



## GreenDataNet

# **D3.14 – Summary of Smart Grid Configuration Tool Specification and Implementation**

[Final]

Authors

Prof. Brunelli Davide – UNITN

Prof. Petri Dario, Maurizio Rossi, Ivan Minakov and Davide Sartori – UNITN

Rev 0.2

## TABLE OF CONTENTS

<b>TABLE OF CONTENTS</b> .....	<b>2</b>
<b>REVISION SHEET</b> .....	<b>3</b>
<b>KEY REFERENCES AND SUPPORTING DOCUMENTATIONS</b> .....	<b>4</b>
<b>1. INTRODUCTION</b> .....	<b>5</b>
1.1 Document Purpose.....	5
1.2 Definition, acronyms and abbreviations .....	7
1.2.1 Key Acronyms and Abbreviations.....	7
1.3 Document overview .....	7
<b>2. SIMULATING FRAMEWORK</b> .....	<b>8</b>
2.1 Reinforcement Learning Engine .....	8
2.1.1 Formal Description .....	8
2.1.2 SGCT Implementation.....	9
2.1.3 User Selectable Rewards .....	10
2.2 Modular design .....	10
2.2.1 Forecasting .....	11
2.2.2 Batteries .....	11
2.2.3 Smart Grid Interaction.....	11
<b>3. SGCT TOOL</b> .....	<b>12</b>
3.1 SW Implementation .....	12
3.1.1 List of Files .....	12
3.2 User Guide.....	13
<b>4. CONCLUSIONS</b> .....	<b>17</b>

## REVISION SHEET

Revision Number	Date	Brief summary of changes
Rev 0.1	30/11/2015	Baseline document
Rev 0.2	9/12/2015	Final version with integration of partners comments

## KEY REFERENCES AND SUPPORTING DOCUMENTATIONS

- [1] Farhangi, Hassan. The path of the smart grid. *Power and Energy Magazine, IEEE*, 2010, 8.1: 18-28.
- [2] International Electrotechnical Commission (IEC). IEC Smart Grid Roadmap. [[PDF ONLINE](#)] last checked 16<sup>th</sup> Nov. 2015.
- [3] Cluster for Energy and Environment (CLEEN). D1.3.1 Smart Grid Standardization Analysis, February 2012. [[PDF ONLINE](#)] last checked 16<sup>th</sup> Nov. 2015.
- [4] Sutton, R. S., and Barto A. G. *Reinforcement learning: An introduction*. Vol. 1. No. 1. Cambridge: MIT press, 1998.
- [5] Kaelbling, L. P. et al. Reinforcement learning: A survey. *Journal of artificial intelligence research*, 1996, 237-285.
- [6] Lewis, F. L. et al. Reinforcement learning and feedback control: Using natural decision methods to design optimal adaptive controllers. *Control Systems, IEEE*, 2012, 32.6, pp. 76-105.
- [7] Hsu, R. C. et al. A Reinforcement Learning-Based ToD Provisioning Dynamic Power Management for Sustainable Operation of Energy Harvesting Wireless Sensor Node. *Emerging Topics in Computing, IEEE Transactions on*, 2014, 2.2, pp. 181-191.
- [8] Das, A. et al. Hardware-software interaction for run-time power optimization: A case study of embedded Linux on multicore smartphones. *Low Power Electronics and Design (ISLPED), Proceedings of the 2015 IEEE/ACM International Symposium on*, 22-24 July 2015.
- [9] D. Baert and A. Vervaet. Lead-acid battery model for the derivation of Peukert's law. *Electrochimica Acta*, 44.20, pp. 3491-3504, 1 June 1999.
- [10] MathWorks Inc. About the Matlab Runtime.  
[http://mathworks.com/help/compiler\\_sdk/cxx/about-the-matlab-runtime.html](http://mathworks.com/help/compiler_sdk/cxx/about-the-matlab-runtime.html)

## 1. INTRODUCTION

The *smart grid* can be defined as the electric energy network of the future, where bidirectional flow of energy will be possible both at transmission and distribution levels, where end users will become prosumers (double role of producers and consumers of electrical energy) and information about network health, power flows, statistics and consumption will be exchanged and be available in real-time [1]. A smart grid is based on a concept of bidirectional electricity flow where buildings can generate and store electric energy, while some years ago they were simply supposed to be consumers. Now, the local generation and storage of electric power can transform real estates equipped with hybrid electrical systems into a source of earnings. Moreover, smart grids impact positively on users: work and home environments, in the majority of the world, are isolated energy consuming units lacking in sustainability and efficiency. In the near future, intelligent grids will interact directly with users, contributing to save energy and to use it in a more efficient way. In such a vision, any node of the network will be an active element of the smart grid. Finally, a smart grid must facilitate the optimal management of the hybrid loads (offering demand-side management and other ancillary services), exploiting the information processed from the meters, and should avoid energy waste, overload and voltage drops. The coordination of its components is fundamental to improve efficiency and to facilitate the use of intelligence and energy efficient policies.

In addition to renewable integration and distributed storage, data centers need to deal with sharing and exploiting energy information to and from the facility in order to insert smoothly within the smart grid infrastructure. The number of different components and devices that are part of a data center introduces a level of complexity that requires innovative ICT solutions that do not exist today. GreenDataNet aims at developing a multi-level optimization framework based on data center workload management, renewable energy sources and smart grid configuration, to ease demand response and load management scenarios. The solution developed in the present Smart Grid Configuration Tool (SGCT) has been designed to operate with several varying factors such as dynamic workload, renewable energy produced by PV systems, available grid power and energy storage capabilities.

Many research efforts have been put in standardization and definition of the ICT infrastructure of the smart grid to convey information along with power [2]-[3], but the *struggle* is far from being over. Actually no standards exist yet for automated testing of smart grid ready equipment, tools and infrastructures; for this reason we defined our own interaction strategy and set of commands to demonstrate the outcome of such interaction rather than evaluating the goodness of a specific ICT protocol.

### 1.1 DOCUMENT PURPOSE

This deliverable presents the public summary of the design and implementation of the Smart Grid Configuration Tool (SGCT) developed within the GreenDataNet project. This tool, built with Matlab® environment has been made portable and redistributable according to the principle of the Open Source SW that drives the project. Accordingly to GDN framework, it exploits, as input/historical data, the PV production forecasts provided by the tool presented in D3.4 and electrical consumption forecasts provided by the tool described in D3.2. The tool implements the analytical models of data center components specified in D2.2 and the battery models presented in D3.6.

Differently from the SEMS (D3.7) and AEMS (D3.11) SW tools, that provide optimal scheduling of the resources, the SGCT provides a long term prediction (up to 14 days) of the data center energy utilization in terms of expected input from the Grid, battery reservoir utilization, health and state of charge, data center energy requirements and PV production/utilization, with hourly time-slots (tunable). The goal is to provide the smart grid operator a tool to ease the resource planning and scheduling at a higher level w.r.t. the data center itself. With this tool the operators can also provide *commands* to the data center management system. Since smart grid doesn't exist yet, we defined our own communication strategy and simulate interactions for the sake of completeness.

In this deliverable we are targeting the smart grid integration with a single data center as scenario, while the distributed interface tool will be the target of D3.15 (and the public summary in D3.16). Together, both tools, will realize the Smart Grid Energy Optimization Tool which placement in the GDN framework is shown in Figure 1.

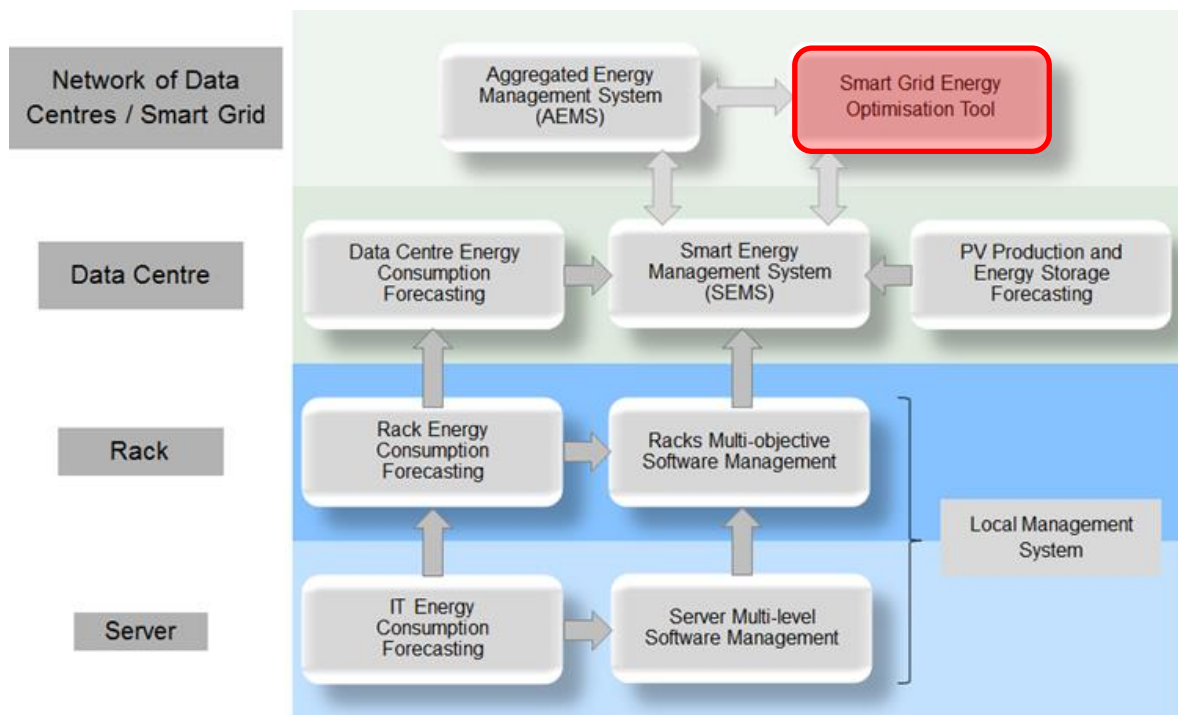


Figure 1. SGCT placement in the GDN project framework

## 1.2 DEFINITION, ACRONYMS AND ABBREVIATIONS

### 1.2.1 KEY ACRONYMS AND ABBREVIATIONS

AC	Alternating Current
AEMS	Aggregated Energy Management System
BMS	Battery Management System
CER	Cooling Efficiency Ration
DC	Direct Current
DCIM	Data Center Infrastructure Management
DoD	Depth of Discharge
DOW	Description of Work
EC	European Commission
EES	Electric Energy Storage
HVAC	Heating, Ventilation and Air Conditioning
ICT	Information and Communication Technology
IT	Information Technology
PDU	Power Distribution Unit
PUE	Power Usage Effectiveness
PV	Photovoltaic
RL	Reinforcement Learning
SDK	Software Development Kit
SEMS	Smart Energy Management System
SGCT	Smart Grid Configuration Tool
SoC	State of Charge
SW	Software
UPS	Uninterruptible Power Supply
WP	Work Package

## 1.3 DOCUMENT OVERVIEW

This document is organized as follows. Section 2 describes the theoretical background of the control system engine, core of the SGCT, the modular design of the simulating framework and its input/output interface. Section 3 presents a detailed User Guide for the tool while Section 4 concludes the deliverable.

## 2. SIMULATING FRAMEWORK

The Smart Grid Energy Optimization Tool represents the interface between an urban green data center and the smart grid. Its role is to interact with smart grid facilities and exchange information to maximize the benefit out of the integration into the grid itself.

The SGCT is a simulating tool that allows evaluating the performance in terms of energy of an urban green data center. Depending on the goodness of the available forecasts, and the availability of historical data, it is possible to evaluate the performance of different, selectable, data center energy management schemes, in a time horizon that is once again selectable by the user. Thanks to the simulating engine based on a custom developed reinforcement learning algorithm, and three different optimization criteria, a single simulation run takes short time (in the order of seconds) allowing to accommodate a broad range of scenarios.

### 2.1 REINFORCEMENT LEARNING ENGINE

Reinforcement Learning is a machine learning method developed in the computational intelligence community to solve a problem faced by an agent that interacts with a dynamic environment and must learn the best behavior through rewards and punishments, consequences of its actions. It was inspired by natural learning processes that characterize human beings and animals [4] – [7]. This approach has been widely used and demonstrated able to find the optimal solution for a wide variety of control problems both for continuous- and discrete-time dynamical systems [6], moreover it can be run on-line [8] (without off-line model construction phase) thanks to the high speed of convergence of the training phase and the reduced memory requirement with respect to other machine learning algorithms (e.g. Neural Networks, Deep Learning, etc.).

#### 2.1.1 FORMAL DESCRIPTION

The Reinforcement Learning it is a heuristic method where a decision-making *agent*, observes the *state of the environment*  $S$ , takes *actions*  $a$ , and receives *rewards*  $r$  (or *punishments*) for its actions, trying to find a solution for a problem in the defined environment. In such method the agent finds the optimal control strategy after a number of *trial-and-error stages* during which it accumulates rewards until a given condition is met; this is called the learning phase. After the learning is completed the agent exploits the accumulated knowledge to implement the optimal control actions in the dynamic environment, in the so called exploiting phase.

We selected the **Q-learning** [7] algorithm to exploit rewards and punishments obtained from the environment, since the transition probability from one state to another is unknown, and the best action cannot be determined based only on the observation of the current state.

The Q-learning algorithm iteratively populates a Q-matrix which dimensions are the cardinality of the state space  $S$  and that of the action space  $A$ , respectively. Each element of the matrix is a coefficient that represents the accumulative rewards for a specific action taken in a specific state of the environment. It processes the rewards through the  $Q(s, a)$  function, which depends on the current state  $s \in S$  and the action  $a \in A$ , to iteratively build the coefficients that represents the knowledge of the agent. The **Q-function**  $Q(s, a)$  is defined as [7]:



$$Q(s_t, a_t) = (1 - \eta)Q(s_t, a_t) + \eta \left[ r_{t+1} + \gamma \max_{a_{t+1} \in A} Q(s_{t+1}, a_{t+1}) \right]$$

where the subscript  $t$  refers to the time-slot,  $Q(s_t, a_t)$  is the accumulative reward updated at time-slot  $t$  for the action  $a_t$  taken in state  $s_t$  upon which the system evolves in state  $s_{t+1}$ . The reward  $r_{t+1}$  is the reward gained in the transition from state  $s_t$  to  $s_{t+1}$  with action  $a_t$ ; while  $\eta$  and  $\gamma$  parameters are called learning factor and discount rate respectively, and varies in the interval  $[0,1]$ . Basically the Q-function computes a weighted sum of the previous accumulated rewards plus a combination of the actual reward and the maximum rewards expected to obtain in the future transition. The initialization of the weighting parameters directly affects the behavior of the learning process: the  $\eta$  parameter affects the speed of learning, the closer it is set to 1 the faster is the convergence, conversely the closer it is set to 0 the better is the convergence. The  $\gamma$  discount rate controls the weight of the future rewards in the update, if it is set to 0 only the immediate rewards count.

---

### 2.1.1.2 SGCT IMPLEMENTATION

Although the RL method looks simple and straightforward, its complexity resides in the modeling phase of the dynamical system, hence the definition of the state space  $\mathcal{S}$ , the action space  $\mathcal{A}$  and ultimately the reward  $r$  attribution function. In the specific case of the GDN framework, the goal of the SGCT was to model a single urban data center equipped with a PV plant and batteries as the energy storage system, according to the model presented in D2.2 and summarized in Figure 2. By taking as a reference the modeling proposed in [7] and [8] we considered as relevant environmental factors for the SGCT agent, the state of the batteries ( $s_{EES}$ ), the availability of the green energy ( $s_{GREEN}$ ) and the price ( $s_{PRICE}$ ) of energy taken from the grid, while the agent can decide, as action  $\mathbf{a}$ , whether to charge, discharge or don't use (standby) the batteries. Formally, the state space  $\mathcal{S}$  is the union of the three sets just introduced and defined as  $s_{EES}:\{full, discharge, standby, charge, empty\}$ ;  $s_{GREEN}:\{zeroPV, lowPV, fullPV\}$  and  $s_{PRICE}:\{low, high\}$ . In details, the availability of the green energy from the PV has been related with the workload, for this reason *lowPV* state means that the available green energy in the current time-slot is not sufficient to supply the load alone. The energy price state has been defined according to a threshold to accommodate for both free energy markets or not, in particular it could be possible to select two or more thresholds leading to more states, but this is not suggested since it would result in an exponential increase in the number of global state for the system. With this specific SGCT environment definition, the total number of states is equal to 30 ( $5 \times 3 \times 2$ ) and the number of action 3, resulting in a Q-matrix of dimensions  $30 \times 3$ .

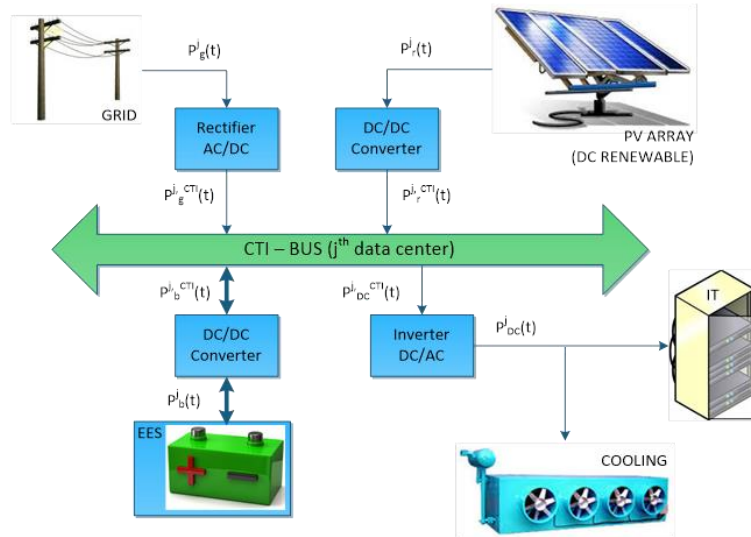


Figure 2. Model of a data center system as presented in D2.2

### 2.1.3 USER SELECTABLE REWARDS

The definition of the rewards attribution function has quite relevant importance to achieve an adequate goodness of the self-learned control policy. Since in the SGCT implementation the actions an agent can perform are related with the batteries utilization, conversely the rewards must take into account a list of parameters, in addition to the energy storage utilization, that the system should optimize to differentiate the target policy. This means, for example, that a system manager can choose to optimize the battery usage to maximize their lifetime by pushing the system to follow continuous charge/discharge cycles (i). Another policy could favor the minimization of the money spent for the energy intake from the grid, postponing the recharge of the batteries from the grid during time-slots when the price is supposed to be lower, while allowing the intake only to supply the load, which should have the highest priority in every policy (ii). The last example is of an optimization policy that tries to maximize the utilization of the green energy provided by the PV array when the workload is not enough to utilize it; in this case the batteries can be recharged only during high irradiance (high PV production) time-slots (iii). These 3 policies were implemented in the SGCT and, as a design choice, batteries were favored as primary supply source for the load with respect to the grid.

## 2.2 MODULAR DESIGN

In order to accommodate for the GDN framework the SGCT has been designed in modular fashion, this will allow refining and introducing new features and models to accommodate different scenario. In this way the tasks that are concurrently under development in the project will be easily ported in the Smart Grid Energy Optimization Tool.

---

### 2.2.1 FORECASTING

The GDN framework extensively exploits forecasts to achieve the target optimization goals, specifically for photovoltaic production (D3.4) and workload estimation (D2.5). Two different, well-known and widely adopted prediction schemes were implemented to test the performance, the *oracle* and the *persistence* approaches, with the oracle, one assumes that the predictor is omniscient and the prediction 100% correct; with the persistence approach, the value of a variable in the current time-slot is replicated and used as prediction for the next slot.

---

### 2.2.2 BATTERIES

We used the Peukert's law to model the electric energy storage units (as presented in [9]), to accommodate for a large variety of battery technologies. This solution allows to tune the SGCT with any kind of storage system type (Li-ion, Lead-Acid...) and size (batteries can be arranged in any kind of matrix  $1 \times n$ ,  $m \times 1$  or  $m \times n$ ) provided that one knows the relevant parameters (Peukert number, rated capacity, reference discharge currents...).

---

### 2.2.3 SMART GRID INTERACTION

Many research efforts are currently put in the design of the smart grid particularly in the definition of the ICT related standards for the realization of the communication infrastructure that must be reliable, scalable and modular at the same time. Currently we cannot rely on an established testing infrastructure to evaluate the goodness of our smart grid interface. Based on the literature review, we envisioned what the relevant information for the smart grid operators can be and what kind of commands they can issue to obtain ancillary services from the distributed premises, in order to achieve peak shaving and load balancing (for example). For these reasons, in the SGCT, we included a module that allows exporting simulation results in both graphical and textual format, allowing the *simulated operators* to put smart grid commands (instructions) in the shape of tags, directly on top of the output time-series. The objective of this module will be (supposedly in the final demonstrator) to process and forward these commands to the relevant GDN modules that schedule resources.

In the current implementation, the simulation results represent, in the form of time series, the expected power balance of the data center for the given target time-horizon. In this way the smart grid operator can have an immediate estimation of what will be the expected power consumption of the data center and its impact on the grid. On top of this information the utility can introduce smart grid commands using the available tags: *reduce* the load on the grid by  $x\%$ , *shift* the load by  $n$  time-slots and finally the operator can state a *margin* of operation, in percentage, to allow the data center to increase its load.

## 3. SGCT TOOL

The SGCT is a SW that allows predicting and simulating the behavior of a data center equipped with PV modules and EES units.

### 3.1 SW IMPLEMENTATION

The SGCT has been implemented in Matlab® R2015b development environment. Although the SW is available only after purchasing a license, the resulting application can be made portable and freely reusable by others in the form of Matlab® APP® or standalone application, thanks to the freely available runtime libraries provided by MathWorks.com [10]. In the following user guide will be presented the SGCT interface which does not depend on the distribution package.

#### 3.1.1 LIST OF FILES

The main files contained in the project folder are:

- 1) *sgct.m*: the script that opens the SGCT
- 2) *GUI.m*: the implementation of the main UI
- 3) *rl.m*: the implementation of the RL Engine
- 4) *annotate.m*: the implementation of the secondary UI

The other files contain the implementation of the data center components' models and auxiliary files for managing predictions and configuration parameters:

- 5) *initGlobals.m*:  
initialization of global variables and parameters
- 6) *initRIGlobals.m*:  
initialization of global variable specific to the RL engine
- 7) *loadData.m*:  
function to load and process the reference datasets in standalone mode
- 8) *configGlobalEES.m*:  
function to configure EES (batteries in this case) parameters based on the input data provided by users through the UI
- 9) *ees.m*:  
function that implements the Peukert model for computation of batteries' runtime parameters like SoC, SoH and residual capacity
- 10) *annotate.m*:  
function that implements the secondary UI to simulate the interaction with the smart grid operator
- 11) *predict.m*:  
function that implements the forecasting algorithm
- 12) *pvarray.m*:  
function that implements the PV array model to compute the green energy produced starting from the irradiance and environmental temperature

13) *rewardEES.m*:

function that implements the computation of the reward for the EES management

14) *rewardExc.m*:

function that implements the computation of the reward for the minimization of the unused energy produced by the PV plant

15) *rewardGrid.m*:

function that implements the computation of the reward for the minimization of the energy intake from the grid.

Finally a *reference.datasets* folder is included in the software package that contains the test datasets used to test, in standalone mode (namely without the interaction with the other GSN framework tools), the features of SGCT.

## 3.2 USER GUIDE

Depending on the distribution, the SGCT can be opened directly from MATLAB Command Window by typing the *sgct* command, or double clicking the icon for the standalone/APP versions. The main application window is depicted in Figure 3. The interface is organized in two boxes, the left one is used to present the simulated results while on the right side are hosted the simulator controls. The different graphical results can be accessed through the tab selector.

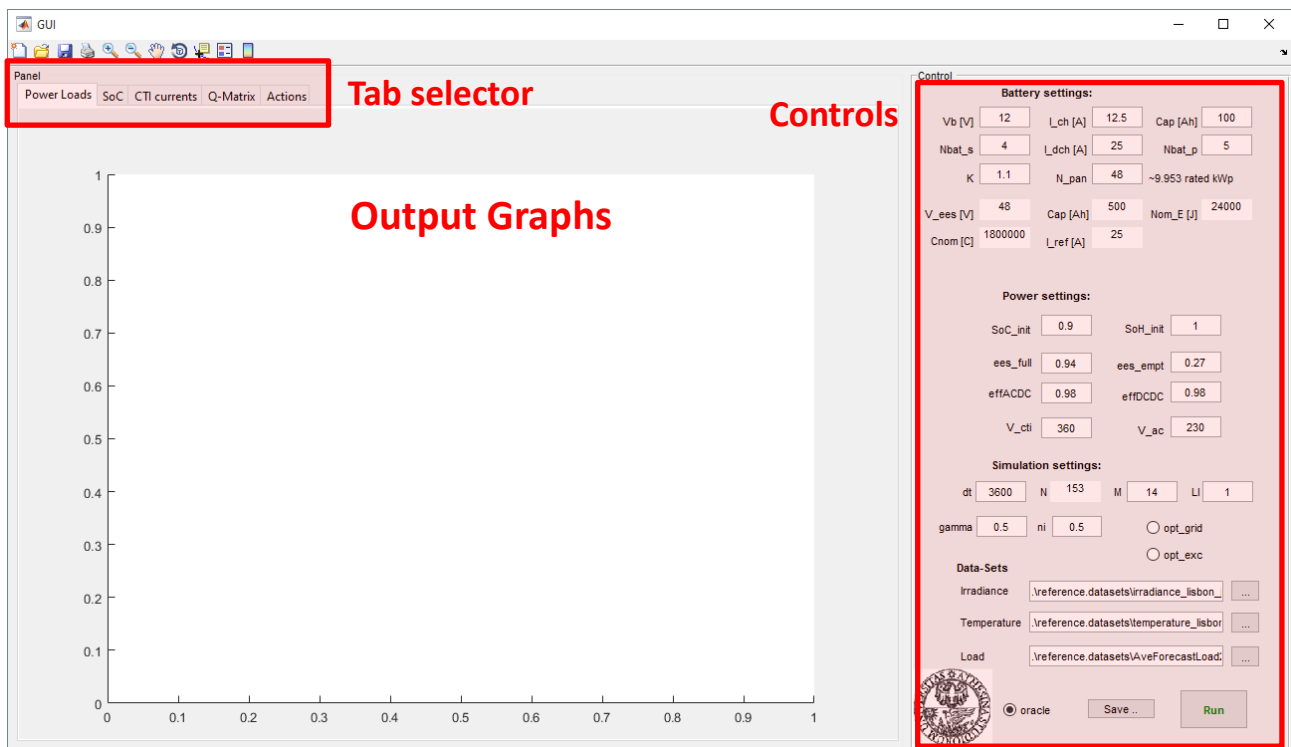


Figure 3. Main SGCT GUI

The Controls section allows configuring the HW configuration of the data center and the simulation parameters. As depicted in Figure 4, the controls are grouped in five blocks:

1. *EES & PV Controls*

These inputs control the battery parameters (size, capacity and arrangement) and the size of the PV plant (in terms of number of PV panels).

It includes a list of auxiliary parameters that are automatically updated by the system when something in the upper controls is modified.

2. *System Parameters*

These inputs are related with some HW configuration parameters like the efficiency of the power converters employed and the voltage on the CTI (both were considered fixed, not function of the power conveyed since both are strongly device dependent and their characterization is out of the scope of the current deliverable).

3. *Simulation Parameters*

These controls allow tuning global simulation parameters like the duration of a time-slot, the horizon and the RL engine optimization scheme.

4. *Input Data-sets*

These three input fields are used to select the data-sets to use for test.

5. *Main Control buttons*

This section includes a *radio-button selector* for the forecasting algorithm to use (oracle or persistence that will be extended according to future decision of the consortium based on the work done in other deliverables); the *run button* to execute the simulation and the *save button* which allows to simultaneously save the results and open the secondary UI to simulate the smart grid interaction.

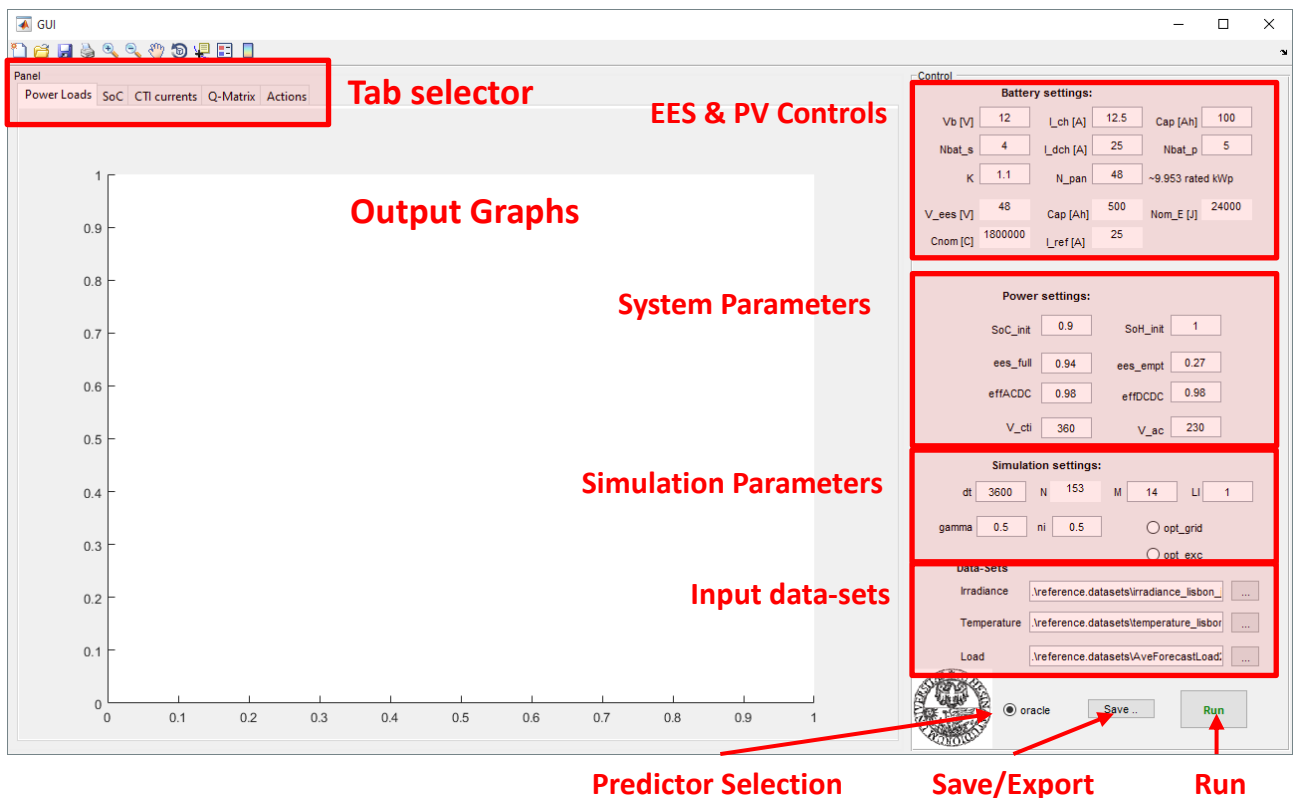


Figure 4. Main SGCT GUI, details of the commands' interface

Detailed list of available controls (those not already covered):

1. *EES & PV Controls*

*V<sub>b</sub>*: voltage of a single battery unit;

*Cap*: reference rated capacity of a single battery unit;

*Nbat<sub>s</sub>*: number of batteries in series (to obtain a specific EES total voltage);

*Nbat<sub>p</sub>*: number of batteries in parallel (to obtain a specific total capacity and reference current);

*I<sub>ch</sub> / I<sub>dch</sub>*: respectively charge current and maximum discharge current (which depends on the load);

*K*: Peukert index to use in the EES model

*N<sub>pan</sub>*: Number of PV panels that compose the array (on the right of this input box is automatically updated the rated kWp power for the selected size).

a. *Auxiliary parameters automatically updated:*

*V<sub>ees</sub>*: rated total voltage level of the EES;

*Cap*: nominal total capacity of the EES;

*C<sub>nom</sub>*: nominal total charge of the EES;

*Nom<sub>E</sub>*: nominal total energy of the EES;

*I<sub>ref</sub>*: reference discharge current based on the computation of the discharge rate, of the total capacity, in 20hrs.

2. *System parameters*

*SoC<sub>init</sub> / SoH<sub>init</sub>*: state-of-charge/health of the EES at the beginning of the simulation;

*ees<sub>full</sub> / ees<sub>empty</sub>*: thresholds to define the maximum depth-of-discharge of the EES;

*effACDC / effDCDC*: efficiencies of the power converters, based on the conversion AC-to-DC (and vice-versa) or DC-to-DC;

*V<sub>cti</sub> / V<sub>ac</sub>*: fixed voltages for the computation of the power (W and VA respectively).

3. *Simulation Parameters*

*dt*: duration of a time-slot in seconds;

*N / M*: length of the simulation in number of days (N is fixed equal to 5 months in this implementation);

*Ll*: RL engine configuration parameter for the control of the learning process (actually not used, left for future improvements);

*gamma / ni*: Q-function parameters;

*opt<sub>grid</sub> / opt<sub>exc</sub>*: selectors for the RL optimization selection, the EES optimization, the basic one is always performed along with this selection.

After the configuration of the relevant parameters and the selection of the test data-sets, by pressing the *run button* the system evaluates the results and updates the GUI in few seconds. An example of the GUI view after the simulation run is depicted in Figure 5, by playing with the tab selector on top of the graphs it is possible to go through all the system variables.

Once the simulation is completed by pressing the *save button* it is possible to export the results in xml format (the system asks for a name of the output file). This choice automatically opens the secondary GUI called "*annotate*" which is essentially a lightweight xml file editor. Figure 6 presents this interface, which has been chosen to simulate the interaction of the smart grid operator with the GDN framework. After the in-

teraction with the xml file (as presented in section 2.2.3) one must save the modification with the dedicated save button on the top-right corner and then close the *annotate* GUI.

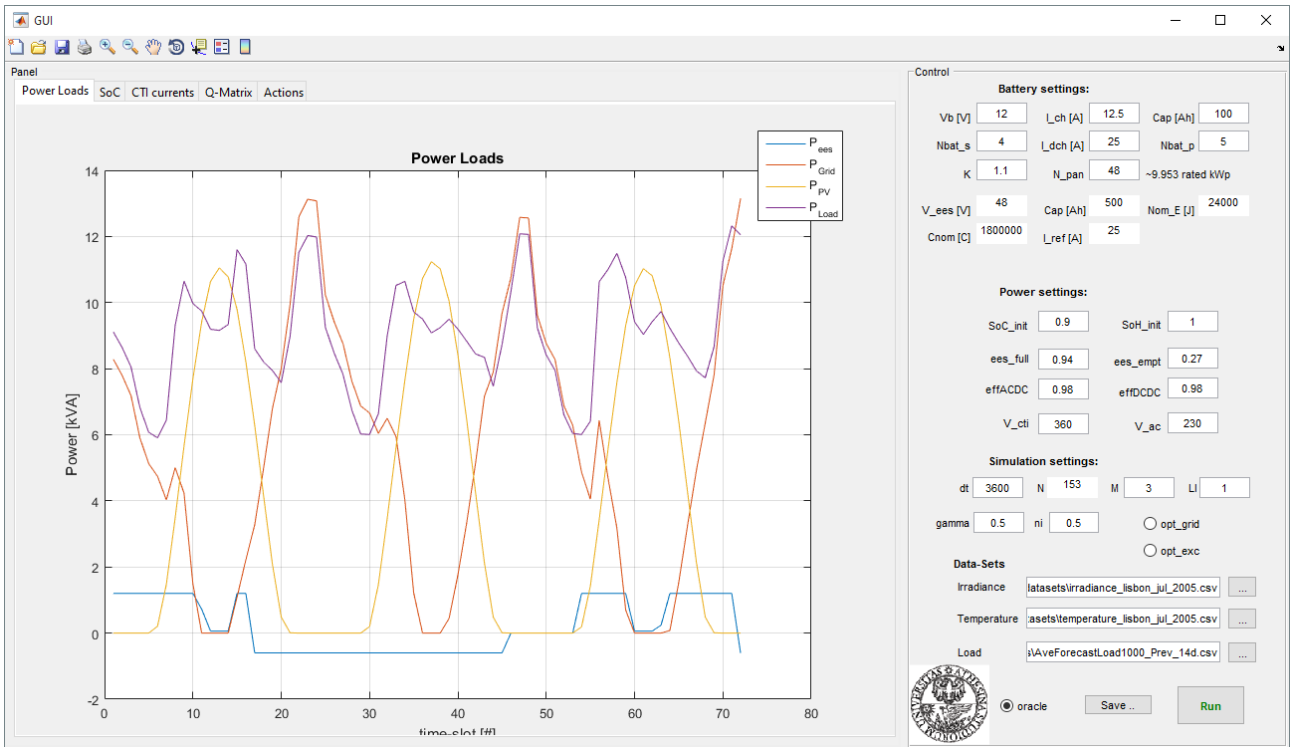


Figure 5. Main SGCT GUI after simulation run

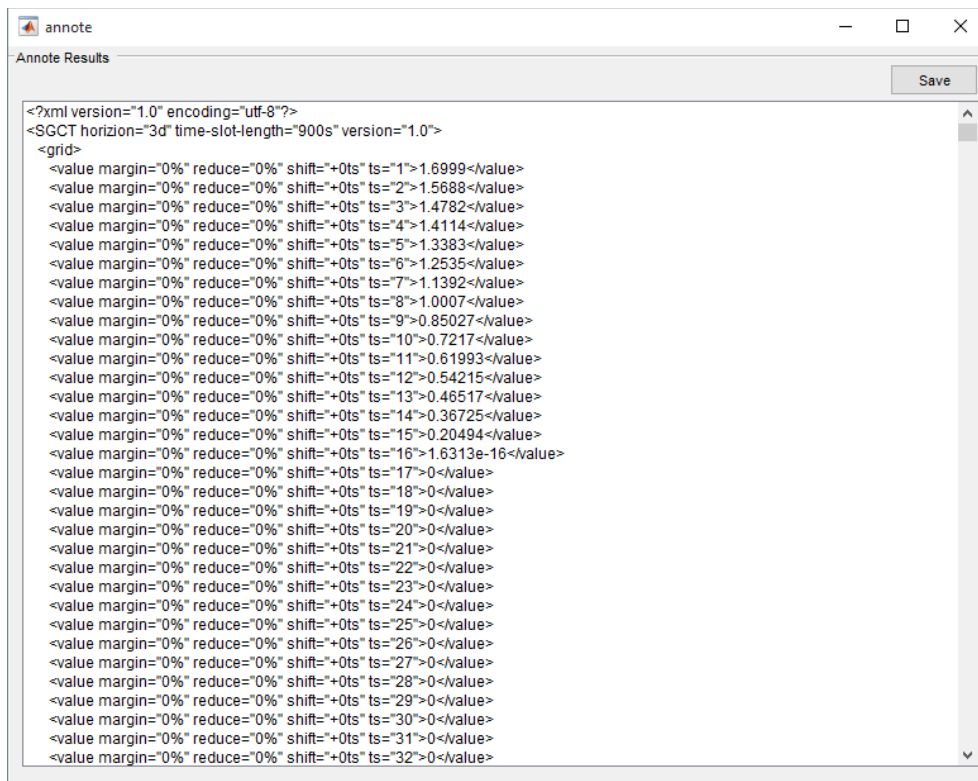


Figure 6. Secondary SGCT GUI for the introduction of smart grid related commands in the output data-set



## 4. CONCLUSIONS

In this public deliverable we presented the Smart Grid Configuration Tool, the first step towards the integration of the smart grid within the GDN framework. In this work we targeted the single data center scenario, while the distributed interface tool will be the target of D3.15. Together, both tools, will realize the Smart Grid Energy Optimization Tool target of Task 3.3.2 according to the GreenDataNet project.

Differently from the SEMS (D3.7) and AEMS (D3.11) SW tools, that provide optimal scheduling of the resources, the SGCT will provide a long term prediction of the data center energy utilization (Grid, batteries' utilization, health and state of charge, data center energy requirements and PV production) with hourly time-slots (tunable).

This SW represent the first stage of the development of the Smart Grid Energy Optimization Tool that will provide the smart grid operator with a set of SWs for the resource planning and scheduling at a higher level w.r.t. the data center itself. For this reason we included in the SGCT a dedicated interface to implement the interaction between smart grid operator and the data center management system (with a predefined set of commands).