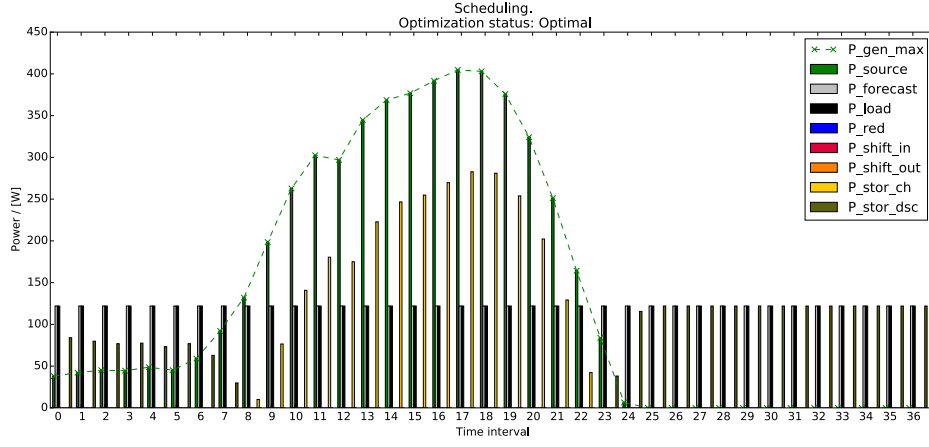
### 7.4.3 Soc estimator

In figure 7.10 the estimate soc for the time instant 9:03 UTC (axis=1) and both the previous and next time instants are shown. At the time instant the load changes abruptly, it is visible that the wind SOC estimator has an error and estimate a SOC close to 10 while the maximum value is 1. The error could be given by the abrupt variation in the load demand that suddenly increases the power required by the battery of the wind DC. The photovoltaic battery does not encounter the same problem because the photovoltaic generator output is not null as for the wind generator.

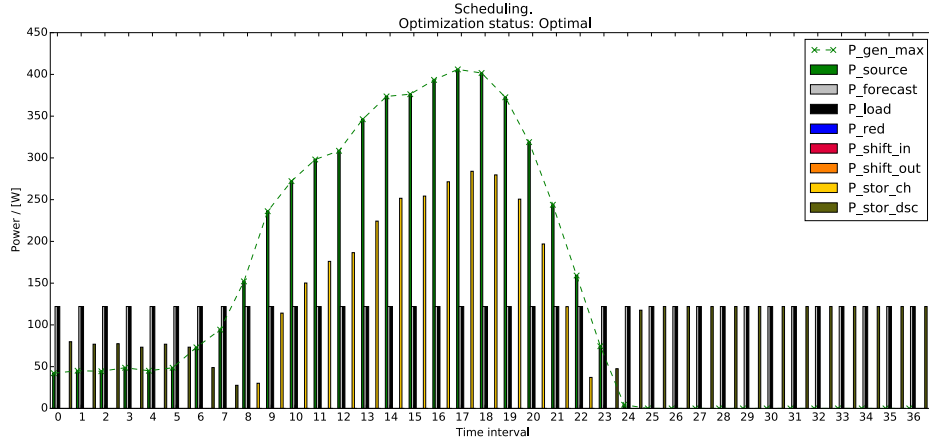
In the next time instant (axis=2) the soc estimator adjusts the value because the load is constant. This leads to the conclusion that the transient part of the battery model required a better tuning of its parameters or a more robust algorithm.

### 7.4.4 Variation of the forecasted wind generation

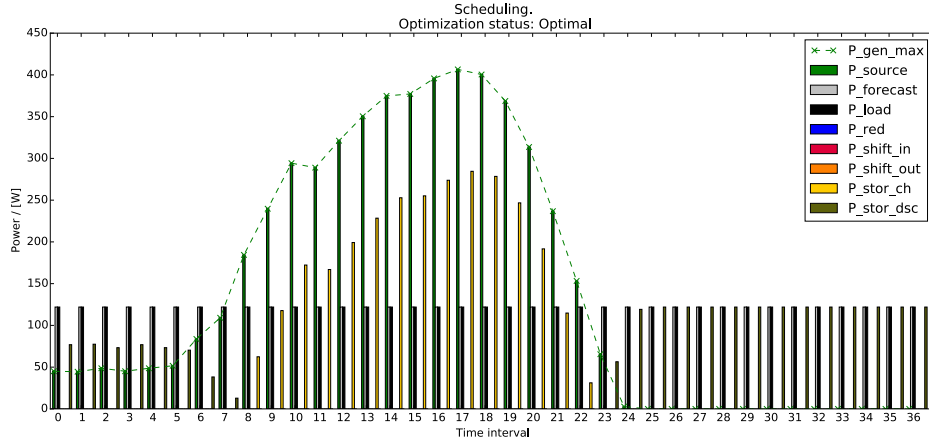
The this case it is shown how the rolling horizon allows to use a simple forecasting method as the persistence. In figure 7.11 the sequential runs of the EMS are displayed for the time interval between 14:05 UTC and 14:15 UTC. At time 14:10 UTC the forecasted source power almost double its values so that the charging power is triplicated and the load curtailment in the future intervals canceled. It is evident that the change is given by the variation in the wind forecasting, in fact during the night hours in which no source power was forecasted in the previous run, now it presents a constant source power of about 350 W. This variation in the forecasted value is caused by the persistence method used to forecast the wind generation: at time 14:10 UTC the wind forecast system uses one of the last sampled wind speed higher than  $5 \text{ m s}^{-1}$ , superior to the cut-in velocity, (see figure 7.2) so the simple wind generator model outputs a not negative power. In reality the power generator does not produce any power because of the start up dynamic (as seen in figure 7.6). Given the higher frequency of the lower wind speed, in the next run (time 14:14 UTC) the wind forecasted adjusts the output of the wind generator to zero, and consequently the EMS resets the future scheduling to the scheduling antecedent the erroneous wind forecasting. This result shows that even if the wind generator model and the wind forecast method are very simple, the overall functionality of the EMS is not compromised by forecast errors.



(a) Time of scheduling run: 8:37 UTC

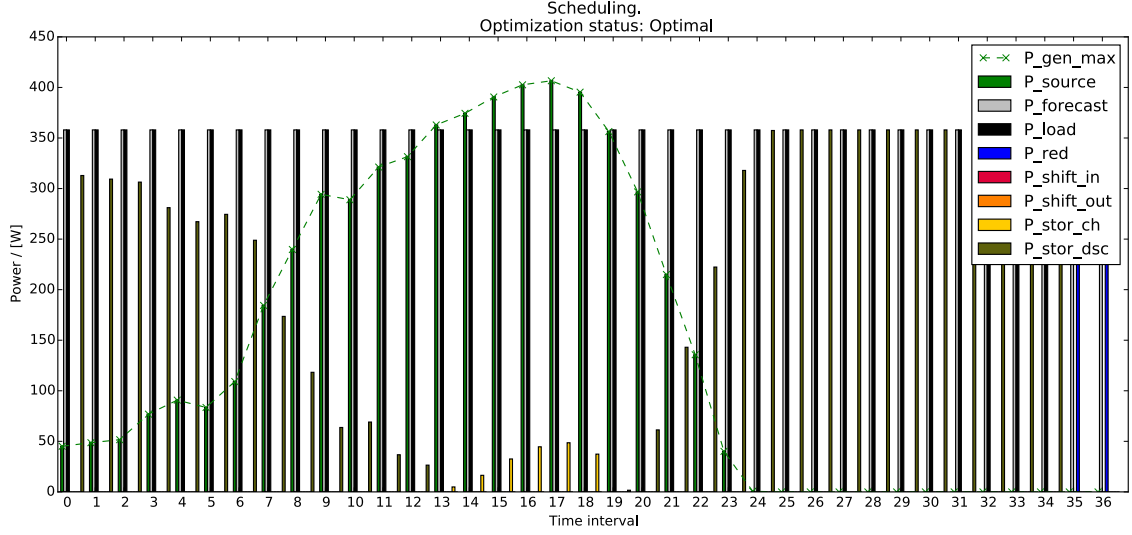


(b) Time of scheduling run: 8:42 UTC

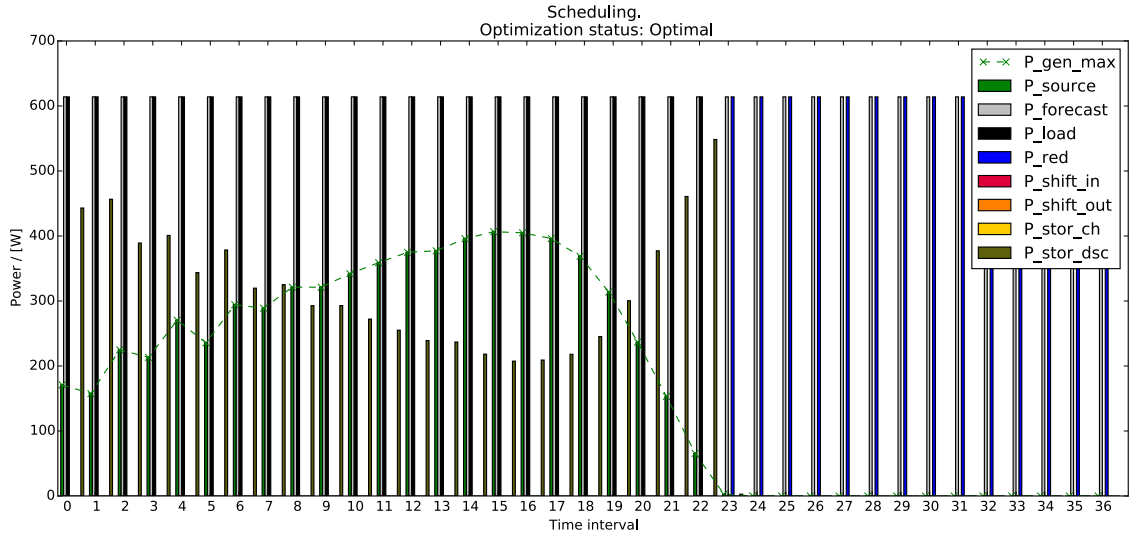


(c) Time of scheduling run: 8:47 UTC

Figure 7.8: First three scheduling of the EMS with constant load at 120 W

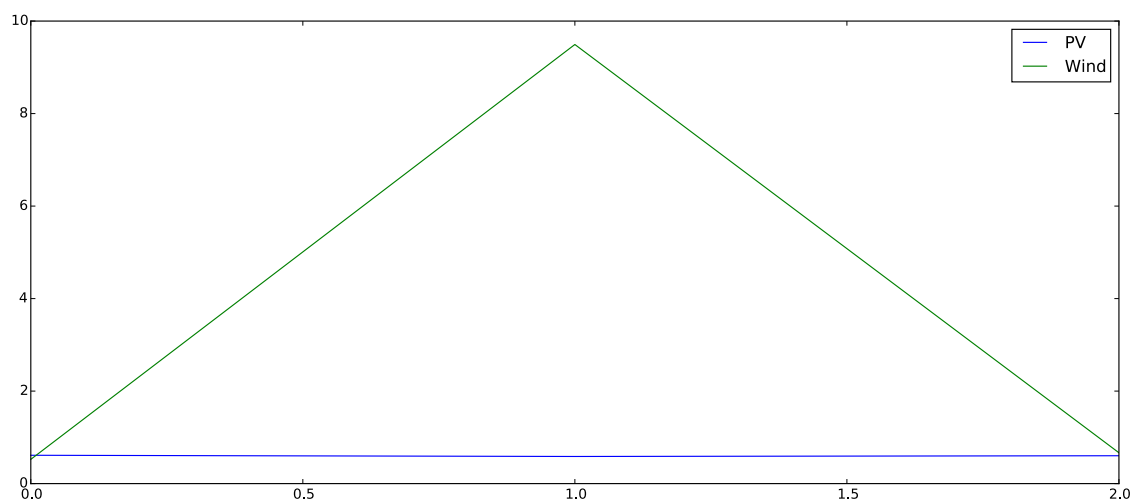


(a) Time of scheduling run: 9:03 UTC

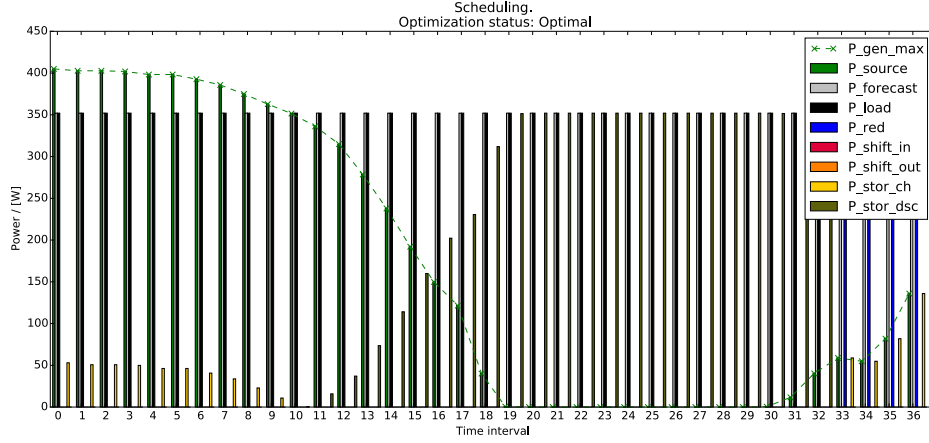


(b) Time of scheduling run: 9:48 UTC

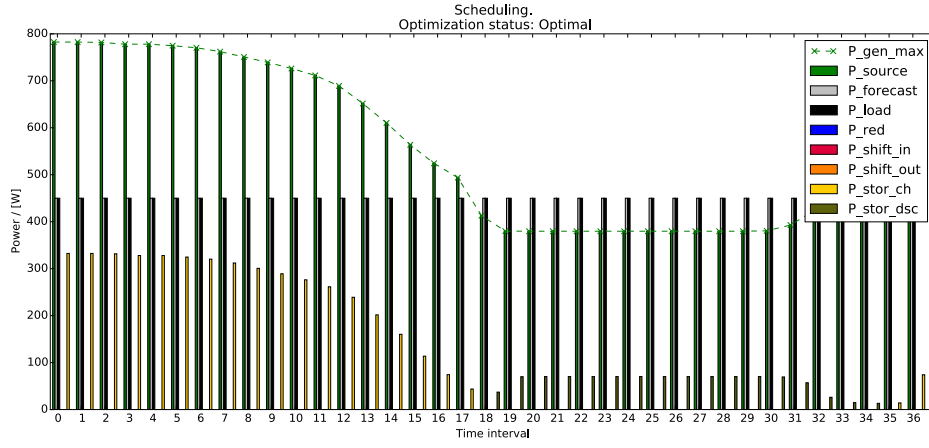
**Figure 7.9:** Abrupt changes in load demand, the scheduling is adapted to the new state of microgrid.



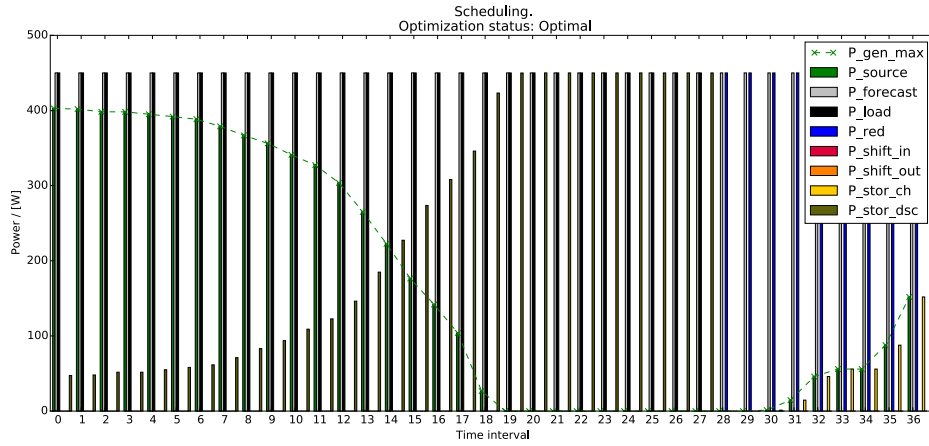
**Figure 7.10:** Error in the evaluation of the SOC for the wind battery due to an abrupt change in the load demand at 9:03 UTC.



(a) Time of scheduling run: 14:05 UTC



(b) Time of scheduling run: 14:10 UTC



(c) Time of scheduling run: 14:15 UTC

**Figure 7.11:** Temporary variation of the scheduling due to a change in the forecasted wind generation based on the persistence method.

## Chapter 8

# Conclusions

In the previous chapters, a simple Energy Management System (EMS) for a small isolated microgrid was developed and tested to understand the main problems of a EMS with a global overview and not just a focus on a small section of the system, thus a deeper analysis of the single fields touched in the present work seems to be the logical continuation. The extension of the work can be divided into four stages with an increasing complexity starting from the improvement of the present system and reaching an expansion to other elements of a microgrid.

The first stage consists only in a more detailed verification of the present system. In fact the results of the test presented in chapter 7, even if the EMS is capable of managing the microgrid scheduling, reveals a weakness: the SOC estimator does not work properly with abrupt variation in power exchange with the storage system. The weakness could be given by the storage model and the estimation method. Considering that they were successfully used in the original authors works, more likely it is given by errors in measures used to tune the model parameters and to estimate the SOC in real time (see section 4.5) because the laboratory set-up did not allowed direct measure of all the input and output electrical quantities. A first step would be to use the EMS in a laboratory with a better measure set-up; then in case of persistence of the weakness, a second step would be the tuning of the model parameter not using the microgrid operational data but using specific method as described in [6, 35, 18]. A last step of this stage would be a long run test that spans over a few days of microgrid operation.

The second stage is an enhancement of the present system that should delete the second weakness of the actual EMS which is the wind generator forecasting system. This weakness is mainly given by the wind generator model which does not consider the dynamic inertia of the wind turbine overestimating the real energy yield and in minimal part by the persistence forecasting method. Some dynamic models for WECS are present in literature as [75, 31, 7] but all of them require information about mechanic, electric and magnetic proprieties of the WECS that are not usually given by the manufactures thus they must be obtained by laboratory analysis. After the upgrade of the wind generator model, always keeping in mind the balance between complexity and improvement as pointed out in section 5.4, an improvement of the forecasting method can be taken into consideration.

The third stage consists in an expansion of the present work. This is driven by the consideration that the economical and technical optimal operation of a microgrid is obtain in system with a combined generation of thermal and electrical power as pointed out in section 1.2. In order to obtain a EMS of an optimal microgrid, models of cogeneration and trigeneration units and their integration in the already developed EMS can be considered as in [87, 76, 61, 32] with the addition other types of electrical and thermal energy storage systems.

The fourth and last stage would be firstly a change in the size of the microgrid with the addition of more sources and loads and eliminating the simplification imposed in section 3.4 mainly with the introduction of power losses and physical constrains of the electrical network.

# Bibliography

- [1] Suleiman Abu-Sharkh and Dennis Doerffel. “Rapid test and non-linear model characterisation of solid-state lithium-ion batteries”. In: *Journal of Power Sources* 130.1–2 (2004), pp. 266–274. ISSN: 0378-7753. DOI: <http://dx.doi.org/10.1016/j.jpowsour.2003.12.001>. URL: <http://www.sciencedirect.com/science/article/pii/S0378775303011455>.
- [2] Thomas Ackermann, Göran Andersson, and Lennart Söder. “Distributed generation: a definition”. In: *Electric Power Systems Research* 57.3 (2001), pp. 195–204. ISSN: 0378-7796. DOI: [http://dx.doi.org/10.1016/S0378-7796\(01\)00101-8](http://dx.doi.org/10.1016/S0378-7796(01)00101-8). URL: <http://www.sciencedirect.com/science/article/pii/S0378779601001018>.
- [3] SMA Solar Technology AG. *Sunny Boy SB 1100LV Inverter for Photovoltaic Plants Installation Guide*. 2006.
- [4] SMA Solar Technology AG. *Type Test Certification Test Result Sheet - SUNNY BOY 1100LV and WINDY BOY 1100LV*. 2009. URL: [http://files.sma.de/dl/22485/ZERT\\_IR\\_SBWB1100LV-ZEN094011.pdf](http://files.sma.de/dl/22485/ZERT_IR_SBWB1100LV-ZEN094011.pdf) (visitato il 2015-02-19).
- [5] Carla Alvial-Palavicino et al. “A methodology for community engagement in the introduction of renewable based smart microgrid”. In: *Energy for Sustainable Development* 15.3 (2011). Special issue on off-grid electrification in developing countries, pp. 314–323. ISSN: 0973-0826. DOI: <http://dx.doi.org/10.1016/j.esd.2011.06.007>. URL: <http://www.sciencedirect.com/science/article/pii/S0973082611000469>.
- [6] G. Aurilio et al. “A battery equivalent-circuit model and an advanced technique for parameter estimation”. In: *Instrumentation and Measurement Technology Conference (I2MTC), 2015 IEEE International*. 2015-05, pp. 1705–1710. DOI: 10.1109/I2MTC.2015.7151537.
- [7] Binayak Bhandari et al. “Mathematical modeling of hybrid renewable energy system: A review on small hydro-solar-wind power generation”. In: *international journal of precision engineering and manufacturing-green technology* 1.2 (2014), pp. 157–173.

- [8] Valerio Lo Brano et al. "An improved five-parameter model for photovoltaic modules". In: *Solar Energy Materials and Solar Cells* 94.8 (2010). National Conference on the Emerging Trends in the Photovoltaic Energy and Utilization, pp. 1358–1370. ISSN: 0927-0248. DOI: <http://dx.doi.org/10.1016/j.solmat.2010.04.003>. URL: <http://www.sciencedirect.com/science/article/pii/S0927024810001686>.
- [9] R.J. Brodd. "{SECONDARY} {BATTERIES} | Overview". In: *Encyclopedia of Electrochemical Power Sources*. Ed. by Editor-in-Chief: Jürgen Garche. Amsterdam: Elsevier, 2009, pp. 254–261. ISBN: 978-0-444-52745-5. DOI: <http://dx.doi.org/10.1016/B978-044452745-5.00125-8>. URL: <http://www.sciencedirect.com/science/article/pii/B9780444527455001258>.
- [10] Tony Burton. *Wind energy handbook*. Chichester, West Sussex: Wiley, 2011. ISBN: 978-1-119-99272-1.
- [11] B. Canizes et al. "Resource scheduling in residential microgrids considering energy selling to external players". In: *Power Systems Conference (PSC), 2015 Clemson University*. 2015-03, pp. 1–7. DOI: 10.1109/PSC.2015.7101700.
- [12] Wen-Yeau Chang. "The State of Charge Estimating Methods for Battery: A Review". In: *ISRN Applied Mathematics* (2013), p. 7. URL: <http://dx.doi.org/10.1155/2013/953792%20%J%20953792>.
- [13] Min Chen and G.A. Rincon-Mora. "Accurate electrical battery model capable of predicting runtime and I-V performance". In: *Energy Conversion, IEEE Transactions on* 21.2 (2006-06), pp. 504–511. ISSN: 0885-8969. DOI: 10.1109/TEC.2006.874229.
- [14] Gianfranco Chicco and Pierluigi Mancarella. "Distributed multi-generation: A comprehensive view". In: *Renewable and Sustainable Energy Reviews* 13.3 (2009), pp. 535–551. ISSN: 1364-0321. DOI: <http://dx.doi.org/10.1016/j.rser.2007.11.014>. URL: <http://www.sciencedirect.com/science/article/pii/S1364032107001578>.
- [15] S Chowdhury. *Microgrids and active distribution networks*. London: Institution of Engineering and Technology, 2009. ISBN: 978-1-84919-014-5.
- [16] CIRCUITOR. *POWER ANALYZER CVM-1D*. 2013. URL: <http://circuitor.com/docs/M98236001-03.pdf> (visitato il 2015-02-25).
- [17] Charles Desoer. *Basic circuit theory*. New York Sydney: McGraw-Hill, 1969. ISBN: 9780070851832.
- [18] Lalitha Devarakonda and Tingshu Hu. "Algebraic method for parameter identification of circuit models for batteries under non-zero initial condition". In: *Journal of Power Sources* 268 (2014), pp. 928–940. ISSN: 0378-7753. DOI: <http://dx.doi.org/10.1016/j.jpowsour.2014.06.069>. URL: <http://www.sciencedirect.com/science/article/pii/S0378775314009318>.
- [19] J.A. Kratochvill D.L. King W.E. Boyson. *Photovoltaic Array Performance Model*. Tech. rep. Sandia National Laboratories, 2004.

- [20] Aron P Dobos. “PVWatts Version 1 Technical Reference”. In: (2013).
- [21] K. Dvijotham and R. Sharma. “Battery life estimation in a real-time energy management system”. In: *Power and Energy Society General Meeting (PES), 2013 IEEE*. 2013-07, pp. 1–5. doi: 10.1109/PESMG.2013.6672464.
- [22] Albert Einstein. “Über einem die Erzeugung und Verwandlung des Lichtes betreffenden heuristischen Gesichtspunkt”. In: *Annalen der Physik* 4 (1905).
- [23] Institute for Energy European Commission Joint Research Centre and Transport. *Photovoltaic Geographical Information System (PVGIS)*. 2005. URL: <http://re.jrc.ec.europa.eu/pvgis/> (visitato il 2015-02-02).
- [24] GNB exide. *Industrial Batteries Network Power Classic Solar*. URL: <http://www.exide.com/Media/files/Downloads/IndustEuro/Classic%20Solar.pdf> (visitato il 2015-02-03).
- [25] Antonio Expósito. *Electric energy systems : analysis and operation*. Boca Raton: CRC Press, 2009. ISBN: 978-0-8493-7365-7.
- [26] John Forrest and Robin Lougee-Heimer. “CBC user guide”. In: *INFORMS Tutorials in Operations Research* (2005), pp. 257–277. URL: <http://www.coin-or.org/Cbc/cbcuserguide.html> (visitato il 2015).
- [27] Python Software Foundation. *Python 3.4.3 Language Reference*. 2015. URL: <https://www.python.org/> (visitato il 2015).
- [28] WG Früh. “Evaluation of simple wind power forecasting methods applied to a long-term wind record from Scotland”. In: 2012.
- [29] Glosser.ca. *Artificial neural network with layer coloring*. Ed. by Wikimedia. 2013. URL: [https://commons.wikimedia.org/wiki/File:Colored\\_neural\\_network.svg](https://commons.wikimedia.org/wiki/File:Colored_neural_network.svg).
- [30] L. Gomes et al. “Distributed, Agent-Based Intelligent System for Demand Response Program Simulation in Smart Grids”. In: *Intelligent Systems, IEEE* 29.1 (2014-01), pp. 56–65. ISSN: 1541-1672. doi: 10.1109/MIS.2013.2.
- [31] F.M. Gonzalez-Longatt, P. Wall, and V. Terzija. “A simplified model for dynamic behavior of permanent magnet synchronous generator for direct drive wind turbines”. In: *PowerTech, 2011 IEEE Trondheim*. 2011-06, pp. 1–7. doi: 10.1109/PTC.2011.6019425.
- [32] Wei Gu et al. “Modeling, planning and optimal energy management of combined cooling, heating and power microgrid: A review”. In: *International Journal of Electrical Power Energy Systems* 54 (2014), pp. 26–37. ISSN: 0142-0615. doi: <http://dx.doi.org/10.1016/j.ijepes.2013.06.028>. URL: <http://www.sciencedirect.com/science/article/pii/S0142061513002883>.
- [33] Nikos Hatziargyriou. *Microgrids : architectures and control*. Chichester, West Sussex, U.K: Wiley, 2014. ISBN: 978-1-118-72068-4.

- [34] John E Hay and John A Davies. “Calculation of the solar radiation incident on an inclined surface”. In: *Proc. of First Canadian Solar Radiation Data Workshop* (Eds: JE Hay and TK Won), Ministry of Supply and Services Canada. Vol. 59. 1980.
- [35] Tingshu Hu, B. Zanchi, and Jianping Zhao. “Simple Analytical Method for Determining Parameters of Discharging Batteries”. In: *Energy Conversion, IEEE Transactions on* 26.3 (2011-09), pp. 787–798. ISSN: 0885-8969. DOI: 10.1109/TEC.2011.2129594.
- [36] Thomas Huld et al. “Mapping the performance of {PV} modules, effects of module type and data averaging”. In: *Solar Energy* 84.2 (2010), pp. 324–338. ISSN: 0038-092X. DOI: <http://dx.doi.org/10.1016/j.solener.2009.12.002>. URL: <http://www.sciencedirect.com/science/article/pii/S0038092X0900293X>.
- [37] IEC. *IEC 61853-1 Photovoltaic (PV) module performance testing and energy rating - Part 1: Irradiance and temperature performance measurements and power rating*. norma. International Electrotechnical Commission, 2011.
- [38] Robert Saint James Momoh Sakis Meliopoulos. *Centralized and Distributed Generated Power Systems - A Comparison Approach*. white paper. Power Systems Engineering Research Center, 2012. URL: [https://duckduckgo.com/l/?kh=-1&uddg=http://www.pserc.wisc.edu/documents/publications/papers/fgwhitepapers/Momoh\\_Future\\_Grid\\_White\\_Paper\\_Gen\\_Analysis\\_June\\_2012.pdf](https://duckduckgo.com/l/?kh=-1&uddg=http://www.pserc.wisc.edu/documents/publications/papers/fgwhitepapers/Momoh_Future_Grid_White_Paper_Gen_Analysis_June_2012.pdf) (visitato il 2014-11-04).
- [39] Eric Jones, Travis Oliphant, Pearu Peterson, et al. *SciPy: Open source scientific tools for Python*. 2001. URL: <http://www.scipy.org/>.
- [40] George Kariniotakis. *DA1 Digital Models for Micro Sources*. white paper. Micro-Grids Project, 2003. URL: [http://www.microgrids.eu/micro2000/delivarables/Deliverable\\_DA1.pdf](http://www.microgrids.eu/micro2000/delivarables/Deliverable_DA1.pdf).
- [41] Charles Kirubi et al. “Community-Based Electric Micro-Grids Can Contribute to Rural Development: Evidence from Kenya”. In: *World Development* 37.7 (2009), pp. 1208–1221. ISSN: 0305-750X. DOI: <http://dx.doi.org/10.1016/j.worlddev.2008.11.005>. URL: <http://www.sciencedirect.com/science/article/pii/S0305750X08003288>.
- [42] Michael Kleehammer. *pyodbc: A Python DB API 2 module for ODBC*. 2008. URL: <https://github.com/mkleehammer/pyodbc>.
- [43] David Kriesel. *A Brief Introduction to Neural Networks*. 2007. URL: <http://www.dkriesel.com>.
- [44] John D Kueck et al. “Microgrid energy management system”. In: *Oak Ridge, TN: Oak Ridge National Laboratory Report ORNL/TM-2002/242* (2003).
- [45] D. Linden and T.B. Reddy. *Handbook of Batteries*. McGraw-Hill handbooks. McGraw-Hill, 2001. ISBN: 9780071414753.

- [46] T. Logenthiran, D. Srinivasan, and Tan Zong Shun. “Demand Side Management in Smart Grid Using Heuristic Optimization”. In: *Smart Grid, IEEE Transactions on* 3.3 (2012-09), pp. 1244–1252. ISSN: 1949-3053. DOI: 10.1109/TSG.2012.2195686.
- [47] Luminous. *WHISPER 200 - WIND CHARGE CONTROLLER*. URL: [thesolarstore.com/manuals/Brochure%20W200%20WCC%20\(NEW\).pdf](http://thesolarstore.com/manuals/Brochure%20W200%20WCC%20(NEW).pdf) (visitato il 2015-02-18).
- [48] A Luque. *Handbook of photovoltaic science and engineering*. Chichester, West Sussex, U.K: Wiley, 2011. ISBN: 978-0-470-97466-7.
- [49] M.P. Marietta, M. Graells, and J.M. Guerrero. “A rolling horizon rescheduling strategy for flexible energy in a microgrid”. In: *Energy Conference (ENERGYCON), 2014 IEEE International*. 2014-05, pp. 1297–1303. DOI: 10.1109/ENERGYCON.2014.6850590.
- [50] Gilbert Masters. *Renewable and efficient electric power systems*. Hoboken, NJ: John Wiley Sons, 2004. ISBN: 0-471-28060-7.
- [51] Eugene L Maxwell. *A quasi-physical model for converting hourly global horizontal to direct normal insolation*. Tech. rep. Solar Energy Research Inst., Golden, CO (USA), 1987.
- [52] Wes McKinney. “Data Structures for Statistical Computing in Python”. In: *Proceedings of the 9th Python in Science Conference*. Ed. by Stéfan van der Walt and Jarrod Millman. 2010, pp. 51–56.
- [53] Hugo Morais, Pedro Faria, and Zita Vale. “Demand response design and use based on network locational marginal prices”. In: *International Journal of Electrical Power Energy Systems* 61 (2014), pp. 180–191. ISSN: 0142-0615. DOI: <http://dx.doi.org/10.1016/j.ijepes.2014.03.024>. URL: <http://www.sciencedirect.com/science/article/pii/S0142061514001264>.
- [54] Hugo Morais et al. “Optimal scheduling of a renewable micro-grid in an isolated load area using mixed-integer linear programming”. In: *Renewable Energy* 35.1 (2010), pp. 151–156. ISSN: 0960-1481. DOI: <http://dx.doi.org/10.1016/j.renene.2009.02.031>. URL: <http://www.sciencedirect.com/science/article/pii/S0960148109001001>.
- [55] David M. Sharman Niall M. McMahon Peter R. Burton. “On Electrodynamic Braking for Small Wind Turbines”. 2014. URL: [http://www.niallmcmahon.com/electrodynamic\\_braking\\_niall\\_mcmahon.pdf](http://www.niallmcmahon.com/electrodynamic_braking_niall_mcmahon.pdf) (visitato il 2015-05-26).
- [56] Torben Skov Nielsen et al. “A new reference for wind power forecasting”. In: *Wind Energy* 1.1 (1998), pp. 29–34. ISSN: 1099-1824. DOI: 10.1002/(SICI)1099-1824(199809)1:1<29::AID-WE10>3.0.CO;2-B. URL: [http://dx.doi.org/10.1002/\(SICI\)1099-1824\(199809\)1:1%3C29::AID-WE10%3E3.0.CO;2-B](http://dx.doi.org/10.1002/(SICI)1099-1824(199809)1:1%3C29::AID-WE10%3E3.0.CO;2-B).

- [57] D.E. Olivares, C.A. Canizares, and M. Kazerani. “A centralized optimal energy management system for microgrids”. In: *Power and Energy Society General Meeting, 2011 IEEE*. 2011-07, pp. 1–6. doi: 10.1109/PES.2011.6039527.
- [58] D.E. Olivares, C.A. Cañizares, and M. Kazerani. “A Centralized Energy Management System for Isolated Microgrids”. In: *Smart Grid, IEEE Transactions on* 5.4 (2014-07), pp. 1864–1875. issn: 1949-3053. doi: 10.1109/TSG.2013.2294187.
- [59] T. Orhan et al. “A feasibility study on microgrid for various Islands in Australia”. In: *Power Engineering Conference (AUPEC), 2014 Australasian Universities*. 2014-09, pp. 1–8. doi: 10.1109/AUPEC.2014.6966604.
- [60] Manfred Padberg and Giovanni Rinaldi. “A branch-and-cut algorithm for the resolution of large-scale symmetric traveling salesman problems”. In: *SIAM review* 33.1 (1991), pp. 60–100.
- [61] R. Palma-Behnke et al. “A Microgrid Energy Management System Based on the Rolling Horizon Strategy”. In: *Smart Grid, IEEE Transactions on* 4.2 (2013-06), pp. 996–1006. issn: 1949-3053. doi: 10.1109/TSG.2012.2231440.
- [62] Perez. “Dynamic global-to-direct irradiance conversion models”. eng. In: *ASHRAE Transactions* 98.1 (1992). ID: unige:38583, pp. 354–369. URL: <http://archive-ouverte.unige.ch/unige:38583>.
- [63] Richard Perez et al. “A new operational model for satellite-derived irradiances: description and validation”. In: *Solar Energy* 73.5 (2002), pp. 307–317. issn: 0038-092X. doi: [http://dx.doi.org/10.1016/S0038-092X\(02\)00122-6](http://dx.doi.org/10.1016/S0038-092X(02)00122-6). URL: <http://www.sciencedirect.com/science/article/pii/S0038092X02001226>.
- [64] Progea. *Movicon 11*. URL: <http://www.progea.com/en-us/products/scadahmimovicon11.aspx>.
- [65] S. Ramos et al. “A data-mining based methodology for win forecasting”. In: *Intelligent System Application to Power Systems (ISAP), 2011 16th International Conference on*. 2011-09, pp. 1–6. doi: 10.1109/ISAP.2011.6082223.
- [66] D.A.J. Rand and P.T. Moseley. “{SECONDARY} {BATTERIES} – LEAD – {ACID} {SYSTEMS} | Overview”. In: *Encyclopedia of Electrochemical Power Sources*. Ed. by Editor-in-Chief: Jürgen Garche. Amsterdam: Elsevier, 2009, pp. 550–575. isbn: 978-0-444-52745-5. doi: <http://dx.doi.org/10.1016/B978-044452745-5.00126-X>. URL: <http://www.sciencedirect.com/science/article/pii/B978044452745500126X>.
- [67] Muhammad Rashid. *Power electronics handbook devices, circuits, and applications*. Burlington, MA: Butterworth-Heinemann, 2011. isbn: 978-0-12-382036-5.
- [68] Gordon Reikard. “Predicting solar radiation at high resolutions: A comparison of time series forecasts”. In: *Solar Energy* 83.3 (2009), pp. 342–349. issn: 0038-092X. doi: <http://dx.doi.org/10.1016/j.solener.2008.08.007>. URL: <http://www.sciencedirect.com/science/article/pii/S0038092X08002107>.

- [69] DT Reindl, WA Beckman, and JA Duffie. “Evaluation of hourly tilted surface radiation models”. In: *Solar Energy* 45.1 (1990), pp. 9–17.
- [70] Ankit Rohatgi. *WebPlotDigitizer*. Version 3.8. 2015. URL: <http://arohatgi.info/WebPlotDigitizer>.
- [71] Satec. *PM130 PLUS Powermeter Series PM130P/PM130E/PM130EH Installation and Operation Manual*. 2012. URL: <http://satec-global.com/UserFiles/File/SATEC/files/Manuals/PM130%20PLUS/PM130%20PLUS.pdf> (visitato il 2015-02-25).
- [72] Aden Seaman, Thanh-Son Dao, and John McPhee. “A survey of mathematics-based equivalent-circuit and electrochemical battery models for hybrid and electric vehicle simulation”. In: *Journal of Power Sources* 256 (2014), pp. 410–423. ISSN: 0378-7753. DOI: <http://dx.doi.org/10.1016/j.jpowsour.2014.01.057>. URL: <http://www.sciencedirect.com/science/article/pii/S0378775314000810>.
- [73] M. Silva, H. Morais, and Z. Vale. “An integrated approach for distributed energy resource short-term scheduling in smart grids considering realistic power system simulation”. In: *Energy Conversion and Management* 64 (2012). {IREC} 2011, The International Renewable Energy Congress, pp. 273–288. ISSN: 0196-8904. DOI: <http://dx.doi.org/10.1016/j.enconman.2012.04.021>. URL: <http://www.sciencedirect.com/science/article/pii/S0196890412002087>.
- [74] M. Silva et al. “Short-term scheduling considering five-minute and hour-ahead energy resource management”. In: *Power and Energy Society General Meeting, 2012 IEEE*. 2012-07, pp. 1–8. DOI: [10.1109/PESGM.2012.6345571](https://doi.org/10.1109/PESGM.2012.6345571).
- [75] Mohit Singh and Surya Santoso. *Dynamic models for wind turbines and wind power plants*. National Renewable Energy Laboratory, 2011.
- [76] P. Skolnik et al. “Cogeneration units simulation models library”. In: *Process Control (PC), 2013 International Conference on*. 2013-06, pp. 252–256. DOI: [10.1109/PC.2013.6581418](https://doi.org/10.1109/PC.2013.6581418).
- [77] S.S. Soman et al. “A review of wind power and wind speed forecasting methods with different time horizons”. In: *North American Power Symposium (NAPS), 2010*. 2010-09, pp. 1–8. DOI: [10.1109/NAPS.2010.5619586](https://doi.org/10.1109/NAPS.2010.5619586).
- [78] W. De Soto, S.A. Klein, and W.A. Beckman. “Improvement and validation of a model for photovoltaic array performance”. In: *Solar Energy* 80.1 (2006), pp. 78–88. ISSN: 0038-092X. DOI: <http://dx.doi.org/10.1016/j.solener.2005.06.010>. URL: <http://www.sciencedirect.com/science/article/pii/S0038092X05002410>.
- [79] Steca. *User and Installer Manual*. URL: [www.solarlink.de/PDF-Files/Steca/TAROM\\_Emanual.pdf](http://www.solarlink.de/PDF-Files/Steca/TAROM_Emanual.pdf) (visitato il 2015-02-19).
- [80] J.S. Roy Stuart A. Mitchell. *PuLP A python Linear Programming API*. Ed. by COIN-OR Computational Infrastructure for Operations Research. Version 1.6.0. 2015. URL: <https://github.com/coin-or/pulp>.

- [81] Studer. *User and Installer Manual Sine wave inverter SI SERIES*. 2013. URL: <http://www.studer-inno.com/upload/folders/405.pdf> (visitato il 2015-02-20).
- [82] Studer. *User's and installer's Manual COMPACT - C 2600-24*. 2005. URL: <http://www.studer-inno.com/upload/folders/2855.pdf> (visitato il 2015-02-20).
- [83] Studer. *User's and installer's Manual XP-COMPACT XPC 2200-24*. 2005. URL: <http://www.studer-inno.com/upload/folders/2858.pdf> (visitato il 2015-02-20).
- [84] Xingguo Tan, Qingmin Li, and Hui Wang. "Advances and trends of energy storage technology in Microgrid". In: *International Journal of Electrical Power Energy Systems* 44.1 (2013), pp. 179–191. ISSN: 0142-0615. DOI: <http://dx.doi.org/10.1016/j.ijepes.2012.07.015>. URL: <http://www.sciencedirect.com/science/article/pii/S0142061512003754>.
- [85] Daniel Tenfen and Erlon Cristian Finardi. "A mixed integer linear programming model for the energy management problem of microgrids". In: *Electric Power Systems Research* 122 (2015), pp. 19–28. ISSN: 0378-7796. DOI: <http://dx.doi.org/10.1016/j.epsr.2014.12.019>. URL: <http://www.sciencedirect.com/science/article/pii/S0378779614004659>.
- [86] A. G. Ter-Gazarian. *Energy Storage for Power Systems (2nd Edition)*. Institution of Engineering and Technology, 2012. ISBN: 978-1-84919-220-0.
- [87] C. Thammasorn. "Generation unit commitment in microgrid with renewable generators and CHP". In: *Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), 2013 10th International Conference on*. 2013-05, pp. 1–6. DOI: 10.1109/ECTICon.2013.6559661.
- [88] JL Threlkeld and RC Jordan. "Direct solar radiation available on clear days". In: *Heat, Piping Air Cond.* 29.12 (1957).
- [89] Z. Vale et al. "MASCEM: Electricity Markets Simulation with Strategic Agents". In: *Intelligent Systems, IEEE* 26.2 (2011-03), pp. 9–17. ISSN: 1541-1672. DOI: 10.1109/MIS.2011.3.
- [90] Brian D. Vick, Daryl R. Myers, and William E. Boyson. "Using direct normal irradiance models and utility electrical loading to assess benefit of a concentrating solar power plant". In: *Solar Energy* 86.12 (2012). Solar Resources, pp. 3519–3530. ISSN: 0038-092X. DOI: <http://dx.doi.org/10.1016/j.solener.2012.03.010>. URL: <http://www.sciencedirect.com/science/article/pii/S0038092X12001156>.
- [91] N.I. Voropai et al. "Development of power supply to isolated territories in Russia on the bases of microgrid concept". In: *Power and Energy Society General Meeting, 2012 IEEE*. 2012-07, pp. 1–5. DOI: 10.1109/PESGM.2012.6344612.

- [92] A. Vosloo and K.A. Raji. “Intelligent central energy management system for remote community microgrid”. In: *Domestic Use of Energy (DUE), 2015 International Conference on the*. 2015-03, pp. 137–140. DOI: 10.1109/DUE.2015.7102973.
- [93] S. van der Walt, S.C. Colbert, and G. Varoquaux. “The NumPy Array: A Structure for Efficient Numerical Computation”. In: *Computing in Science Engineering* 13.2 (2011-03), pp. 22–30. ISSN: 1521-9615. DOI: 10.1109/MCSE.2011.37.
- [94] Martin Winter and Ralph J Brodd. “What are batteries, fuel cells, and supercapacitors?” In: *Chemical reviews* 104.10 (2004), pp. 4245–4270.
- [95] Laurent Xavier. *QElectroTech*. 2008. URL: <http://qelectrotech.org>.
- [96] Georgi Hristov Yordanov. “Characterization and analysis of photovoltaic modules and the solar resource based on in-situ measurements in Southern Norway”. PhD thesis. Norwegian University of Science, Technology Faculty of Information Technology, Mathematics, and Electrical Engineering Department of Electric Power Engineering, 2012. URL: <http://www.diva-portal.org/smash/record.jsf?pid=diva2:589641&dswid=-107>.
- [97] Hanlei Zhang and Mo-Yuen Chow. “Comprehensive dynamic battery modeling for PHEV applications”. In: *Power and Energy Society General Meeting, 2010 IEEE*. 2010-07, pp. 1–6. DOI: 10.1109/PES.2010.5590108.

# Annex Unit Commitment

September 20, 2015

```
In [1]: import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import pyEMS
%matplotlib inline

In [2]: random = np.random.RandomState(seed=12345) # fixed seed for reproducibility of the random func

In [3]: def plot_power(node):
    fig1 = plt.figure(figsize=(17, 4))
    for load in node.list_elements('load'):
        plt.plot(load.requested, label=(load.name + ' power requested'))

    for source in node.list_elements('source'):
        plt.plot(source.power_max, label=(source.name + ' maximum power'))

    plt.title('Power')
    plt.xlabel('Time interval')
    plt.ylabel('Power / W')
    plt.grid(True)
    plt.legend(loc='upper center', bbox_to_anchor=(0.5, -0.1),
               fancybox=True, shadow=True, ncol=5)

def plot_cost(node):
    fig2 = plt.figure(figsize=(17, 4))
    for load in node.list_elements('load'):
        plt.plot(load.shift_cost, label=(load.name + ' shift cost'))
        plt.plot(load.curtail_cost, label=(load.name + ' Load curtail cost'))

    for source in node.list_elements('source'):
        plt.plot(source.cost, label=(source.name + ' cost'))

    plt.title('Cost')
    plt.xlabel('Time interval')
    plt.ylabel('Cost / (€ kWh-1)')
    plt.grid(True)
    plt.legend(loc='upper center', bbox_to_anchor=(0.5, -0.1),
               fancybox=True, shadow=True, ncol=5)

def plot_power_max_requested(node):
    fig3 = plt.figure("Power", figsize=(17, 7))
    for load in node.list_elements('load'):
        plt.plot(load.requested, label=(load.name + ' power requested'))
```

```

sum_source = np.zeros(np.size(node.list_elements('source')[0].committed))
for source in node.list_elements('source'):
    plt.plot(source.power_max, '--', label=(source.name + ' maximum power'))
    plt.plot(source.committed, label=(source.name + ' power committed'))
    sum_source += np.array(source.committed)

if np.size(node.list_elements('storage')) is not 0:
    sum_storage = np.zeros(np.size(node.list_elements('storage')[0].committed_dsc))
    for storage in node.list_elements('storage'):
        plt.plot(storage.committed_ch, label=(storage.name + ' charging power'))
        plt.plot(storage.committed_dsc, label=(storage.name + ' discharging power'))
        sum_storage += np.array(storage.committed_dsc)
    plt.plot(sum_source + sum_storage, label='Total sources power')

plt.title('Power')
plt.xlabel('Time interval')
plt.ylabel('Power / W')
plt.grid(True)
plt.legend(loc='upper center', bbox_to_anchor=(0.5, -0.1),
          fancybox=True, shadow=True, ncol=5)

```

## 1 Examples for the Unit Commitment module

In the following notebook we'll show some basic examples cases of the Unit Commitment done by the module scheduling in order to demonstrate both its use than its validity.

### 1.1 Loads examples

A common basic scenario is set up in order to evaluate how the optimization process of the Unit Commitment allocates the resources changing the characteristics of the load. The scenario presents:

- a window of analysis of ten equal time intervals,
- one variable source that has the same power of load in the first half of the time intervals and twice the load power in the second half,
- one constant load,
- cost of source variable and higher than shift and curtail cost in the first half of time intervals while lower and than both of them in the second half.

Starting from the common scenario, three different typologies of loads are used: a non controllable load, a shiftable load and a curtailable load.

While the hourly cost of the generator is equal for all the scenarios, the cost of the load shifting and load curtailment change between the scenarios.

The resolution of a Unit Commitment problem starts with the definition of the time intervals for which the commitment is searched. The window list contains ten equal intervals of 3600 seconds each one so that the total time length is of ten hours; note that each time intervals can also have different length but here they are constant to keep the problem simple.

```

In [4]: time_intervals = 10
        half_time_intervals = int(time_intervals / 2)
        window = [3600] * time_intervals # s

```

After the window object, the second required element is the node to which sources, storages and loads are connected. The element is created using the node Class in module elements of the pyEMS package; this class required just a name to be initialized. Here a node called *Node example* is created.

```
In [5]: node = pyEMS.elements.Node(name='Node example')
```

The just created node is empty.

```
In [6]: node.list_all_elements()
```

The Node Class as all the other classes in the `elements` module, when generates a new object, assigns a universally unique identifier (UUID) to the object so that to different elements can have the same name and still be distinguishable.

```
In [7]: node2 = pyEMS.elements.Node(name='Node example')
        print(node.name, node.uuid)
        print(node2.name, node2.uuid)
```

```
Node example 135aef79-ba8c-4dcc-837e-12ac1ce07f2f
```

```
Node example d2093840-70a2-40d4-b266-45916a0736fb
```

The next step is to create the elements connected to the node. Given that in the following section different typologies of load are used while the source element does not change, the latter is created first.

The source element is created with the `Source` Class with the following four inputs: the name of the element, the low bound, the up bound and the generation cost of power for each time interval.

The created `source` is then added to the node.

```
In [8]: unit_power= 0.5  # W
        unit_cost = 1.0  # € Wh-1

        source = pyEMS.elements.Source(name='Source example',
                                         power_min=[0] * time_intervals,
                                         power_max=([unit_power] * half_time_intervals +
                                                    [2 * unit_power] * half_time_intervals),
                                         cost=([8 * unit_cost] * half_time_intervals +
                                                [0 * unit_cost] * half_time_intervals))

        node.add_element(source)
        node.list_all_elements()
```

```
Element name: Source example UUID: 3a95c05e-0d65-4d04-a5fc-a349a1bfcd55
```

Before the Unit Commitment process, the source has no power committed.

```
In [9]: source.committed
```

```
Out[9]: [0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
```

### 1.1.1 Load not modifiable

The scenario with a load without shift neither curtail capabilities shows that even if it was cheaper curtail/shift the load the algorithm does not allow it.

The scenario presents:

- a not controllable load (not shiftable neither curtailable),
- cost of curtail and cost of shift constant with the shift cost higher than curtail cost.

The load object is created by the `Load` Class setting its name `name`, the forecasted or planned load profile for each time interval `requested` then the Demand Side Management option allowed and their cost. The load element is then added to the node.

```
In [10]: load_not_mod = pyEMS.elements.Load(name='Load not modificable',
                                             requested=[unit_power] * time_intervals, # W
                                             curtailable=False,
                                             shiftable=False,
                                             curtail_cost=[2 * unit_cost] * time_intervals,
                                             shift_cost=[4 * unit_cost] * time_intervals)

node.add_element(load_not_mod)
node.list_all_elements()
```

Element name: Source example UUID: 3a95c05e-0d65-4d04-a5fc-a349a1bfcd55

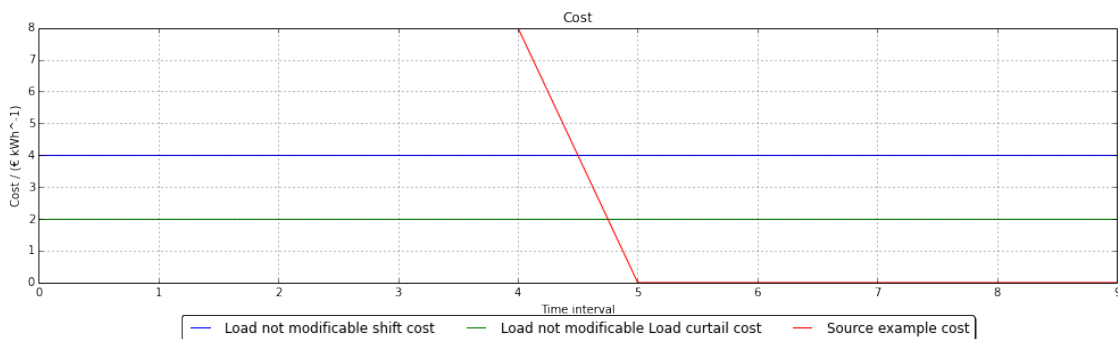
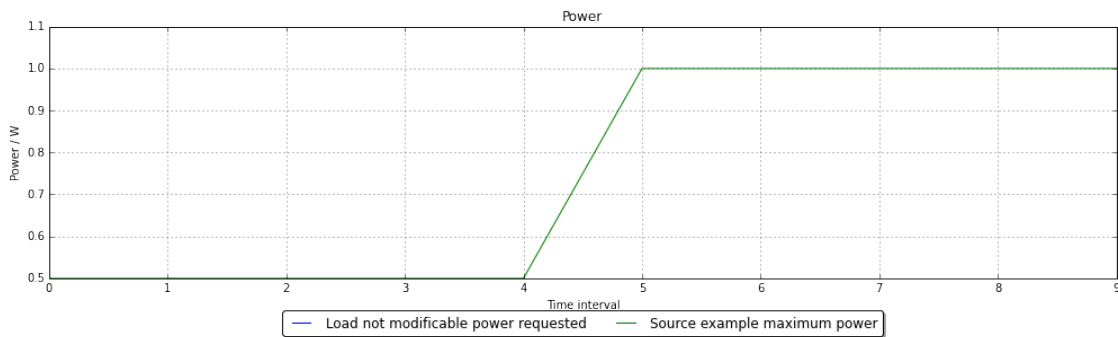
Element name: Load not modificable UUID: 99db080f-55e8-4ce9-a606-e78c72aea3f3

As shown for the source, the load has no power committed before the Unit Commitment.

```
In [11]: load_not_mod.committed(window)
```

```
Out[11]: [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]
```

```
In [12]: plot_power(node)
         plot_cost(node)
```



The Unit Commitment optimization is performed by the `unit_commitment` function in the `scheduling` module. The function requires a time reference given by the `window` element and the `node` element on which perform the optimization.

```
In [13]: pyEMS.scheduling.unit_commitment(window, node)
```

```
Status: Optimal
Solution time: 0.03003200000000028 s
Cost: 20.0
```

The function save the result of the optimization directly in the elements objects in the node and print the **Status** of the optimization process, the time required to obtain the solution **Solution time** in seconds and the total **Cost** of the scheduled Unit Commitment.

```
In [14]: source.committed
```

```
Out[14]: [0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5]
```

```
In [15]: load_not_mod.committed(window)
```

```
Out[15]: [0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5]
```

```
In [16]: load_not_mod.shift_in_power(window)
```

```
Out[16]: [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]
```

```
In [17]: load_not_mod.shift_out_power()
```

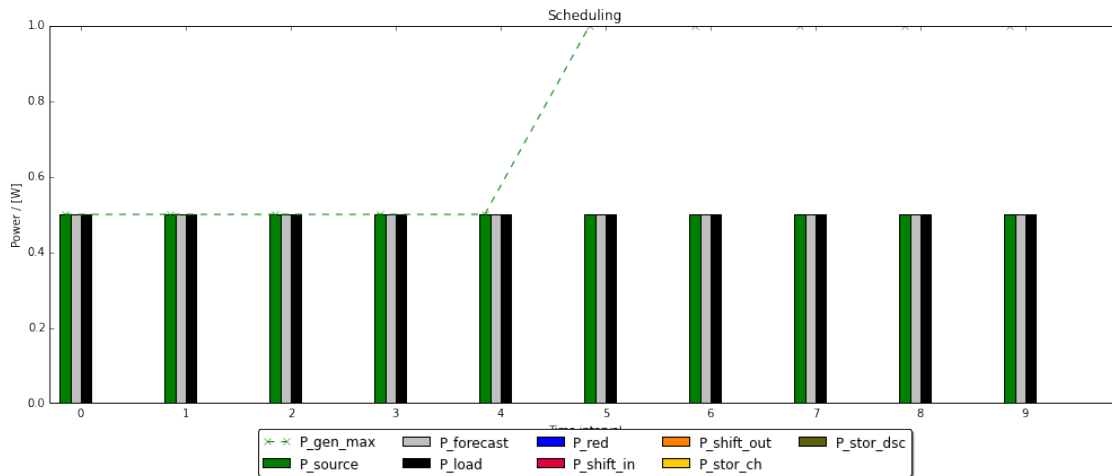
```
Out[17]: [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]
```

```
In [18]: load_not_mod.curtail_power()
```

```
Out[18]: [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]
```

The results can be plotted using the function `scheduling_plot` of the `plots` module.

```
In [19]: pyEMS.plots.scheduling_plot(window, node)
```



The result of the Unit Commitment shows that the final load after optimization `P_load` is equal to the load before the optimization `P_forecasted`.

```
In [20]: load_not_mod.shift_table()
```

```
[1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]
[0.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]
[0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]
[0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]
[0.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0]
[0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 0.0]
[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.0]
[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0]
[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0]
[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0]
```

The load table has all and only diagonal elements not null because the scheduling cannot change the time intervals of the loads.

Before changing case study with an other load, the actual load is removed from the node.

```
In [21]: node.remove_element(load_not_mod)
```

```
In [22]: node.list_all_elements()
```

```
Element name: Source example UUID: 3a95c05e-0d65-4d04-a5fc-a349a1bfcd55
```

### 1.1.2 Load only shiftable

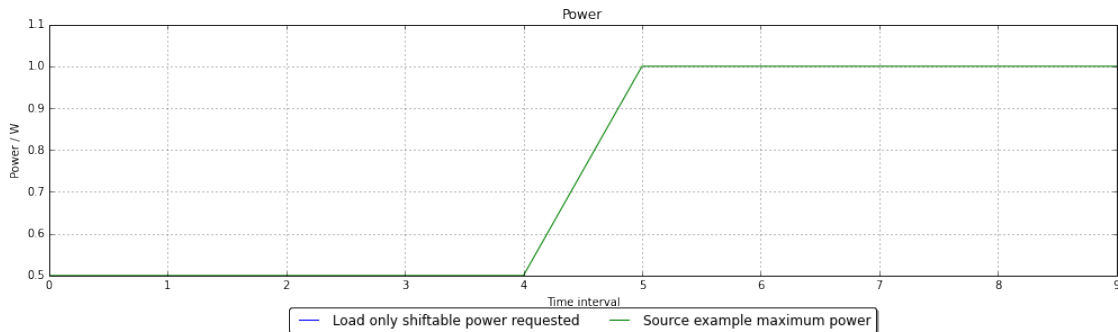
The scenario presents:

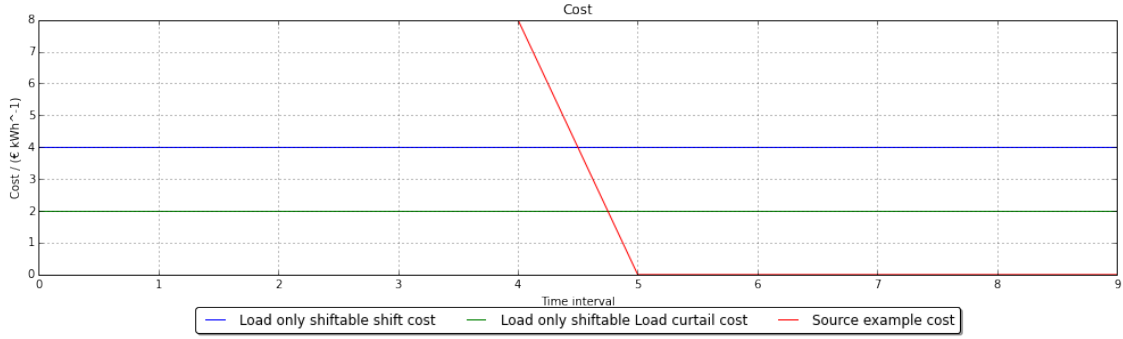
- load shiftable but not curtailable,
- cost of curtail and cost of shift constant with the shift cost higher than curtail cost.

```
In [23]: load_only_shift = pyEMS.elements.Load(name='Load only shiftable',
        requested=[unit_power] * time_intervals, # W
        curtailable=False,
        shiftable=True,
        curtail_cost=[2 * unit_cost] * time_intervals,
        shift_cost=[4 * unit_cost] * time_intervals)
```

```
node.add_element(load_only_shift)
```

```
In [24]: plot_power(node)
        plot_cost(node)
```





If the load is only shiftable.

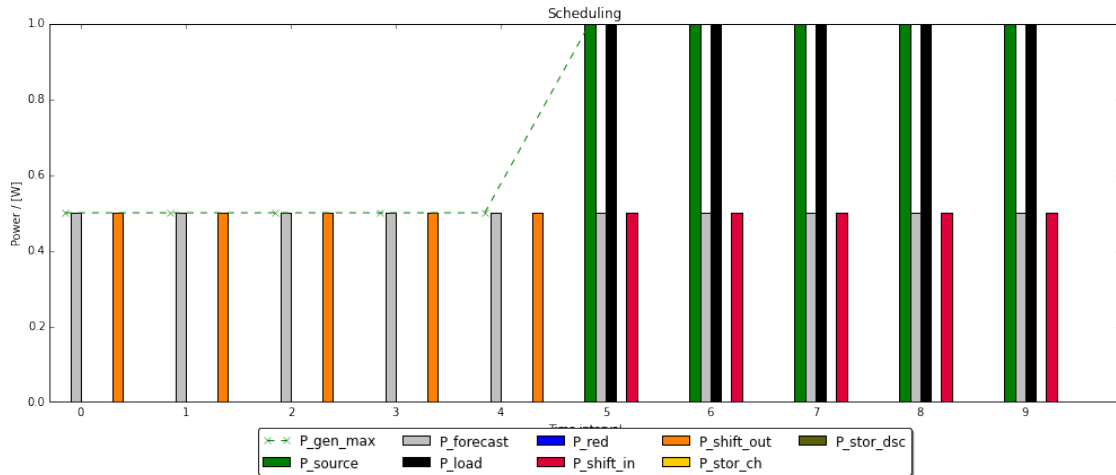
```
In [25]: pyEMS.scheduling.unit_commitment(window, node)
```

Status: Optimal

Solution time: 0.02538800000000041 s

Cost: 10.0

```
In [26]: pyEMS.plots.scheduling_plot(window, node)
```



The Unit Commitment optimization shifted the forecasted load from first half of the time window to the second one where the hourly cost of the power source is null. The scenario was designed so that the capacity of the source can satisfies the load demand, thus there are no physical constrains and the optimization has only economical reasons. Unlike the not controllable load scenario where the total cost of operation was given by the cost of the source, in this case the total cost is given by the shift cost because the source has no generation cost when satisfies the load.

```
In [27]: load_only_shift.shift_table()
```

```
[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0]
[0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 0.0]
[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0]
[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.0]
```

```

[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0]
[0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 0.0]
[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.0]
[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0]
[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0]
[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0]

```

The table show that the shift is not necessary sequential because the shift cost is constant in all the window and generator is constant in each half window thus the total cost is not effected by the order of relocation of the energy slot, matematically it means that the global optimum of the optimization process is not unique and the solver gives the first optimum met.

The Unit Commitment algorithm chose to shift the load instead of using the more expensive original scheduling but it cannot use the cheaper solution which was the load curtailment. In order to show that the forbidding of the curtailment was a constrain in the problem, the scenario is rerun setting both shift and curtailment as allowed.

```
In [28]: load_only_shift.curtailable=True
```

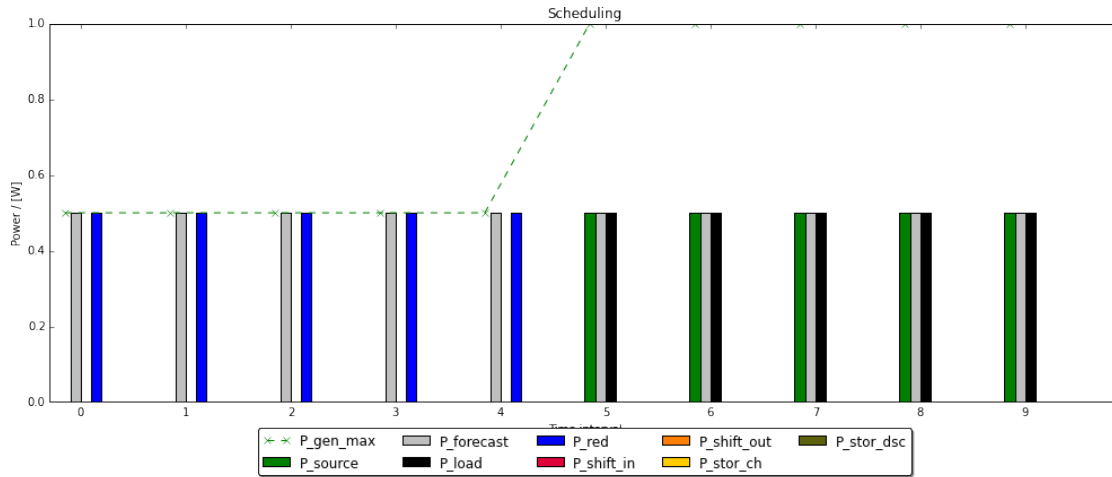
```
In [29]: pyEMS.scheduling.unit_commitment(window, node)
```

Status: Optimal

Solution time: 0.026217000000000823 s

Cost: 5.0

```
In [30]: pyEMS.plots.scheduling_plot(window, node)
```



```
In [31]: load_only_shift.shift_table()
```

```

[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]
[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]
[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]
[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]
[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]
[0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 0.0]
[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.0]
[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0]
[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0]
[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0]

```

In this case the Unit Commitment chose to use the curtailment option during the time intervals with high generation cost while in the intervals with zero generation cost the load is refed because the cost of curtailment is not null.

### 1.1.3 Load only curtailable

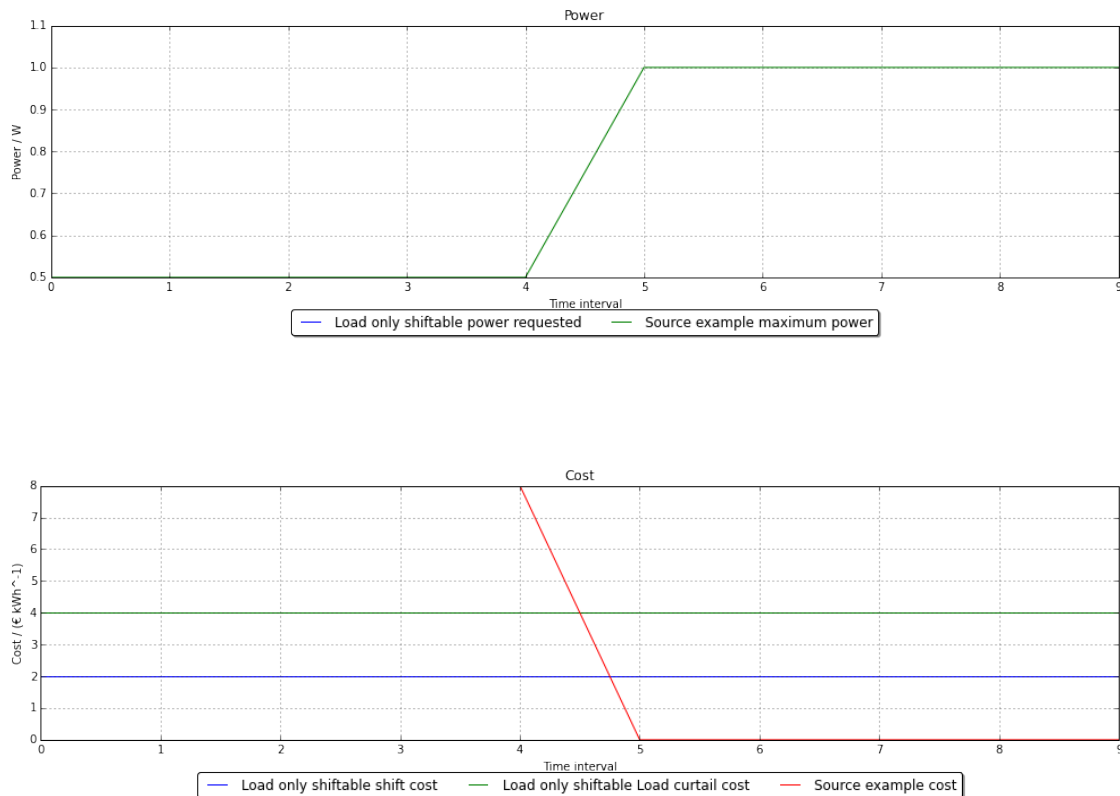
The scenario presents:

- load not shiftable but curtailable
- cost of curtail and cost of shift constant with the curtail cost higher than shift cost.

```
In [32]: load_only_curtail = pyEMS.elements.Load(name='Load only shiftable',
        requested=[unit_power] * time_intervals, # W
        curtailable=True,
        shiftable=False,
        curtail_cost=[4 * unit_cost] * time_intervals,
        shift_cost=[2 * unit_cost] * time_intervals)

node.remove_element(load_only_shift)
node.add_element(load_only_curtail)
```

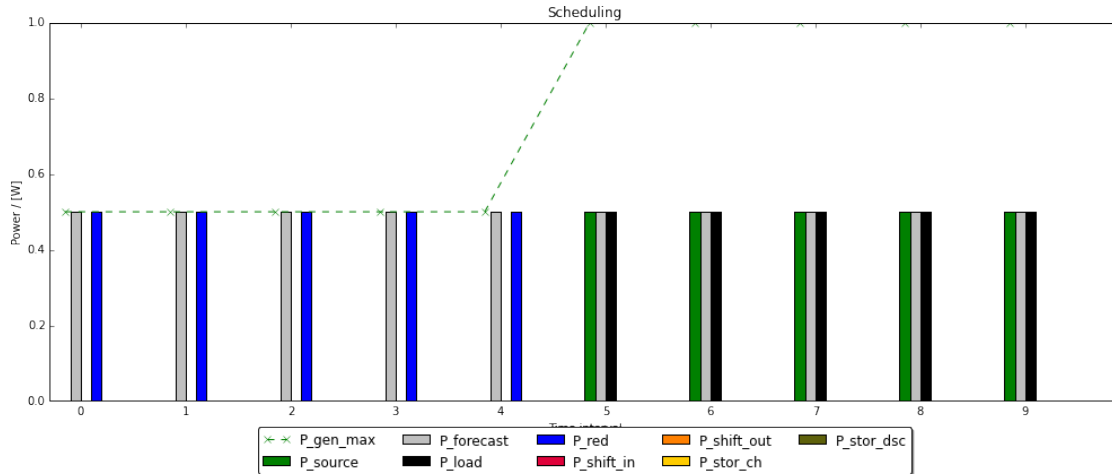
```
In [33]: plot_power(node)
        plot_cost(node)
```



```
In [34]: pyEMS.scheduling.unit_commitment(window, node)
```

```
Status: Optimal
Solution time: 0.029779000000001332 s
Cost: 10.0
```

```
In [35]: pyEMS.plots.scheduling_plot(window, node)
```



In this scenario the Unit Commitment chose a curtailment of the load even if it has a higher cost than the load shift because of the imposed constrain on this option.

```
In [36]: load_only_curtail.shift_table()
```

```
[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]
[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]
[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]
[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]
[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]
[0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0]
[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 0.0]
[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.0]
[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0]
[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0]
```

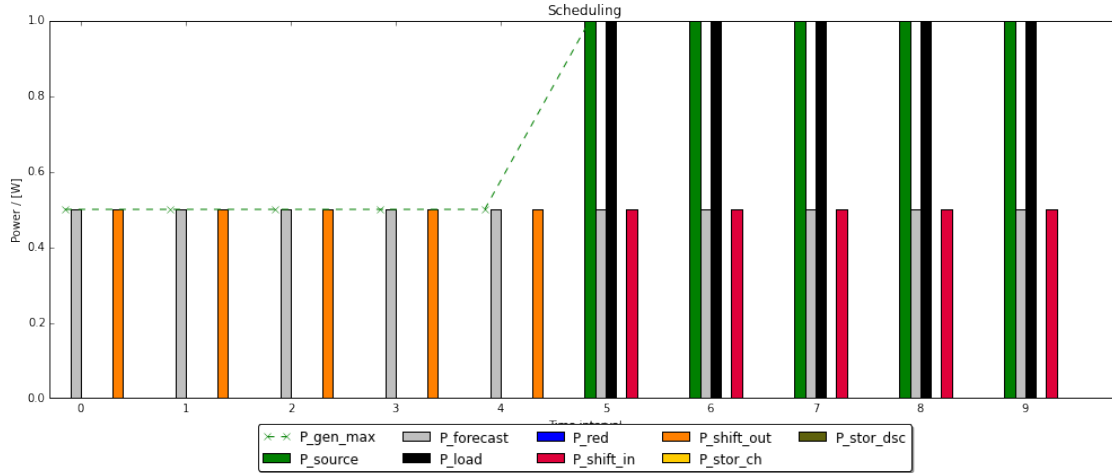
Removing the prohibition on the shift of the load so that both the load control option are allowable, the algorithm find a better optimum scheduling with a shift of the load to the hour with null generation cost.

```
In [37]: load_only_curtail.shifttable = True
```

```
In [38]: pyEMS.scheduling.unit_commitment(window, node)
```

```
Status: Optimal
Solution time: 0.02525600000000061 s
Cost: 5.0
```

```
In [39]: pyEMS.plots.scheduling_plot(window, node)
```



```
In [40]: load_only_curtail.shift_table()
```

```
[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0]
[0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 0.0]
[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0]
[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0]
[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.0]
[0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 0.0]
[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.0]
[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0]
[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0]
[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0]
```

## 1.2 Sources and storage examples

In the following it is shown how it is possible to set the dispatching priority to a renewable source over a conventional one just setting a difference between the generation costs of the sources.

Set the priority of the renewable source, the role of energy buffer of a storage system is inlustrated, showing how the storage allows to use all the renewable potential.

The scenario presents:

- a window of analysis of ten equal time intervals,
- a conventional source `con_source` with a capacity greather than the load demand,
- a renewable source `res_source` with a capacity inferior to the load demand,
- a variable not controllable load `load` with a randomly generated power profile.

```
In [41]: time_intervals = 10
         window = [3600] * time_intervals # s
         unit_power= 0.5 # W
         unit_cost = 1.0 # € Wh-1

         node_source = pyEMS.elements.Node(name='Node for source example')

         load = pyEMS.elements.Load(name='Load',
                                   requested= random.randint(0, 10, time_intervals) * unit_power, # W
                                   curtailable=False,
```

```

        shiftable=False,
        curtail_cost=[unit_cost] * time_intervals,
        shift_cost=[unit_cost] * time_intervals)

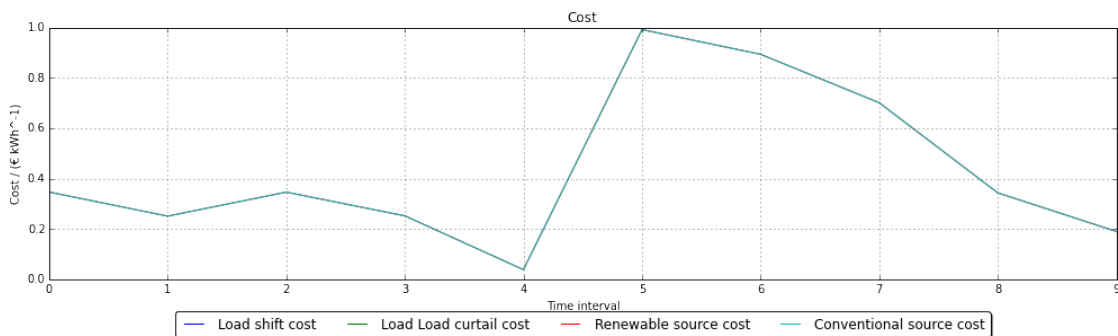
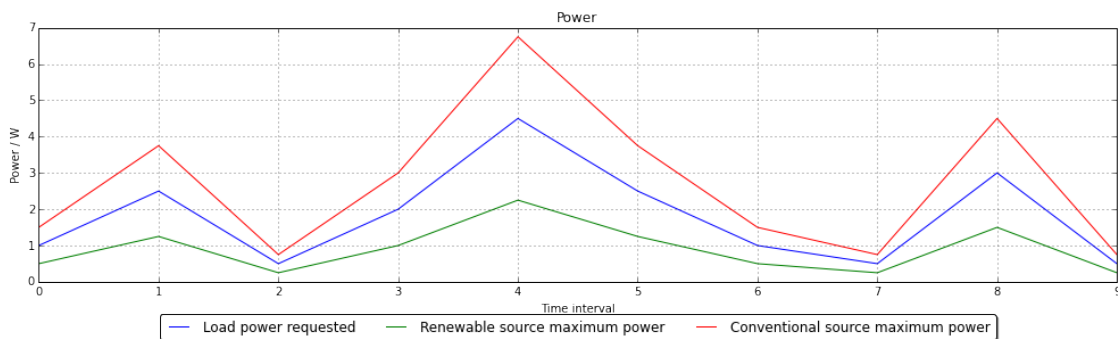
    con_source = pyEMS.elements.Source(name='Conventional source',
                                       power_min=[0] * time_intervals,
                                       power_max=1.5 * load.requested,
                                       cost=[random.uniform(1, 0) * unit_cost for _ in range(time_

    res_source = pyEMS.elements.Source(name='Renewable source',
                                       power_min=[0] * time_intervals,
                                       power_max=0.5 * load.requested,
                                       cost=con_source.cost)

    node_source.add_element(load)
    node_source.add_element(con_source)
    node_source.add_element(res_source)

In [42]: plot_power(node_source)
         plot_cost(node_source)

```



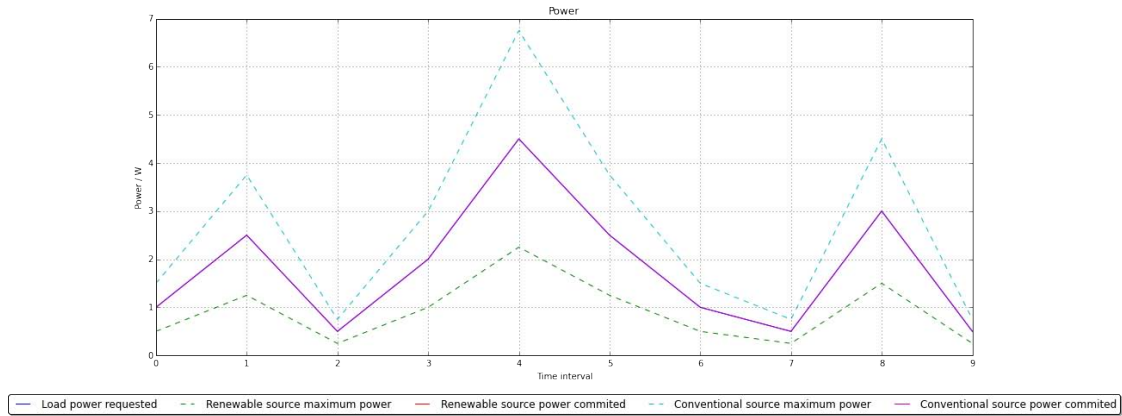
```
In [43]: pyEMS.scheduling.unit_commitment(window, node_source)
```

Status: Optimal

Solution time: 0.03711299999999973 s

Cost: 6.67555463904

In [44]: `plot_power_max_requested(node_source)`



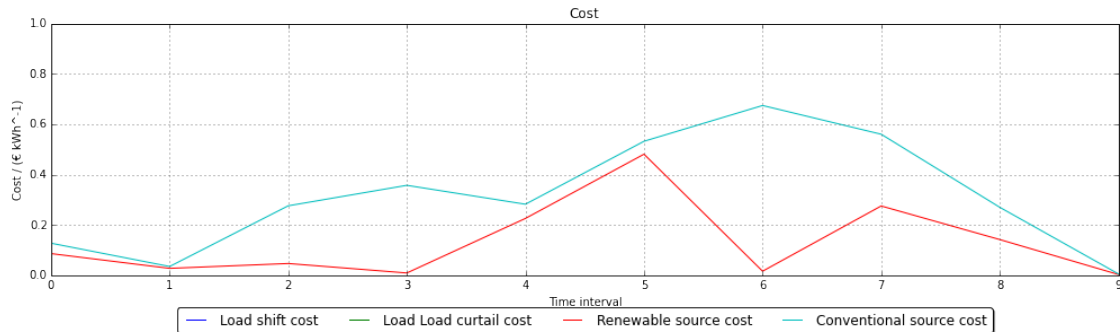
The plot shows that the Unit Commitment algorithm chose to use only the conventional source, which is capable of feed the load alone, because the costs of the two source is equal.

### 1.2.1 Renewable source priority

To obtain a priority of the renewable source, the unit cost of this source is set lower than the conventional source by a random real number  $x \in [0, 1]$  with uniform distribution.

```
In [45]: con_source.cost = [random.uniform(1, 0) * unit_cost for _ in range(time_intervals)]
         res_source.cost = [p * random.uniform(0, 1) for p in con_source.cost]
```

In [46]: `plot_cost(node_source)`



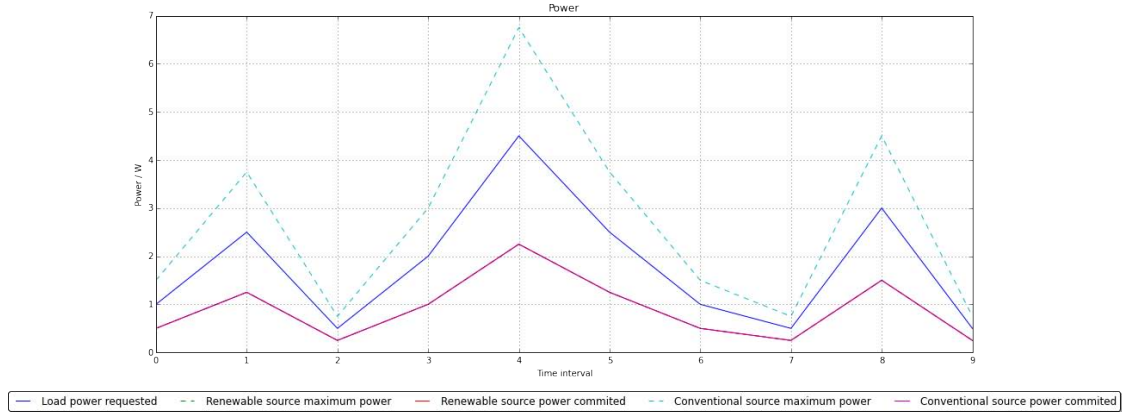
In [47]: `pyEMS.scheduling.unit_commitment(window, node_source)`

Status: Optimal

Solution time: 0.0351600000000119 s

Cost: 4.22153654987

In [48]: `plot_power_max_requested(node_source)`



The plot shows that the Unit Commitment algorithm chose to use all the renewable potential (the line of the Renewable source power is on the top of the Renewable source maximum power) and then part of the Conventional source so that the total sources power is align with the load power request.

### 1.2.2 Use of all the renewable potential

To show that a storage system can increase the renewable source energy, it is necessary to set a load power profile which is lower of the renewable source in some time intervals and higher than it in other time intervals: the storage system will act as an energy time buffer between the power generation and the load request.

The storage system object is created with the **Storage** Class of the **elements** module. It requires a name, power bounds, cost and efficiency of the discharge and charge processes for each time interval, the energy bounds of the storage and its initial energy.

```
In [49]: res_source.power_max = random.permutation(load.requested)
```

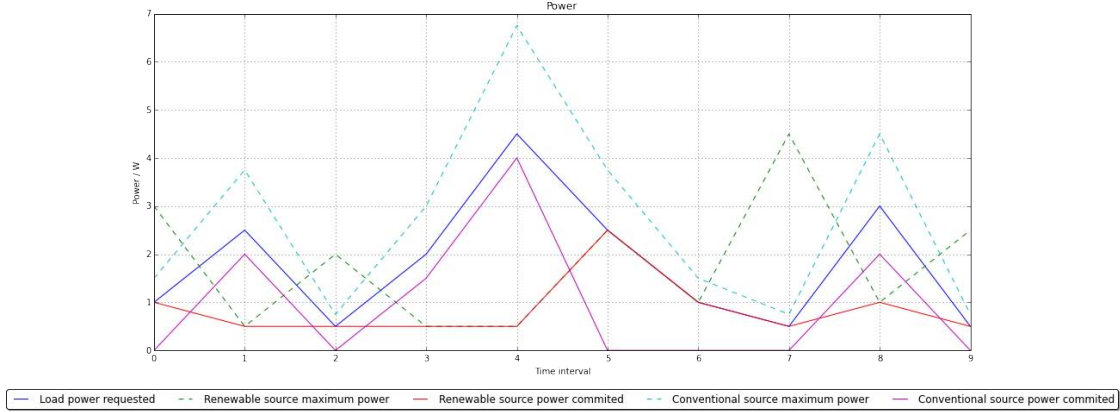
```
storage = pyEMS.elements.Storage(name='Storage',
                                   power_dsc_min=[0] * time_intervals,
                                   power_dsc_max=[100] * time_intervals,
                                   power_ch_min=[0] * time_intervals,
                                   power_ch_max=[100] * time_intervals,
                                   cost_dsc=[0] * time_intervals,
                                   cost_ch=[0] * time_intervals,
                                   efficiency_dsc=1.0,
                                   efficiency_ch=1.0,
                                   energy_min=0,
                                   energy_max=100000000,
                                   energy_start=0)
```

The first Unit Commitment is done without any storage system.

```
In [50]: pyEMS.scheduling.unit_commitment(window, node_source)
```

```
Status: Optimal
Solution time: 0.0347179999999803 s
Cost: 4.02109507258
```

```
In [51]: plot_power_max_requested(node_source)
```



It is possible to notice that without the storage system it is not assured that all the possible renewable storage is utilized: if in a time interval the load power is higher of the renewable source potential, the conventional source is required while if the load is lower than the renewable source potential all the power between the source potential and the actual source power is wasted.

The storage system is insered in the system.

```
In [52]: node_source.add_element(storage)
```

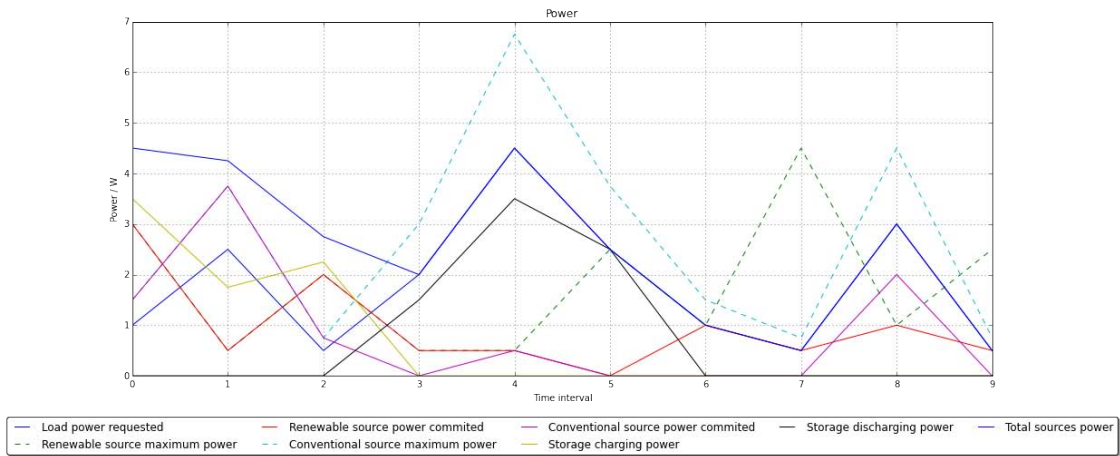
```
pyEMS.scheduling.unit_commitment(window, node_source)
```

Status: Optimal

Solution time: 0.048408000000002005 s

Cost: 1.99775831797

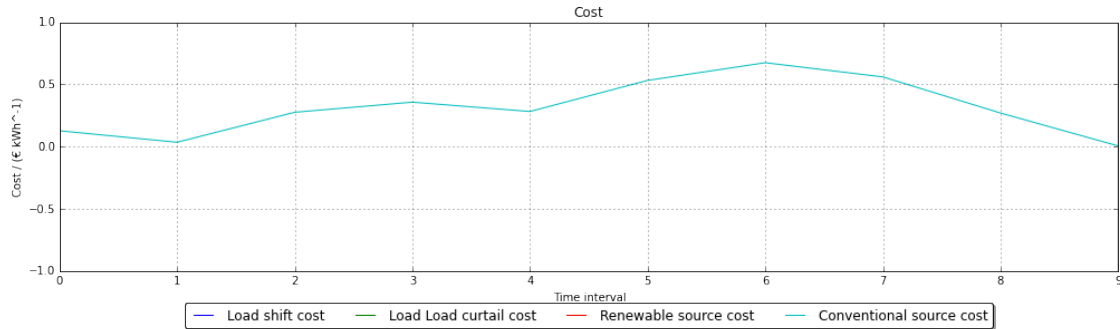
```
In [53]: plot_power_max_requested(node_source)
```



The plot shows that the renewable potential is not completely used:

```
In [54]: res_source.cost = [-1] * time_intervals
```

```
In [55]: plot_cost(node_source)
```



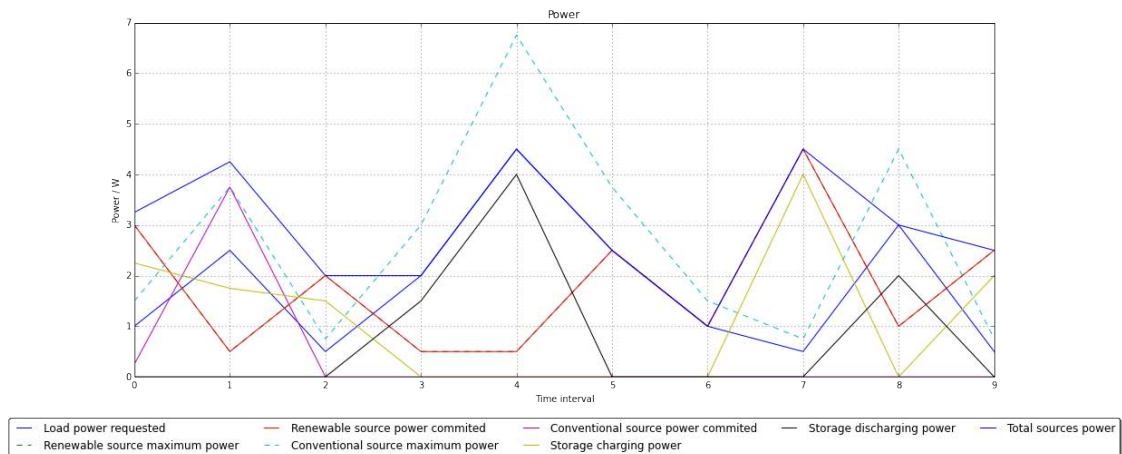
```
In [56]: pyEMS.scheduling.unit_commitment(window, node_source)
```

Status: Optimal

Solution time: 0.04520199999999974 s

Cost: -17.8354724686

```
In [57]: plot_power_max_requested(node_source)
```



The last plot shows that if the cost of the renewable source is set negative, that is a profit not a cost, the algorithm uses all the renewable resource. Obviously setting the renewable cost as a profit makes not comparable the total cost of the last optimization with the total costs of the past optimization and it does not mean it is the best economic choice.

```
In [58]: %load_ext watermark
          %watermark -a "Nicola Tomasone" -u -e -t -v -m -g -p numpy,scipy,pandas,pyEMS
```

Nicola Tomasone

Last updated: 20/09/2015 22:39:24

CPython 3.4.3

IPython 4.0.0

numpy 1.9.2

scipy 0.16.0  
pandas 0.16.2  
pyEMS 0.1

compiler : GCC 4.2.1 (Apple Inc. build 5577)  
system : Darwin  
release : 10.8.0  
machine : i386  
processor : i386  
CPU cores : 2  
interpreter: 64bit  
Git hash : f905ffeca27f8dcd4ab4ea78a9cd672475e8d

# Annex Photovoltaic model

September 20, 2015

```
In [1]: import numpy as np
import pandas as pd
import pvlib
import pytz
import pickle

import pyEMS
%matplotlib inline
```

## 0.1 Import data from GECAD database

The meteorological data are collected by a Davis Vantage Pro2 Plus device installed in the facility and stored in the local private GECAD database and in the public databases of [MeteoISEP](#) and [WeatherUnderground](#).

The electrical data are collected by a PLC in the GECAD renewable laboratory (aka laboratory F514) from the instruments of the laboratory and saved in the local GECAD database.

The photovoltaic power regulator device is directly connected to the storage system which acts as Direct Current (DC) bus, thus the output of the photovoltaic system depends on the voltage of the storage; if storage has a high State of Charge the power output of the photovoltaic system is limited. In order to obtain useful data the query is done on day 2015-09-03 when an experiment was performed.

In order to deal with the Daylight saving time (DST) the recorded data are queried with the UTC time, thus the time is first localized to the Europe/Lisbon time-zone and then converted to the UTC time.

```
In [2]: utc = pytz.utc
local = pytz.timezone('Europe/Lisbon')
```

```
In [3]: time_interval = (local.localize(pd.datetime(2015, 9, 3, 0, 0)).astimezone(utc),
local.localize(pd.datetime(2015, 9, 3, 23, 59)).astimezone(utc))
```

```
In [4]: with open('../shared_files/connection_string.pickle', 'rb') as f:
connection_string = pickle.load(f)
```

```
In [5]: #try:
df_weather = pyEMS.gecad_tools.weather_data(time_interval, connection_string)
#except:
#    df_weather = pd.read_csv('Photovoltaic_model_test_df_weather.csv', index_col='DateUTC',
#    # find how to handle timezone in readed csv file
#    # because now it has no timezone and the later
#    # localization fail
#try:
df_electrical = pyEMS.gecad_tools.electrical_data(time_interval, connection_string)
#except:
#    df_electrical = pd.read_csv('Photovoltaic_model_test_df_electrical.csv', index_col='Da
```

## 0.2 Define the system space and time coordinates

The evaluation algorithms of the Direct Normal Irradiance (BNI) and Direct Irradiance on the module (BMI) required information on the local geographical position, local time and modules orientation to calculate the relative position between Sun and photovoltaic modules.

```
In [6]: loc = pvlib.location.Location(name='ISEP', latitude=41.179, longitude=-8.607, tz='Europe/Lisbon')
        times = df_weather.index
        times_loc = times.tz_convert(loc.pytz)
        surface_tilt = 40
        surface_azimuth = 188 # check this value
```

### 0.2.1 Weather data

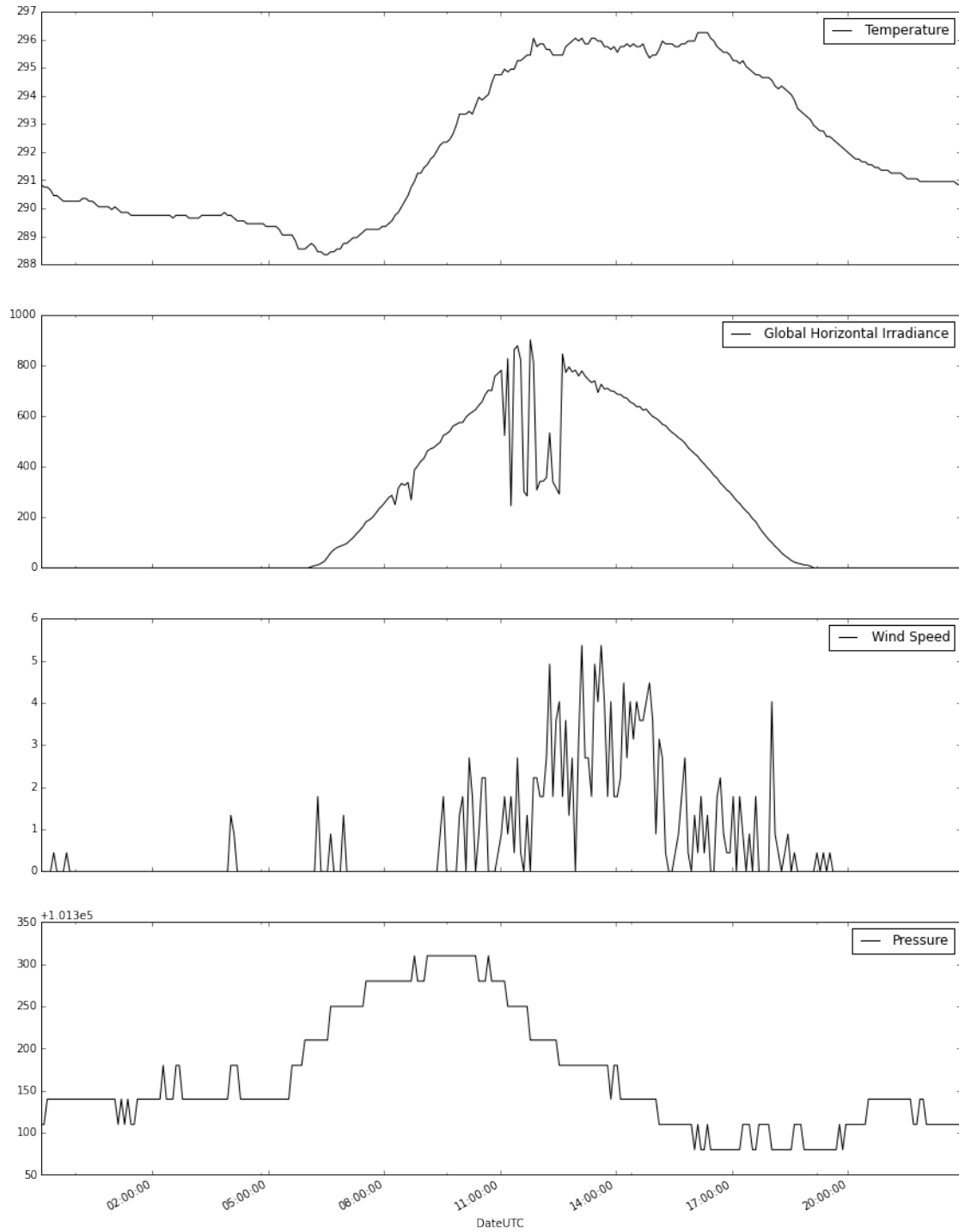
```
In [7]: df_weather.describe()
```

```
Out[7]:
```

	Temperature	Global Horizontal Irradiance	Wind Speed	Pressure
count	284.000000	284.000000	284.000000	284.000000
mean	292.156690	225.183099	0.671264	101465.739437
std	2.604886	278.143088	1.226417	68.691576
min	288.350000	0.000000	0.000000	101380.000000
25%	289.750000	0.000000	0.000000	101410.000000
50%	291.350000	35.000000	0.000000	101440.000000
75%	295.075000	462.500000	0.888889	101510.000000
max	296.250000	902.000000	5.361111	101610.000000

```
In [8]: df_weather.plot(figsize=(15,21), subplots=True, legend=True)
```

```
Out[8]: array([<matplotlib.axes._subplots.AxesSubplot object at 0x1094202e8>,
               <matplotlib.axes._subplots.AxesSubplot object at 0x109188240>,
               <matplotlib.axes._subplots.AxesSubplot object at 0x109111128>,
               <matplotlib.axes._subplots.AxesSubplot object at 0x1094a6470>], dtype=object)
```



The **Global Horizontal Irradiance** has almost the classical shape of a clear sky condition except for the central hours of the day where the clouds effect is noticeable.

## 0.2.2 Electrical data

The recorded electrical values for the photovoltaic generator are the DC bus voltage F1\_U1 and the DC output current of the photovoltaic power device F1\_I1, then DC power is then indirectly obtain multiplying the two measures.

```
In [9]: df_electrical.dropna().head(3)
```

```
Out[9]:
```

	P load pv	P load wind	P source pv	P battery pv \
TimeCol				
2015-09-02 23:00:10	0	0	0.0000	69.0000
2015-09-02 23:00:20	0	0	0.7995	68.2005
2015-09-02 23:00:30	0	0	0.7995	68.2005

	P source wind	P battery wind	U battery pv \
TimeCol			
2015-09-02 23:00:10	0	-50	26.65
2015-09-02 23:00:20	0	-50	26.65
2015-09-02 23:00:30	0	-50	26.65

	U battery wind	I battery pv	I battery wind
TimeCol			
2015-09-02 23:00:10	26.75	2.589118	-1.869159
2015-09-02 23:00:20	26.80	2.559118	-1.865672
2015-09-02 23:00:30	26.80	2.559118	-1.865672

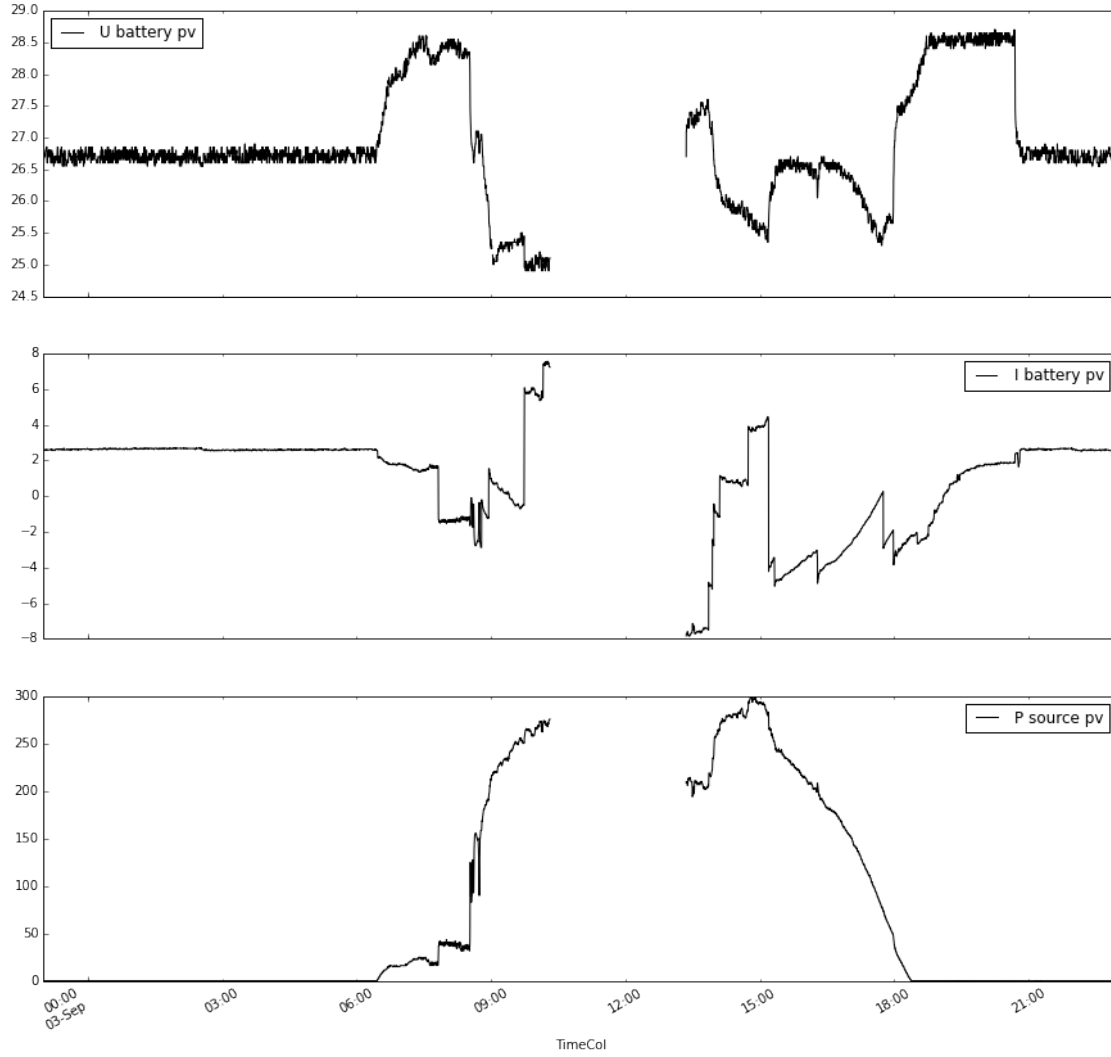
```
In [10]: df_electrical[['U battery pv', 'I battery pv', 'P source pv']].describe()
```

```
Out[10]:
```

	U battery pv	I battery pv	P source pv
count	7530.000000	7529.000000	7530.000000
mean	26.869396	0.915107	67.699802
std	0.906833	2.731091	101.289737
min	24.900000	-7.830000	0.000000
25%	26.600000	-0.869545	0.801000
50%	26.700000	2.544627	0.853500
75%	27.150000	2.619254	152.856000
max	28.700000	7.553513	299.022000

```
In [11]: df_electrical[['U battery pv', 'I battery pv', 'P source pv']].plot(figsize=(15, 15), subplots=
```

```
Out[11]: array([<matplotlib.axes._subplots.AxesSubplot object at 0x10172cba8>,  
                <matplotlib.axes._subplots.AxesSubplot object at 0x106dd28d0>,  
                <matplotlib.axes._subplots.AxesSubplot object at 0x1092b7080>], dtype=object)
```



From the above plots of the Global Horizontal Irradiance (ghi) and the DC power output pv power recorded can be noticed that the DC power have not the same trend until time 8:30 UTC: the former has a smoother trend while the latter has a step at the aforementioned time instant; the reason, in accordance with what written in the beginnign, is that the experiment was started at that time.

In addition the values between 10:30 UTC and 13:50 UTC are missing due to a ethernet connection problem in the ISEP net.

### 0.3 Evaluate the Sun position for the given time interval

The first point is the evaluation of Sun position during the time of analysis. The algorithm used is the *NREL's SPA* presented in [Reda, I.; Andreas, A. (2003). Solar Position Algorithm for Solar Radiation Applications. 55 pp.; NREL Report No. TP-560-34302, Revised January 2008]; it requires the local time, therefore the input time is in local time then the output is converted to UTC time.

```
In [12]: sun_position = pvlib.solarposition.get_solarposition(time=times_loc, location=loc, method='nrel')
         sun_position.index = sun_position.index.tz_convert('UTC')
```

## 0.4 Evaluate irradiance components

### 0.4.1 Direct normal irradiance (bni)

The direct normal irradiance (bni), the irradiation received per unit area by a surface that is always held perpendicular to the rays that come in a straight line from the direction of the sun at its current position in the sky, is computed by means of the DISC model.

```
In [13]: disc_model = pvlib.irradiance.disc(ghi=df_weather['Global Horizontal Irradiance'],
                                             zenith=sun_position['zenith'],
                                             times=times,
                                             pressure=df_weather['Pressure'])

In [14]: irradiance = df_weather.copy().drop(['Temperature', 'Wind Speed', 'Pressure'], axis=1)

In [15]: irradiance.rename(columns={'Global Horizontal Irradiance': 'ghi'}, inplace=True)

In [16]: irradiance['bni'] = disc_model['dni'].copy()
```

### 0.4.2 Direct horizontal irradiance (bhi)

The direct horizontal irradiance is computed

```
In [17]: irradiance['bhi'] = disc_model['dni'] * pvlib.tools.cosd(sun_position['zenith'])
```

### 0.4.3 Direct diffuse irradiance (dhi)

```
In [18]: irradiance['dhi'] = irradiance['ghi'] - irradiance['bhi']
```

### 0.4.4 Direct module irradiance (bmi)

```
In [19]: irradiance['bmi'] = pvlib.irradiance.beam_component(surface_tilt=surface_tilt,
                                                             surface_azimuth=surface_azimuth,
                                                             solar_zenith=sun_position['zenith'],
                                                             solar_azimuth=sun_position['azimuth'],
                                                             dni=disc_model['dni'])
```

### 0.4.5 Diffuse module irradiance (dmi)

The diffuse module irradiance (dmi) is evaluated by means of the isotropic model.

```
In [20]: irradiance['dmi'] = pvlib.irradiance.isotropic(surface_tilt=40,
                                                         dhi=irradiance['dhi'])
```

### 0.4.6 Reflected module irradiance (rmi)

The reflected module irradiance (rmi) is evaluated for a surface with a reflective coefficient  $\text{albedo} = 0.25$ .

```
In [21]: irradiance['rmi'] = pvlib.irradiance.grounddiffuse(surface_tilt=surface_tilt,
                                                             ghi=irradiance['ghi'],
                                                             albedo=0.25,
                                                             surface_type=None)
```

### 0.4.7 Global module irradiance (gmi)

The sum of the three module irradiances gives the total irradiance over the module surface for each time instant.

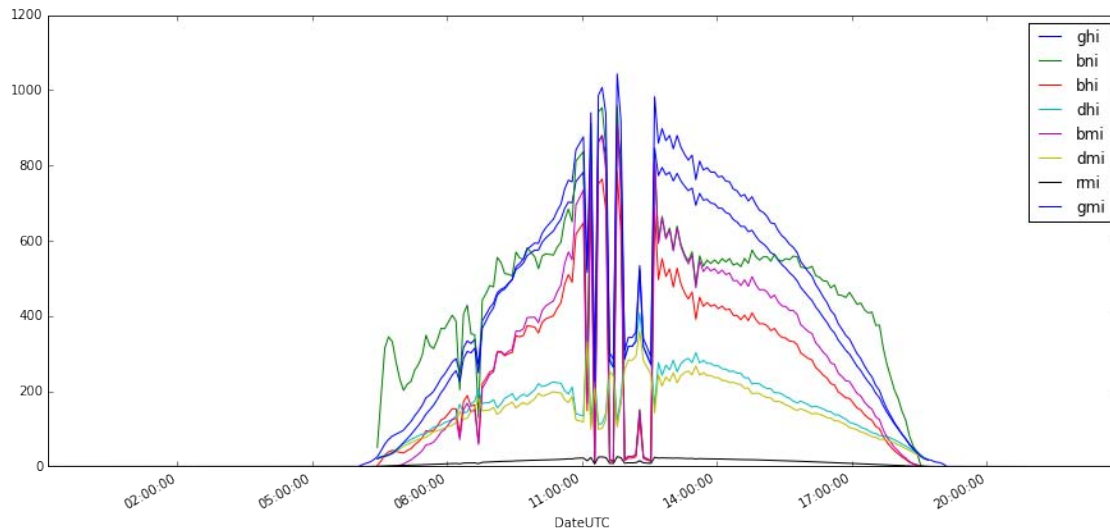
```
In [22]: irradiance['gmi'] = irradiance['bmi'] + irradiance['dmi'] + irradiance['rmi']
```

#### 0.4.8 Plots of irradiances

All the irradiance:

```
In [23]: irradiance.plot(figsize=(15, 7))
```

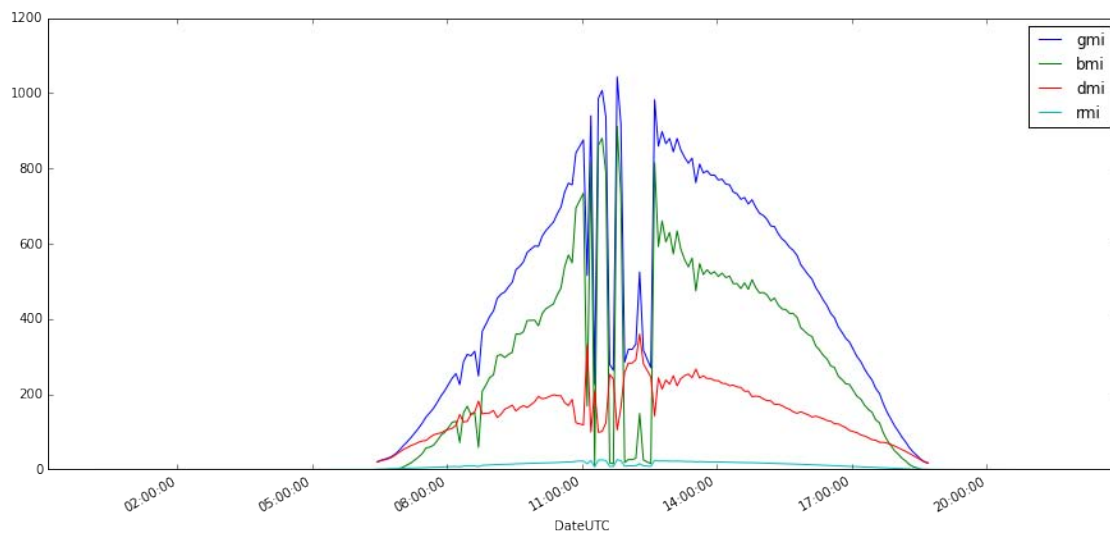
```
Out[23]: <matplotlib.axes._subplots.AxesSubplot at 0x10172c438>
```



Only the modules irradiances:

```
In [24]: irradiance[['gmi', 'bmi', 'dmi', 'rmi']].plot(figsize=(15, 7))
```

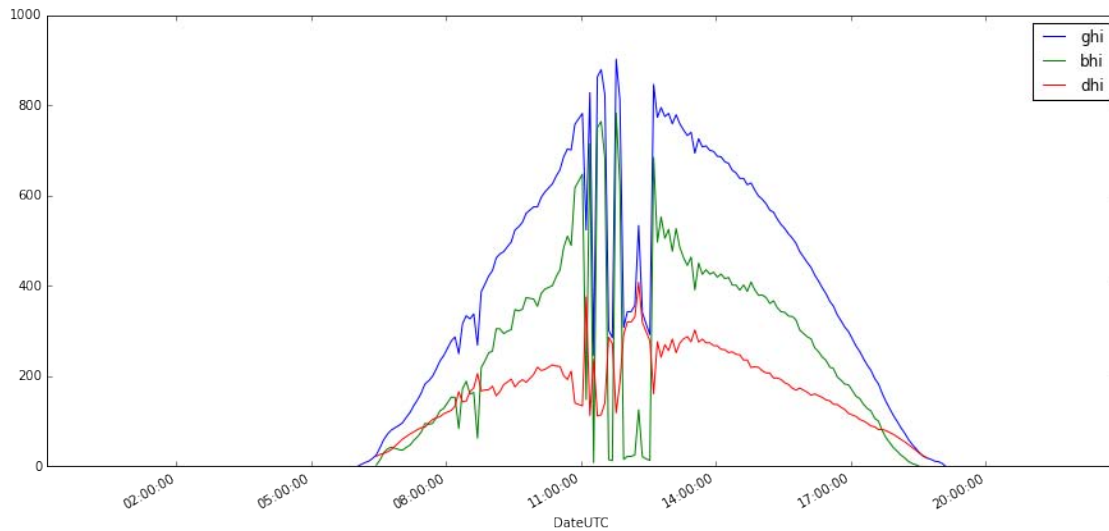
```
Out[24]: <matplotlib.axes._subplots.AxesSubplot at 0x109b2f550>
```



Only the horizontal irradiances:

```
In [25]: irradiance[['ghi', 'bhi', 'dhi']].plot(figsize=(15, 7))
```

Out [25]: <matplotlib.axes.\_subplots.AxesSubplot at 0x1047966d8>



## 0.5 Evaluate power output of photovoltaic system

The power of the photovoltaic system is evaluated with the PVGIS power performance model.

```
In [26]: n_modules = 3
         power_std = 200 # [W]
         module_surface = 1.425 * 0.990 # [m^2]
         array_surface = module_surface * n_modules
         print('The total surface of the pv modules is:', array_surface, '[m^2]')
```

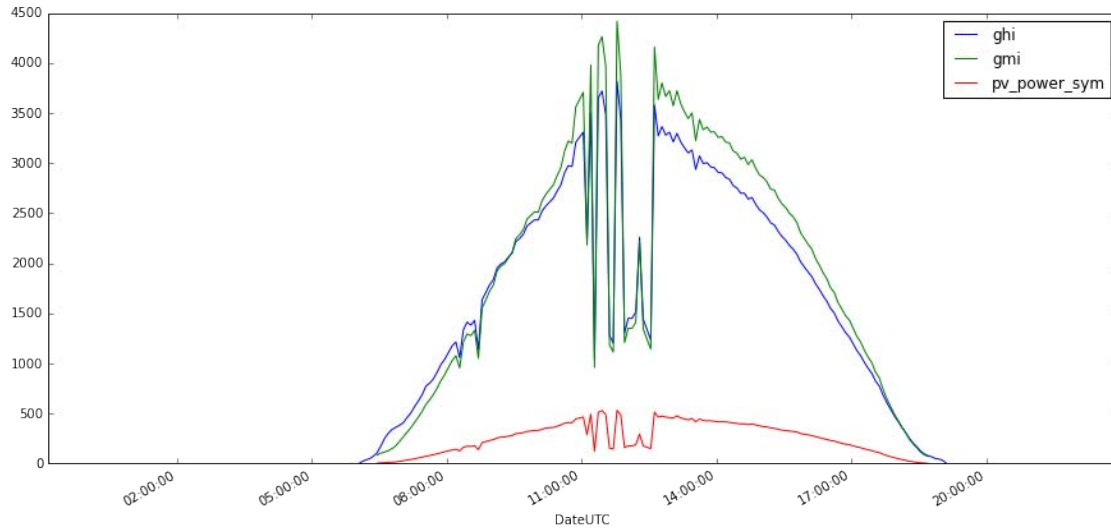
The total surface of the pv modules is: 4.23225 [m<sup>2</sup>]

```
In [27]: pv_power_sym = pyEMS.models.photovoltaic_generator(gmi=irradiance['gmi'],
                                                            temp_amb=df_weather['Temperature'],
                                                            wind_speed=df_weather['Wind Speed'],
                                                            power_std=power_std,
                                                            n_modules=n_modules,
                                                            power_conv_efficiency=1.0)
```

To estimate the efficiency of the simulated photovoltaic system the maximum power reaching the modules is calculated multiplying the irradiance for the total surface of the photovoltaic modules.

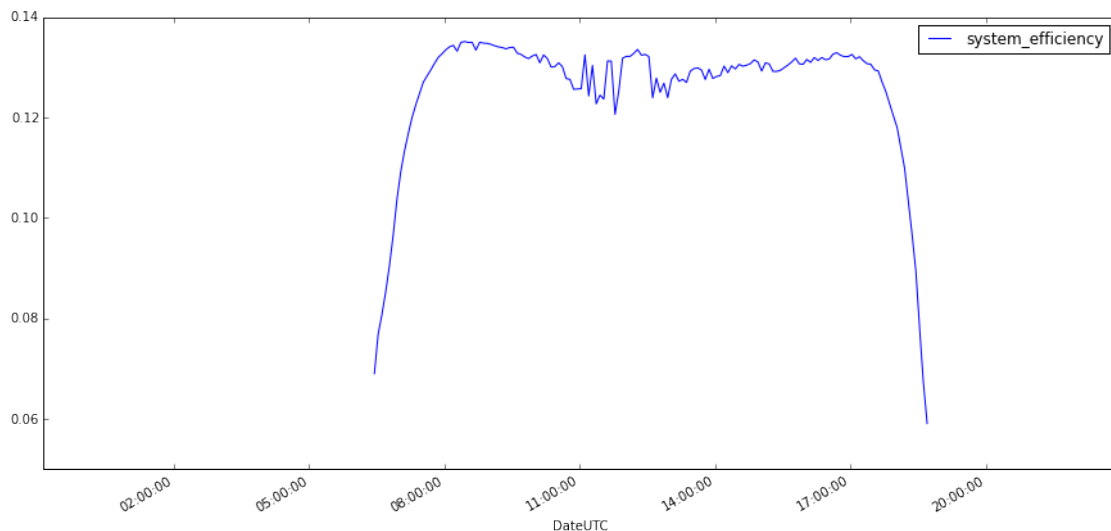
```
In [28]: (irradiance['ghi'] * array_surface).plot(figsize=(15,7), legend=True) # Solar power on the mo
         (irradiance['gmi'] * array_surface).plot(figsize=(15,7), legend=True)
         pv_power_sym.plot(figsize=(15,7), legend=True)
```

Out [28]: <matplotlib.axes.\_subplots.AxesSubplot at 0x104789898>



```
In [29]: eff_pv_system = pd.Series(pv_power_sym / (irradiance['gmi'] * array_surface), name='system_eff')
         eff_pv_system.plot(figsize=(15,7), legend=True)
```

```
Out[29]: <matplotlib.axes._subplots.AxesSubplot at 0x109dcf630>
```

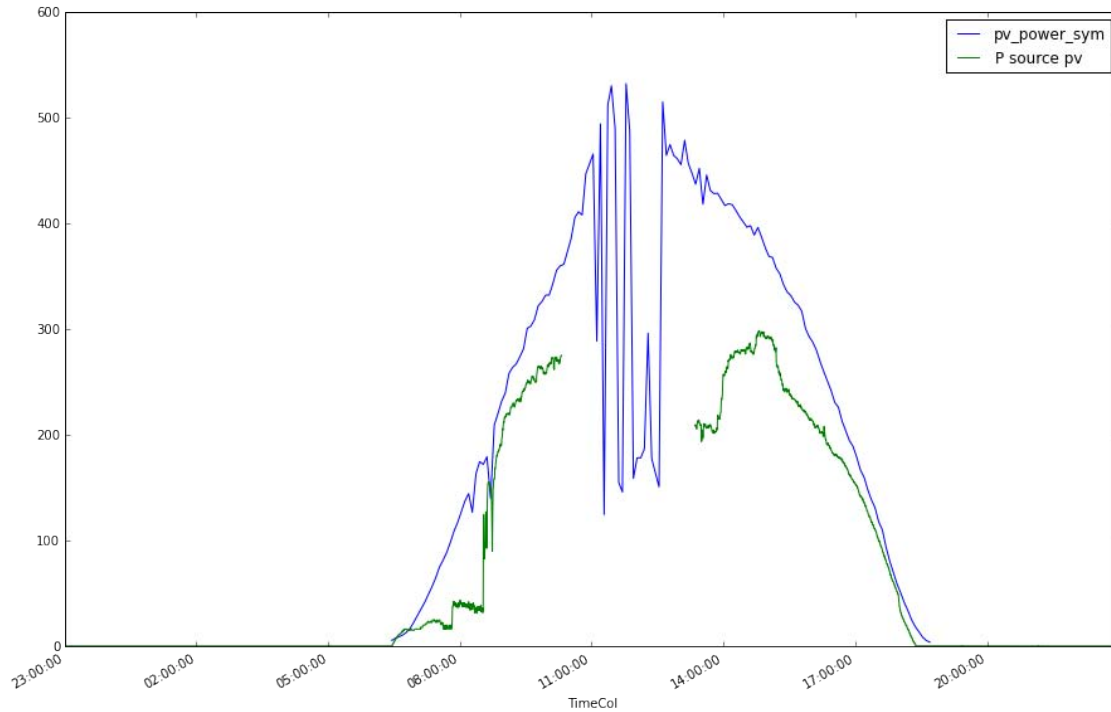


The efficiency is between the 12% and 14% during the hours with light, the result is resonable for the crystalline silicon technology of the modules.

## 0.6 Evaluate the power recorded by the database for the pv

```
In [30]: pv_power_sym.plot(figsize=(15,10), legend=True)
         df_electrical['P source pv'].plot(figsize=(15,10), legend=True)
```

```
Out[30]: <matplotlib.axes._subplots.AxesSubplot at 0x109dfb0f0>
```



Looking to the plot above, it is important to highlight two aspects:

1. the central hours are not useful because of missing electrical data, the sharp step down is given by a clud from in the simulated power and by missing data in the recorded power,
2. the maximum power of the recorded power is 300 W, instead of more than 500 W of the simulated, because the maximum load in the laboratory is of 300 W.

The only comparable quantities are the trends of the two quantities, trends that are similar.

```
In [31]: %load_ext watermark
         %watermark -a "Nicola Tomasone" -u -e -t -v -m -g -p numpy,scipy,pandas,pvlib,pytz
```

Nicola Tomasone

Last updated: 20/09/2015 22:40:10

CPython 3.4.3

IPython 4.0.0

numpy 1.9.2

scipy 0.16.0

pandas 0.16.2

pvlib 0.2.1

pytz 2015.4

compiler : GCC 4.2.1 (Apple Inc. build 5577)

system : Darwin

release : 10.8.0

machine : i386

processor : i386

CPU cores : 2  
interpreter: 64bit  
Git hash : f905ffeca27f8dcd4ab4ea78a9cd672475e8d

# Annex NRFM

September 20, 2015

```
In [1]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
import sys
sys.path.append('../code')
import pyEMS
```

## 1 Comparison between the New Reference Forecast Model and the Persistence Model

In the following notebook the New Reference Forecasted Model (NRFM), proposed in [NJM+98] as replacement of the Persistence model for the benchmark of more complex forecasting models, is compared with the latter method through the use of the Root Mean Square of the error.

The comparison was done using two data set (training set and testing set) of the wind speed recorded by the local weather station as it is recommended in [Mad04]. The training set is composed by forty-eight hours records while the testing set is composed by the twenty-four hours records; both sets have the same sampling frequency of 0,003 Hz. The wind speed samples are used to computed the wind power production through the wind generator model, then the RMSs of the forecasted series created with the three models and the real series are computed.

```
In [2]: def nrm_correlation(TimeSeries, k, x_mean):
        """
        Compute the correlation factor of the New Reference Model.

        Parameters
        -----
        TimeSeries : numpy.array
            Time series of recorded values.
        k : float
            Total number of required correlation coefficients.
        x_mean : float
            Mean value to use in the NRFM.

        Returns
        -----
        a_k : numpy.array
            Array of correlation factors for the 'TimeSeries' input.

        """
        N = np.size(TimeSeries)
        assert k <= N, "k must be less than TimeSeries lenght N "
```

```

if k == N:
    a_k = 0.0
else:
    diff_vec = TimeSeries - (np.ones(N) * x_mean)

    a_k = np.dot(diff_vec[0:(N - 1 - k)], diff_vec[k: (N - 1)]) / np.sum(diff_vec[: (N - 1 - k)])

return a_k

def nrm(x_t, TimeSeries, k):
    """
    Compute the wind power forecast following the New Reference Model.

    Parameters
    -----
    x_t : float
        Actual value of the aleatory variable for which the forecast asked.
    TimeSeries : numpy.array
        Time series of recorded values.
    k : float
        Total number of required correlation coefficients.

    Returns
    -----
    x_t_k : numpy.array
        Forecasted values of the aleatory variable for the future k-th instant.
    """
    x_mean = np.mean(TimeSeries)
    a_k = nrm_correlation(TimeSeries, k, x_mean)
    x_t_k = a_k * x_t + (1 - a_k) * x_mean

    return x_t_k

def RMS(a, b, k):
    """
    Compute the MSE of 'a' over 'b' inputs up to the 'k' sample.

    Parameters
    -----
    a, b : numpy.array
        Two series for which the RMS is computed.
    k : float
        Length of the required RMS.

    Returns
    -----
    RMS : numpy.array
        RMS up to the 'k' sample.
    """
    size = np.size(k)
    RMS = np.zeros(size)
    for k in range(0, size):
        RMS[k] = np.sqrt(np.mean((a[0: k] - b[0: k]) ** 2))

```

```
return RMS
```

## 1.1 Input data

The training set is composed of 48 hour samples of the wind speed between 2015-04-06 00:00 and 2015-04-07 23:59, while the test set is composed of 24 hour samples of the wind speed of day 2015-04-08 00:00.

```
In [3]: try:
```

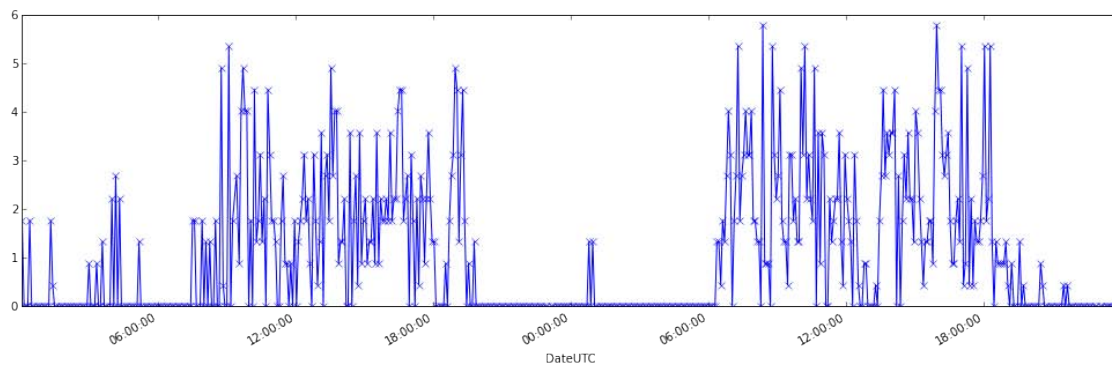
```
    data_training = pyEMS.gecad_tools.weather_data(pd.datetime(2015, 4, 6, 0, 0),
                                                    pd.datetime(2015, 4, 7, 23,59))
```

```
except:
```

```
    data_training = pd.read_csv('NRFM_notebook_data_training.csv', index_col='DateUTC', parse_d
```

```
    data_training['Wind Speed'].plot(figsize=(17,5), style='-x')
```

```
Out[3]: <matplotlib.axes._subplots.AxesSubplot at 0x108f56f98>
```



```
In [4]: try:
```

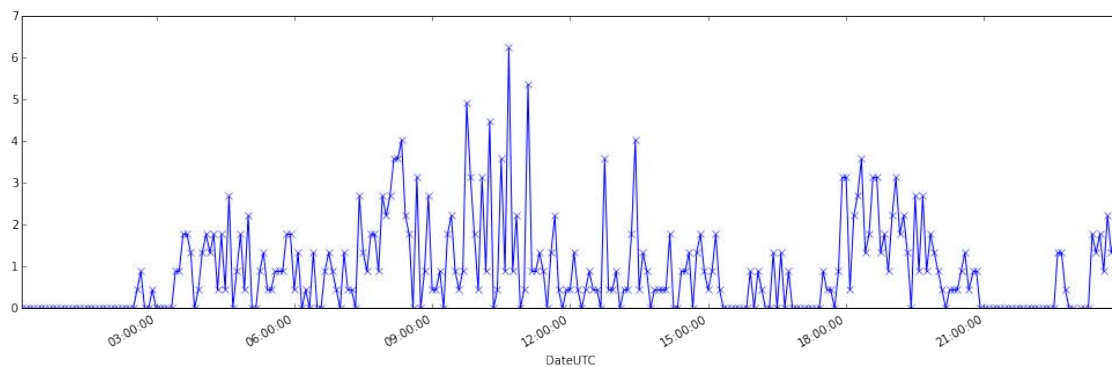
```
    data_testing = pyEMS.gecad_tools.weather_data(pd.datetime(2015, 4, 8, 0, 0),
                                                    pd.datetime(2015, 4, 8, 23,59))
```

```
except:
```

```
    data_testing = pd.read_csv('NRFM_notebook_data_testing.csv', index_col='DateUTC', parse_d
```

```
    data_testing['Wind Speed'].plot(figsize=(17,5), style='-x')
```

```
Out[4]: <matplotlib.axes._subplots.AxesSubplot at 0x109064e48>
```



Statistical data of the training set:

```
In [5]: print('Training:', data_training['Wind Speed'].describe())
```

```
Training: count      571.000000
mean         1.056188
std          1.433739
min          0.000000
25%          0.000000
50%          0.000000
75%          1.777778
max          5.805556
Name: Wind Speed, dtype: float64
```

Statistical data of the test set:

```
In [6]: print('Testing:', data_testing['Wind Speed'].describe())
```

```
Testing: count      286.000000
mean         0.883741
std          1.088803
min          0.000000
25%          0.000000
50%          0.444444
75%          1.333333
max          6.250000
Name: Wind Speed, dtype: float64
```

## 1.2 Evaluation of the output wind power

The two wind speed series are used as inputs of the wind generator model to compute the resulting DC power.

```
In [7]: wind_training = np.reshape(data_training['Wind Speed'].values, (np.size(data_training['Wind Speed']),))
p_training = pyEMS.models.wind_generator(wind_training)

wind_testing = np.reshape(data_testing['Wind Speed'].values, (np.size(data_testing['Wind Speed']),))
p_testing = pyEMS.models.wind_generator(wind_testing)
```

## 1.3 RMS for minimum wind speed

```
In [8]: start = p_testing.argmax()

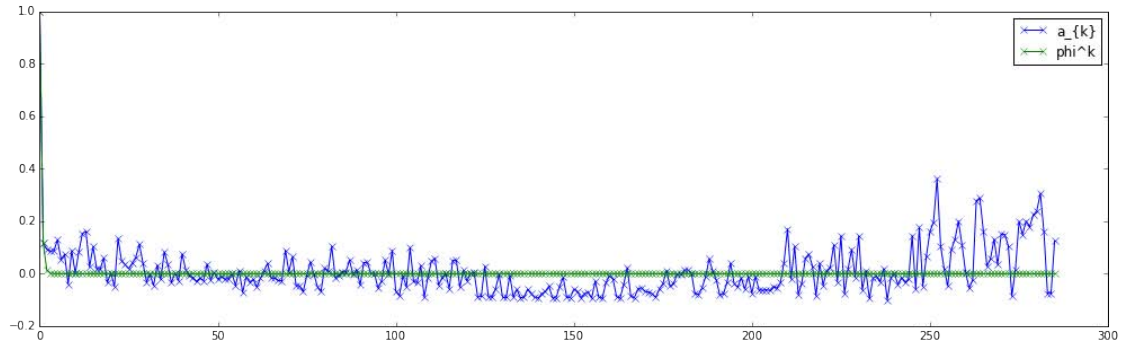
end = np.size(p_testing)
resolution = end - start
time_range = range(start, end)
k = np.arange(0, resolution)
```

Evaluation of the correlation coefficients  $a_k$  for all the time series and with a constant forecast window from the training set (see [NJM+98]).

```
In [9]: a_k = [nrm_correlation(p_training, k, np.mean(p_training))] for k in range(0, resolution)
phi = [nrm_correlation(p_training[k:], 1, np.mean(p_training[k:]))**k for k in range(0, resolution)]
```

```
fig = plt.figure(figsize=(17, 5))
plt.plot(a_k, '-x', label='a_{k}')
plt.plot(phi, '-x', label='phi^k')
plt.legend()
```

Out[9]: <matplotlib.legend.Legend at 0x109c7d160>



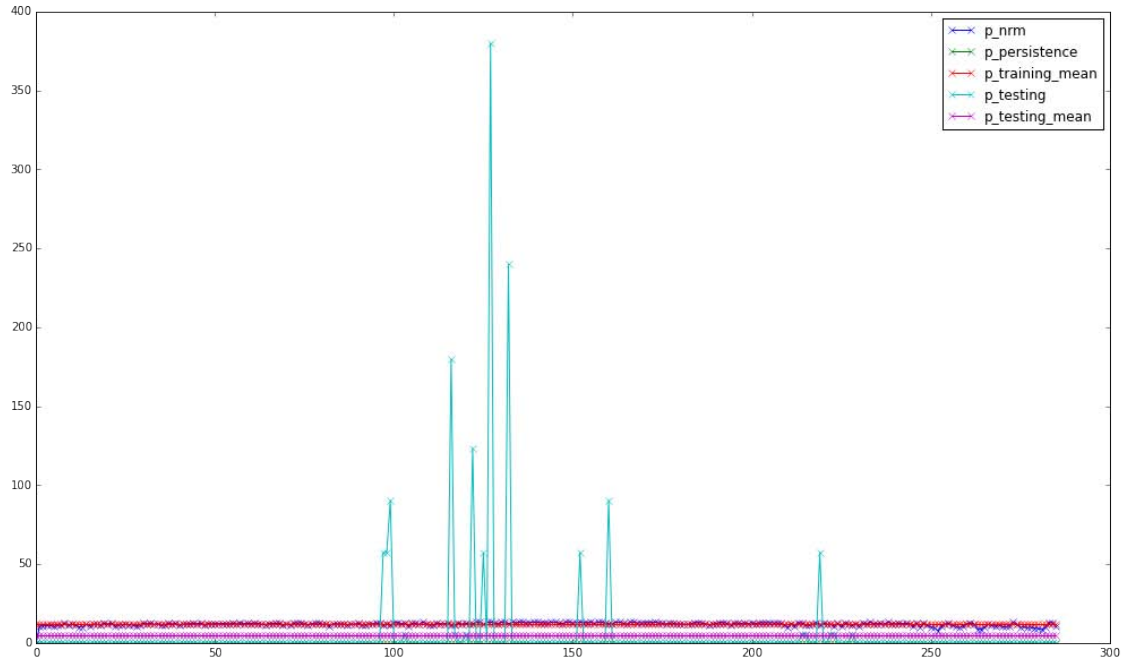
Evaluation of the forecasted values for the three models.

```
In [10]: p_start = p_testing[start]
print('v_start:', wind_testing[start], 'm s^-1')
p_nrm = np.array([nrm(p_start, p_training, k) for k in range(0, resolution)])
p_persistence = np.ones(np.size(p_nrm)) * p_start
p_testing_mean = np.ones(np.size(p_nrm)) * np.mean(p_testing)
p_training_mean = np.ones(np.size(p_nrm)) * np.mean(p_training)

fig = plt.figure(figsize=(17, 10))
plt.plot(p_nrm, '-x', label='p_nrm')
plt.plot(p_persistence, '-x', label='p_persistence')
plt.plot(p_training_mean, '-x', label='p_training_mean')
plt.plot(p_testing, '-x', label='p_testing')
plt.plot(p_testing_mean, '-x', label='p_testing_mean')
plt.legend()
```

v\_start: 0.0 m s<sup>-1</sup>

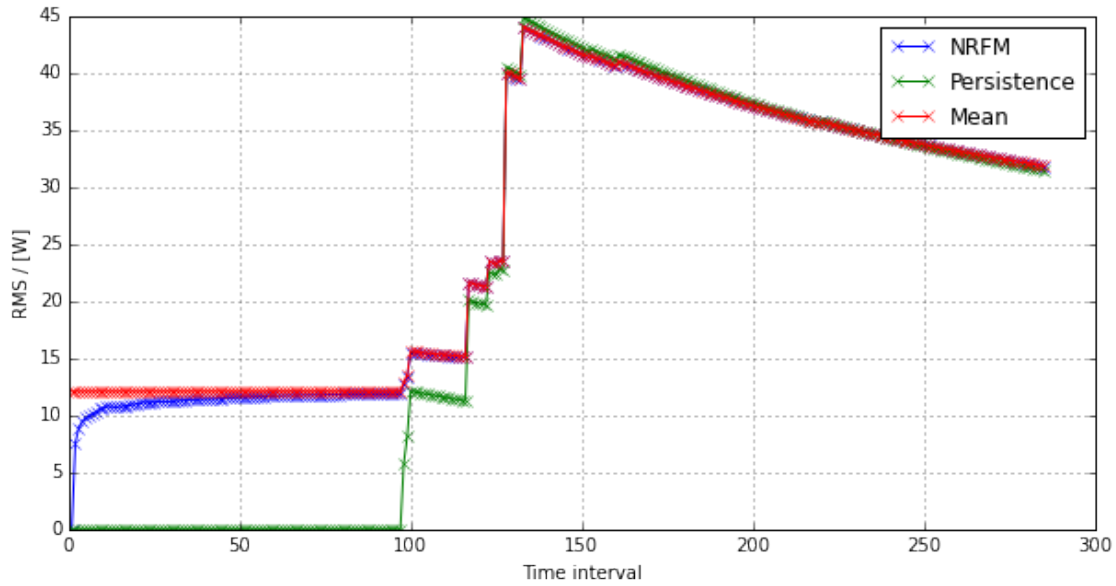
Out[10]: <matplotlib.legend.Legend at 0x109e31b38>



```
In [11]: RMS_nrm = RMS(p_nrm, p_testing, k)
        RMS_persistence = RMS(p_persistence, p_testing, k)
        RMS_mean =RMS(p_training_mean, p_testing, k)
```

```
/Users/nicolatomasone/miniconda3/envs/thesis/lib/python3.4/site-packages/numpy/core/_methods.py:59: RuntimeWarning: Mean of empty slice.
  warnings.warn("Mean of empty slice.", RuntimeWarning)
```

```
In [12]: fig = plt.figure(figsize=(10, 5))
        plt.plot(RMS_nrm, '-x', label='NRFM')
        plt.plot(RMS_persistence, '-x', label='Persistence')
        plt.plot(RMS_mean, '-x', label='Mean')
        plt.ylabel('RMS / [W]')
        plt.xlabel('Time interval')
        plt.grid()
        plt.legend()
        plt.savefig('../images/rms_0_initial_wind_speed.pdf',bbox_inches='tight')
```



Persistence model has a lower RMS than NRMF.

#### 1.4 RMS for mean wind speed

```
In [13]: start2 = np.abs(p_testing - p_testing.mean()).argmin() # value closer to mean of series
end2 = np.size(p_testing)
resolution2 = end2 - start2
```

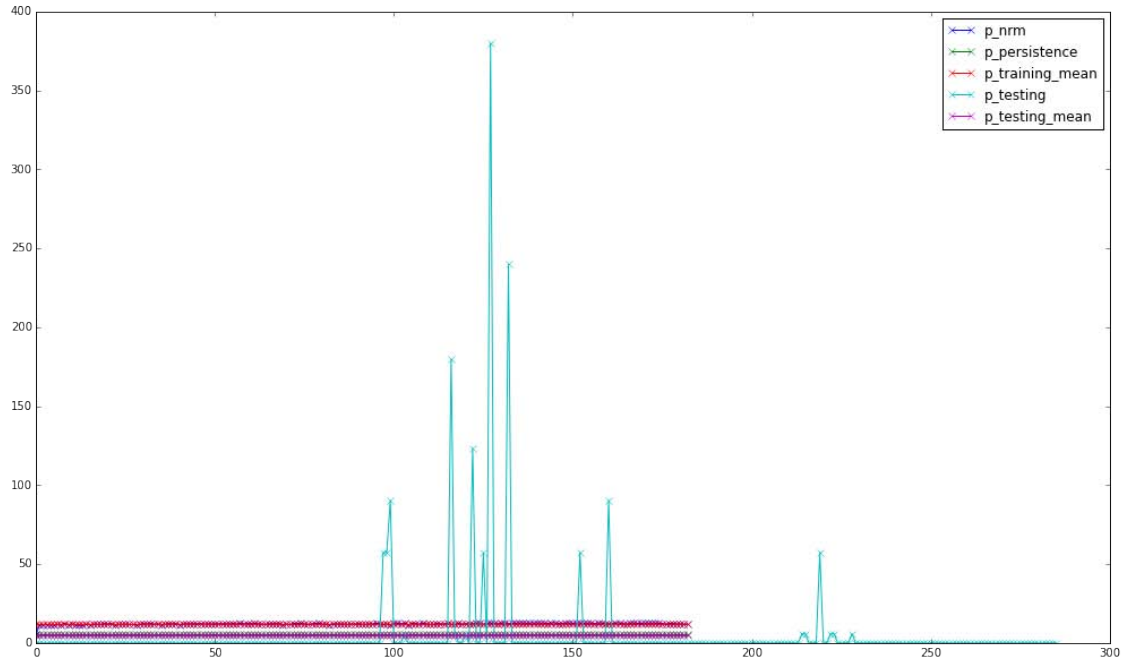
```
time_range2 = range(start2, end2)
k2 = np.arange(0, resolution2)
```

```
p_start2 = p_testing[start2]
p_nrm2 = np.array([nrm(p_start2, p_training, k2) for k2 in range(0, resolution2)])
p_persistence2 = np.ones(np.size(p_nrm2)) * p_start2
p_testing_mean2 = np.ones(np.size(p_nrm2)) * np.mean(p_testing)
p_training_mean = np.ones(np.size(p_nrm2)) * np.mean(p_training)
```

```
In [14]: print('v_start:', wind_testing[start2], 'm s^-1')
fig = plt.figure(figsize=(17, 10))
plt.plot(p_nrm2, '-x', label='p_nrm')
plt.plot(p_persistence2, '-x', label='p_persistence')
plt.plot(p_training_mean, '-x', label='p_training_mean')
plt.plot(p_testing, '-x', label='p_testing')
plt.plot(p_testing_mean2, '-x', label='p_testing_mean')
plt.legend()
```

```
v_start: 3.13888888889 m s^-1
```

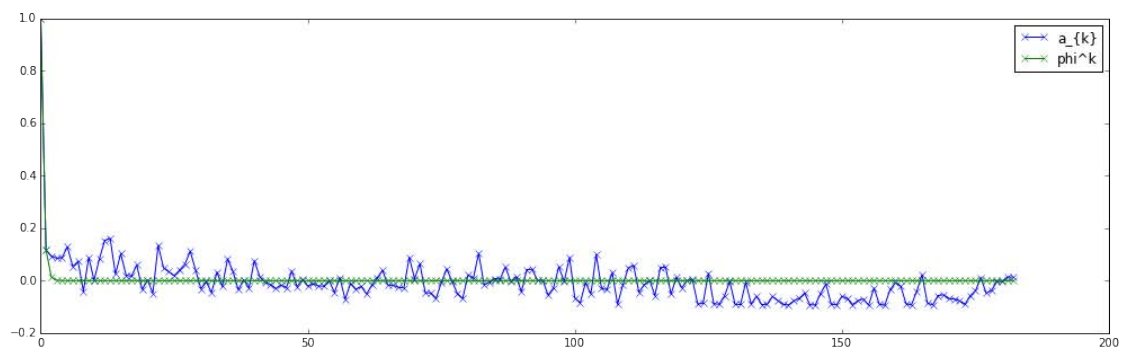
```
Out[14]: <matplotlib.legend.Legend at 0x10a5a5ac8>
```



```
In [15]: a_k2 = [nrm_correlation(p_training, k, np.mean(p_training)) for k in range(0, resolution2)]
          phi2 = [nrm_correlation(p_training[k:], 1, np.mean(p_training[k:]))*k for k in range(0, resolution2)]

          fig = plt.figure(figsize=(17, 5))
          plt.plot(a_k2, '-x', label='a_{k}')
          plt.plot(phi2, '-x', label='phi^k')
          plt.legend()
```

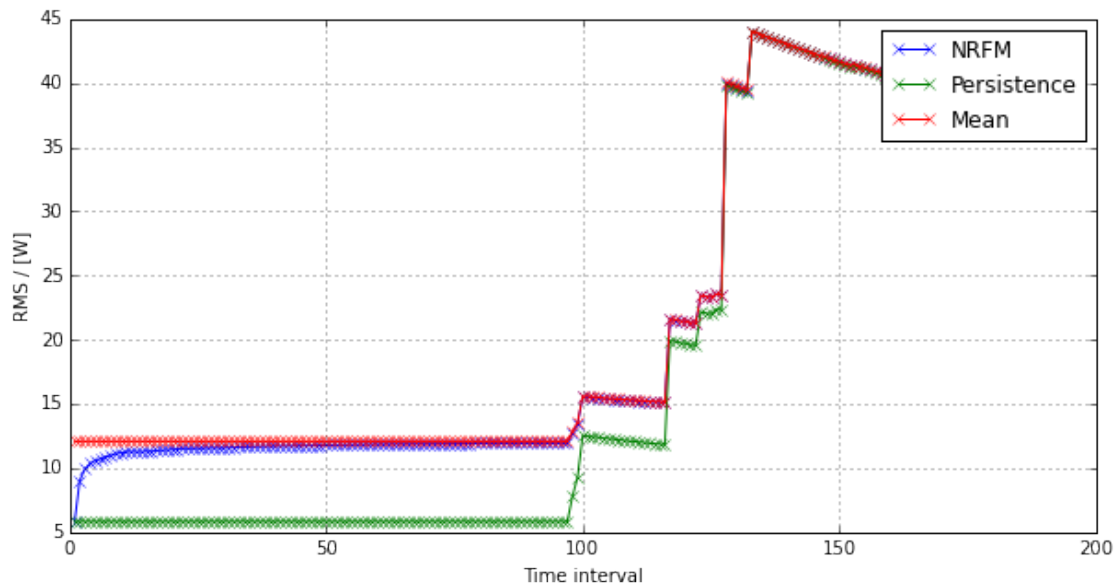
Out[15]: <matplotlib.legend.Legend at 0x10ab26da0>



```
In [16]: RMS_nrm2 = RMS(p_nrm2, p_testing, k2)
          RMS_persistence2 = RMS(p_persistence2, p_testing, k2)
          RMS_mean2 = RMS(p_training_mean, p_testing, k2)
```

/Users/nicolatomasone/miniconda3/envs/thesis/lib/python3.4/site-packages/numpy/core/\_methods.py:59: RuntimeWarning: Mean of empty slice.

```
In [17]: fig = plt.figure(figsize=(10, 5))
plt.plot(RMS_nrm2, '-x', label='NRFM')
plt.plot(RMS_persistence2, '-x', label='Persistence')
plt.plot(RMS_mean2, '-x', label='Mean')
plt.ylabel('RMS / [W]')
plt.xlabel('Time interval')
plt.grid()
plt.legend()
plt.savefig('../images/rms_initial_wind_speed_mean_series.pdf',bbox_inches='tight')
```



Persistence model has a lower RMS than NRFM.

```
In [18]: %load_ext watermark
%watermark -a "Nicola Tomasone" -u -e -t -v -m -g -p numpy,scipy,pandas,pyEMS
```

Nicola Tomasone

Last updated: 20/09/2015 23:19:19

CPython 3.4.3

IPython 4.0.0

numpy 1.9.2

scipy 0.16.0

pandas 0.16.2

pyEMS 0.1

compiler : GCC 4.2.1 (Apple Inc. build 5577)

system : Darwin

release : 10.8.0

machine : i386

processor : i386

CPU cores : 2

interpreter: 64bit

Git hash : 8e950b17b84b326c4998f6f30bccda1ba0cdb9eb

## 1.5 References

[NJM+98] Torben Skov Nielsen, Alfred Joensen, Henrik Madsen, Lars Landberg, and Gregor Giebel. A new reference for wind power forecasting. *Wind Energy*, 1(1):29–34, 1998.

[Mad04] H. Madsen. A protocol for standardizing the performance evaluation of short-term wind power prediction models, 2004.

# Annex electrical database function

September 20, 2015

```
In [1]: import numpy as np
import pandas as pd
from scipy import signal
import pyodbc
import pickle
from IPython.core.display import Image
%matplotlib inline

import pyEMS
```

## 1 Construction of the database function for electrical data

In the following notebook, after an analysis of data related to the electrical quantities in the GECAD Renewable Energy laboratory (AKA laboratory F514), it is written a function that fetches the data from the GECAD database and gives back elaborated meaningful data.

### 1.1 Fetching data from database

In order to properly handle the time variables regardless the user time zone and the laboratory time zone and regardless the present or absence of the Daylight Saving Time (DST), all the time variables must refer to the Coordinated Universal Time (UTC).

It is chosen a time interval in which an experiment was performed so that a practical case is shown.

```
In [2]: pd.Timestamp
```

```
Out[2]: pandas.tslib.Timestamp
```

```
In [3]: date = pd.to_datetime(('2015-03-04 00:00', '2015-03-05 00:00')).tz_localize('Europe/Lisbon').tz
```

The data are fetched from the GECAD database using the `pyodbc` and the `pandas` packages: the first package creates the data connection with the database server while the second saves the fetched data in a `pandas DataFrame` object that simplifies the later data processing.

**Note 1:** The connection is performed over the GECAD VPN so that the user computer must be connected to that VPN in order to reach the GECAD database.

**Note 2:** Here the connection string, the string with the address and the login information, is loaded from a binary file because it cannot be published for security reasons while in the final implementation of the function it is an input string.

```
In [4]: with open('../shared_files/connection_string.pickle', 'rb') as f:
connection_string = pickle.load(f)
```

```
In [5]: conn_remote = pyodbc.connect(connection_string)
curs_remote = conn_remote.cursor()
time_format = "%Y-%m-%d %H:%M"
```

```

queryF1 = ("""
        SELECT TimeCol, F1_U1, F1_I1, F1_U2, F1_I2
        FROM GID_F.dbo.F1_Analyzer
        WHERE TimeCol > ? AND TimeCol < ?
        ORDER BY TimeCol;
        """)

queryF2 = ("""
        SELECT TimeCol, F2_U1N, F2_U2N, F2_U3N, F2_P1, F2_P2, F2_P3, F2_S1, F2_S2, F2_S3
           F2_I1, F2_I2, F2_I3, F2_Freq
        FROM GID_F.dbo.F2_Analyzer
        WHERE TimeCol > ? AND TimeCol < ?
        ORDER BY TimeCol;
        """)

queryF3 = ("""
        SELECT TimeCol, F3_U1N, F3_U2N, F3_U3N, F3_P1, F3_P2, F3_P3, F3_S1, F3_S2, F3_S3
           F3_I1, F3_I2, F3_I3, F3_Freq
        FROM GID_F.dbo.F3_Analyzer
        WHERE TimeCol > ? AND TimeCol < ?
        ORDER BY TimeCol;
        """)

queryF4 = ("""
        SELECT TimeCol, F4_U1N, F4_U2N, F4_U3N, F4_P1, F4_P2, F4_P3, F4_S1, F4_S2, F4_S3
           F4_I1, F4_I2, F4_I3, F4_Freq
        FROM GID_F.dbo.F4_Analyzer
        WHERE TimeCol > ? AND TimeCol < ?
        ORDER BY TimeCol;
        """)

queryF5 = ("""
        SELECT TimeCol, F5_PF, F5_Q_L, F5_P, F5_I_MD, F5_ActEnr, F5_Q, F5_UN, F5_S, F5_Q
        FROM GID_F.dbo.F5_Analyzer
        WHERE TimeCol > ? AND TimeCol < ?
        ORDER BY TimeCol;
        """)

queryF6 = ("""
        SELECT TimeCol, F6_PF, F6_Q_L, F6_P, F6_I_MD, F6_ActEnr, F6_Q, F6_UN, F6_S, F6_Q
        FROM GID_F.dbo.F6_Analyzer
        WHERE TimeCol > ? AND TimeCol < ?
        ORDER BY TimeCol;
        """)

queryF7 = ("""
        SELECT TimeCol, F7_PF, F7_Q_L, F7_P, F7_I_MD, F7_ActEnr, F7_Q, F7_UN, F7_S, F7_Q
        FROM GID_F.dbo.F7_Analyzer
        WHERE TimeCol > ? AND TimeCol < ?
        ORDER BY TimeCol;
        """)

query_result = []
for query in (queryF1, queryF2, queryF3, queryF4, queryF5, queryF6, queryF7):

```

```

query_result.append(pd.read_sql(query,
                                conn_remote,
                                index_col='TimeCol',
                                params=[pd.datetime.strptime(date[0], time_format),
                                        pd.datetime.strptime(date[-1], time_format)],
                                parse_dates=['TimeCol']))

curs_remote.close()
conn_remote.close()

```

## 1.2 Post processing fetch data

Values for voltages, currents and frequency are store as integer where the last two digits are decimals due to the PLC implementation, so that they required a factor conversion of 1/100.

```
In [6]: def SI_normalize(pandas_DataFrame, string, conversion_factor):
```

```
    """
```

```
    Find the request column and apply the conversion.
```

```
    Find the column which has the request string in its
    name and multiply the founded column for the
    conversion factor.
```

```
    Parameters
```

```
    -----
```

```
    panda_DataFrame : pandas.DataFrame
```

```
        Dataframe on which apply the research an conversion.
```

```
    string : str
```

```
        String to look for in the columns names.
```

```
    conversion_factor : float
```

```
        Factor to multiply the founded column.
```

```
    Return
```

```
    -----
```

```
    pandas_DataFrame : pandas.DataFrame
```

```
        Initial dataframe 'pandas_DataFrame' converted.
```

```
    """
```

```
    for column_name in pandas_DataFrame.columns:
```

```
        if string in column_name:
```

```
            pandas_DataFrame[column_name] = (
                pandas_DataFrame[column_name] *
                conversion_factor)

```

```
    return pandas_DataFrame
```

```
conversion_factor = 1 / 100
```

```
string_list = 'U', 'I', 'Freq'
```

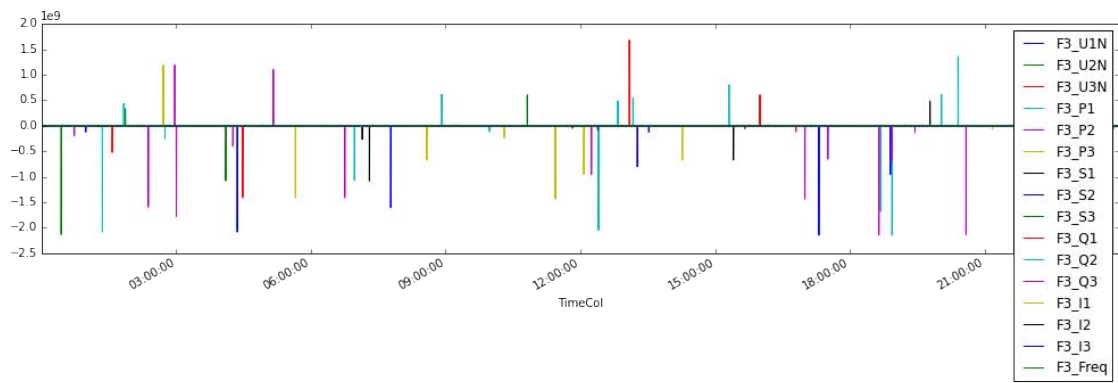
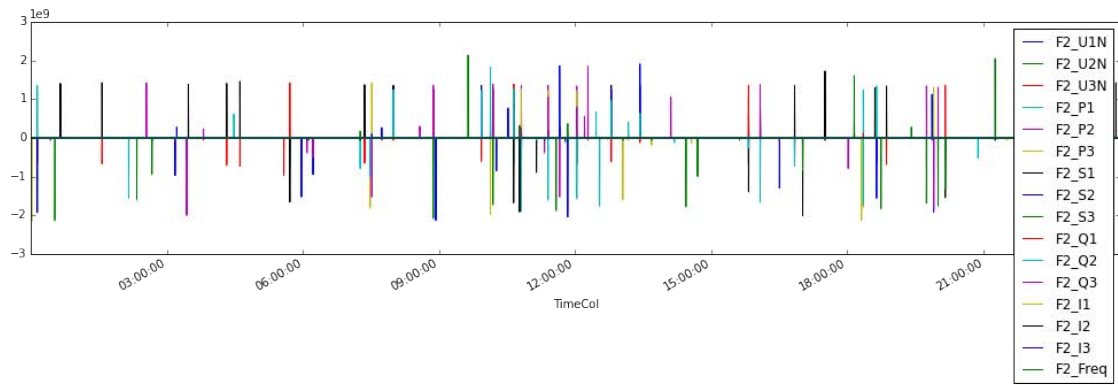
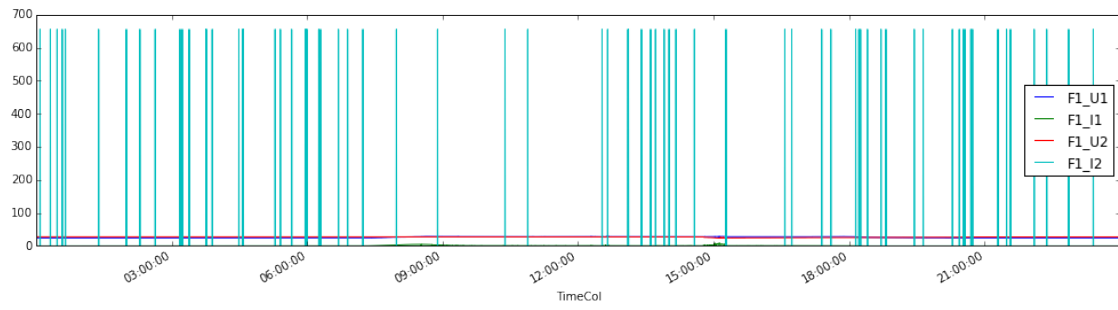
```
for query in query_result:
```

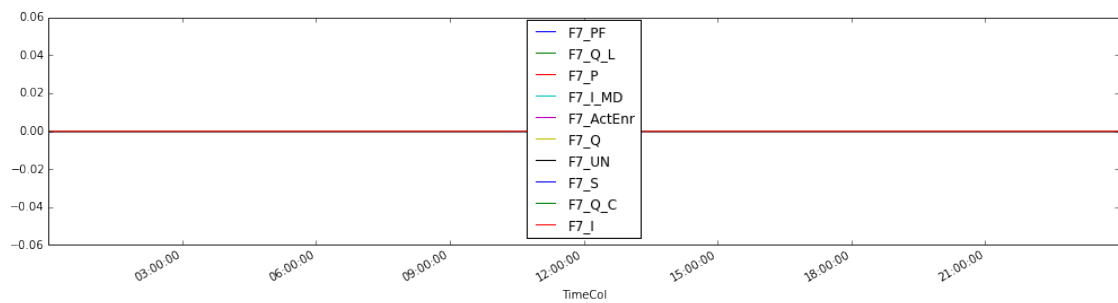
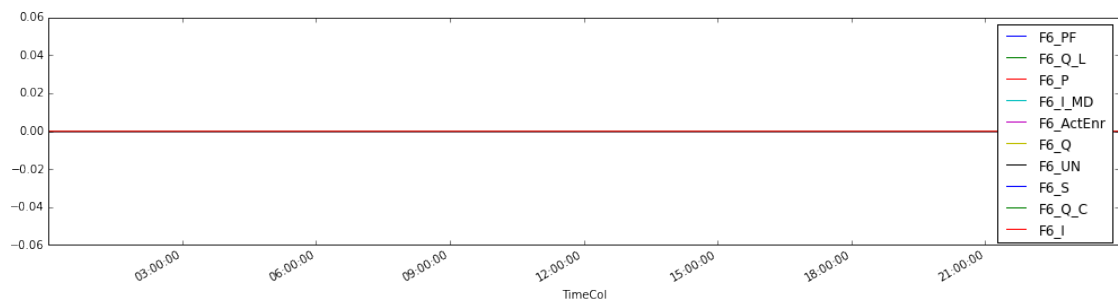
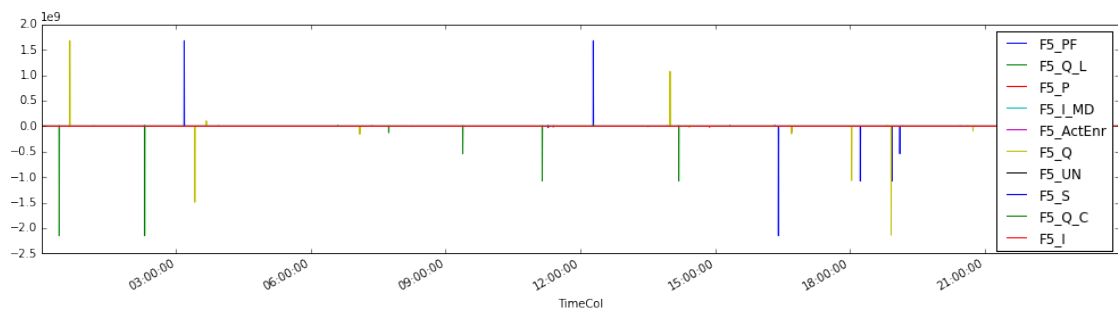
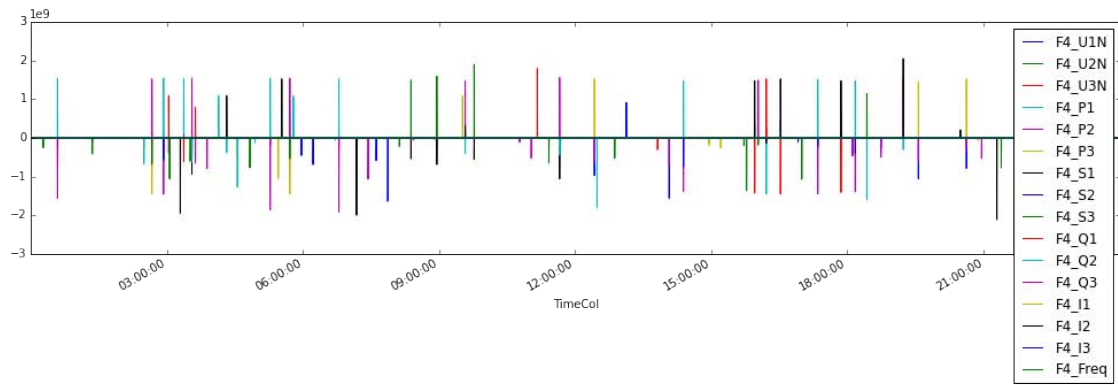
```
    for string in string_list:
```

```
        SI_normalize(pandas_DataFrame=query,
                    string = string,
                    conversion_factor=conversion_factor)

```

```
In [7]: for query in query_result:
        query.plot(figsize=(17, 4))
```

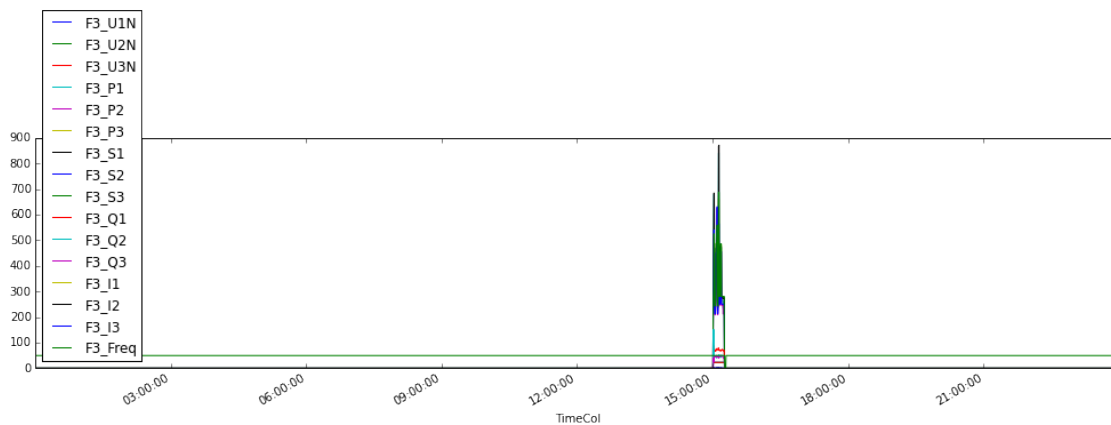
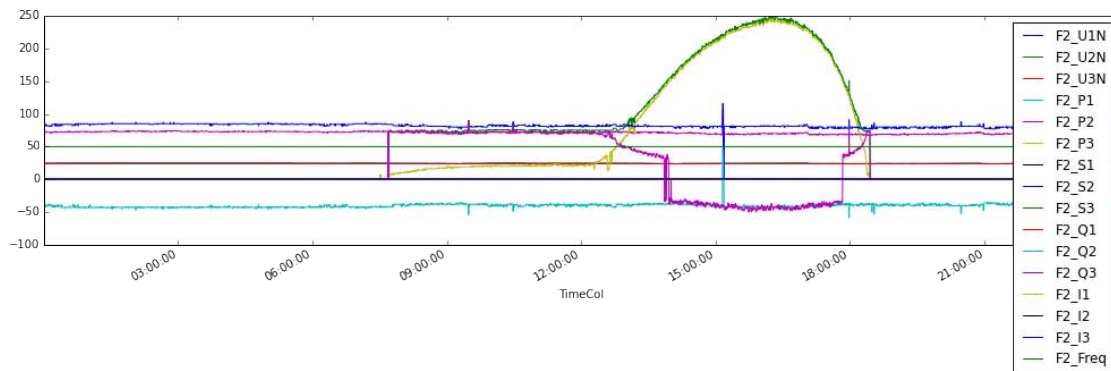
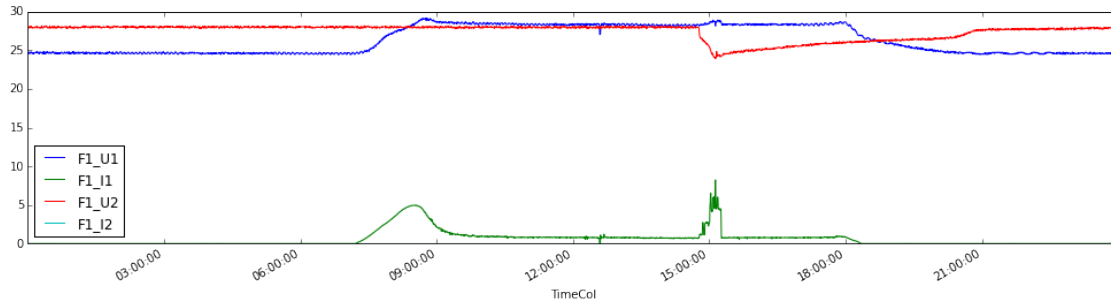


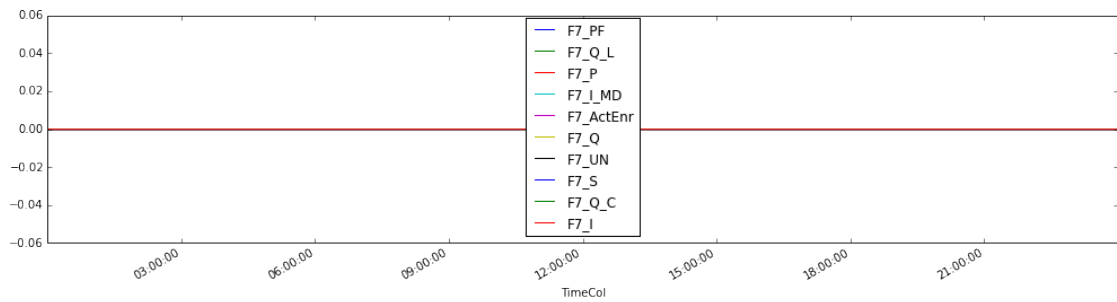
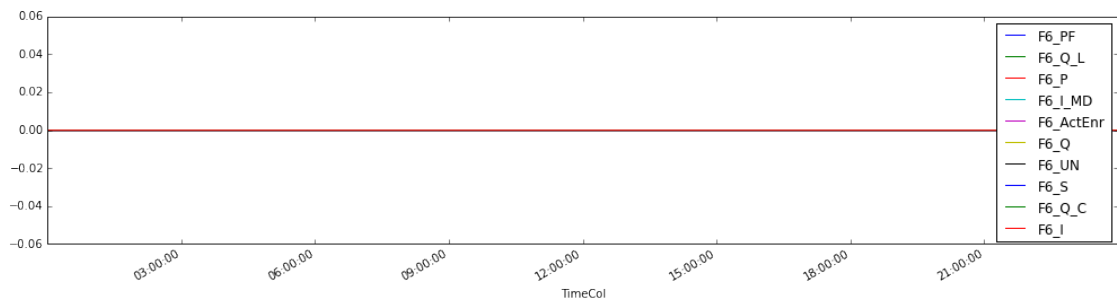
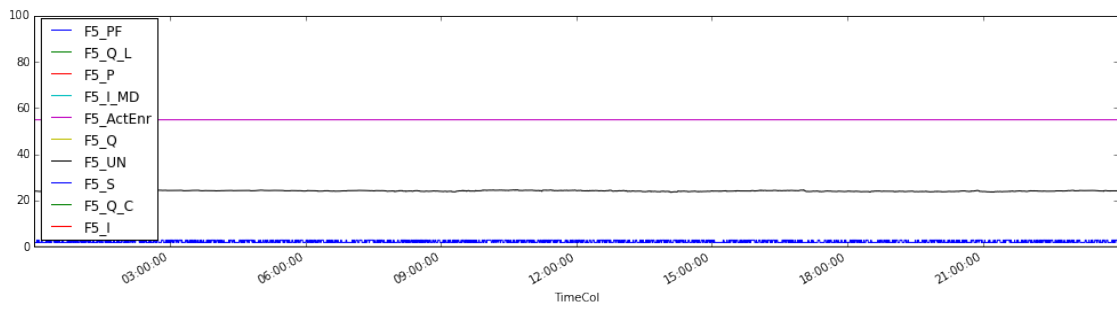
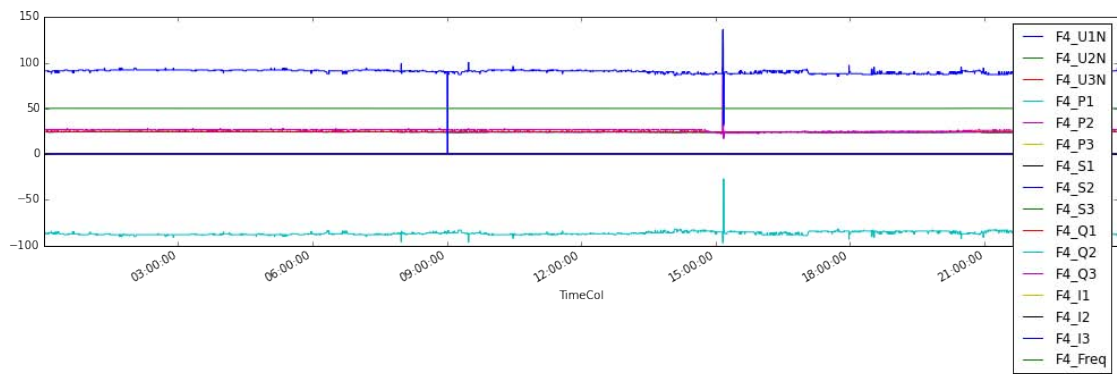


The plots show the presence of pitch noise in the dataset. Applying a median filter (see 1 and 2) to get rid off the noise.

```
In [8]: query_filtered = []
        for query in query_result:
            query_filtered.append(query.apply(lambda x: signal.medfilt(x, 7)))
```

```
In [9]: for query in query_filtered:
        query.plot(figsize=(17, 4))
```





The measured data sent to the PLC board by different measure instruments, are not time aligned because the link bus can work only with a sequential transmission of data. The minimum sampling frequency is found and then used to resample all the data to have the same time index for all the measures.

```
In [10]: sampling_min = int(np.ceil(np.min([(np.diff(query.index.values).min() / np.timedelta64(1, 's'))
                                             for query in query_filtered])))
        str(sampling_min) + 's'
```

```
Out[10]: '10s'
```

```
In [11]: query_resampled = []
        for query in query_filtered:
            query_resampled.append(query.resample(str(sampling_min) + 's')) # It loses timezone!

        (queryF1_resampled, queryF2_resampled,
         queryF3_resampled, queryF4_resampled,
         queryF5_resampled, queryF6_resampled, queryF7_resampled) = query_resampled
```

The recorded data are now usable for later analysis.

As example, the first ten records of data for the F1 and F2 measure instruments are shown.

```
In [12]: queryF1_resampled.head(10)
```

```
Out[12]:
```

	F1_U1	F1_I1	F1_U2	F1_I2
TimeCol				
2015-03-04 00:00:00	24.5	0.03	27.90	0
2015-03-04 00:00:10	24.6	0.03	27.90	0
2015-03-04 00:00:20	24.7	0.03	28.00	0
2015-03-04 00:00:30	24.7	0.03	28.05	0
2015-03-04 00:00:40	24.6	0.03	28.00	0
2015-03-04 00:00:50	24.7	0.03	28.00	0
2015-03-04 00:01:00	24.7	0.03	28.05	0
2015-03-04 00:01:10	24.7	0.03	28.00	0
2015-03-04 00:01:20	24.6	0.03	28.00	0
2015-03-04 00:01:30	24.6	0.03	27.90	0

```
In [13]: queryF2_resampled.head(10)
```

```
Out[13]:
```

	F2_U1N	F2_U2N	F2_U3N	F2_P1	F2_P2	F2_P3	F2_S1	\
TimeCol								
2015-03-04 00:00:00	24.27	24.43	24.29	0	72	0	0	
2015-03-04 00:00:10	24.30	24.43	24.31	0	72	0	0	
2015-03-04 00:00:20	24.30	24.43	24.31	0	72	0	0	
2015-03-04 00:00:30	24.30	24.43	24.31	0	73	0	0	
2015-03-04 00:00:40	24.29	24.45	24.30	0	73	0	0	
2015-03-04 00:00:50	24.28	24.45	24.30	0	73	0	0	
2015-03-04 00:01:00	24.28	24.45	24.30	0	73	0	0	
2015-03-04 00:01:10	24.28	24.45	24.30	0	73	0	0	
2015-03-04 00:01:20	24.26	24.45	24.27	0	73	0	0	
2015-03-04 00:01:30	24.25	24.45	24.26	0	73	0	0	

	F2_S2	F2_S3	F2_Q1	F2_Q2	F2_Q3	F2_I1	F2_I2	F2_I3	\
TimeCol									
2015-03-04 00:00:00	82	0	0	-39	0	0	0.34	0	

2015-03-04 00:00:10	82	0	0	-40	0	0	0.34	0
2015-03-04 00:00:20	82	0	0	-40	0	0	0.34	0
2015-03-04 00:00:30	82	0	0	-40	0	0	0.34	0
2015-03-04 00:00:40	85	0	0	-41	0	0	0.35	0
2015-03-04 00:00:50	85	0	0	-41	0	0	0.35	0
2015-03-04 00:01:00	85	0	0	-41	0	0	0.35	0
2015-03-04 00:01:10	82	0	0	-40	0	0	0.34	0
2015-03-04 00:01:20	82	0	0	-39	0	0	0.34	0
2015-03-04 00:01:30	82	0	0	-40	0	0	0.34	0

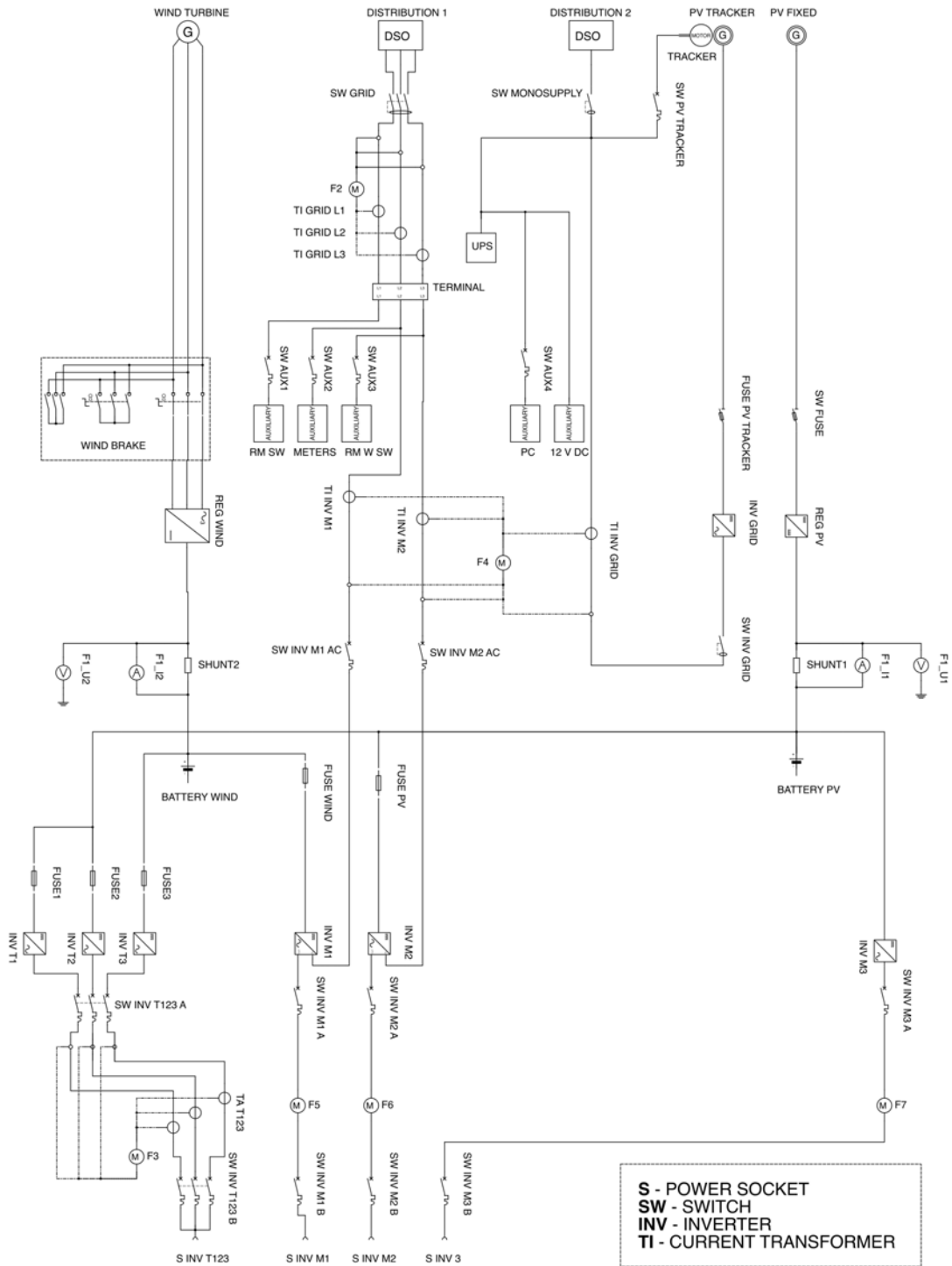
	F2_Freq
TimeCol	
2015-03-04 00:00:00	50.00
2015-03-04 00:00:10	50.00
2015-03-04 00:00:20	50.01
2015-03-04 00:00:30	50.01
2015-03-04 00:00:40	50.00
2015-03-04 00:00:50	50.00
2015-03-04 00:01:00	50.00
2015-03-04 00:01:10	49.99
2015-03-04 00:01:20	49.99
2015-03-04 00:01:30	49.99

### 1.3 Data elaborating

In the following the post processed data are used to computed electrical quantity not directly measured but useful for the laboratory analysis. The electrical diagram of the laboratory is used as reference.

```
In [14]: Image("../images/gecad_labF514_ele_diagram.png")
```

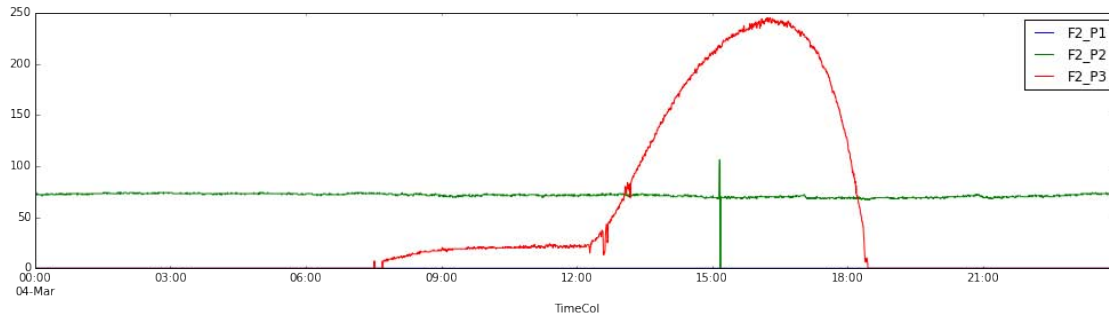
```
Out[14]:
```



There is no measure of the output currents from the batteries so it is done considering the power net exchanges and divided for the DC buses voltages.

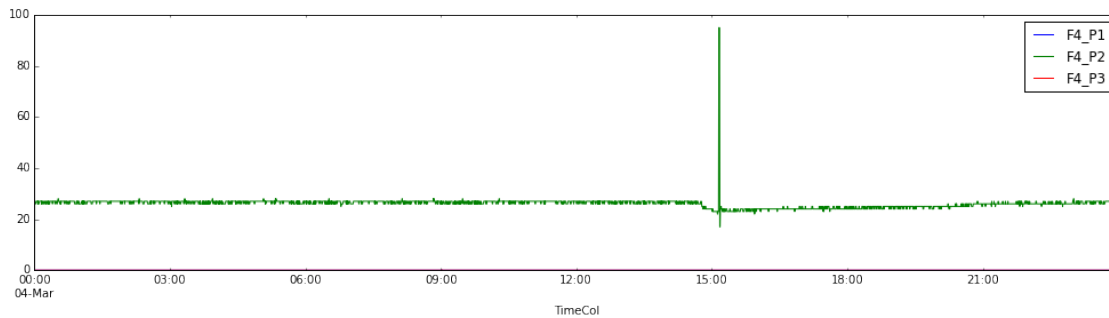
```
In [15]: queryF2_resampled[['F2_P1', 'F2_P2', 'F2_P3']].plot(figsize=(17, 4))
```

```
Out[15]: <matplotlib.axes._subplots.AxesSubplot at 0x10a873e48>
```



```
In [16]: queryF4_resampled[['F4_P1', 'F4_P2', 'F4_P3']].plot(figsize=(17, 4))
```

```
Out[16]: <matplotlib.axes._subplots.AxesSubplot at 0x10d89b630>
```



Considering the bus DC, the flow entering are negative, leaving are positive.

```
In [17]: elaborated = pd.DataFrame(queryF6_resampled['F6_P'] +
                                   queryF3_resampled['F3_P1'] +
                                   queryF3_resampled['F3_P2'] +
                                   queryF7_resampled['F7_P']
                                   , columns=['P load pv'])
```

```
In [18]: elaborated['P load wind'] = (queryF3_resampled['F3_P2'] + queryF5_resampled['F5_P'])
```

```
In [19]: elaborated['P source pv'] = (queryF1_resampled['F1_U1'] *
                                       queryF1_resampled['F1_I1'])
```

```
elaborated['P battery pv'] = (elaborated['P load pv'] -
                              elaborated['P source pv'] -
                              queryF2_resampled['F2_P1'])
```

```

In [20]: elaborated['P source wind'] = (queryF1_resampled['F1_U2'] *
                                         queryF1_resampled['F1_I2'])

        elaborated['P battery wind'] = (elaborated['P load wind'] -
                                         elaborated['P source wind'] -
                                         queryF2_resampled['F2_P2'])

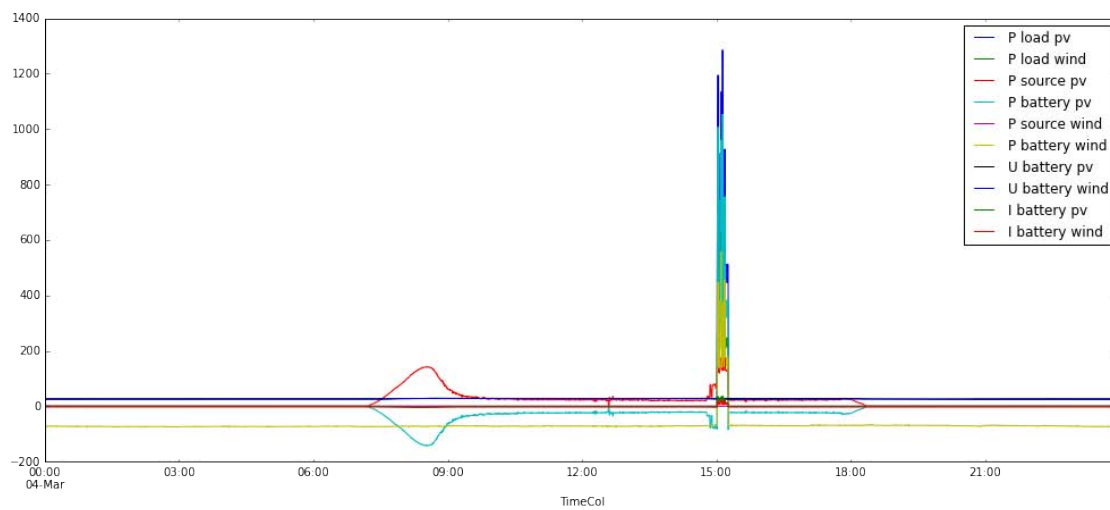
In [21]: elaborated[['U battery pv', 'U battery wind']] = queryF1_resampled[['F1_U1', 'F1_U2']]

In [22]: elaborated['I battery pv'] = elaborated['P battery pv'] / elaborated['U battery pv']
        elaborated['I battery wind'] = elaborated['P battery wind'] / elaborated['U battery wind']

In [23]: elaborated.plot(figsize=(17, 7))

Out[23]: <matplotlib.axes._subplots.AxesSubplot at 0x10a336b00>

```

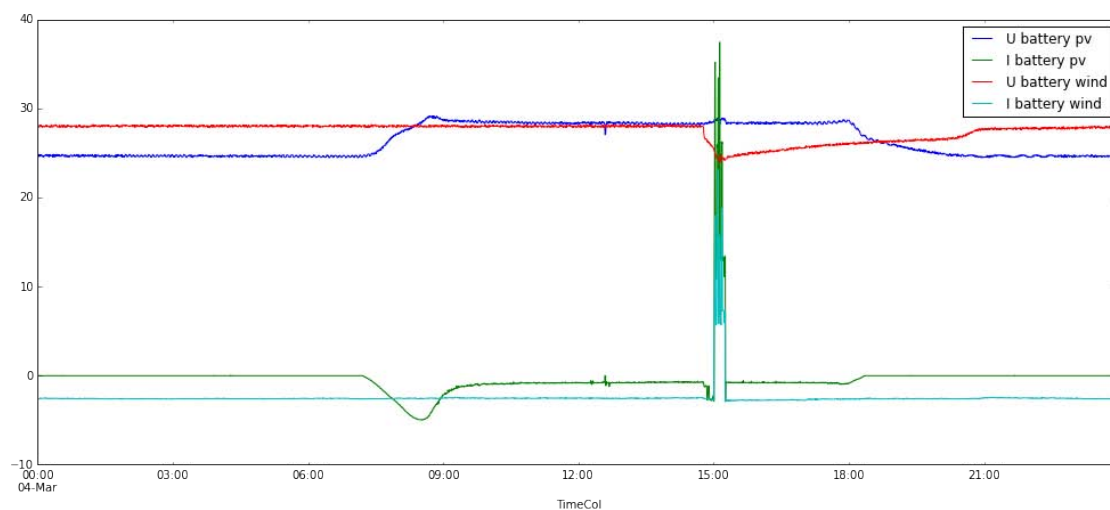


```

In [24]: elaborated[['U battery pv', 'I battery pv', 'U battery wind', 'I battery wind']].plot(figsize=

Out[24]: <matplotlib.axes._subplots.AxesSubplot at 0x10b36a9b0>

```



```
In [25]: %load_ext watermark
         %watermark -a "Nicola Tomasone" -u -e -t -v -m -g -p numpy,scipy,pandas,pyEMS
```

Nicola Tomasone

Last updated: 20/09/2015 22:35:10

CPython 3.4.3

IPython 4.0.0

numpy 1.9.2

scipy 0.16.0

pandas 0.16.2

pyEMS 0.1

compiler : GCC 4.2.1 (Apple Inc. build 5577)

system : Darwin

release : 10.8.0

machine : i386

processor : i386

CPU cores : 2

interpreter: 64bit

Git hash : f905ffecaaffa27f8dcd4ab4ea78a9cd672475e8d