POLITECNICO DI TORINO

Corso di Laurea Magistrale in Ingegneria Energetica e Nucleare

Innovazione nella produzione di energia

Tesi di Laurea Magistrale

DEMOSOFC project data analysis: development of a Visual Basic tool for import, evaluation and visualization of plant parameters and performance indicators

Relatori
Massimo Santarelli
Marta Gandiglio

Candidato
Davide Medina

Table of contents

I. Nomenclature

WWTP	Waste Water Treatment Plant
SOFC	Solid Oxide Fuel Cell
DEMOSOFC	Cogeneration plant based on three SOFC modules, with a biogas clean-up system and a heat recovery circuit
CONVION	Convion Ltd., Finnish provider of the SOFC modules
QUALVISTA	Qualvista Ltd, Finnish responsible for biogas constituents monitoring system
SMAT	Società Metropolitana Acque Torino S.p.A, Italian operator of the Collegno (TO) WWTP
BIOKOMP	Biokomp Srl, Italian supplier of the clean-up system
CKC	Activated carbons designed for hydrogen sulphide removal from biogas stream
C64	Activated carbons designed for siloxanes removal from biogas stream
PLC	Programmable Logic Controller
KPI	Key Performance Indicator
HRU	Heat recovery unit
DST	Daylight Saving Time
CSV	Comma Separated Value file format
UPS	Uninterruptable power supply
STP	Standard temperature and pressure (T=0 [°C], P=1 [bar])

II. Physical quantities

Methane molecular weight	$MW_{CH_4} = 16.04 \left[\frac{g}{mol}\right]$
Carbon dioxide molecular weight	$MW_{CO_2} = 44.01 \left[\frac{g}{mol}\right]$
Sulphur molecular weight	$MW_S = 32.07 \left[\frac{g}{mol}\right]$
Silicon molecular weight	$MW_{Si} = 60.08 \left[\frac{g}{mol}\right]$
Octamethylcyclotetrasiloxane ($C_8H_{24}O_4Si_4$) molecular weight	$MW_{D4} = 292.62 \left[\frac{g}{mol}\right]$
Glycol water mixture at 30 [$\%_{vol}$] specific heat capacity	$c_{p\,glycol\,water} = 3.765 \left[\frac{kJ}{kg \cdot K}\right]$
Glycol water mixture at 30 [$\%_{vol}$] density	$\rho_{glycol\,water} = 1024 \left[\frac{kg}{m^3}\right]$
Water specific heat capacity	$c_{p\,water} = 4.18 \left[\frac{kJ}{kg \cdot K}\right]$
Water density	$\rho_{water} = 999 \left[\frac{kg}{m^3}\right]$
Methane lower heating value	$LHV = 50{,}000 \left[\frac{kJ}{kg}\right]$
Specific CKC equivalent sulphur adsorption capacity	$q_{CKC} = 84 \left[\frac{mg}{g}\right]$
Specific C64 equivalent D4 adsorption capacity	$q_{C64} = 150 \left[\frac{mg}{g}\right]$

III. List of figures

IV. List of tables

1. Abstract

The DEMOSOFC project, part of the Horizon 2020 research and innovation programme funded by the European Union, is the first example in Europe of a high-efficiency cogeneration plant revolving around medium size fuel cell modules fed by biogas.

The scope of this work has been to develop a user-friendly tool to be used by the plant operators to analyse the plant performance and extract raw data for maintenance and diagnostic activities. The tool is based on the Visual Basic for Applications environment implemented in the Microsoft Office suite. The tool comprises several routines and user interfaces integrated into an Excel file, with also a couple of Word files used as report templates.

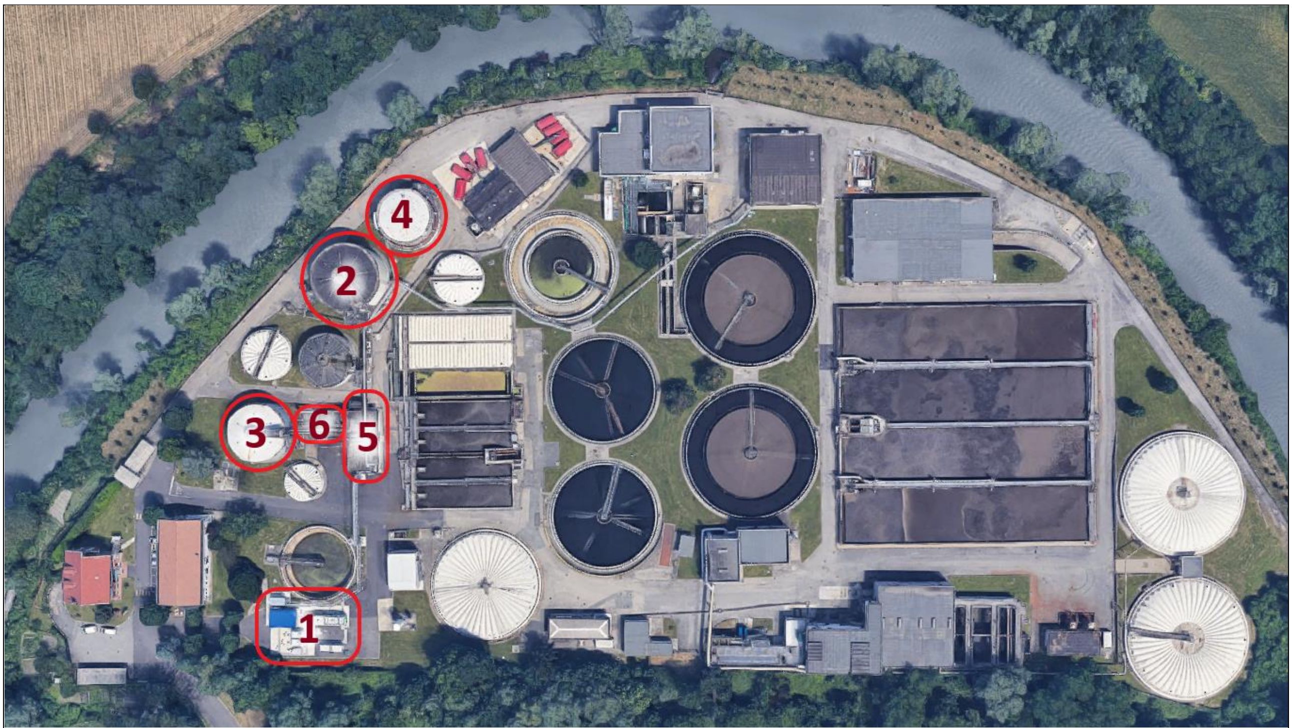
The available commands are meant to be employed by the plant operators to automatically perform operations and manipulate the data files generated by the Programmable Logic Controller installed on site. Plant parameters collected therein can be imported, arranged and stored. From this information, Key Performance Indicators and the operation mode of each module can be evaluated. Other commands allow to visualize significant information over customizable time periods, both graphically and numerically, with the possibility to respectively export them as images or reports.

The designed tool will help plant operators and authorized personal that might want to monitor the performances of the main equipment of the project: the clean-up section, the fuel cell modules and the heat recovery system. It will expectantly provide useful information for the future improvement and employment of this kind of high-efficiency cogeneration system, both on a technical and economic point of view.

2. Introduction

2.1. Plant description

The DEMOSOFC project is part of the European Union's Horizon 2020 research and innovation programme. It consists of a high efficiency cogeneration system based on the solid oxide fuel cells (SOFC) technology, fuelled by the biogas produced from the waste water treatment plant (WWTP) of Collegno, Piedmont, north-west of Italy. The partners are *Società Metropolitana Acque Torino* (SMAT) as owner of the WWTP and end-user of the energy produced, *Convion Ltd* (CONVION) as provider of the SOFC modules, *VTT Technical Research Centre Of Finland Ltd* for the analysis of the emissions and the *Imperial College of Science Technology and Medicine* for the business analysis. The project is coordinated by the Energy Department of *Politecnico di Torino.* Figure 1 presents an aerial view of the establishment, highlighting the main infrastructure related to the DEMOSOFC operation.



LEGEND:
① – DEMOSOFC area
② – Primary digester
③ – Secondary digester
④ – Gasometer
⑤ – Pumps and sludge heat exchanger utility room
⑥ – Boilers utility room

Figure 1: Aerial view of the waste water treatment plant

The waste water treatment plant receives from neighbouring municipalities around 1500 [m^3/h] of wastewater from the sewers. The stream has to undergo several stages of filtering, deposition and microbiological treatment before reaching the physical, chemical and biological requirements prescribed by the "*Testo unico ambientale*" D.Lgs. 3/04/2006, n. 152: purified water can then be reused in nearby factories or discharged in the adjacent river Dora Riparia.

Around 250 [m^3/d] of residues from the waste water treatment are collected in a stream of sludge and sent to a thickener that takes it up to ~2 [%] of solid fraction. Before being sent to the anaerobic digester, the sludge has to be heated to a temperature of ~38 [°C]: heat is required to maintain the content of the reactor between 30÷45 [°C], compensating for the thermal losses of the structure towards the environment. Temperature is one of the stability parameters of the biological processes taking place inside the digester, promoting the growth and the metabolic activity of specific bacterial populations. Consequently, the composition of the biogas produced by the anaerobic digestion changes.

Biogas is mainly a mixture of methane, carbon dioxide and water vapour. The digester management aims at maximizing the yield of methane, which currently settles between 57.4.÷72.9 [%] with an average of 64.6 [%]. Biogas is stored in a water-sealed gasometer. It consists in an open bottom vessel that rises over a deep tank of water as the volume of the gas entering the structure increases. Water prevents the biogas from leaking into the environment: besides storing the gas, the tank's design serves to establish the pressure of the gas system. Pressure is also the driving force of the cap movement, which is guided by helical runners placed on the lateral surface of the vessel. A flaring system can burn excess biogas if the gas volume exceeds the design limit of 1400 [m^3]: this happens especially during summer, when the thermal demand of the digester is lowered by higher solar irradiance and higher ambient air temperature.

Digestate, the solid part resulting from the process, is centrifuged to separate the moisture from the solid fraction, which must be then disposed of as a special waste. Figure 2 summarizes the operation of the WWTP.

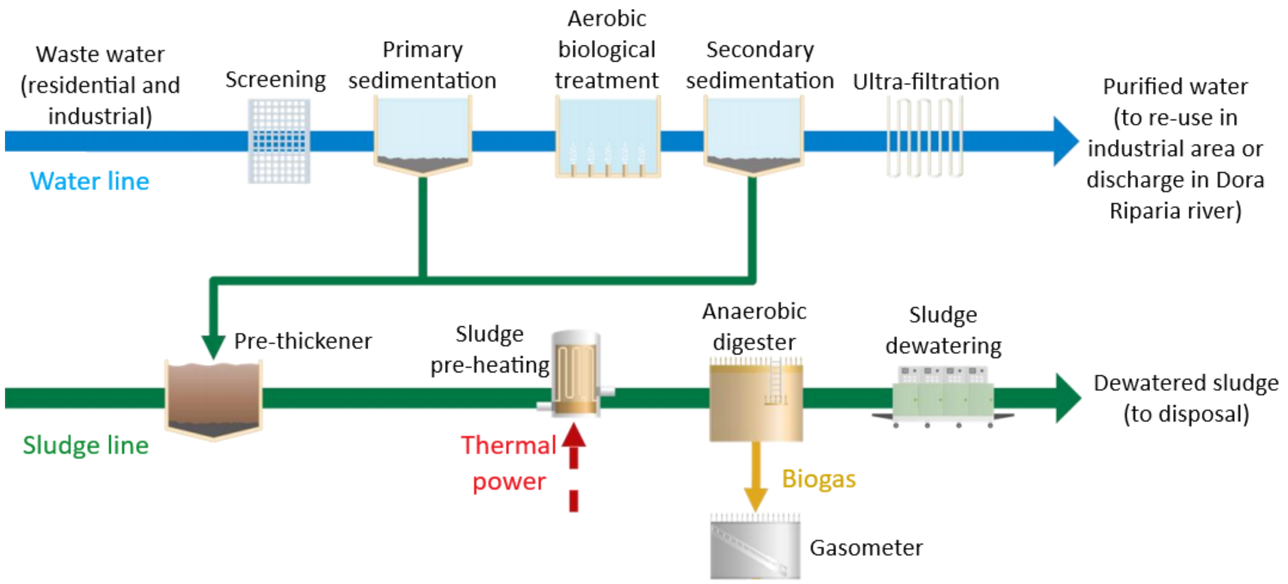


Figure 2: Waste water treatment plant operation

The heat required to sustain the process has been obtained from the combustion of methane (1) in two boilers, one fed with biogas and the other with natural gas bought from the national grid.

$$CH_4 + 2O_2 \rightarrow CO_2 + H_2O_{(g)} \quad \Delta H_{(STP)} = -50 \left[\frac{MJ}{kg}\right] \tag{1}$$

The integration of the SOFC modules allows to exploit part of the energy connected to methane oxidation to directly generate electrical power through electrochemical reactions, which is then self-consumed inside the plant. A heat recovery system ultimately transfers the sensible heat of the gaseous exhausts to the sludge. Figure 3 outlines the integration of the DEMOSOFC installation in the mass and energy flows of the WWTP.

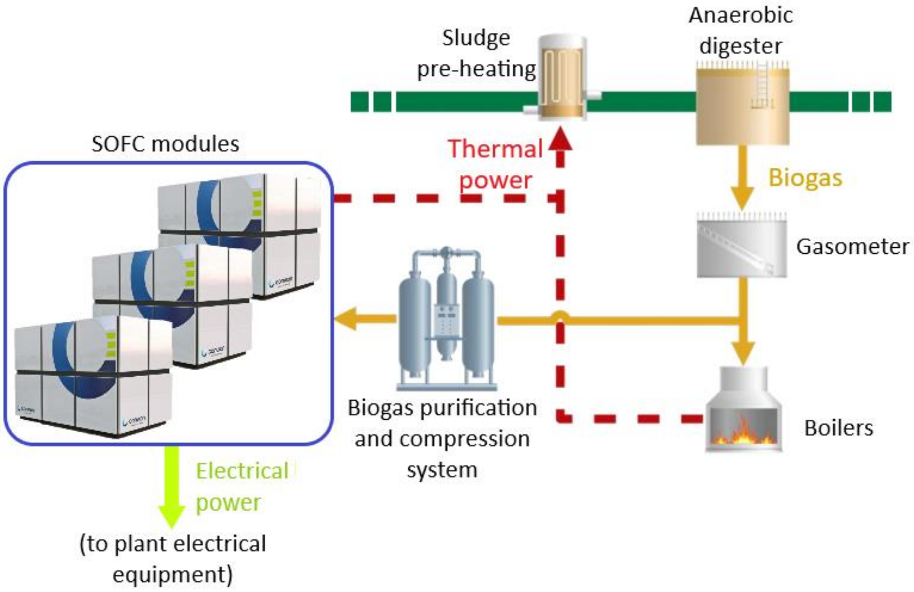


Figure 3: Sludge heating system and DEMOSOFC integration

Before being fed into the modules, biogas must undergo a thorough purification process to remove contaminants that are dangerous for the fuel cells performance, as they can clog or disable active reaction sites. Siloxanes, organic silicon compounds, do not take part in the electrochemical reaction but can deposit silica on the anode and electrolyte pores. Similarly, hydrogen sulphide (H_2S) can be chemisorbed by the nickel catalyst typically employed in the anode with the formation of Ni_3S_2 , disabling active sites.

The concentrations of these substances in the biogas are relatively low (around 4 [mg_{Si}/m^3] for silicon compunds and 35 [ppm] for hydrogen sulphide) with respect to the main gaseous species in the mixture, but they are nonetheless harmful. Therefore, a clean-up system provided by *Biokomp Srl* (BIOKOMP) has been installed near the modules in the DEMOSOFC area highlighted in Figure 1. The system is entirely fitted into an ATEX certified container, and it includes a chiller for the removal of water traces, a series of reactors containing activated carbons for the removal of contaminants, a compressor for increasing biogas pressure to operative conditions (around 4 [bar(g)]) and a metering system for biogas composition (Figure 4).

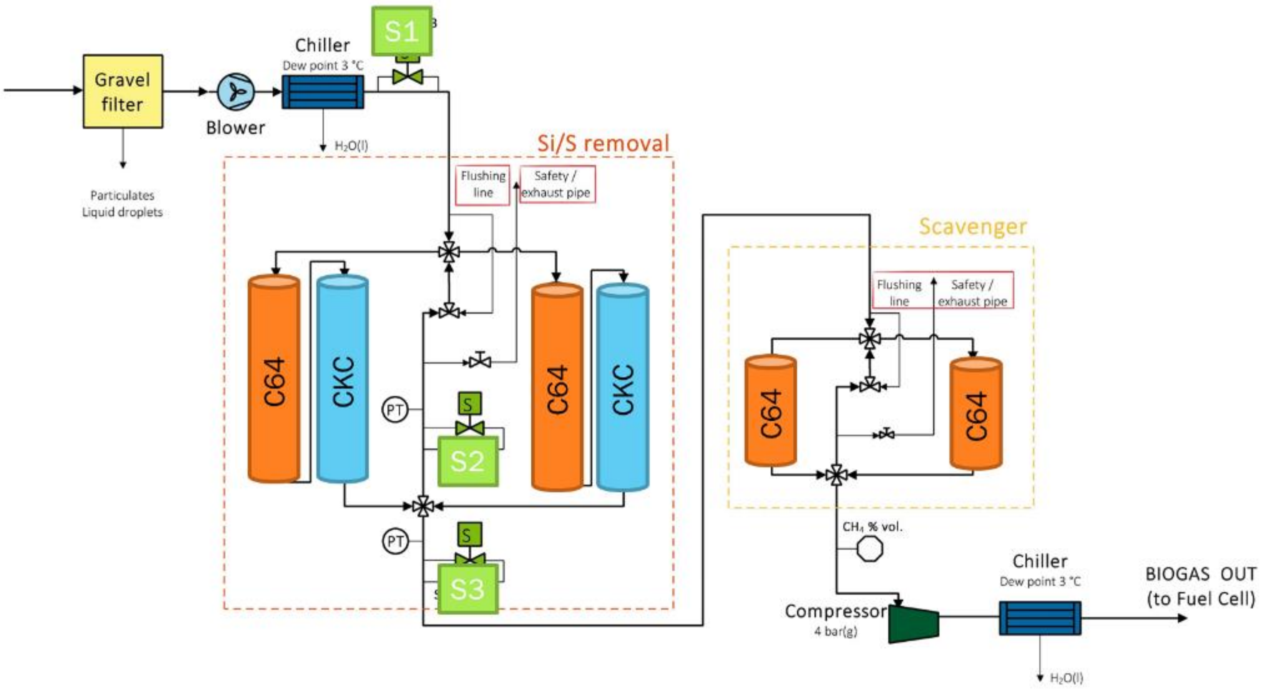


Figure 4 Layout of the clean-up system

Clean-up adsorbents are provided by *AirDep Srl*. CKC is an activated carbon of mineral origin and it is particularly suitable for the removal of hydrogen sulphide (H_2S), low molecular weight mercaptans, acid gases and organic sulphurs. C64 is another activated carbon and it is suitable for the adsorption of volatile and incondensable compounds, such as siloxanes. Both sorbents have similar carbon contents with similar elemental composition, but different structures, specific surface areas, and micropore volumes. Both sorbents predicted adsorption capacity q has been assumed equal to the one found through screening tests performed in the presence of 0.1 [%] of O_2 [x]: oxygen availability in small quantities, in the range of 0.01÷0.33 [%], is confirmed by the laboratory analysis performed in the previous years [x].

CKC equivalent sulphur adsorption capacity $$q_{CKC} = 84 \left[\frac{mg}{g}\right]$$

C64 equivalent D4 adsorption capacity $$q_{C64} = 150 \left[\frac{mg}{g}\right]$$

Predicted adsorption capacities are available in terms of equivalent sulphur and equivalent D4 respectively. Octamethylcyclotetrasiloxane ($C_8H_{24}O_4Si_4$), also called D4, is an organosilicon compound belonging to the cyclic dimethylsiloxane group. Together with D3 and D5, they are the most abundant siloxane species contained in the digester biogas [x].

The first two series of reactors operate in series in a lead-and-lag configuration, such that the leading reactors adsorb the majority of the contaminants: the configuration can be changed through valves, allowing to perform maintenance or to replace saturated carbons on each component individually. The last couple of reactors acts as “scavengers”, preventing any contaminants breakthrough from the previous system.

In the plant area dedicated to the DEMSOFC project, it was forecasted the installation of three SOFC modules, provided by CONVION. Modules are set in parallel and they are able to operate autonomously. As of March 2020, only two of them have been installed, but they have never been operated together. According to the project nomenclature, module №3 has been the first to be installed and operated. The relative data sheet is presented in Figure 5. It has a nominal electrical power of 58 [kW], while the second, module №2, has a nominal electrical power of 45 [kW]. The two modules have the same number of cells stacks but different nominal power since two different producers manufactured them. The three modules are expected to cover about 30 [%] of the plant total electrical needs.

Performance	**Targets**
Net power output	58kW (3x400-440V AC 50/60Hz)
Energy efficiency (LHV)	
Electrical (net , AC)	> 53 %
Total (exhaust 40°C)	> 80 %
Heat recover	
Exhaust gas flow	650 kg/h
Exhaust gas temperature	222 °C
Emissions	
NOx	< 2 ppm
Particulates(PM10)	< 0.09 mg/kWh
CO_2 (NG, nominal load)	354 kg/MWh
CO_2 (with heat recovery)	234 kg/MWh
Fuels	Natural gas, City gas, Biogas
Dimensions (L x W x H)	
power unit	3,5 x 1,9 x 2,3 m
aux. equipment	2,4 x 0,6 x 2,2 m
Noise level	< 70 dB(A) at 1 m
Installation	Indoor / outdoor
Temperature	-20 – +40°C

Figure 5: Convion C50 module data sheet

Although the internal working of the modules is unknown, each of them is at least equipped with stacks of solid oxide fuel cells, an air blower, electrical resistances and a heat recovery unit (HRU). Each fuel cell is an electrochemical reactor: electrical power is generated exploiting the electronic transfer occurring in the redox reaction between the oxygen contained in the cathodic air stream and the hydrogen in the anodic stream (2a+2b).

$$H_2 + O_2^- \rightarrow H_2O_{(g)} + 2e^- \quad (2a)$$

$$\frac{1}{2}O_2 + 2e^- \rightarrow O_2^- \quad (2b)$$

$$H_2 + \frac{1}{2}O_2 \rightarrow H_2O_{(g)} \quad \Delta H_{(STP)} = -286 \left[\frac{kJ}{mol_{H_2}}\right] \quad (2a+2b)$$

Hydrogen is produced through reforming, either wet (3) or dry (4), of the methane fraction of biogas. Steam required for the reforming is usually obtained by recirculating part of the anodic exhaust stream. It is not known if the modules include a dedicated reforming reactor or if the reactions are carried out directly on the cells surface.

$$CH_4 + CO_2 \rightarrow 2H_2 + 2CO \quad \Delta H_{(STP)} = +247 \left[\frac{kJ}{mol_{CH_4}}\right] \tag{3}$$

$$CH_4 + H_2O_{(g)} \rightarrow 3H_2 + CO \quad \Delta H_{(STP)} = +206 \left[\frac{kJ}{mol_{CH_4}}\right] \tag{4}$$

Carbon monoxide can also act as fuel like hydrogen, but the water gas shift reaction (5) has a faster kinetic and its contribution is therefore negligible [x].

$$CO + H_2O_{(g)} \rightarrow CO_2 + H_2 \quad \Delta H_{(STP)} = -41 \left[\frac{kJ}{mol_{CO}}\right] \tag{5}$$

The modules must be operated at high temperature, around 700÷800 [°C], to enable proper activity of the ceramic layers that make up the fuel cells. Although the overall process generates heat, the reforming reactions are endothermic, therefore high temperature helps to increase the kinetic. While the module temperature can be controlled by removing heat through air flow variation, during the start-up phase the module need to be initially heated by the electrical resistances, absorbing heat from the grid, at least until the reactions can sustain themselves.

Since the voltage difference that drives the hydrogen redox reaction also depends on the reactants concentration on the two electrodes surface, fuel flow is purposely set to have unspent reactants when leaving the stack, ensuring uniform performance of the cells. Modules are therefore usually equipped with an afterburner that carries out the mixing and the combustion of the excess fuel and air streams. The heat recovery unit (HRU) absorbs the heat from the gaseous exhaust, partially employing it within the module to pre-heat reactants flows. However, most of it is recovered by the external heat recovery circuit, releasing the exhaust in the atmosphere at 70÷80 [°C].

The external heat recovery circuit spans from the DEMOSOFC area to the pumps and sludge heat exchanger utility room (number 1 and 5 in Figure 1). The thermal fluid employed is a mono-ethylene glycol water mixture at 30 [$\%_{vol}$], whose data are provided by the producer *Fimi S.p.A*.

Glycol water mixture at 30 [$\%_{vol}$] specific heat capacity $\quad c_{p_{glycol\ water}} = 3.765 \left[\frac{kJ}{kg \cdot K}\right]$

Glycol water mixture at 30 [$\%_{vol}$] density $\quad \rho_{glycol\ water} = 1024 \left[\frac{kg}{m^3}\right]$

Each module internal heat exchanger (respectively identified as HRU1, HRU2, HRU3) is served by a branch of the circuit composed of a set of twin pumps, thermostats, a flow meter and a three-way valve for partial recirculation. The main piping then collect and carry the thermal fluid for more than 50 [m] on dedicated racks mounted outdoor to a tube in tube heat exchanger (HRU4). Here, heat is transferred to a mixture of sludge, pre-emptively uniformed in a macerator, and industrial water. Another set of twin pumps, thermostats and flow meter are installed on the glycol water side in the utility room. An hydraulic separator is installed near the branches serving the modules: this component puts in contact supply and return glycol water, exploiting the thermal stratification between the two to balance pressure losses across the system, and avoiding flow in inactive modules. A scheme of the overall circuit is presented in Figure 6.

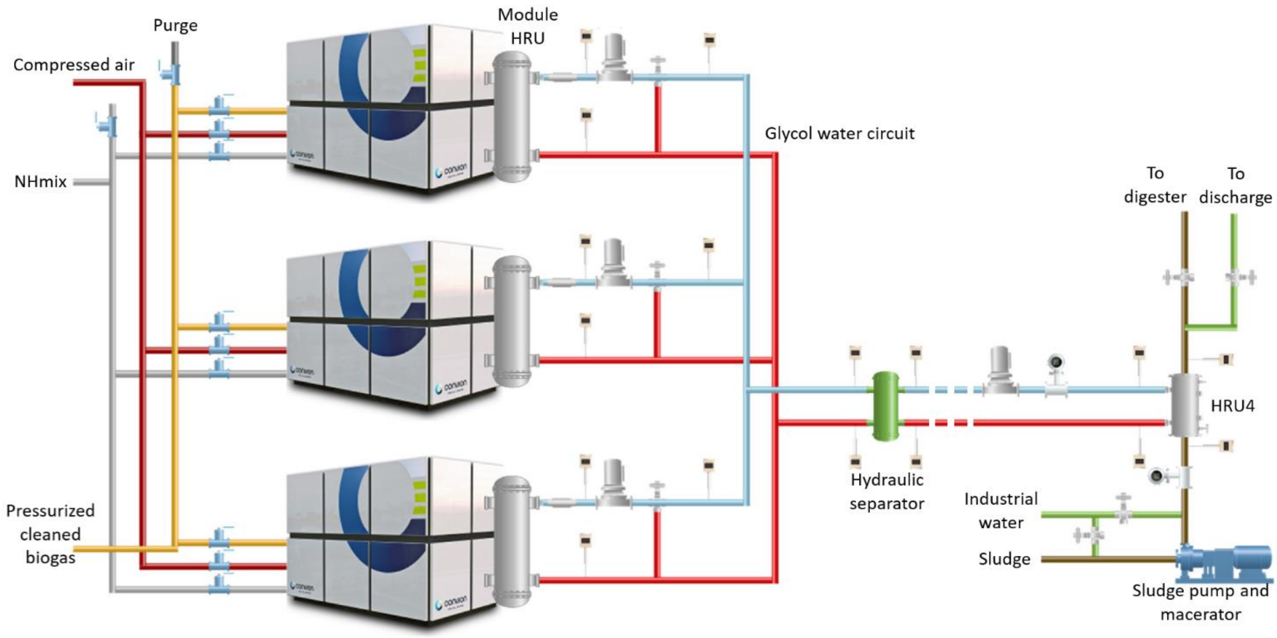


Figure 6: Scheme of modules feed and external thermal recovery circuit

In the dedicated DEMOSOFC area, besides the modules and the BIOKOMP container, there are other structures dedicated to the balance of plant components. A technical building hosts the hydraulic components of the glycol water circuit serving the modules mentioned before, together with a control room, containing the dashboard that acts as interface for the operator, and an electrical room. In the latter are installed the inverters, the electrical cabinets, medium-voltage switchgear, power meters and two electric sinks of 20 [kW] each. An uninterruptable power supply (UPS) with the dedicated batteries are installed in a smaller prefabricated just outside of the building. All of these equipment are connected among them and to the grid as in Figure 7.

Inverters convert the DC power output of the modules to medium voltage AC current. During the start-up of the modules, the grid supplies power to both the modules and the auxiliary equipment for several hours. While the operations are nominal, the grid powers the auxiliaries through the UPS main line and the module send their power to the WWTP line.

In case of failure of the external grid, a couple of relays can disconnect the system allowing the modules to work in island mode. In island mode, the modules reduce their power output in order to just maintain the auxiliaries operating through the UPS by-pass line, avoiding to switch off the entire system, which would need time and could damage the cells. In case more than one module is on, the excess of power is dissipated by the two electric sinks. The UPS role is to handle the transient condition, avoiding voltage drops and guaranteeing the stability of the power supply thanks to some stacks of batteries.

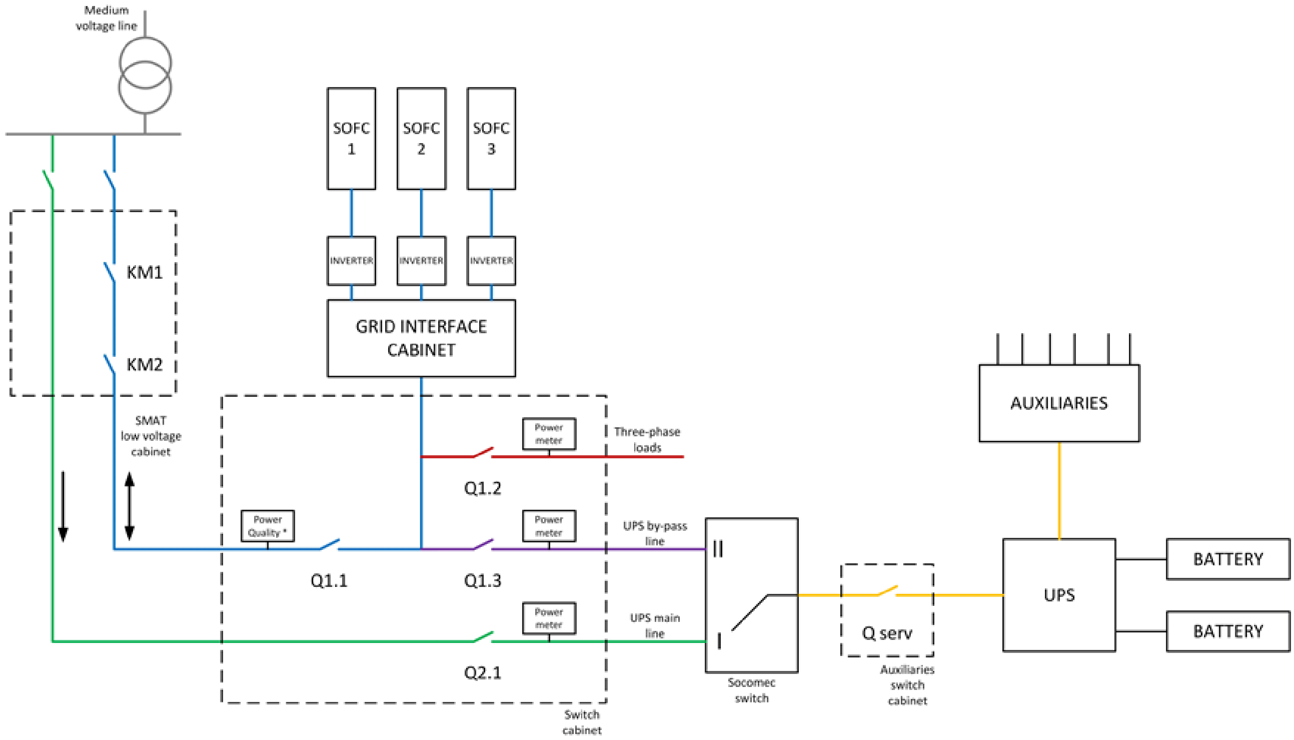


Figure 7: Electrical layout schematization

The auxiliaries include lights, electric sockets, the air conditioning systems of the technical building system, the dashboard transformer, all the pumps, the chiller, compressor and air conditioning of the clean-up container, the control panel, the macerator and the air compressor. The air compressor, which supplies the module with the air needed during start-up or anodic channels purge, mainly supplies the entire WWTP and can be manually loaded on the external grid or on the modules, through a specific switch. The compressor is usually loaded on the module to keep it operative during blackouts and relative island event.

In the same area there is also a prefabricated that holds technical gases tanks, namely nitrogen and a mixture of hydrogen and nitrogen (95% N_2, 5% H_2, abbreviated as NH_{mix} for future reference). These are also used by the modules during emergency procedures to respectively purge the biogas piping of the clean-up container and keep the modules in hot standby. This last procedure is meant to keep the fuel cell stack at high temperature providing only a small amount of fuel, until operation can be safely restored.

2.2. Toolkit scope

The scope of this work has been to develop a user-friendly tool to be used by the plant operators to collect and organize all of the raw information made available by the Programmable Logic Controller (PLC) installed in the control room and to analyse the plant performance for maintenance and diagnostic activities.

The tool is based on the Visual Basic for Applications environment implemented in the Microsoft Office suite. The tool comprises several routines and user interfaces integrated into an Excel file, with also a couple of Word files used as report templates.

In a dedicated tab called "DEMOSOFC" added to the Excel file ribbon, four new commands have been implemented (Figure 8). All of them will prompt a specific user form, while to start the actual procedures some input from the user is required.

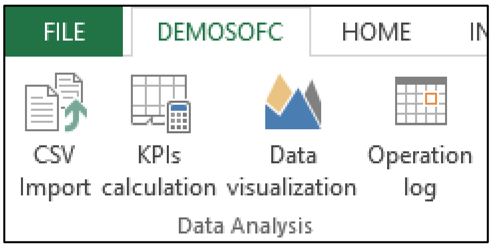


Figure 8: Custom tab and commands in Excel file ribbon

The "CSV Import" command allows the user to select one or more files at the same time. It is meant to work with the files produced by the PLC, generated with the .CSV extension. These files will be opened one at a time and the data contained within will be loaded in order in the workbook in the corresponding worksheet. A status bar in the bottom left corner keeps updating to show the procedure progress. User intervention will be required only if some duplicate data is found. When the procedure is over, a message box will appear signalling correct execution.

The "KPIs calculation" command allows the user to select a year for which Key Performance Indicators (KPIs) will be evaluated. The procedure is kept separated from the previous one as Excel processing may slow down when code is run for too long. These parameters are extremely useful in determining plant performance and operation issues and help with predictive maintenance. The results will again be saved in the corresponding worksheet. A status bar in the bottom left corner keeps updating to show the procedure progress. When the procedure is over, a message box will appear signalling correct execution.

The "Data visualization" command allows the user to either create or edit an existing chart. Charts add up to the sheet list in the workbook but do not interfere with the other procedures. Two main types of graphics can be displayed: a wide selection of plant parameters and KPIs can either be plotted with respect to time as line charts or with respect to the same plethora of outputs as a scatter plot. Both can be displayed for a customizable time period, although it is limited within the same year for technical issues. Together with the chart, some overview parameters will be evaluated and shown inside the user form. These values are only temporary, and closing the user form will delete them: a dedicated page allows the user to export them and the chart itself as a brief Word report, which is based on a template saved in the same folder as the workbook. The chart may also be saved individually as a .PNG image. While the user form is open, interaction with the workbook is not permitted, but a dedicated button allows the user to freely modify the chart without losing overview data.

The "Operation log" command allows the user to see overview data for a selected year or for the entire lifetime of the plant (starting from 2018). Again, these can be exported into a similar Word report. The user form contains more information on nominal operation periods.

Aside from these new commands, the file is still an Excel workbook, so the operator may still create worksheets and make calculations on available data.

3. Materials and methodology

3.1. Available data and import procedure

The DEMOSOFC Programmable Logic Controller (PLC) collects information from various field sensors scattered across the plant and also receives data from CONVION PLC in Finland, which monitors the internal operation of the SOFC modules (Figure 9).

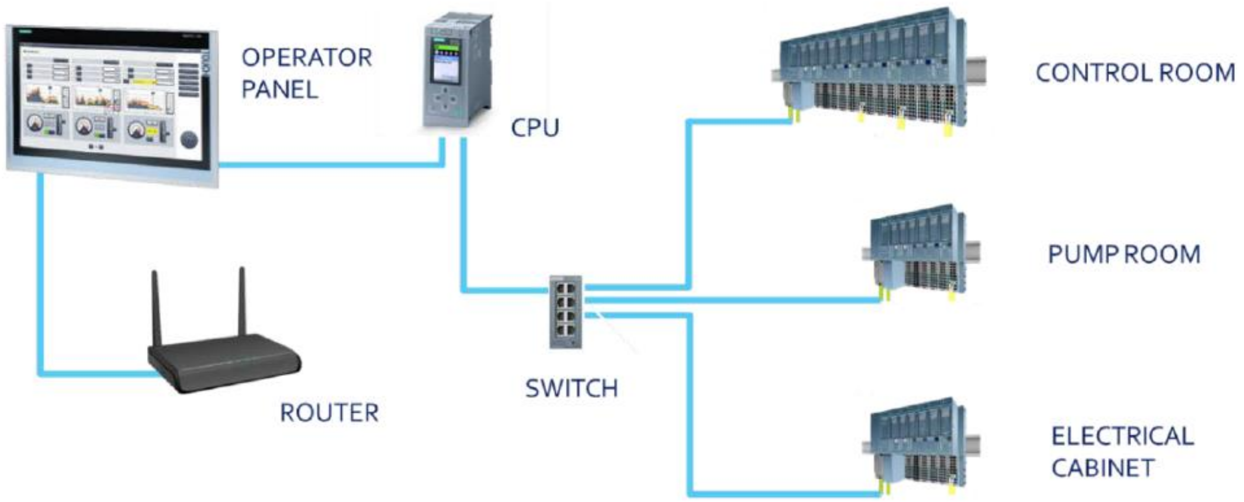


Figure 9: DEMOSOFC Programmable Logic Controller layout

Data is automatically logged by the software every ~10 minutes, divided into four categories:

- 91 analogic measures, of which:
  - 35 parameters about fluids and equipment controlled by SMAT (labelled as FIW)
  - 43 parameters about fluids and electrical measures related to the SOFC modules (labelled as ETH)
  - 5 parameters about the clean-up compressor
  - 8 values of the speed set-point for the pumps on the water-glycol circuit
- 29 electrical measures coming from the two electrical cabinets of DEMOSOFC plant and WWTP
- 125 binary digital signals related to commands and equipment status
- 5 biogas composition measures per 8 sample points, although only three of them are connected to the clean-up system at the same time

In the following sections, instead of employing the nomenclature used by the PLC, the aforementioned measures and signals will be recalled for simplicity with a letter according to the data category - A for analogic, E for electric, D for digital, G for biogas composition - and a number (e.g. A1 instead of FIW_512_AP01_PRESSURE_DeEPT003). The complete list of information can be found in Appendix 8.1.

These information can be extracted by the operator from the PLC internal memory as a series of Comma Separated Value (CSV) files. The first command of the toolkit allows the user to select these files and start an import procedure that will ultimately allow storing the data in a properly available form in the workbook itself.

Cycling through each CSV file, the algorithm will first detect the correct file category and year, assigning or creating the relative worksheet. The log is pre-emptively checked to remove any useless record due to disconnections or failed measurements. Then, for each date that is identified while cycling through the file, the corresponding data are moved in bulk to a new row into the worksheet. If the number of measures appear different from the standard values mentioned above, then the assignment is performed individually.

Any duplicate date which may be detected as already present in the worksheet will trigger a user form to appear, leaving to the operator the choice to either overwrite or ignore the record(s), or abort the import procedure entirely, closing the corresponding file (Figure 10). Duplicates due to Daylight Saving Time (DST) ending in October are accounted for, adding the suffix "DST" to the relative dates in the workbook.

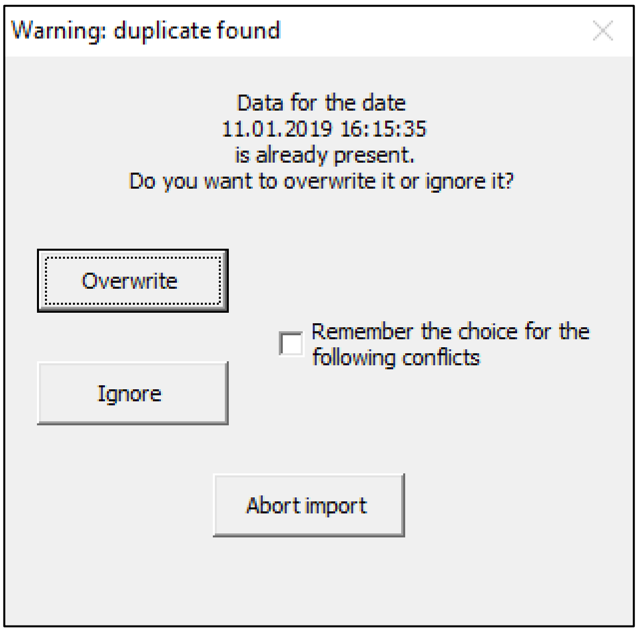


Figure 10: Example for duplicate data user form

Biogas composition logs are checked beforehand to ensure that they contain useful data and are otherwise closed. Measures from different sample points also appear to be sometimes saved with a 1÷2 seconds latency from one another and are therefore checked to determine whether the date has already been stored in the worksheet.

At every change in year or data category, the contents of the corresponding worksheet are ordered chronologically. Once the import procedure is over, worksheets are ordered alphabetically, and a message box will notify the user that it has been successfully completed. The complete routine is attached in Appendix 8.3.1.

3.2. Key Performance Indicators evaluation

The second command of the toolkit will prompt a user form (Figure 11) that recaps the number of records of each data category for the selected year and allows to perform the evaluation of the Key Performance Indicators (KPIs) for each time step. The number of KPIs that has already been calculated also appear in the list; therefore, the operator may choose to run the routine only for newly imported time steps.

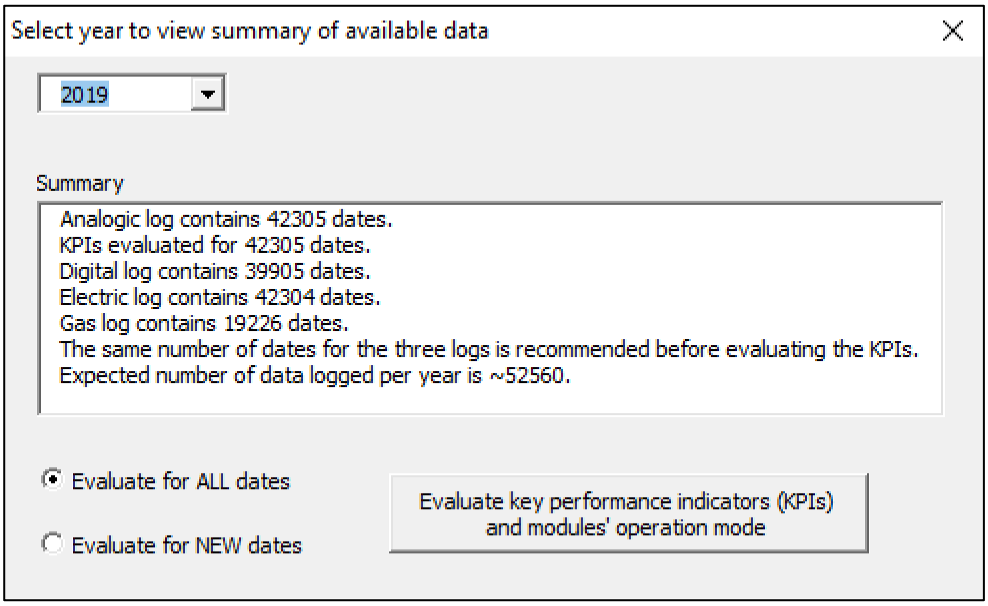


Figure 11: Key Performance Indicators user form - Example for the year 2019

The routine (Appendix 8.3.2) performs the evaluation of 35 KPIs, handling missing data, divisions by zero and unreliable values of efficiencies. The analogic data category worksheet has been used as reference for the time steps, since all of the KPIs relate to one or more of its measures. KPIs are directly written in the corresponding cell as values, since the use of a high number of Excel formulas is not recommended due to the fallout on calculation speed and memory usage.

Another parameter which is evaluated is the elapsed time between one time step and the previous one; this will be used to determine whether the PLC failed to collect or to write data with its designed frequency. Again, this calculation accounts for DST changes, assigning the standard value of 10 minutes both in March and October.

The KPIs are listed and coded in Table 1 and described in the following paragraphs.

Key Performance Indicator	Unit of measure	Code
Biogas volumetric flow rate total	[Nm3/h]	K1
Biogas volumetric flow rate mod.1	[Nm3/h]	K2
Biogas volumetric flow rate mod.2	[Nm3/h]	K3
Biogas volumetric flow rate mod.3	[Nm3/h]	K4
Methane mass flow rate mod.1	[kg/h]	K5
Methane mass flow rate mod.2	[kg/h]	K6
Methane mass flow rate mod.3	[kg/h]	K7
Biogas power input mod.1	[kW]	K8
Biogas power input mod.2	[kW]	K9
Biogas power input mod.3	[kW]	K10
Global power generation	[kW]	K11
Global power exchange	[kW]	K12
Electrical efficiency mod.1	[%]	K13
Electrical efficiency mod.2	[%]	K14
Electrical efficiency mod.3	[%]	K15
Thermal power removed from mod.1	[kW]	K16
Thermal power removed from mod.2	[kW]	K17
Thermal power removed from mod.3	[kW]	K18
Thermal power removed from modules	[kW]	K19
Thermal power transfered from water/glycol to sludge	[kW]	K20
Thermal power received by sludge from water/glycol	[kW]	K21
Thermal efficiency of heat exchange system	[%]	K22
Thermal efficiency mod.1	[%]	K23
Thermal efficiency mod.2	[%]	K24
Thermal efficiency mod.3	[%]	K25
Global efficiency mod.1	[%]	K26
Global efficiency mod.2	[%]	K27
Global efficiency mod.3	[%]	K28
Auxiliary equipment power consumption	[kW]	K29
Net electrical power output of modules	[kW]	K30
Net electrical efficiency of plant	[%]	K31
Net thermal efficiency of plant	[%]	K32
Net global efficiency of plant	[%]	K33
Equivalent sulphur adsorption rate	[mg/h]	K34
Equivalent D4 adsorption rate	[mg/h]	K35

Table 1: Key Performance Indicators

Biogas volumetric flow rate

Starting from the biogas mass flow rate $\dot{m}_{biogas,mod}$ being fed into each module (A60, A61, A62), available in [kg/h], the volumetric flow rate $\dot{V}_{biogas,mod}$, expressed in [Nm^3/h], is evaluated for each module (K1, K2, K3) as:

$$\dot{V}_{biogas,mod} = \dot{m}_{biogas,mod} \cdot \frac{22.414}{MW_{biogas}} \tag{6}$$

The molecular weight of the biogas stream MW_{biogas} is computed assuming that the gas is a mixture of methane and carbon dioxide, therefore overlooking the contributions due to oxygen or contaminants:

$$MW_{biogas} = \frac{\%_{vol,CH_4}}{100} \cdot MW_{CH_4} + \left(1 - \frac{\%_{vol,CH_4}}{100}\right) \cdot MW_{CO_2} \tag{7}$$

Methane molecular weight $MW_{CH_4} = 16.04 \left[\frac{g}{mol}\right]$

Carbon dioxide molecular weight $MW_{CO_2} = 44.01 \left[\frac{g}{mol}\right]$

The measure of volumetric content of methane in the biogas $\%_{vol,CH4}$, available in [%], is taken from the ETH fast sensor (A91) managed by CONVION, but in case this data is missing the value from the DeFIT005 sensor (A26) managed by SMAT is used, lowered by 1.5 [%] according to previous reliability observations [x].

The sum of the three volumetric flow rates (K4) is also saved as an additional KPI.

Methane mass flow rate

The mass flow rate of methane $\dot{m}_{CH4,mod}$ being fed in each module (K5, K6, K7), expressed in [kg/h], is evaluated with the same assumptions previously declared as:

$$\dot{m}_{CH_4,mod} = \dot{m}_{biogas,mod} \cdot \frac{\%_{vol,CH_4}}{100} \cdot MW_{CH_4} \tag{8}$$

Biogas power input

The thermochemical power input $P_{CH4,mod}$, expressed in [kW], associated with the lower heating value of methane entering each module (K8, K9, K10) is evaluated from the corresponding mass flow rate as:

$$P_{CH_4,mod} = \dot{m}_{CH_4,mod} \cdot \frac{LHV}{3600} \tag{9}$$

Methane lower heating value $LHV = 50{,}000 \left[\frac{kJ}{kg}\right]$

Thermal power removed from module

The thermal power, expressed in [kW], recovered from the gaseous exhaust stream of each module $P_{th,mod}$ (K16, K17, K18) is evaluated on the glycol water side of each HRU as:

$$P_{th,mod} = \frac{(c_p \cdot \rho)_{glycol\ water} \cdot \left(\dot{V} \cdot (T_{out} - T_{in})\right)_{glycol\ water,mod}}{3600} \tag{10}$$

The fluid temperature increase $(T_{out} - T_{in})_{glycol\ water,mod}$ is evaluated according to the readings (A5-A4, A8-A7, A10-A9) of the thermometers installed before and after the corresponding internal heat exchanger of the modules. The glycol water volumetric flow rate $\dot{V}_{glycol\ water,mod}$ of each module (A18, A19, A20) is available in [m^3/h]. The flow meters appear to display a fixed value of 0.3 [m^3/h] when the pumps are inactive, most likely due to the instrument sensibility. These KPIs are therefore evaluated only if the pumps are operating and the temperature difference is positive.

The sum of the three thermal powers removed from the modules (K19) is also saved as an additional KPI.

Thermal power transferred from glycol water to sludge

Similarly $P_{th,hex,glycol}$ (K20), the thermal power actually delivered to the sludge through the heat exchanger HRU4, evaluated in [kW] on the glycol water side, is computed as:

$$P_{th,hex,glycol} = \frac{\left(c_p \cdot \rho\right)_{glycol\ water} \cdot \left(\dot{V} \cdot (T_{in} - T_{out})\right)_{glycol\ water,HRU4}}{3600} \tag{11}$$

The total glycol water volumetric flow rate $\dot{V}_{glycol\ water,HRU4}$, available in [m^3/h], is measured by a dedicated flow meter (A34). Digital signals of the twin pumps located near the heat exchanger (D106, D107) are used to determine if at least one is active. This KPI is again evaluated only if the pumps are operating and the temperature difference (A28-A29) is positive.

Thermal power received by sludge from glycol water

$P_{th,hex,sludge}$ (K21), the thermal power transferred through the heat exchanger HRU4, evaluated in [kW], is computed on the sludge side as:

$$P_{th,hex,sludge} = \frac{\left(c_p \cdot \rho\right)_{sludge\ water} \cdot \left(\dot{V} \cdot (T_{out} - T_{in})\right)_{sludge\ water,HRU4}}{3600} \tag{12}$$

The flow rate $\dot{V}_{sludge\ water,HRU4}$, available in [m^3/h], is measured by a flow meter (A35) placed after the collector joining sludge and industrial water. It has been assumed that the fluid equates to pure water, since most of the time the heat exchanger is fed with industrial water, and the treated sludge has a low solid fraction nonetheless.

Water specific heat capacity	$c_{p\,sludge\ water} = 4.18 \left[\frac{kJ}{kg \cdot K}\right]$
Water density	$\rho_{sludge\ water} = 999 \left[\frac{kg}{m^3}\right]$

This KPI is evaluated only if the pumps on the glycol water side are operating and the temperature increase measured by the thermostats on the sludge side (A31-A30) is positive.

Global power generation

The global power generation $P_{el,gen}$ (K11), expressed in [kW], is evaluated as the sum of the electrical powers produced by the modules:

$$P_{el,gen} = \sum_{1\ to\ 3} P_{el,mod}\ \left(if\ P_{el,mod} > 0\right) \tag{13}$$

Electrical power readings $P_{el,mod}$ of each module (A36, A37, A38) is accounted as production only if positive. Negative readings are possible during non-nominal operations, as the module can absorb power from the grid to feed its electrical components.

Global power exchange

The global power exchange $P_{el,ex}$ (K12), expressed in [kW], is evaluated as the sum of the electrical power produced or absorbed by the modules:

$$P_{el,ex} = \sum_{1\ to\ 3} P_{el,mod} \tag{14}$$

Auxiliary equipment power consumption

Current and voltage readings from the power meter of electric panel Q2 (E28, E29), available in [A] and [V] respectively, allow to determine the auxiliary equipment power consumption $P_{el,aux}$ (K29), expressed in [kW], as:

$$P_{el,aux} = \frac{\sqrt{3} \cdot (V \cdot I)_{power\ meter\ Q2}}{1000} \quad (15)$$

The equipment connected to this electric panel is all part of the DEMOSOFC installation. The main consumptions are related to the clean-up system, which include the chiller, the biogas blower and the biogas compressor. The major power absorber though, is the air compressor that provides compressed air to the WWTP. The pneumatic circuit has been extended to the modules, as compressed air is required during non-nominal operation to purge biogas from the anode channels. Despite the marginal utilization of the air compressor for the DEMOSOFC project, its entire contribution can be accounted among the auxiliaries consumptions if charged on the electric panel Q2. A complete list of the auxiliaries equipment connected to the electric panel Q2 is available in Appendix 8.2.

Net electrical power generation

The net electrical power generation $P_{el,net}$ (K30), expressed in [kW], is evaluated as difference between the global power exchange (K12) and the auxiliary equipment power consumption (K29) described above:

$$P_{el,net} = P_{el,ex} - P_{el,aux} \quad (16)$$

This KPI represents the amount of power made available by the modules for use inside the plant, already accounting for the equipment dedicated to the DEMOSOFC project. When the value is negative, it represent the amount of power withdrawn from the national grid for the operation of modules and auxiliaries.

Electrical efficiency of module

Electrical efficiency $\varepsilon_{el,mod}$, expressed in [%], is evaluated for each module (K13, K14, K15) as the ratio between the corresponding produced electrical power (A36, A37, A38) and the biogas power input (K8, K9, K10):

$$\varepsilon_{el,mod} = \frac{P_{el,mod}}{P_{CH_4,mod}} \cdot 100 \quad (17)$$

These values are evaluated only when the corresponding module is producing power and there is fuel flow.

Thermal efficiency of module

Thermal efficiency $\varepsilon_{th,mod}$, expressed in [%], is evaluated for each module (K22, K23, K24) as the ratio between the corresponding thermal power recovered (K16, K17, K18) and the biogas power input (K8, K9, K10):

$$\varepsilon_{th,mod} = \frac{P_{th,mod}}{P_{CH_4,mod}} \cdot 100 \quad (18)$$

These values are evaluated only when there is fuel flow. They may exceed 100 [%] since the glycol water can still remove heat from a module exhaust even in the absence of fuel feed: in this scenario the value has been set to 100 [%].

Global efficiency of module

Global efficiency $\varepsilon_{gl,mod}$, expressed in [%], is evaluated for each module (K26, K27, K28) as the sum of the corresponding electrical (K13, K14, K15) and thermal (K22, K23, K24) efficiencies, therefore displaying how much of the thermochemical energy of the biogas stream has been recovered through the whole module:

$$\varepsilon_{gl,mod} = \varepsilon_{th,mod} + \varepsilon_{th,mod} \tag{19}$$

Again, the same observation made for the module thermal efficiency applies for these KPIs.

Thermal efficiency of heat exchange system

The thermal efficiency of the entire heat exchange system $\varepsilon_{th,hex}$ (K22), expressed in [%], is evaluated as the ratio between the thermal power transferred in HRU4 evaluated on the water glycol side (K20) and the sum of the thermal power recovered from the three modules (K19):

$$\varepsilon_{th,hex} = \frac{P_{th,hex,glycol}}{\sum_{1\ to\ 3} P_{th,mod}} \cdot 100 \tag{20}$$

This KPI is complementary to the percentage thermal losses occurring at the hydraulic separator and along the piping connecting the modules to the utility room of the heat exchanger HRU4.

Net electrical efficiency of plant

Net electrical efficiency $\varepsilon_{el,net,plant}$ (K31), expressed in [%], is computed as the ratio between the net electrical power generation (K30) and the overall biogas power input (K8 + K9 + K10):

$$\varepsilon_{el,net,plant} = \frac{P_{el,net}}{\sum_{1\ to\ 3} P_{CH_4,mod}} \cdot 100 \tag{21}$$

This value is evaluated only when the system is producing net power and there is fuel flow in at least a module.

Net thermal efficiency of plant

Net thermal efficiency $\varepsilon_{th,net,plant}$ (K32), expressed in [%], is computed as the ratio between the thermal power transferred in HRU4 evaluated on the water glycol side (K20) and the overall biogas power (K8 + K9 + K10):

$$\varepsilon_{th,net,plant} = \frac{P_{th,hex,glycol}}{\sum_{1\ to\ 3} P_{CH_4,mod}} \cdot 100 \tag{22}$$

Again, the same observation made for the modules thermal efficiency applies for this KPI.

Net global efficiency of plant

Net global efficiency $\varepsilon_{th,net,plant}$ (K33), expressed in [%], is computed as the sum of electrical (K31) and thermal (K32) net efficiencies, therefore displaying how much of the thermochemical energy of the biogas stream sent to the modules has been recovered through the whole system:

$$\varepsilon_{gl,net,plant} = \varepsilon_{el,net,plant} + \varepsilon_{th,net,plant} \tag{23}$$

Equivalent sulphur adsorption rate [mg/h]

The adsorption rate of equivalent sulphur $\dot{m}_{eq,S}$ (K34), expressed in [mg/h], which is captured by the lead-and-lag reactors containing CKC carbons is evaluated from the concentration of hydrogen sulphide χ_{H2S} (G3), available in [ppm], of the total biogas stream sent to the modules (K1), measured before entering the clean-up system:

$$\dot{m}_{eq.S} = \chi_{H_2S} \cdot \frac{MW_S}{22.414} \cdot \sum_{1\ to\ 3} \dot{V}_{biogas,mod} \tag{24}$$

Sulphur molecular weight	$MW_S = 32.07 \left[\frac{g}{mol}\right]$

The assumption is that all of the contaminant is adsorbed: this hypothesis is supported by the data available from the QUALVISTA measuring system, which is less frequent but more accurate than the data logged by the PLC, especially for what concerns the following sample points. This KPI will be used to determine the loading rate of the reactors with respect to their predicted capacity.

Equivalent D4 adsorption rate

Similarly, the adsorption rate of equivalent D4 $\dot{m}_{eq,D4}$ (K35), expressed in [mg/h], which is captured by the lead-and-lag reactors containing C64 carbons is evaluated from the concentration of silicon γ_{Si} (G5), available in [mg/m^3], of the total biogas stream sent to the modules (K1), measured before entering the clean-up system:

$$\dot{m}_{eq.D4} = \gamma_{Si} \cdot \frac{MW_{D4}}{4 \cdot MW_{Si}} \cdot \sum_{1\ to\ 3} \dot{V}_{biogas,mod} \tag{25}$$

Silicon molecular weight	$MW_{Si} = 60.08 \left[\frac{g}{mol}\right]$
Octamethylcyclotetrasiloxane ($C_8H_{24}O_4Si_4$) molecular weight	$MW_{D4} = 292.62 \left[\frac{g}{mol}\right]$

The same assumption made for hydrogen sulphide adsorption have also been made for siloxanes.

3.3. Operation modes and identification algorithm

Once all the KPIs have been evaluated, the same routine proceeds to identify the operation mode of each module for each time step. The modalities considered are described here.

ON

For nominal operation: the module is fed with biogas and ambient air and power production is positive, according to the current set-point and the actual fuel utilization. Current set-point of the modules has been, for most of the time the system has been operative, remotely imposed by CONVION.

OFF

When modules are inactive: this mode is also assigned when the module is run remotely to perform tests without actually producing net power.

ISL
For island mode events: these are expected by the system, to avoid the shutdown of the module in case of external grid failures. The module keeps producing electricity to feed the essential auxiliaries. There is not always a complete disconnection from the grid.

HOT
For hot standby events: this procedure is meant to maintain fuel cells at high temperature by injecting NH_{mix} on the anode side. Meanwhile, biogas feed is cut off and the module stops producing electricity. Once it is over, the stack should be ready to resume nominal operation. This procedure can also be imposed by the operator to allow short-term maintenance on the clean-up section while one or more modules are running.

INT
For generic interruptions: electrical power output drops significantly, and may recover immediately or after some hours. These events are not expected by the system, and their cause is not univocal.

START
For the start-up procedure: before being able to feed the module under nominal operation, it has to be heated up from ambient temperature to around 700 [°C]. The whole procedure should take from 24÷48 hours, but it may actually require up to almost four days. In the first phase, the modules consume power from the grid to produce heat through electrical resistances. Ambient air is blown through the fuel cells cathodic side, acting as a thermal medium. Compressed air is later injected on the anodic side and, as the modules heat up, it is gradually substituted by the fuel, initiating the necessary thermal and chemical conditions in the system. Throughout this phase, the cathode air is maintained at a level capable of diluting fuel below the lower flammability limit at the afterburner, avoiding the accumulation of explosive gas mixture anywhere in the system.

SHUT
For the shutdown procedure: the current in the stack is lowered to zero, and the modules begins to cool down, aided by ambient air ventilation. Fuel flow is also stopped as compressed air injection purges the stack.

The operation mode identification procedure is carried out for each module individually.
Employed analogic readings distinct for each module are:

- Module electrical power output P_{el} [kW] (A36, A37, A38)
- Module compressed air flow intake $\dot{m}_{CA}$ [m^3/h] (A48, A49, A50)
- Module temperature T (A39, A40, A41)
- Module ambient air flow rate intake $\dot{m}_{AA}$ [m^3/h] (A45, A46, A47)

Employed digital data shared by all modules are:

- Active island mode event digital signal (D9)
- Air compressor operation digital signal (D11)
- Common fuel valve DeVG01 open digital signal (D37)

Through a series of logical checks, the operation mode of a module can either be directly identified or it has to be further investigated by controlling data of subsequent time steps through a specific sub-routine: the sub-routine identifies when a given operation mode ends, and assigns the output to the dedicated worksheet cells inbetween. Investigation stops beforehand if there is more than 30 minutes between one time step and the following one, or if the last available record is reached. Data and values employed in these procedures have been based on manual event analysis of the last two years of plant operation. The code for the overarching logical checks is written in Appendix 8.3.2, after the evaluation of KPIs. The aforementioned sub-routines code can be found in Appendix 8.3.3.

The identification of the operation mode is performed based on the continuity of available data. In case the elapsed time between a time step *t* and the previous one is longer than 10 minutes, the detection is based purely on data belonging to the corresponding time step *t*. The logic of this "static" assumption is summarized in the flowchart of Figure 12.

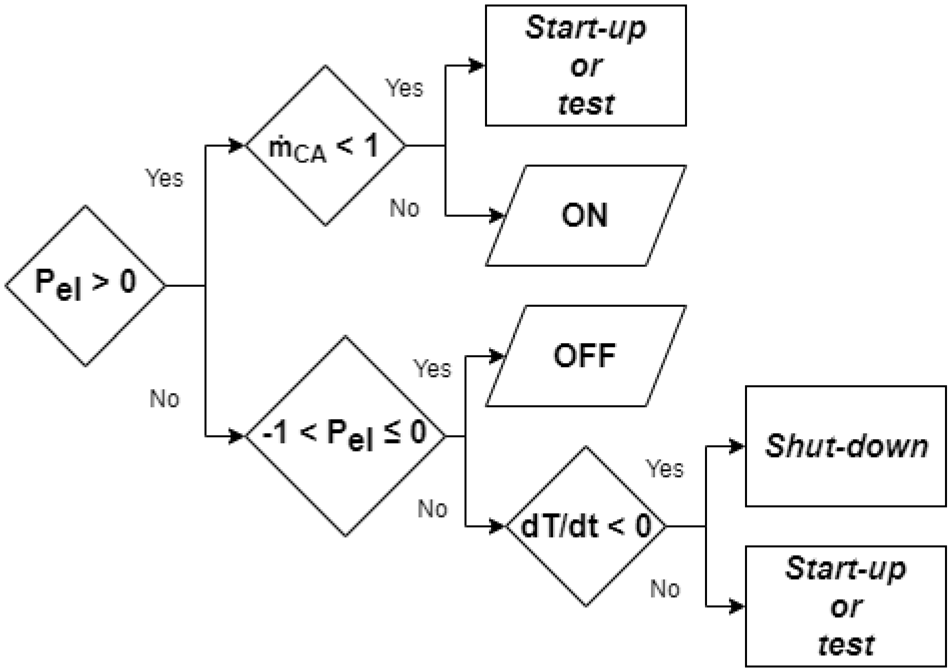


Figure 12: Flowchart for "static" assumption logic

Here, failure or emergency modes such as ISL, HOT and INT have not been considered, as they cannot be properly identified without previous reference. Checks on electrical power output, compressed air flow and module temperature variation with time (dT/dt) are used as control variables.

In case of positive power production, the module can either be starting-up or being already in nominal operation. The "*Start-up or test*" sub-routine investigates the consecutive 100 hours (~4 days) following time *t*, searching for a time step with positive electrical power production, no compressed air flow and a module temperature of at least 700 [°C]. These conditions identify a module that has been successfully activated and is running on nominal conditions, although power output may have not yet reached its peak. If said time step is found, the output of the routine for time steps inbetween is START. If these conditions have not been met by the end of the timespan, then the module has undergone a remote test, and the routine keep searching into consecutive time steps for the moment when the module returns inactive, with a temperature lower than 50 [°C] and close to none electric power and ambient air consumptions. The output in this scenario is OFF for the entire period.

The intensity of power absorption on the other hand, can identify weather the module is inactive or is performing a start-up (or test) or shutdown procedure. The evolution of temperature with time determines if the module is respectively heating up or cooling down.

The "*Shut-down*" sub-routine investigates the consecutive 360 hours (~2 weeks) following time *t*, searching for a time step when the module returns inactive, with a temperature lower than 50 [°C] and close to none electric power and ambient air consumptions. The output inbetween is SHUT.

The temperature variation with time (dT/dt) is evaluated through a dedicated sub-routine as the slope of the simple linear regression of the module temperature in the consecutive 24 hours following time *t*, ignoring any null values that may occur due to logging failures.

Successive time steps follow a "dynamic" assumption logic, as their possible operation modes are defined by the output of the previous one. Again, checks on electrical power output, compressed air flow and temperature variation with time are used as control variables. Figure 13 summarizes this logic. In case the procedure has identified a start-up or shutdown procedure, the obvious prosecution is respectively ON or OFF. In case the system is inactive, a significant power absorption identifies the beginning of a start-up or test procedure. In case the module is operating in nominal conditions, a check on the relative variation of power output between the two consecutive time steps $\Delta P_{el,\%}$ (26) helps in identifying significant power drops, related to a shutdown procedure or a failure or emergency mode: failure or emergency modes can happen in succession before the system manages to restore itself properly.

$$\Delta P_{el,\%} = \frac{P_{el,t} - P_{el,t-1}}{P_{el,t}} \cdot 100 \tag{26}$$

The "*Island mode*" routine investigate the consecutive time steps following time *t* with positive electrical power production, searching for when the compressed air flow stops or when the dedicated digital signal (D9) changes from 0 to -1. The output inbetween is ISL.

The "*Interruption or hot standby*" routine investigate consecutive time steps following time *t*, searching for when electrical power production is restored: altogether, either there is no more compressed air flow, or the digital signal of the air compressor itself is -1. Then, if the digital signal of the main fuel valve changed throughout the event, the output inbetween is HOT. The analogic measure of the NH_{mix} flow (A22) has shown some erratic behaviour and therefore it is not reliable as a hot-standby event indicator. If the previous condition is not met, then the output inbetween is INT.

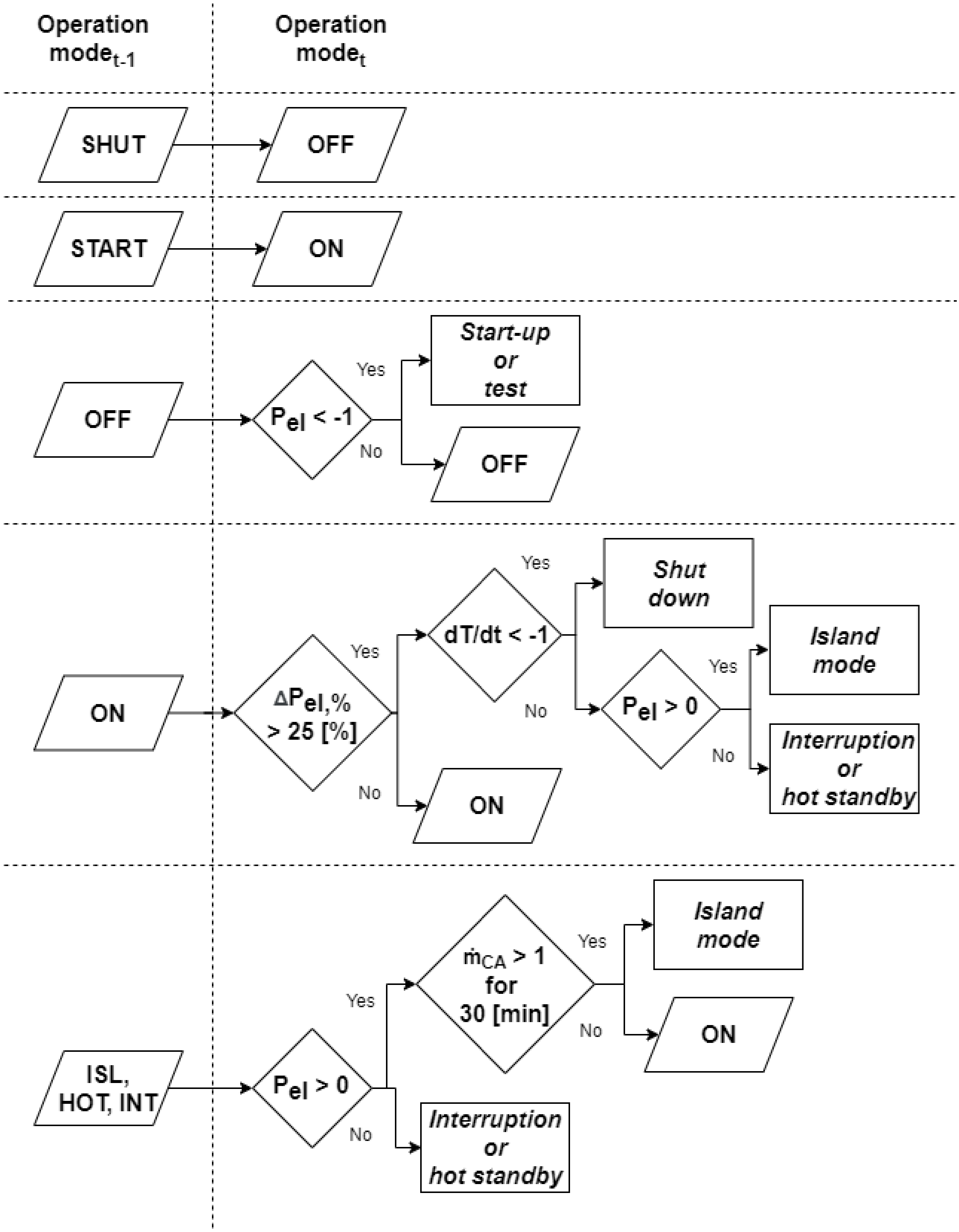


Figure 13: Flowchart for "dynamic" assumption logic

3.4. Data visualization, overview and report procedure

Another command of the toolkit allows easily choosing and displaying the KPIs and some of the most relevant data logged by the PLC. The user can choose up to 8 outputs from 60 measures, plus time for the abscissa (an example is provided in Figure 14). Ordinate data can be shown on the primary or secondary axis. Axis labels are automatically updated to show the correct unit of measure based on the selected output.

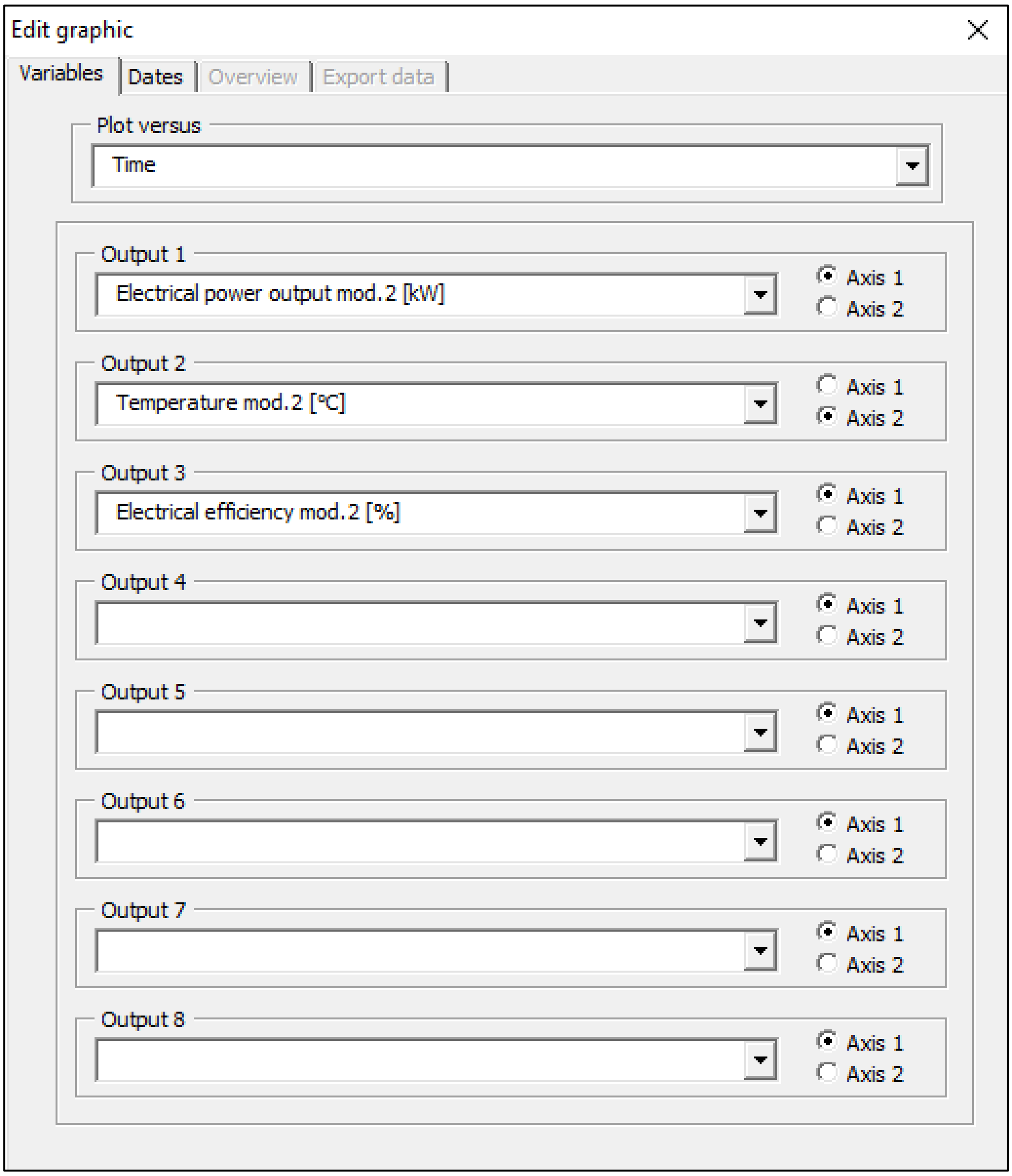


Figure 14: Data visualization user form - Example of outputs choice

The time period to visualize is custom, but it is restricted within the same year, due to Excel data series inability to simultaneously refer to more than one worksheet. When plotting a chart versus time, time periods longer than 1 hour without any data, because of logging failures or data collecting purposefully disabled, are explicitly signalled. An example is provided in Figure 15.

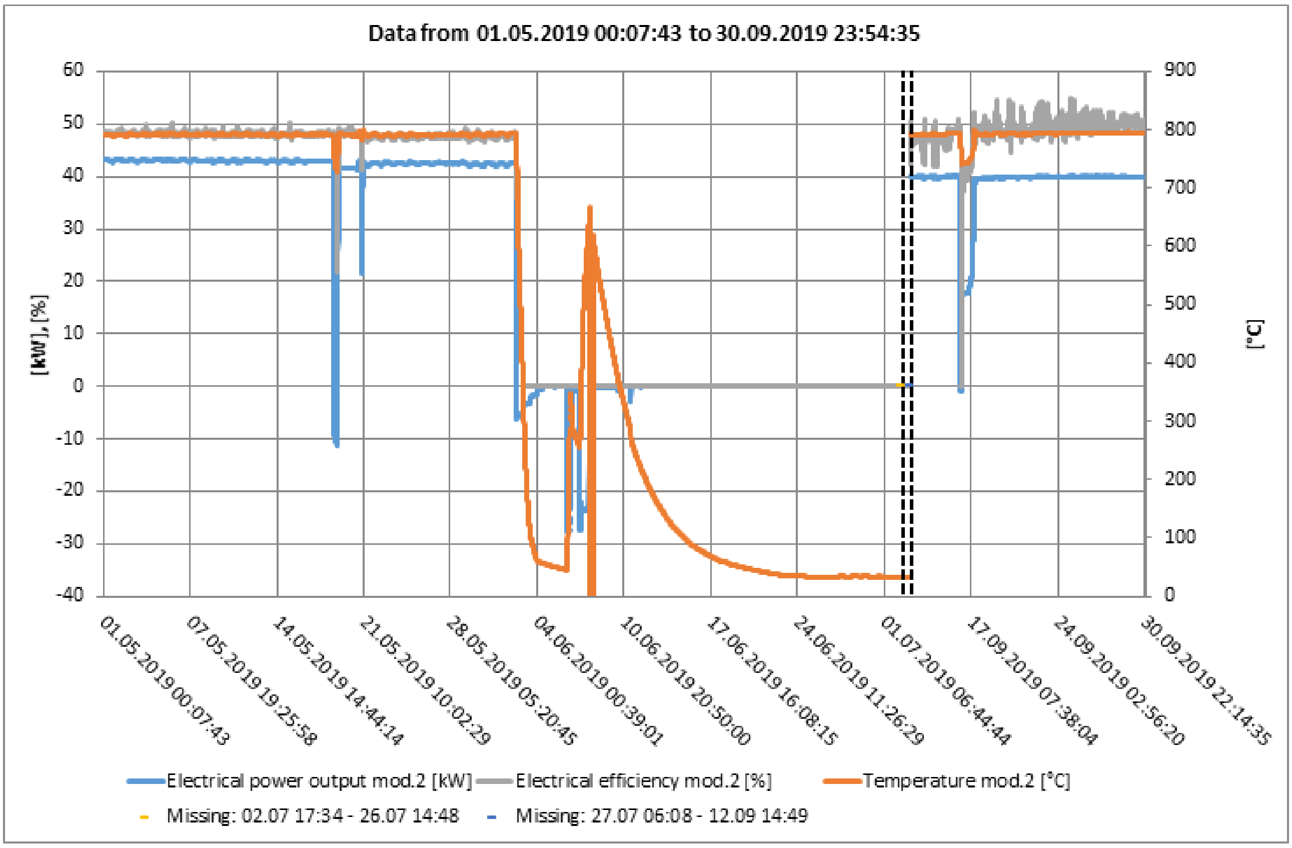


Figure 15: Data visualization chart - Example for the previous outputs choice

As an additional feature, some overview data regarding the selected time period is displayed in the user form. For each of the selected outputs, maximum, minimum and average values are available to properly read the chart. For each module, a recap of the time spent in each operation mode, in detail and overall allows to crosscheck the outputs evolution with the events that have occurred (Figure 16).

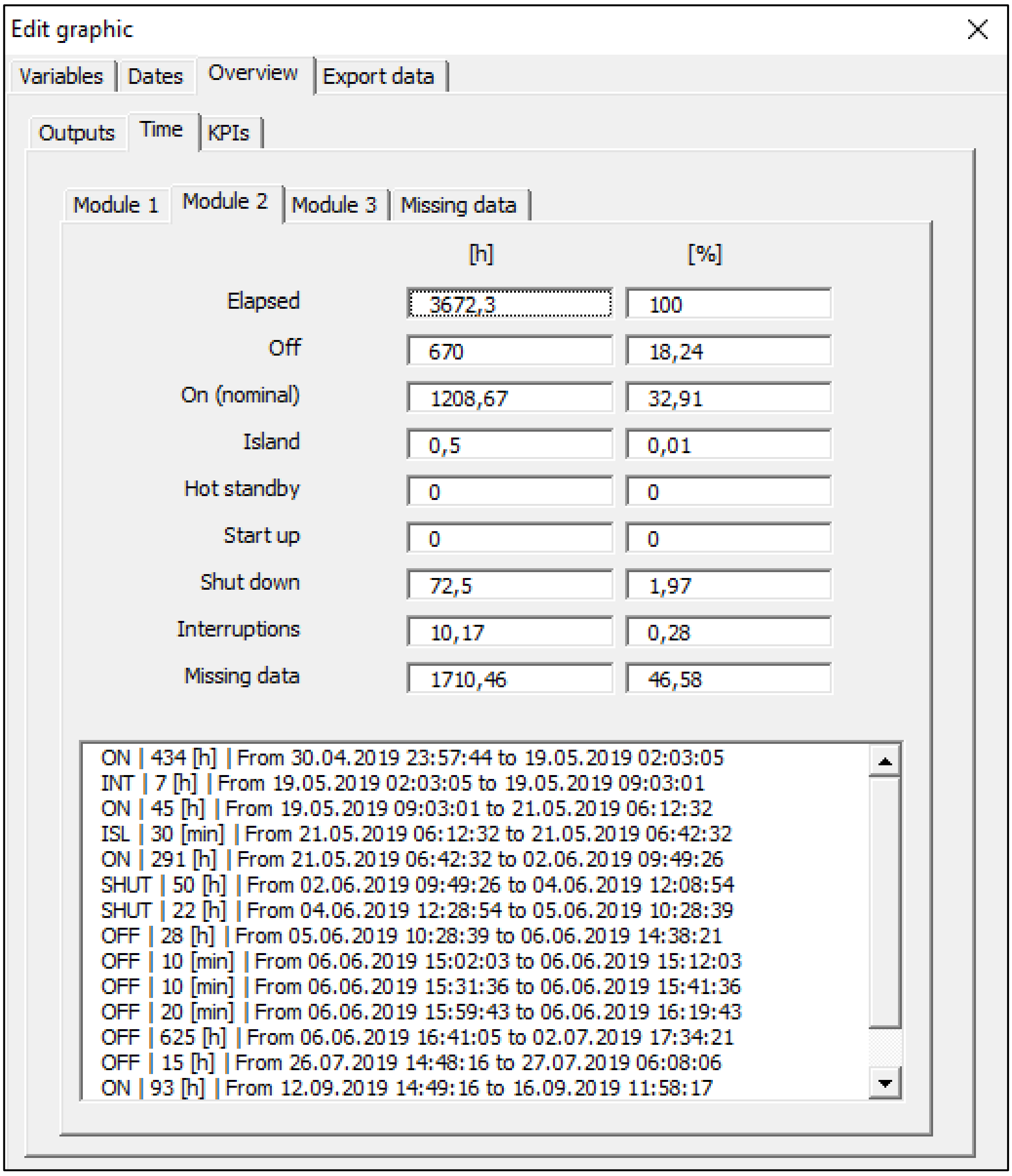


Figure 16: Module 2 operation time overview for the previous example

Time periods with missing data due to logging failures or data collecting purposefully disabled are listed in a separate sheet. Another page (Figure 17) is devoted to displaying time dependant KPIs: most of these are evaluated as a Riemann sum (27) over the stated time period, hence with each measure assumed constant throughout the time step *Δt* of 10 minutes (1/6 of an hour).

$$S(x) = \sum_{i=t_1}^{t_2} x_i \cdot \Delta t = \sum_{i=t_1}^{t_2} \frac{x_i}{6} \qquad (27)$$

Electrical energy production E_{el} , expressed in [kWh], is evaluated from the corresponding electrical power production P_{el} (therefore only if positive) for each module (A36, A37, A38) and as net output (K30):

$$E_{el} = S(P_{el}\ \ If\ P_{el} > 0) \quad (28)$$

Gross electrical energy production is computed as sum of the electrical energy production of the three modules.

Thermal energy production E_{th} , expressed in [kWh], is evaluated from the corresponding thermal power P_{th} removed from each module (K16, K17, K18), globally removed from the modules (K19) and actually transferred to the sludge (K20):

$$E_{th} = S(P_{th}) \quad (29)$$

The capacity factor CF , expressed in [%], of each module is defined as the ratio between the electrical energy produced during nominal operation $E_{el,ON}$ with respect to the energy that would have been produced in the same timespan t_{ON} at nominal power output $P_{el,N}$:

$$CF = \frac{E_{el,ON}}{P_{el,N} \cdot t_{ON}} \cdot 100 \quad (30)$$

Electrical energy produced during nominal operation is evaluated similarly to electrical energy production, with the additional requirement of the module operation mode being identified as ON for the corresponding time step. t_{ON} is the time, expressed in hours, the module has spent running in nominal operation within the stated time period. The nominal power output $P_{el,N}$ is assumed as the one associated with a set-point of 100 [%], therefore 58 [kW] for modules №1 (presumably) and №3, 45 [kW] for module №2, as the actual current set-point is not directly available from the imported data. The plant capacity factor $CF_{modules}$ is also evaluated by properly accounting each module contribution:

$$CF_{modules} = \frac{\sum_{1\ to\ 3} E_{el,ON,mod}}{\sum_{1\ to\ 3} \left(P_{el,N,mod} \cdot t_{ON,mod}\right)} \cdot 100 \quad (31)$$

Carbon dioxide emissions m_{CO2}, expressed in [kg], is evaluated considering both the part already present in the biogas stream sent to the modules $\dot{V}_{biogas,total}$ (K1), available in [m^3/h], and the part related to methane, assuming that the afterburner carries out the complete combustion of any of its traces:

$$m_{CO_2} = \frac{MW_{CO_2}}{22.414} \cdot S\left(\dot{V}_{biogas,total}\right) \quad (32)$$

Carbon dioxide molecular weight $MW_{CO_2} = 44.01 \left[\frac{g}{mol}\right]$

Contaminants adsorbed in the stated time period are evaluated in absolute terms as [mg] of equivalent sulphur $m_{eq,S}$ and equivalent D4 $m_{eq,D4}$ from the corresponding adsorption rates $\dot{m}_{eq,S}$ and $\dot{m}_{eq,D4}$ (K34, K35), available in [mg/h]:

$$m_{eq.S} = S\left(\dot{m}_{eq.S}\right) \quad (33)$$

$$m_{eq.D4} = S\left(\dot{m}_{eq.D4}\right) \quad (34)$$

Contaminants adsorbed in the stated time period are also evaluated in relative terms as loading rates $LR_{eq,S}$ and $LR_{eq,D4}$, expressed in [%], with respect to the predicted adsorption capacity of a single reactor:

$$LR_{eq.S} = \frac{m_{eq.S}}{(q \cdot m)_{CKC}} \cdot 100 \tag{35}$$

$$LR_{eq.D4} = \frac{m_{eq.D4}}{(q \cdot m)_{C64}} \cdot 100 \tag{36}$$

Specific CKC equivalent sulphur adsorption capacity	$q_{CKC} = 84 \left[\frac{mg}{g}\right]$
Specific C64 equivalent D4 adsorption capacity	$q_{C64} = 150 \left[\frac{mg}{g}\right]$
Activated carbons weight of a single reactor	$m_{CKC} = m_{C64} = 250\ [kg]$

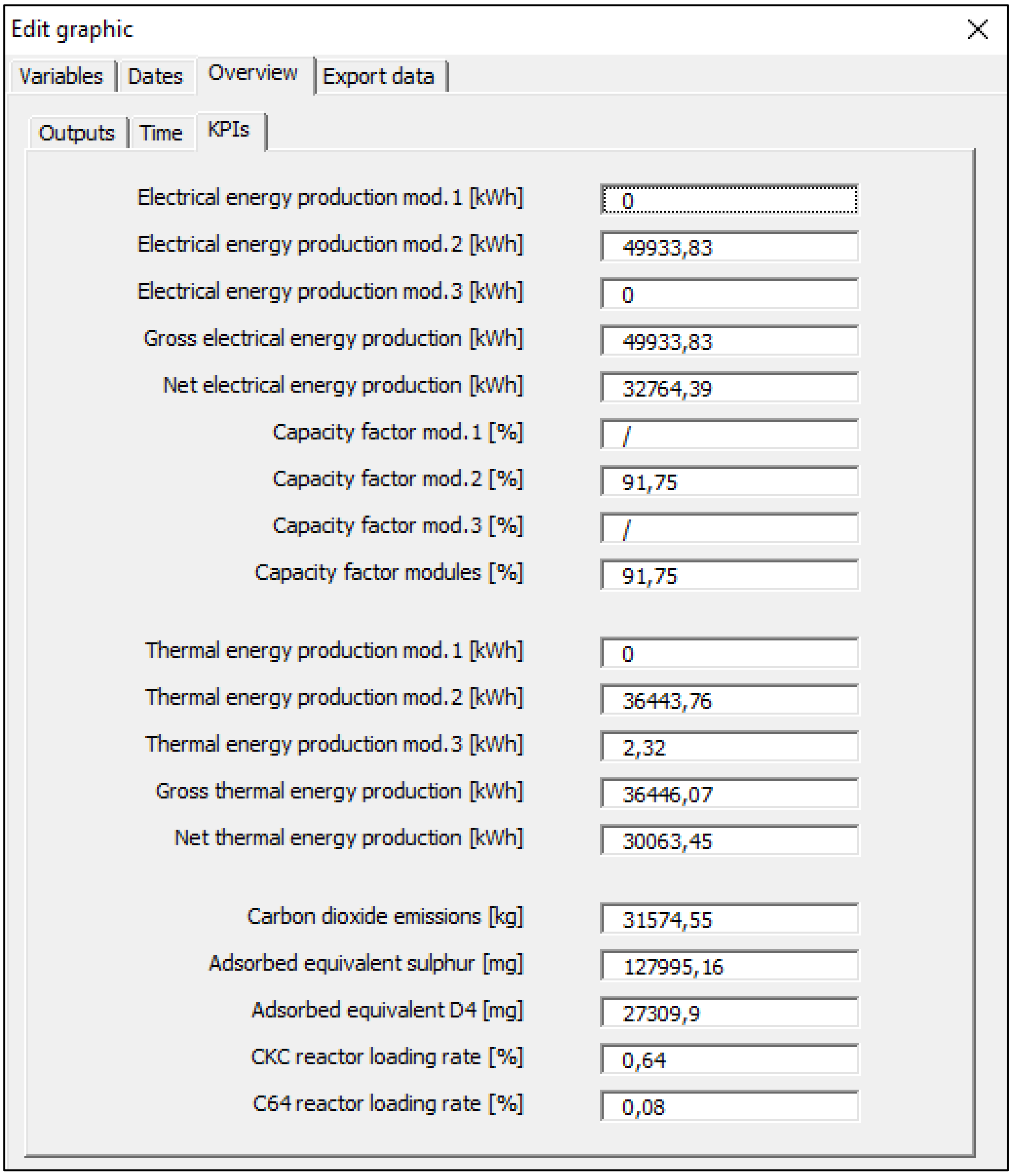


Figure 17: Time dependant KPIs for the previous example

Although most of the graphical adjustments are already implemented in the routine, the chart can also be manually modified by the operator. A couple of export commands allow to save the graph as a PNG picture or create a Word report based on a template file included in the folder, which will present all the previously evaluated information.

The operation log command performs a similar function, showing only overview data for a selected year or for the entire lifetime of the plant. These information are more quickly accessible as they are for the most part evaluated in a worksheet through cell formulas. For each module, average efficiencies are evaluated over the available ON periods and information on selected ON periods are provided in the user form (Figure 18). Again, an export procedure is available to save the desired information in a Word report based on a similar template file included in the folder.

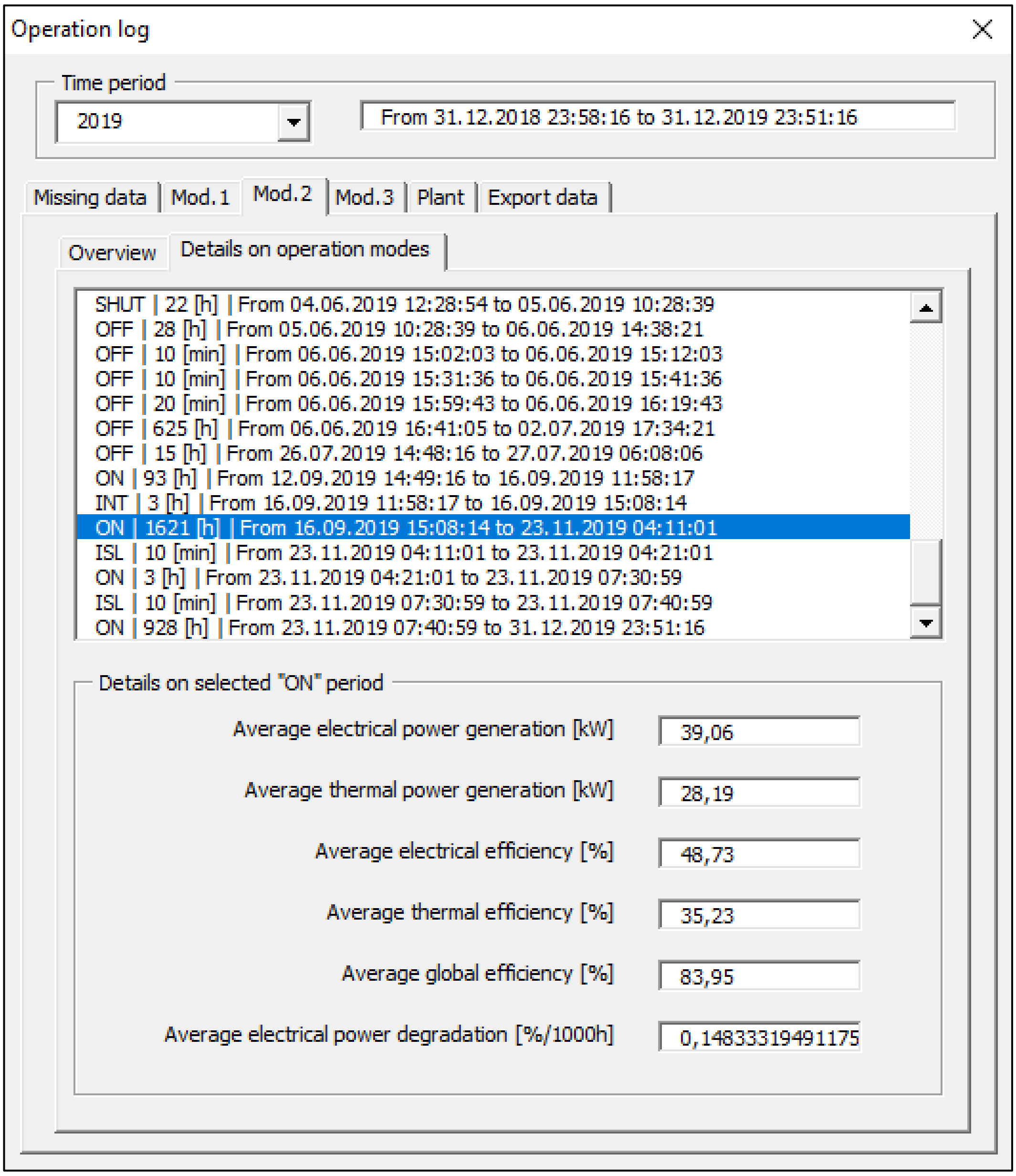


Figure 18: Operation log user form, example for the year 2019

4. Results and discussion

Data presented in this section is mainly dedicated to the year 2019, since previous years have already been discussed by other authors. During this year, only module №2 has been operative, therefore many of the KPIs related to overall plant performance are redundant. In the following, results obtained through the use of the toolkit are presented and discussed, while some further observations and analysis are carried out with additional information not available by the PLC.

4.1. Operation mode analysis

Module №2 (from here on out simply called "module" since it was the only one running during the year 2019) has been shut down on the 19th of December 2018, and it has been restarted on the 17th of February 2019. The imposed current set point is 25 [A], or 100 [%] on a scale where 12 [A] corresponds to 0 [%]: power production settles around the nominal power of 45 [kW]. The module runs until 19th March with only a couple of short interruptions (Figure 19). It can be observed that compressed air injection accompanies part of the start-up procedure and also interruptions. The latter are sometimes paired with a complete disconnection and reconnection with the electrical grid, during which the voltage reading of the modules electrical panel drops to zero.

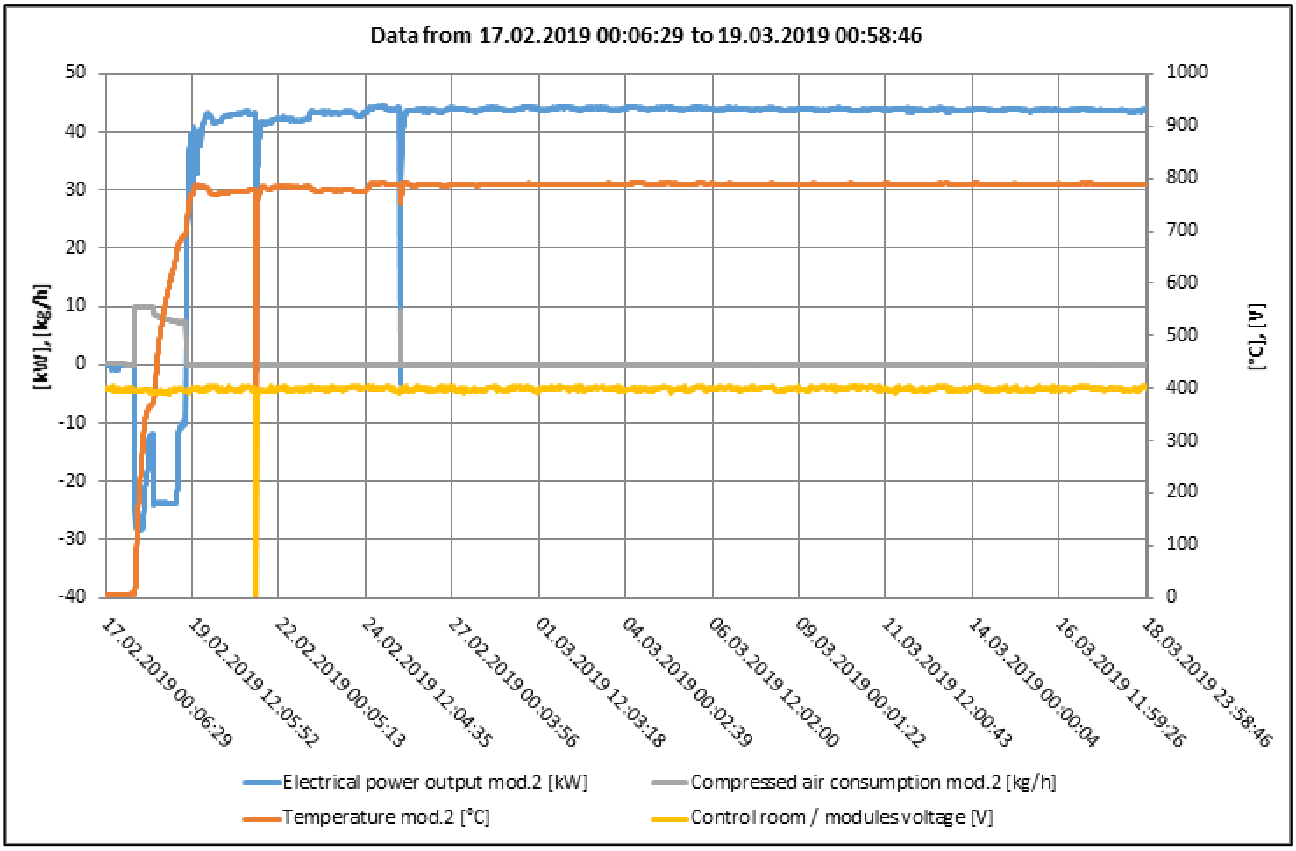


Figure 19: 2019 - First start-up and nominal operation period

Then until the 24th of March the module appears to run at a low power level, with 11 interruptions throughout. During this period, the algorithm could not surmise a unanimous operation mode, as digital signals do not provide any insight on the cause of this behaviour. Whether the mode is considered as ON or ISL, the electrical power output averages around 26 [kW]: after the last interruption, this trend continues until the 27th of March, when the power of the system restores itself without further information (Figure 20).

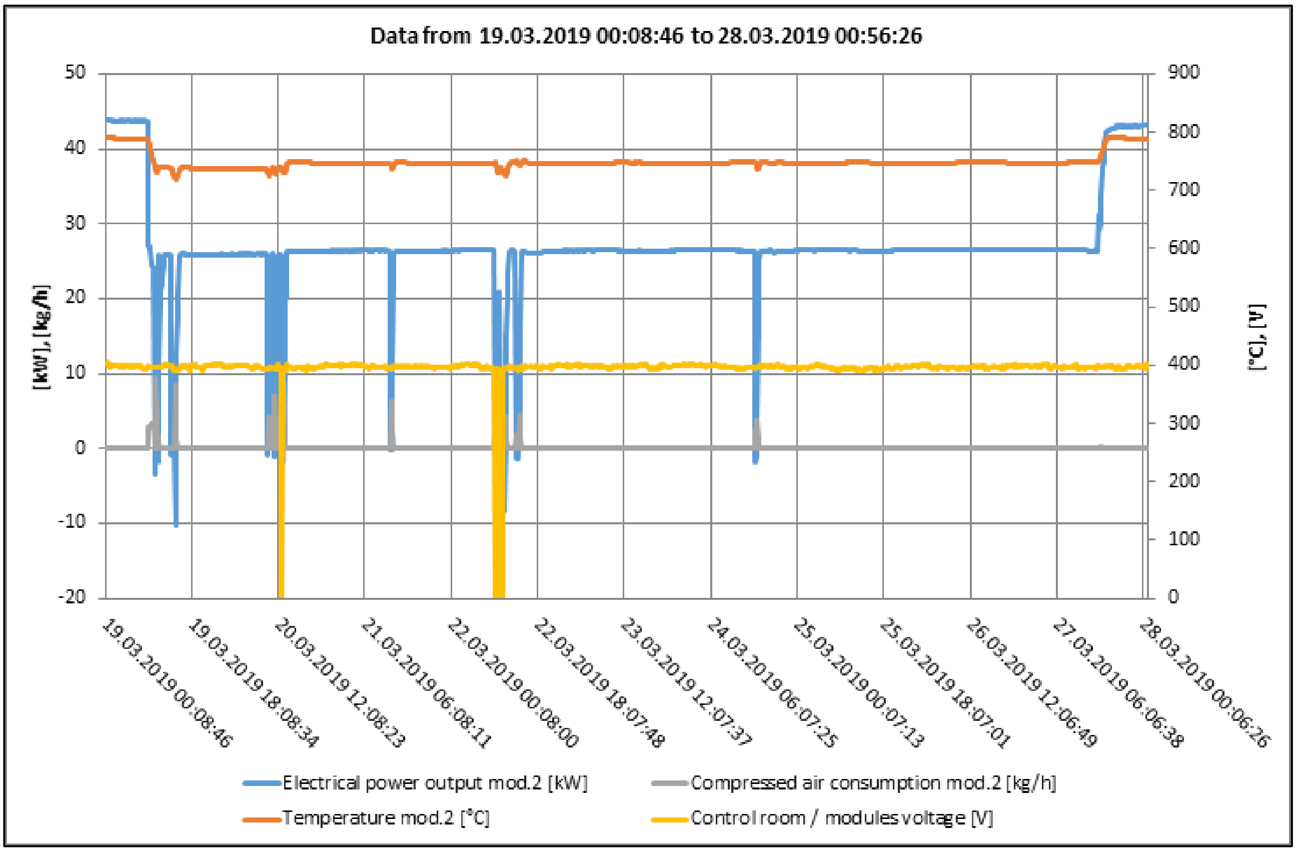


Figure 20: 2019 - March interruptions and low power output period

Operation continues until the shutdown on the 2nd of June. Aside from a couple of short-term island events, the system experiences a similar low power output behaviour on the 5th of April, eventually restored after a complete disconnection and reconnection with the electrical grid, and a 7 hour-long interruption on the 19th of May, followed by a slow recovery of power production due to the stack cooling in the meantime. A few days after the shutdown has completed, the module is powered again, probably as a test, heating up to 668 [°C] before being left to freely cool down (Figure 21).

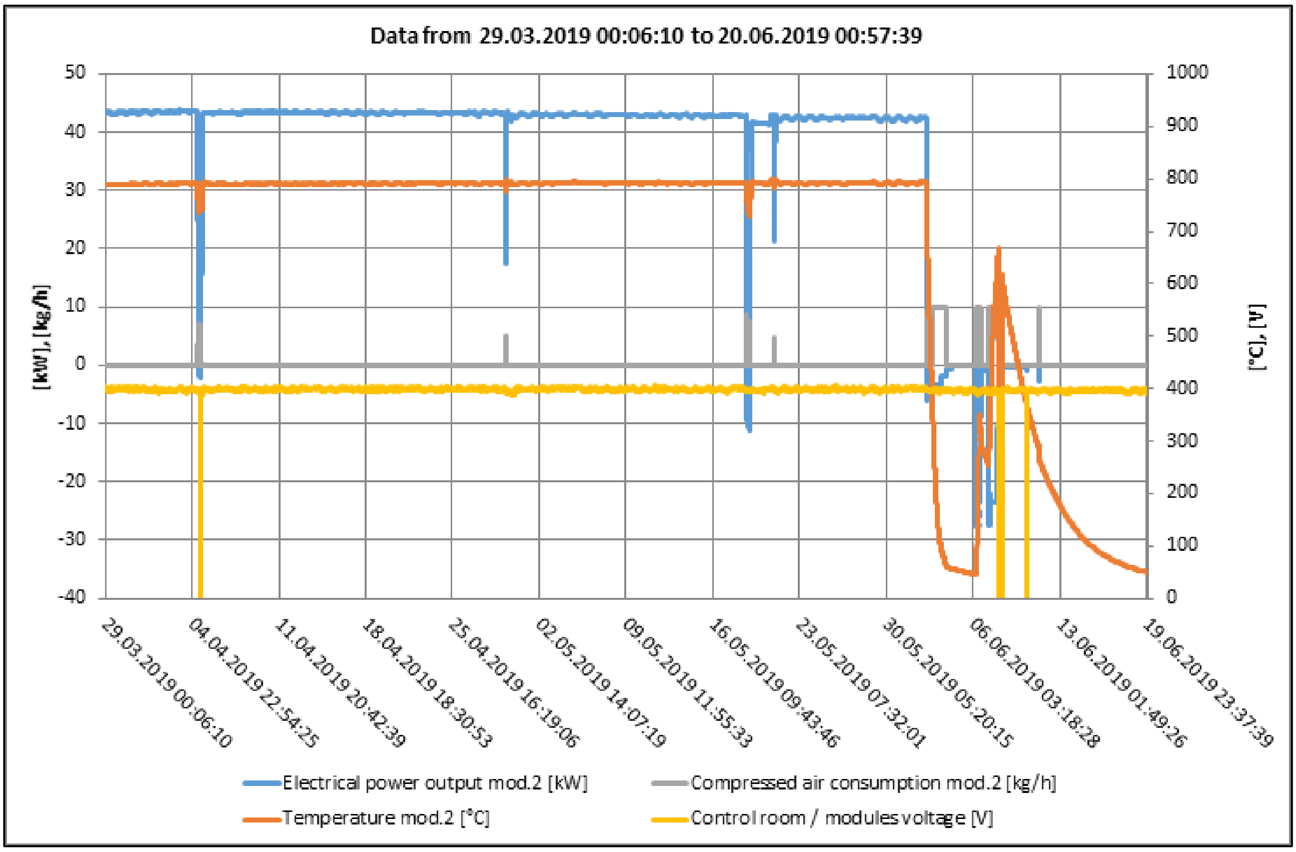


Figure 21: 2019 - Spring operation period, shutdown and test

During summer, the module has been kept inactive: operation has been resumed at the beginning of September, with a current set point of 23.6 [A], corresponding to 89.2 [%]. Data logging resumed on the 12th of September, while the module was already running, therefore there are no information regarding the start-up procedure which actually occurred at the end of August: the relative data was lost due to human error at the plant site. Aside from a 3 hour-long interruption the module continued operation until the end of the year with only a couple of power drops throughout (Figure 22).

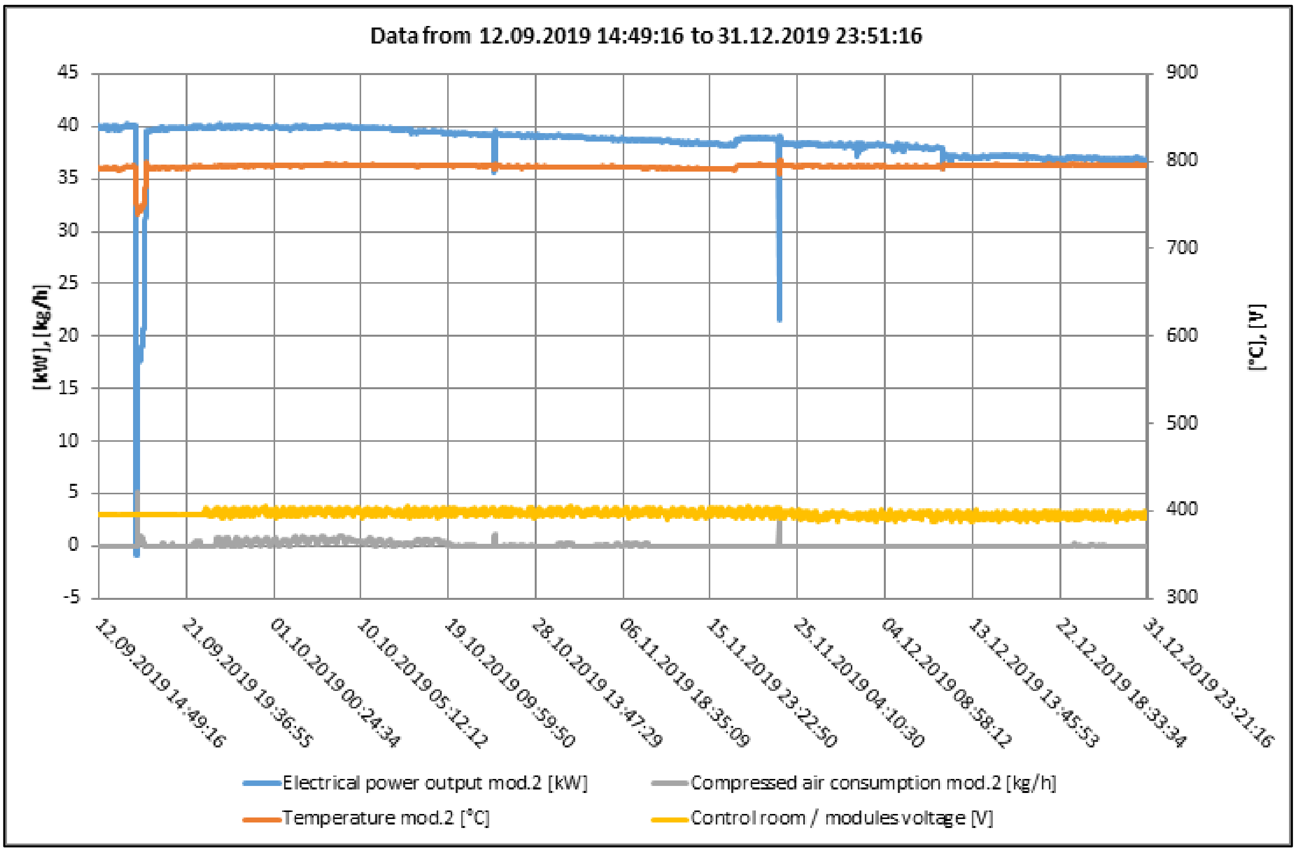


Figure 22: 2019 - Operation after the second start-up until the end of the year

Operation has continued at least until the 11th of February 2020, which is the last time data has been manually extracted from the PLC. In this period, the system has experienced some small power drops and two major interruptions.

Tables 2 and 3 summarize the time spent in each operation mode and the time periods without logged analogic data throughout the year 2019, as extracted from the corresponding Word report.

Missing time periods:
MISS | 10 [min] | From 04.06.2019 12:08:54 to 04.06.2019 12:28:54
MISS | 14 [min] | From 06.06.2019 14:38:21 to 06.06.2019 15:02:03
MISS | 10 [min] | From 06.06.2019 15:12:03 to 06.06.2019 15:31:36
MISS | 8 [min] | From 06.06.2019 15:41:36 to 06.06.2019 15:59:43
MISS | 11 [min] | From 06.06.2019 16:19:43 to 06.06.2019 16:41:05
MISS | 573 [h] | From 02.07.2019 17:34:21 to 26.07.2019 14:48:16
MISS | 1137 [h] | From 27.07.2019 06:08:06 to 12.09.2019 14:49:16

Table 2: 2019 - Time periods with missing data (logging failure or intentionally disabled)

Operation mode	Time [h]	Time [%]
OFF	1818.33	20.76
ON	5097.33	58.19
ISLAND	4.83	0.06
HOT STANDBY	0	0
START UP	35.83	0.41
SHUT DOWN	72.5	0.83
INTERRUPTION	22	0.25
MISSING DATA	1709.05	19.51

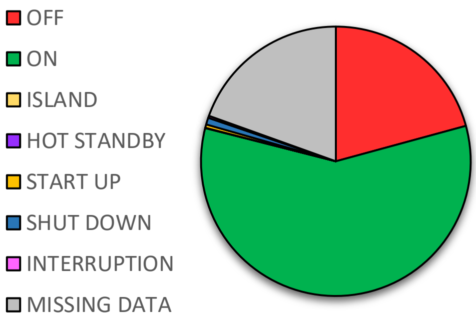


Table 3: 2019 - Operation mode summary

The PLC failed to log data just 5 times, each skipping a single time step. The major missing data periods are those due to the logging routine being intentionally disabled during July and August, as the modules were scheduled to remain inactive, and the first days of September lost due to human error.

Only ~27 hours of nominal operation have been lost because of failure or emergency modes of operation. The module has worked as intended for more than 5097 hours (about 212 days).

4.2. Electrical performance

The module analysis can be split into two time periods, before and after the summer of 2019, to take into consideration the different set-points employed. Average values reported in Table 4 have been evaluated during nominal operation, with the exclusion of the aforementioned lower power output phases. There is no benchmark for the electrical efficiency as the available data sheet (Figure 5) refers to CONVION C50 module with 58 [kW] nominal power.

		Electrical power production [kW]	Electrical efficiency [%]
Period I From 01/01/2019 to 01/07/2019 Set-point: 100 [%]	Maximum	44.68	51.79
	Average	43.18	48.55
Period II From 01/07/2019 to 31/12/2019 Set-point: 89.2 [%]	Maximum	40.25	55.36
	Average	38.71	47.59

Table 4: 2019 - Electrical power and efficiency performance

In both periods, while maximum power production is compatible with the imposed set-point, the average is penalized by the effects of degradation that are easily noticeable in Figure 23. Electrical efficiency shares the same fate, with slightly underwhelming performances with respect to literature reference of 55÷65 [%] [x].

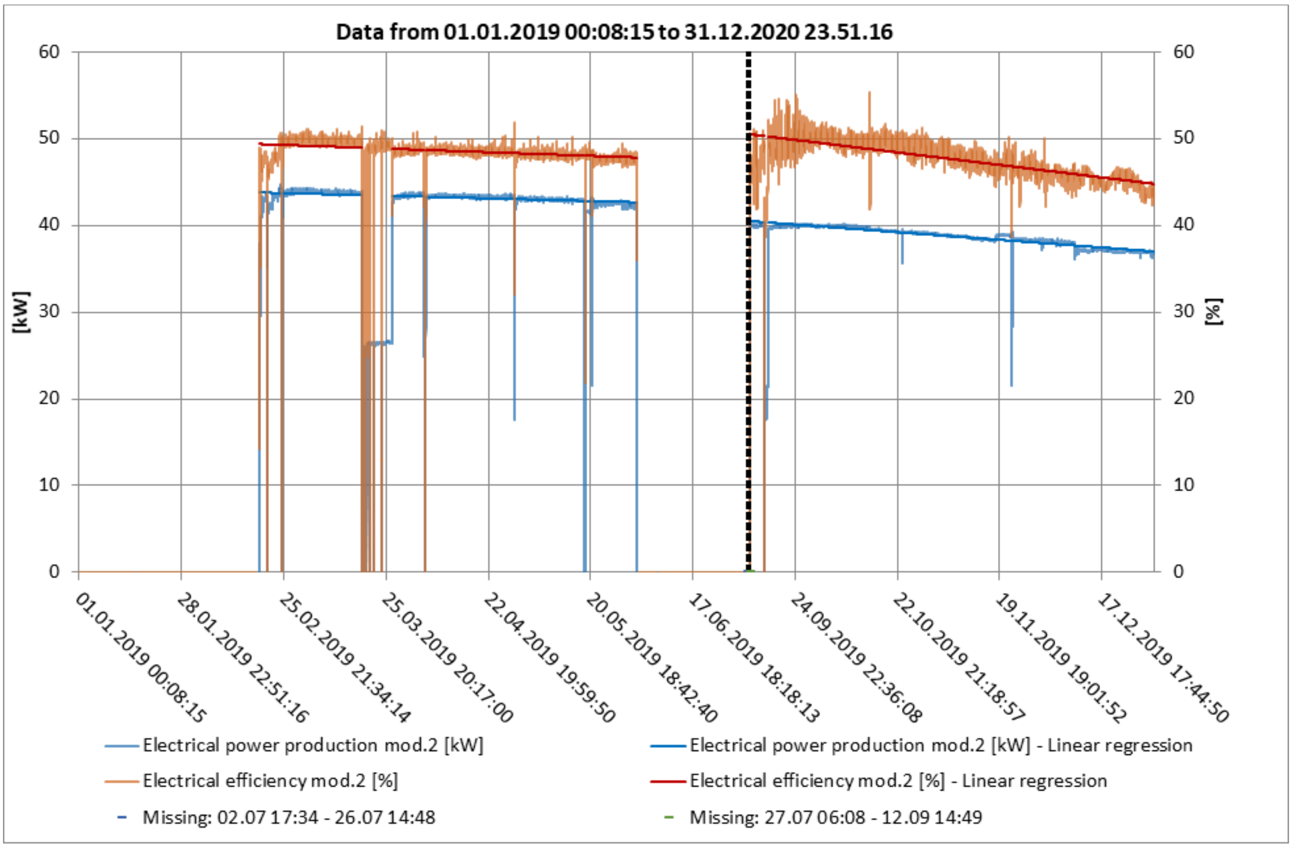


Figure 23: 2019 - Electrical performance and degradation

Fitting with a linear regression each data series on the two time periods with the same assumptions made before, it is possible to estimate said degradation as a percentage drop over 1000 hours of nominal operation [%/1000 h]:

$$D_{P_{el}} = \frac{\Delta P_{el_{LR,t_{ON}}}}{P_{el,N} \cdot \frac{I_{set-point}}{25}} \cdot 100 \cdot \frac{1000}{t_{ON}} \tag{37}$$

$$D_{\varepsilon_{el}} = \Delta\varepsilon_{el_{LR,t_{ON}}} \cdot \frac{1000}{t_{ON}} \tag{38}$$

$\Delta P_{el,LR,tON}$ and $\Delta\varepsilon_{el,LR,tON}$ are respectively the drop in power production, available in [kW], and electrical efficiency, already in [%], over the nominal operation time considered t_{ON} , expressed in hours, estimated as differences of the two ends of the corresponding linear regressions. In equation (37), power production drop D_{Pel} is evaluated in [%] dividing with respect to the expected power output for the given current set-point $I_{set\text{-}point}$, expressed in [A], where $P_{el,N}$ is the nominal power output of 45 [kW] of the module for the 25 [A] set-point.

In period I, power degradation is around 1.1 [%/1000 h], while electrical efficiency degradation is around 0.69 [%/1000 h]. As the slope in Figure 23 suggests, degradation in period II is more prominent, with a 3.13 [%/1000 h] drop for power production and 2.17 [%/1000 h] for electrical efficiency.

Fuel cell degradation in SOFC stacks is a known phenomenon, especially when fuels different from pure hydrogen is employed. Carbon associated with the different gaseous species in the biogas feed may cause solid depositions on the anodic interface, irreversibly reducing the available cell voltage. Likewise, the presence of hydrogen sulphide and siloxanes cause the same issue, but their presence in the modules should be negligible given the extensive clean-up system installed.

While recirculation from the anodic exhaust to the fuel feed introduces oxygen carriers counteracting carbon separation, it also lowers the partial pressure of hydrocarbons, reducing cell voltage. In addition, oxygen-leakage has a substantial effect on the actual fuel utilization, fuel sensitivity and voltage-current characteristics [x]. Also, operating temperature can cause grains coarsening on the electrodes, affecting the available active surface size.

These phenomena may also be unevenly distributed among the cells composing the stack: the internal control system can reduce output current to maintain voltage over a minimum limit while trying to preserve degraded cells from working in possibly harmful conditions. From the outside, this kind of adjustment appears as a power drop. Therefore, the degradation on electrical power production and efficiency does not have a unanimous justification, as it may be simply due to internal regulations, which are inaccessible to the plant operators. The irreversible component of the degradation should be evaluated on the long run, comparing performances with the same set-point right after proper start-up procedures. The causes may eventually be identified when the stack will be taken apart for maintenance or replacement.

From Figure 24, changes in fuel utilization can be confirmed in period II, as the flow rate and power input of the biogas feed increase with time despite the reduction in electrical power output: consequently, this explains the reduction in efficiency. Also from Figure 24, it can be surmised that the odd behaviour of electrical efficiency right after logging resumed in September is due to the oscillating biogas flowmeter readings throughout the month, possibly related to the activation of the secondary digester for maintenance activity on the main one.

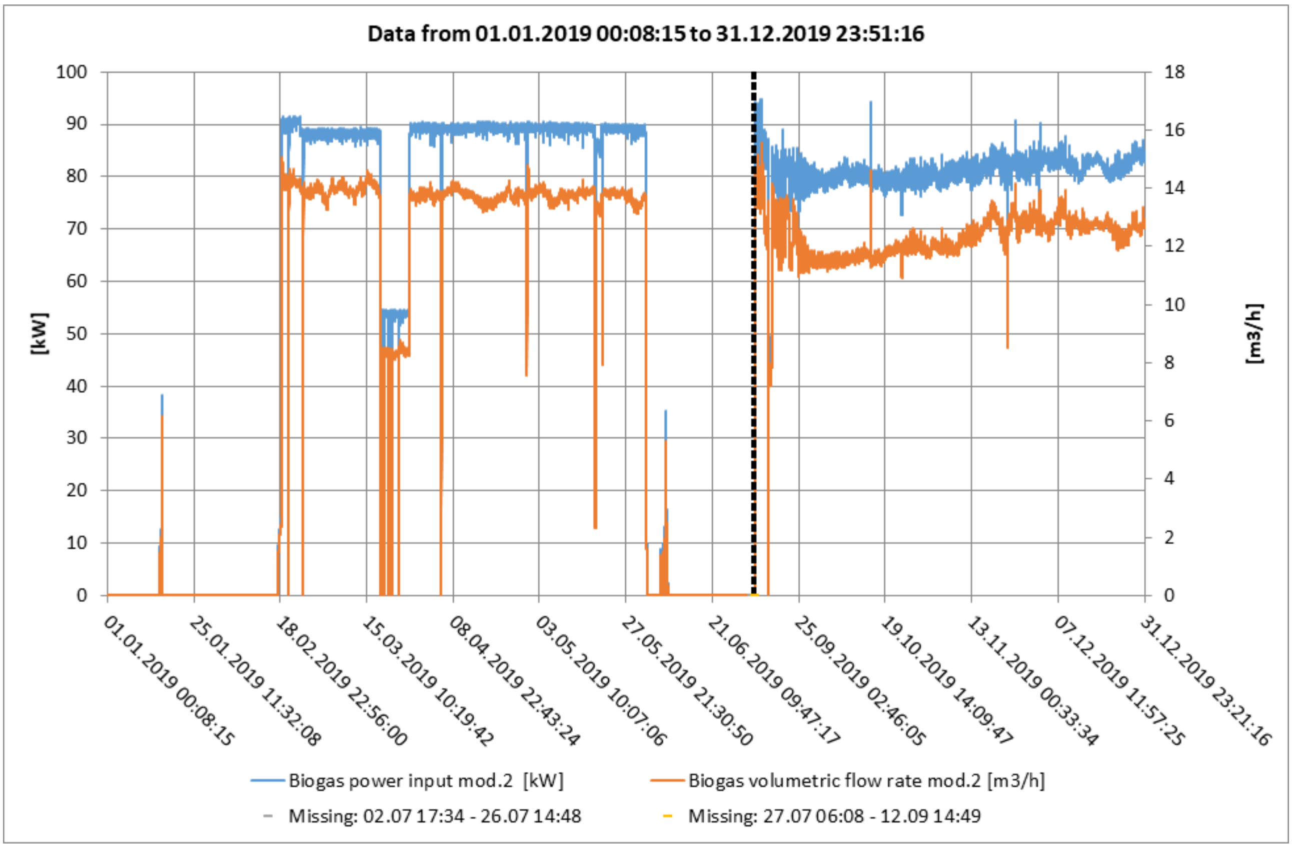


Figure 24: 2019 - Biogas feed for the module

Auxiliary consumptions show a distinct behaviour in the two periods as well. The air compressor is a major piece of equipment, as it mainly supplies the WWTP for normal operation. It can either be manually charged on the SOFC system, therefore being accounted for in the readings of the auxiliaries power meter, or directly on the grid. Charging on the SOFC system is a safety measure required to avoid the indirect halt of the modules induced by the control logic, as an alarm triggers when low pressure is detected in the compressed air circuit. While the average electrical consumptions of the auxiliaries listed in Appendix 8.2 settles around 10.17 [kW], the contribution of the air compressor can raise the power required up to 25.79 [kW]. Figure 25 suggests that the air compressor has been charged on the module only during period I, as auxiliaries consumptions are significantly lower in period II, even considering the lower biogas flow to be treated by the clean-up system due to the change in set-point.

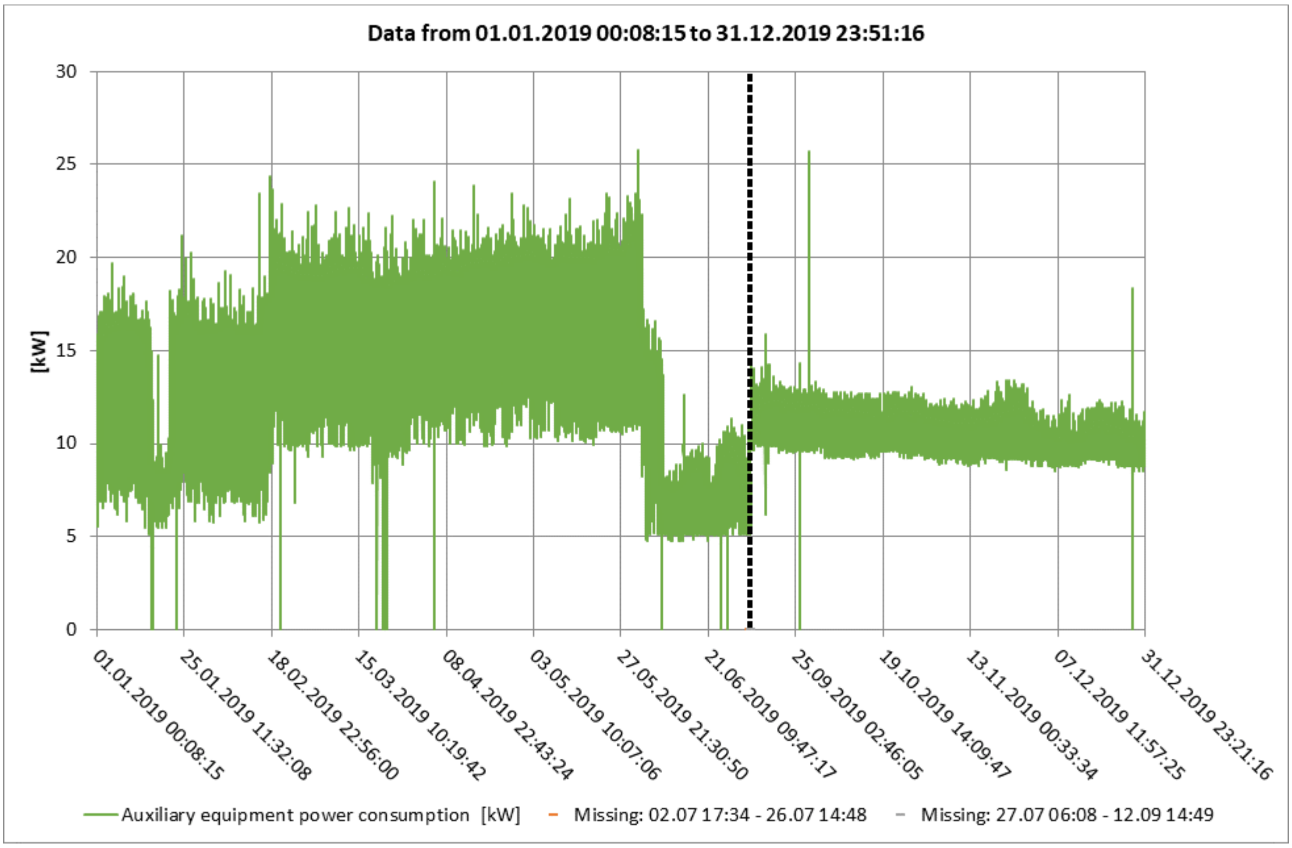


Figure 25: 2019 - Auxiliary equipment consumptions

4.3. Thermal performance

Regarding the thermal recovery from the module exhaust stream, during nominal operation the heat removed in period I has been on average 29.23 [kW], with an average efficiency with respect to the biogas power input of 33.88 [%]. In period II, the figures have been respectively 28.09 [kW] and 34.68 [%]. From Figure 26 it is possible to observe the different trends, comparing them to the yearly evolution of ambient air temperature.

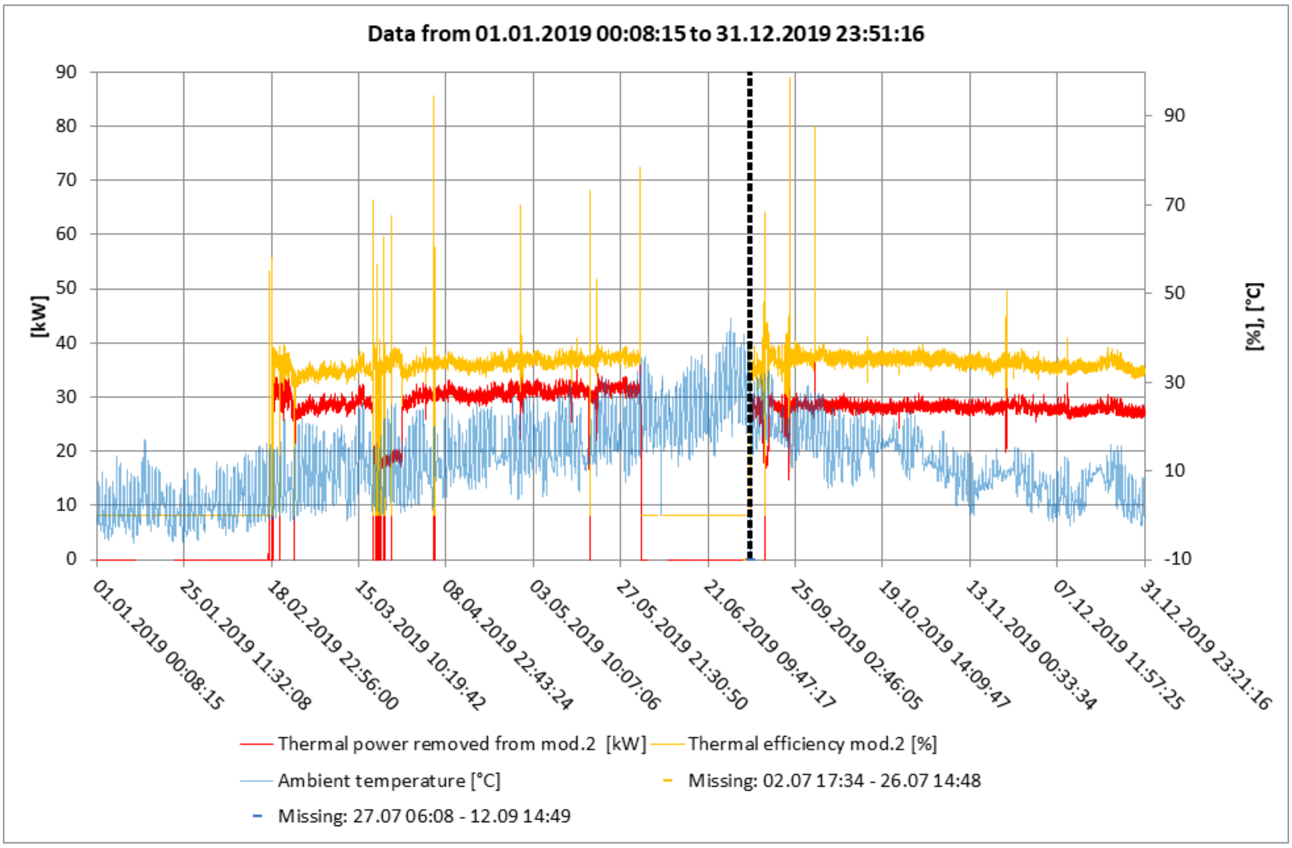


Figure 26: 2019 - Module thermal power recovered, thermal efficiency and ambient air temperature

In period I, as ambient air temperature increases, less power is required to heat up the module to the fuel cell requirement. Also, thermal losses from the module itself towards the environment are reduced: these conditions translate into an increase of thermal performances with time. Viceversa, in period II the drop in ambient air temperature increases the power required from the biogas to heat up the cathodic air stream: this may contribute to the change in fuel utilization mentioned before.

Thermal efficiency of the entire heat exchange circuit while on nominal operation settles around 75.15 [%]. Figure 27 compares said efficiency with ambient air temperature throughout the year. Although a slight dependency on external temperature is present (around 0.156 [kW/°C]) due to losses in the lengthy insulated piping installed outdoor, the main thermal loss has to be attributed to the hydraulic separator. This component partially acts as a mixer heat exchanger in case the supply and return flow rates are different, which is very likely given the poor precision of the flow meters and pumps regulation.

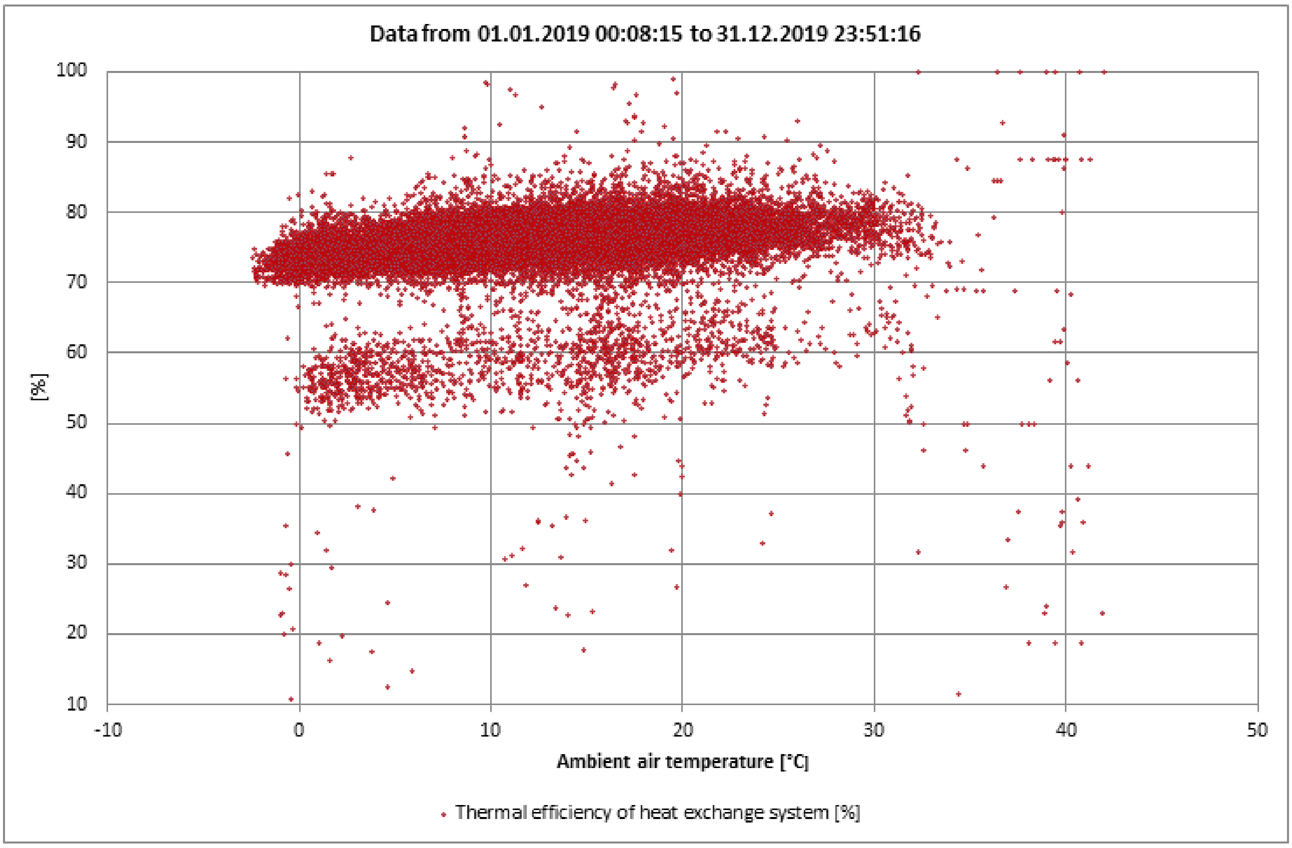


Figure 27: 2019 - Thermal efficiency of heat exchange system with respect to ambient air temperature

Figure 28 shows the thermal powers computed on each side of the sludge heat exchanger. Flow rate measurements oscillations on the sludge side render the thermal power evaluation quite unreliable: this is why net thermal power produced has been evaluated on the glycol water side. Sludge employment in the heat recovery unit has actually been dismissed since residual strands tended to clog the piping due to their small diameter. Industrial water is withdrawn from a well with a highly fluctuating pressure, causing unstable readings.

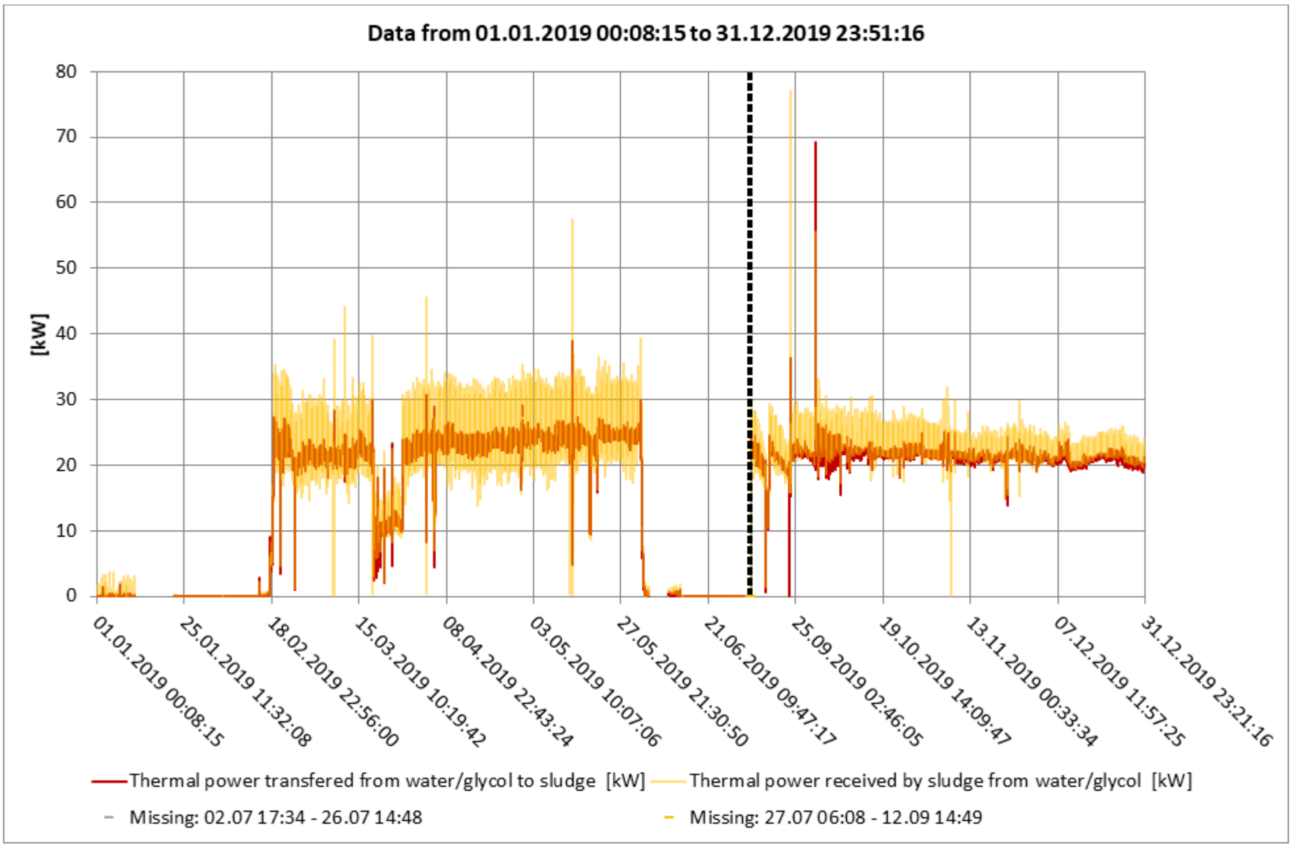


Figure 28: 2019 - Thermal power exchanged on the two sides of the sludge heat exchanger HRU4

In addition, thermal power on the sludge side is on average 1.59 [kW] higher than the one evaluated on the glycol water side. This may be due to a combination of measurements errors and a reduction in the heat capacity of the glycol water mixture (with respect to the one provided by the supplier).

Fluid composition, and therefore its physical properties, can change unpredictably over time. Prolonged exposure to high temperatures (above 104 [°C]) causes glycol degradation and the formation of organic acids [x]. Also the makeup water to adjust for pressure losses in the system causes a dilution in glycol concentration. Therefore, glycol has to be periodically reintegrated in the circuit to avoid reduction or potential removal of the freeze protection.

In Figure 29 the difference between the two computed thermal powers have been evaluated for all of the available data, and compared to the difference in case the glycol water mixture is equated to water (therefore with the two fluids having the same heat capacity). A linear regression for each data series is also shown to better visualize their behaviour.

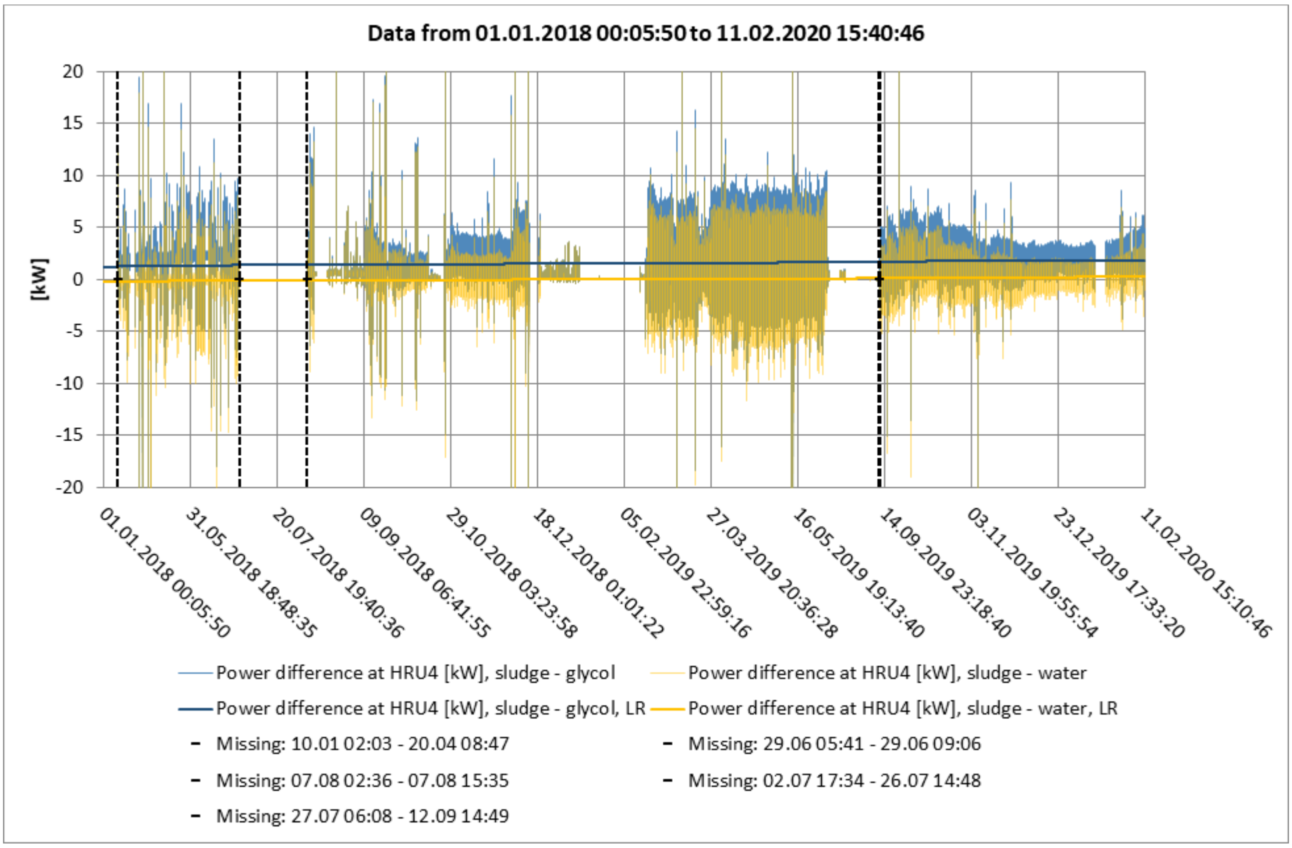


Figure 29: Sludge heat exchanger thermal power discrepancies

Both trends slightly increase with time, corroborating the assumption that a combination of glycol degradation and dilution has been reducing the heat capacity of the fluid. While it appears that there is on average no difference in thermal power at HRU4 considering pure water properties on both sides, it is not safe to assume such a hypothesis in the KPIs evaluation, as the flow rate readings themselves are extremely fluctuating: thermal power on the glycol water side should be considered as an underestimation of the actual heat flow exchanged in each HRU. Glycol concentration can hardly be determined by the available measurements, and direct inspection of the fluid composition should be carried out periodically to avoid freezing issues.

4.4. Plant performance

Table 5 presents overall KPIs for the year 2019, compared to those of 2018 for a first comparison.

Key Performance Indicators		2018	2019
MOD.2	Electrical energy production [MWh]	55.65	204.66
	Capacity factor [%]	96.07	89.15
	Thermal energy production [MWh]	36.05	145.62
	Average (while ON) electrical efficiency [%]	48.95	48.02
	Average (while ON) thermal efficiency [%]	34.73	34.29
	Average (while ON) global efficiency [%]	83.82	82.31
MOD.3	Electrical energy production [MWh]	111.24	/
	Capacity factor [%]	78.12	/
	Thermal energy production [MWh]	69.85	/
	Average (while ON) electrical efficiency [%]	47.98	/
	Average (while ON) thermal efficiency [%]	30.8	/
	Average (while ON) global efficiency [%]	78.76	/
PLANT	Biogas energy input [MWh]	289.53	427.37
	Gross electrical energy production [MWh]	166.89	204.66
	Net electrical energy production [MWh]	106.81	138.52
	Capacity factor modules [%]	83.33	89.15
	Gross thermal energy production [MWh]	105.9	145.62
	Net thermal energy production [MWh]	89.22	110.52
	Biogas treated volume [m^3]	50472	65644
	Carbon dioxide emissions [t]	99.1	128.89
	Adsorbed equivalent sulphur [g]	1539.98	998.16
	Adsorbed equivalent D4 [g]	268.66	125.27
	CKC reactor loading rate [%]	7.61	4.99
	C64 reactor loading rate [%]	0.72	0.36

Table 5: 2018-2019 Key Performance Indicators comparison

In terms of efficiency, the module performed similarly to the previous year, with a lower capacity factor due to the time spent on an 89.2 [%] set-point in the second half of the year.

Regarding the plant as a whole, in 2019, 427.37 [MWh] of methane allowed to produce 204.66 [MWh] of electrical energy and recover 145.62 [MWh] of thermal energy. Respectively accounting for the consumptions of the auxiliaries and the thermal losses along the glycol water distribution circuit, the plant has reached a net electrical efficiency of 32.4 [%], a net thermal efficiency of 27.9 [%] and a net global efficiency of 59.3 [%]. This result, as it has been pointed out before, is penalized by consumptions related to the plant air compressor and the overall auxiliaries, whose contribution should be split among three modules.

The adsorption of contaminants appears to have decreased in 2019, despite the increase in treated biogas, from 50472 [m^3] in 2018 to around 65644 [m^3] in 2019. This is caused in the first place by the lack of data regarding biogas composition for the first half of 2019, and secondly by the lower readings of both contaminants with respect to 2018. Both of these issues can be observed in Figure 30 and Figure 31. To provide a reference, also readings from the QUALVISTA online monitoring system have been added to the charts since, despite being less frequent, they are logged regularly.

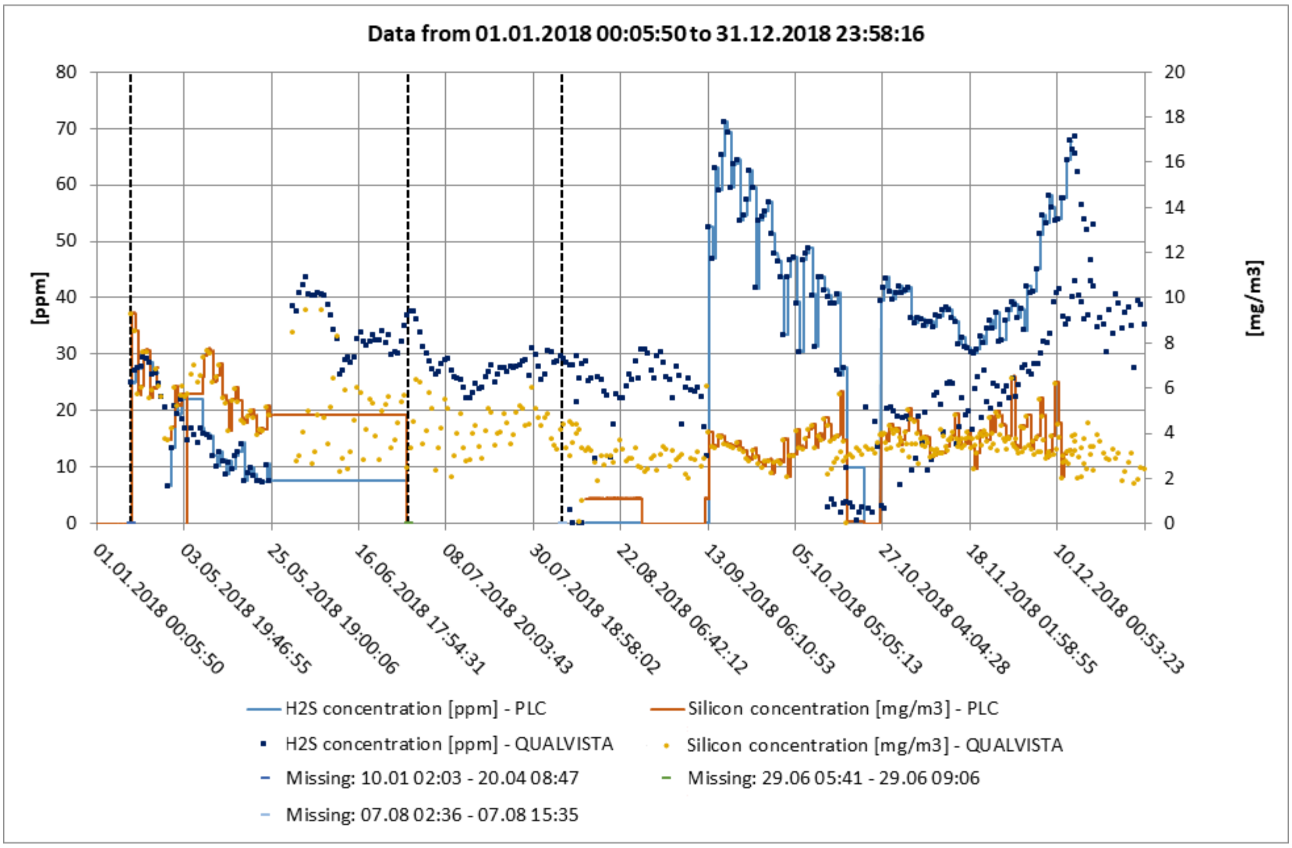


Figure 30: 2018 - Contaminants concentrations readings

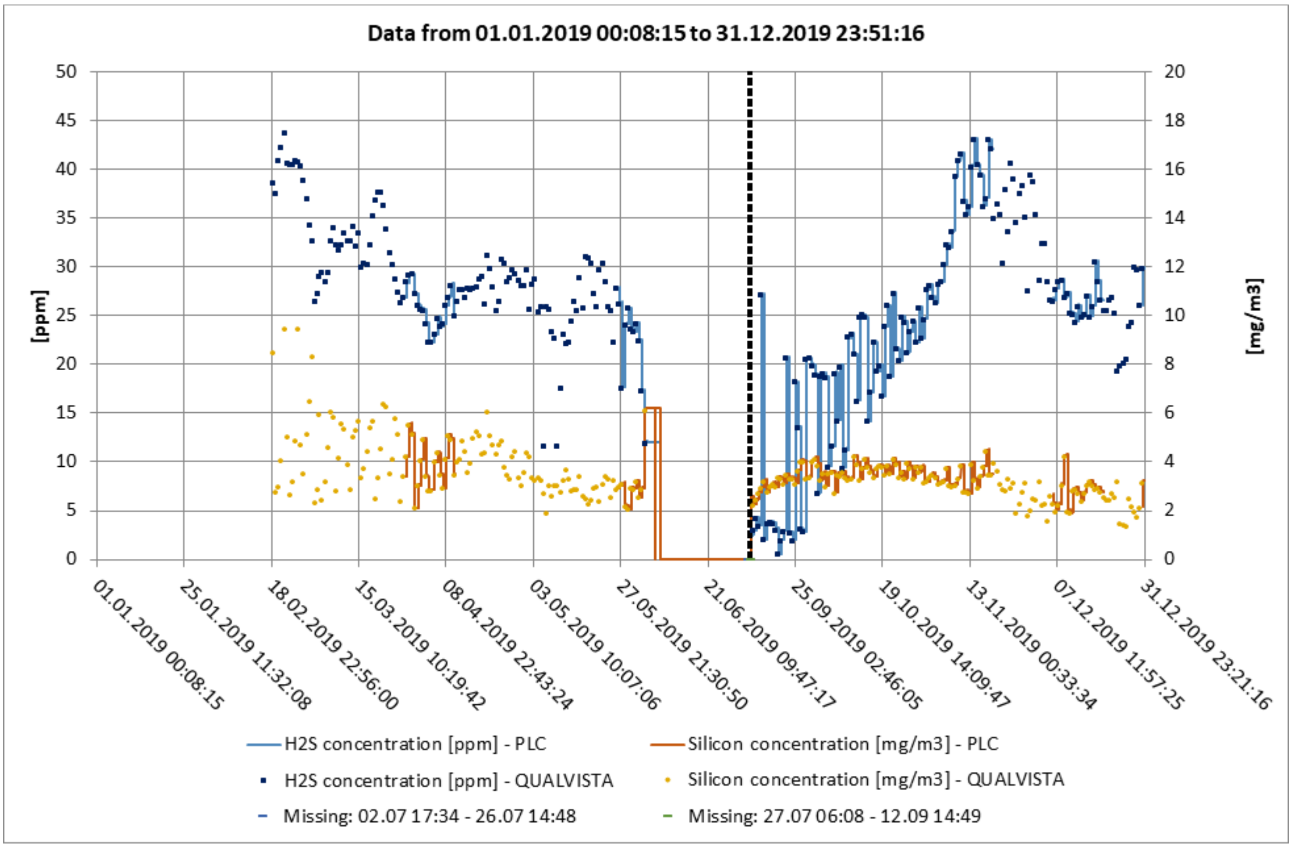


Figure 31: 2019 - Contaminants concentrations readings

More complete figures can be obtained by averaging the readings obtained from the QUALVISTA online monitoring system and referring them to the overall treated biogas (Table 6): results are more comparable between them than the ones obtained through the toolkit, but the difference in contaminants concentration is an issue that is more likely related to the measuring system itself.

Year	2018	2019
Biogas treated volume [m^3]	50472	65644
Weighted average for H2S concentration [ppm]	36.08	25.91
Weighted average for silicon concentration [mg/m^3]	4.21	3.34
Adsorbed equivalent sulphur [g]	2605.37	2433.94
Adsorbed equivalent D4 [g]	258.82	267.03
CKC reactor loading rate [%]	12.41	11.59
C64 reactor loading rate [%]	0.69	0.71

Table 6: Contaminants adsorbtion estimates according to QUALVISTA online measuring system

Despite the lack and uncertainty of data, the values of loading rates can still be useful in determining when a reactor, presumably the leading one if no changes were made in the meantime, is reaching its predicted capacity. In this scenario, the corresponding activated carbons should be replaced or regenerated.

5. Conclusions

The toolkit proved to be a valuable asset in simplifying data management and in speeding up the analysis of plant performance. Nonetheless, due to the nature of the DEMOSOFC project, monitoring and control of the plant are split among the different partners: some key information has to be shared with each other to be able to carry out an effective and complete diagnostic.

Enhancements to the actual code may furtherly help in tackling unexpected logging errors, like missing or extra commas, and correctly identify complex events chains, for example emergency shut down procedures. Some improvements could be implemented also according to the plant operators' feedback, for instance a monthly subdivision of KPIs to help the energy manager in estimating the thermal and electrical coverage of the digester, with the corresponding economic savings.

Hopefully this work will contribute to the continuous improvement and optimization of the modules and the plant itself, providing valuable and easily accessible information to promote the establishment of this cogeneration system in the energy market.

6. Reference

[1] Drago D., *The sulfur issue in fuel cell based cogenerative, systems*, Tesi di dottorato, Politecnico di Torino; 2017.

[2] Capello M., *Analysis of the performance of the first biogas-fed Solid Oxide Fuel Cell plant in Europe (the DEMOSOFC project)*,

[3] Gandiglio M. et al., *Results from an industrial size biogas-fed SOFC plant (the DEMOSOFC project), International*, Journal of Hydrogen Energy, https://doi.org/10.1016/j.ijhydene.2019.08.022

[4] Kevin Huang, *Fuel utilization and fuel sensitivity of solid oxide fuel cells*, Journal of Power Sources, Volume 196, Issue 5, 2011, Pages 2763-2767, ISSN 0378-7753, https://doi.org/10.1016/j.jpowsour.2010.10.077.

[5] Larminie and Dicks, Fuel Cell Handbook, 2000

[6] Patrick McMullen, Dynalene Inc., Glycols Heat Transfer Fluids: A Look at Thermal Conductivity and Thermal Stability April 1, 2012

[7] SMAT Analysis on Digester CA 3033 and CA 3034, 2012

[8] Takaya Ogawa, Mizutomo Takeuchi and Yuya Kajikawa, *Comprehensive Analysis of Trends and Emerging Technologies in All Types of Fuel Cells Based on a Computational Method*, Sustainability 2018, 10(2), 458; https://doi.org/10.3390/su10020458

7. Acknowledgements

I want to dedicate this achievement to the unfailing support of my family, my parents and my better half, without whom this journey would not have been possible. I am extremely thankful for the sacrifices they made for me and for the support they gave me throughout these years.

I want to thank my supervisors, Dr. Gandiglio and Prof. Santarelli for their help and great disposition during these months.

I am also grateful to all my friends, for they are an irreplaceable source of happiness.

This accomplishment would not have been possible without all of them.

From the bottom of my hearth,
Thank you.

8. Appendix

8.1. List of logged data

8.1.1. Analogic data

PLC denomination	Description	Unit of measure	Code
FIW_512_AP01_PRESSURE_DeEPT003	Vessel air pressure	mbar	A1
FIW_514_AP02_PRESSURE_DeEPT004	-	-	A2
FIW_516_HE57_TEMPERATURA_DeTT205	Water temperature after the mix valve HRU2	°C	A3
FIW_518_HE01_TEMPERATURA_DeTT103	Water inlet temperature HRU1	°C	A4
FIW_520_HE02_TEMPERATURA_DeTT104	Water outlet temperature HRU1	°C	A5
FIW_522_HE58_TEMPERATURA_DeTT305	Water temperature after the mix valve HRU3	°C	A6
FIW_524_HE03_TEMPERATURA_DeTT203	Water inlet temperature HRU2	°C	A7
FIW_526_HE04_TEMPERATURA_DeTT204	Water outlet temperature HRU2	°C	A8
FIW_528_HE05_TEMPERATURA_DeTT303	Water inlet temperature HRU3	°C	A9
FIW_530_HE06_TEMPERATURA_DeTT304	Water outlet temperature HRU3	°C	A10
FIW_532_HE07_TEMPERATURA_DeTT001	Water temperature on hot side inlet of hydraulic separator	°C	A11
FIW_534_HE08_TEMPERATURA_DeTT002	Water temperature on cold side outlet of hydraulic separator	°C	A12
FIW_536_HE09_TEMPERATURA_DeTT003	Water temperature on hot side outlet of hydraulic separator	°C	A13
FIW_538_HE10_TEMPERATURA_DeTT004	Water temperature on cold side inlet of hydraulic separator	°C	A14
FIW_540_HE16_PRESSURE_DeEPT013_BIOGAS	Biogas pressure after clean-up	mbar	A15
FIW_542_HE18_PRESSURE_DePT005_ H2O	Water pressure on hot side inlet of hydraulic separator	bar	A16
FIW_544_HE56_TEMPERATURA_DeTT105	Water temperature after the mix valve HRU1	°C	A17
FIW_548_DeFIT101_WaterFlow	Water flow HRU1	m^3/h	A18
FIW_550_DeFIT201_WaterFlow	Water flow HRU2	m^3/h	A19
FIW_552_DeFIT301_WaterFlow	Water flow HRU3	m^3/h	A20
FIW_560_GT34_FLOW METER_GAS_N	N2 flow	l/min	A21
FIW_562_GT35_FLOW METER_GAS_NHmix	NHmix flow	l/min	A22
FIW_576_SG01_LIVELLO_GASOMETRO _DeLT001	Gasometer biogas level	m	A23
FIW_578_SG02_PORTATA _DeFIT005	Biogas flow extracted from the gasometer	m^3/h	A24
FIW_580_SG03_PRESSIONE_FIT005_DePT012	Biogas pressure after extraction from gasometer	mbar	A25
FIW_582_SG04_CH4 DeFIT005	Methane content of biogas	%	A26
FIW_584_SG05_TEMPERATURA_DeFIT005	Biogas temperature after extraction from gasometer	°C	A27
FIW_586_HE11_TEMPERATURA_DeTT005	Water inlet temperature HRU4	°C	A28
FIW_588_HE12_TEMPERATURA_DeTT006	Water outlet temperature HRU4	°C	A29
FIW_590_HE13_TEMPERATURA_DeTT007	Sludge inlet temperature HRU4	°C	A30
FIW_592_HE14_TEMPERATURA_DeTT008	Sludge outlet temperature HRU4	°C	A31
FIW_594_HE15_PRESSURE_DeEPT001_FANGO	Sludge pressure HRU4	bar	A32
FIW_596_HE17_PRESSURE_DeEPT014 _H2O	Water pressure HRU4	bar	A33
FIW_602_HE19_FLOW_METER_DeFT001	Water flow HRU4	m^3/h	A34
FIW_604_HE20_FLOW_METER_DeFT002	Sludge flow HRU4	m^3/h	A35
ETH_SC_101_POWER_OUT_1	Electrical power output mod.1	kW	A36
ETH_SC_102_POWER_OUT_2	Electrical power output mod.2	kW	A37
ETH_SC_103_POWER_OUT_3	Electrical power output mod.3	kW	A38
ETH_SC_104_STACK_TEMPERATURE_1	Temperature mod.1	°C	A39
ETH_SC_105_STACK_TEMPERATURE_2	Temperature mod.2	°C	A40
ETH_SC_106_STACK_TEMPERATURE_3	Temperature mod.3	°C	A41
ETH_SC_107_LOADING_HOURS_1	Loading hours mod. mod.1	h	A42

ETH_SC_108_LOADING_HOURS_2	Loading hours mod. mod.2	h	A43
ETH_SC_109_LOADING_HOURS_3	Loading hours mod. mod.3	h	A44
ETH_SC_110_AMBIENT_AIR_CONSUMPTION_1	Ambient air consumption mod.1	kg/h	A45
ETH_SC_111_AMBIENT_AIR_CONSUMPTION_2	Ambient air consumption mod.2	kg/h	A46
ETH_SC_112_AMBIENT_AIR_CONSUMPTION_3	Ambient air consumption mod.3	kg/h	A47
ETH_SC_113_CMP_AIR_CONSUMPTION_1	Compressed air consumption mod.1	kg/h	A48
ETH_SC_114_CMP_AIR_CONSUMPTION_2	Compressed air consumption mod.2	kg/h	A49
ETH_SC_115_CMP_AIR_CONSUMPTION_3	Compressed air consumption mod.3	kg/h	A50
ETH_SC_116_EXHAUST_inlet_TEMP_1_DeTT102	Exhaust gas inlet temperature mod.1	°C	A51
ETH_SC_117_EXHAUST_inlet_TEMP_2_DeTT202	Exhaust gas inlet temperature mod.2	°C	A52
ETH_SC_118_EXHAUST_inlet_TEMP_3_DeTT302	Exhaust gas inlet temperature mod.3	°C	A53
ETH_SC_119_EXHAUST_outlet_TEMP_1_DeTT101	Exhaust gas outlet temperature mod.1	°C	A54
ETH_SC_120_EXHAUST_outlet_TEMP_2_DeTT201	Exhaust gas outlet temperature mod.2	°C	A55
ETH_SC_121_EXHAUST_outlet_TEMP_3_DeTT301	Exhaust gas outlet temperature mod.3	°C	A56
ETH_SC_125_REACTIV_POWER_OUT_1	Reactive power mod.1	kVAR	A57
ETH_SC_126_REACTIV_POWER_OUT_2	Reactive power mod.2	kVAR	A58
ETH_SC_127_REACTIV_POWER_OUT_3	Reactive power mod.3	kVAR	A59
ETH_SC_128_FUEL_FLOW_RATE_1	Fuel flow rate mod.1	kg/h	A60
ETH_SC_129_FUEL_FLOW_RATE_2	Fuel flow rate mod.2	kg/h	A61
ETH_SC_130_FUEL_FLOW_RATE_3	Fuel flow rate mod.3	kg/h	A62
ETH_SC_131_FUEL_INLET_PRESSURE_1	Fuel inlet pressure mod.1	mbar	A63
ETH_SC_132_FUEL_INLET_PRESSURE_2	Fuel inlet pressure mod.2	mbar	A64
ETH_SC_133_FUEL_INLET_PRESSURE_3	Fuel inlet pressure mod.3	mbar	A65
ETH_SC_134_FUEL_INLET_TEMPERATURE_1	Fuel inlet temperture mod.1	°C	A66
ETH_SC_135_FUEL_INLET_TEMPERATURE_2	Fuel inlet temperture mod.2	°C	A67
ETH_SC_136_FUEL_INLET_TEMPERATURE_3	Fuel inlet temperture mod.3	°C	A68
ETH_SC_137_AMBIENT_TEMPERATURE	Ambient air temperature	°C	A69
ETH_SC_138_AMBIENT_PRESSURE	Ambient air pressure	mbar	A70
ETH_SC_139_AMBIENT_HUMIDTY	Ambient air humidty	%	A71
ETH_SC_140_HRU_WATER_INLET_1	Water inlet temperature mod.1	°C	A72
ETH_SC_141_HRU_WATER_INLET_2	Water inlet temperature mod.2	°C	A73
ETH_SC_142_HRU_WATER_INLET_3	Water inlet temperature mod.3	°C	A74
ETH_SC_143_HRU_WATER_OUTLET_1	Water outlet temperature mod.1	°C	A75
ETH_SC_144_HRU_WATER_OUTLET_2	Water outlet temperature mod.2	°C	A76
ETH_SC_145_HRU_WATER_OUTLET_3	Water outlet temperature mod.3	°C	A77
Modbus_Signals_Clean_UP_MB_CORRENTE_MOTORE	Clean-up compressor power current	A	A78
Modbus_Signals_Clean_UP_MB_PRESSIONE_MANDATA	Clean-up compressor power discharge pressure	bar	A79
Modbus_Signals_Clean_UP_MB_POTENZA	Clean-up compressor power	kW	A80
Modbus_Signals_Clean_UP_MB_TEMPERATURA_POMPANTE	Clean-up compressor temperature	°C	A81
Modbus_Signals_Clean_UP_MB_VELOCITA	Clean-up compressor velocity	rpm	A82
Pump_DePR01_A_SetPoint_Speed_Out	Set point pump A HRU4	%	A83
Pump_DePR01_B_SetPoint_Speed_Out	Set point pump B HRU4	%	A84
Pump_PR101_A_SetPoint_Speed_Out	Set point pump A HRU1	%	A85
Pump_PR301_B_SetPoint_Speed_Out	Set point pump B HRU3	%	A86
Pump_PR101_B_SetPoint_Speed_Out	Set point pump A HRU1	%	A87
Pump_PR201_A_SetPoint_Speed_Out	Set point pump B HRU2	%	A88
Pump_PR301_A_SetPoint_Speed_Out	Set point pump A HRU3	%	A89
Pump_PR201_B_SetPoint_Speed_Out	Set point pump B HRU2	%	A90
ETH_SC_161_%CH4	Methane content of biogas (fast sensor)	%	A91

8.1.2. Digital data

PLC denomination	Code
FDIO.FI0_0_SC01_EMERGENCY_SHUTDOWN	D1
FDIO.FI0_1_SC02_MAINS_PROTECTION	D2
FDIO.FI0_2_SC03_GRID_PARALLEL_MODE_BY_PLC	D3
FDIO.FI0_3_SC04_INTERLOCKING_MAINS_PROTECTION	D4
FDIO.FI0_4_SC05_REMOTE_MODE_ENABLED	D5
FDIO.FI0_5_SC06_REMOTE_MODE_ENABLED	D6
FDIO.FI0_6_SC07_REMOTE_MODE_ENABLED	D7
FDIO.FI0_7_SC18_HOT_IDLE_ACTIVE	D8
FDIO.FI1_0_SC19_ISLAND_MODE_ACTIVE	D9
FDIO.FI1_1_SC20_RUN_ACTIVE_1	D10
FDIO.FI1_2_SC21_RUN_ACTIVE_2	D11
FDIO.FI1_3_SC22_RUN_ACTIVE_3	D12
FDIO.FI1_4_SC23_FAILURE_ACTIVE_1	D13
FDIO.FI1_5_SC24_FAILURE_ACTIVE_2	D14
FDIO.FI1_6_SC25_FAILURE_ACTIVE_3	D15
FDIO.FI10_0_GT20_DeVG08_1_OPEN	D16
FDIO.FI10_1_GT21_DeVG08_1_CLOSE	D17
FDIO.FI10_2_GT22_DeVG01_2_PURGE_OPEN	D18
FDIO.FI10_3_GT23_DeVG01_2_PURGE_CLOSE	D19
FDIO.FI10_4_GT24_DeVG02_2_PURGE_OPEN	D20
FDIO.FI10_5_GT25_DeVG02_2_PURGE_CLOSE	D21
FDIO.FI11_2_GT30_DeVG08_2_PURGE_OPEN	D22
FDIO.FI11_3_GT31_DeVG08_2_PURGE_CLOSE	D23
FDIO.FI2_0_QCS02_STATO	D24
FDIO.FI2_2_QCS04_AUTO	D25
FDIO.FI2_3_QCS05_BYPASS_MAN	D26
FDIO.FI2_4_QCS07_STATO	D27
FDIO.FI2_6_QCS09_AUTO	D28
FDIO.FI2_7_QCS10_BYPASS_MAN	D29
FDIO.FI20_0_AP03_ALIMENTAZIONE_QL5	D30
FDIO.FI20_1_AP04_ALIMENTAZIONE_QS_SOFC	D31
FDIO.FI20_2_AP06_EV_ISLAND_MODE_STATUS_DeVAM001	D32
FDIO.FI20_3_AP08_EV_RING_MODE_STATUS_DeVAM001	D33
FDIO.FI20_4_HE48_DePR01_A_PUMP_STATUS	D34
FDIO.FI20_5_HE52_DePR01_B_PUMP_STATUS	D35
FDIO.FI20_6_HE63_DeP01_POMPA_FANGO_STATUS	D36
FDIO.FI22_0_GT02_DeVG01_1_OPEN	D37
FDIO.FI22_1_GT03_DeVG01_1_CLOSE	D38
FDIO.FI22_2_GT04_DeVG01_3_OPEN	D39
FDIO.FI22_3_GT05_DeVG01_3_CLOSE	D40
FDIO.FI22_4_GT06_DeVG02_1_OPEN	D41
FDIO.FI22_5_GT07_DeVG02_1_CLOSE	D42
FDIO.FI22_6_GT08_DeVG02_3_OPEN	D43
FDIO.FI22_7_GT09_DeVG02_3_CLOSE	D44
FDIO.FI3_0_QCS12_STATO	D45
FDIO.FI3_2_QCS14_AUTO	D46
FDIO.FI3_3_QCS15_BYPASS_MAN	D47
FDIO.FI3_4_QCS17_STATO	D48
FDIO.FI3_6_QCS19_AUTO	D49
FDIO.FI3_7_QCS20_BYPASS_MAN	D50
FDIO.FI30_0_CE06_PRESENZA_TENSIONE_1	D51
FDIO.FI30_1_CE07_PRESENZA_TENSIONE_2	D52
FDIO.FI30_2_CE08_PRESENZA_TENSIONE_3	D53
FDIO.FI30_3_CE09_PRESENZA_TENSIONE_4	D54
FDIO.FI30_4_CE10_KM1_STATUS	D55
FDIO.FI30_5_CE11_KM2_STATUS	D56
FDIO.FI6_1_QCS39_UPS_COMMUTATORE_POS_A	D57

FDIO.FI6_2_QCS40_UPS_COMMUTATORE_POS_B	D58
FDIO.FI6_3_CU03_ALLARME_GENERICO	D59
FDIO.FI6_4_CU04_PRE_ALLARME_GENERICO	D60
FDIO.FI6_5_CU05_MACCHINA_IN_FUNZIONE	D61
FDIO.FI6_6_CU06_ALLBASSA_PRESSIONE_ASPIRAZIONE_COMP	D62
FDIO.FI6_7_HE21_PR101_A_PUMP_STATUS	D63
FDIO.FI7_0_HE25_PR101_B_PUMP_STATUS	D64
FDIO.FI7_1_HE29_PR201_A_PUMP_STATUS	D65
FDIO.FI7_2_HE33_PR201_B_PUMP_STATUS	D66
FDIO.FI7_3_HE37_PR301_A_PUMP_STATUS	D67
FDIO.FI7_4_HE41_PR301_B_PUMP_STATUS	D68
FDIO.FI8_2_GT10_DeVG09_6_OPEN	D69
FDIO.FI8_3_GT11_DeVG09_6_CLOSE	D70
FDIO.FI8_4_GT10_DeVG03_2_OPEN	D71
FDIO.FI8_5_GT11_DeVG03_2_CLOSE	D72
FDIO.FI8_6_GT10_DeVG03_1_OPEN	D73
FDIO.FI8_7_GT11_DeVG03_1_CLOSE	D74
FDIO.FI9_0_GT12_DeVG04_1_OPEN	D75
FDIO.FI9_1_GT13_DeVG04_1_CLOSE	D76
FDIO.FI9_2_GT14_DeVG05_1_OPEN	D77
FDIO.FI9_3_GT15_DeVG05_1_CLOSE	D78
FDIO.FI9_4_GT16_DeVG06_1_OPEN	D79
FDIO.FI9_5_GT17_DeVG06_1_CLOSE	D80
FDIO.FI9_6_GT18_DeVG07_1_OPEN	D81
FDIO.FI9_7_GT19_DeVG07_1_CLOSE	D82
FDIO.FQ0_0_SC08_RUN_COMMAND	D83
FDIO.FQ0_1_SC09_RUN_COMMAND	D84
FDIO.FQ0_2_SC10_RUN_COMMAND	D85
FDIO.FQ0_3_SC11_FORCE_STOP_COMMAND	D86
FDIO.FQ0_4_SC12_FORCE_STOP_COMMAND	D87
FDIO.FQ0_5_SC13_FORCE_STOP_COMMAND	D88
FDIO.FQ0_6_SC14_HOT_STANDBY_COMMAND	D89
FDIO.FQ1_0_QCS01_COMANDO	D90
FDIO.FQ1_1_QCS06_COMANDO	D91
FDIO.FQ1_2_QCS11_COMANDO	D92
FDIO.FQ1_3_QCS16_COMANDO	D93
FDIO.FQ1_4_QCS38_UPS_COMMUTATORE_COMANDO_MAN	D94
FDIO.FQ1_5_CU01_START	D95
FDIO.FQ1_6_CU02_STOP	D96
FDIO.FQ1_7_HE22_PR101_A_PUMP_COMMAND	D97
FDIO.FQ2_1_HE26_PR101_B_PUMP_COMMAND	D98
FDIO.FQ2_3_HE30_PR201_A_PUMP_COMMAND	D99
FDIO.FQ2_4_HE34_PR201_B_PUMP_COMMAND	D100
FDIO.FQ2_5_HE38_PR301_A_PUMP_COMMAND	D101
FDIO.FQ2_6_HE42_PR301_B_PUMP_COMMAND	D102
FDIO.FQ4_0_AP05_EV_ISLAND_MODE_COMMAND_DeVAM001	D103
FDIO.FQ4_1_AP07_EV_RING_MODE_COMMAND_DeVAM001	D104
FDIO.FQ4_2_HE49_DePR01_A_PUMP_COMMAND	D105
FDIO.FQ4_3_HE53_DePR01_B_PUMP_COMMAND	D106
FDIO.FQ4_4_HE62_DeP01_POMPA_FANGO_COMMAND	D107
FDIO.FQ4_5_GT01_DeVG01_1_COMMAND_OPEN	D108
FI4_0_QF14_KA1_CIRCOLATOREDePR101A	D109
FI4_1_QF15_KA2_CIRCOLATOREDePR101B	D110
FI4_2_QF16_KA3_CIRCOLATOREDePR201A	D111
FI4_3_QF17_KA4_CIRCOLATOREDePR201B	D112
FI4_4_QF18_KA5_CIRCOLATOREDePR301A	D113
FI4_5_QF19_KA6_CIRCOLATOREDePR301B	D114
FI4_6_QF20_KA7_CIRCOLATOREDePR01A	D115
FI4_7_QF21_KA8_CIRCOLATOREDePR01B	D116

FI5_0_QF23_KA10_DeP01POMPAFANGHI	D117
FI5_1_QF24_KA11_CLEANUPCONTAINER	D118
FI5_2_QF26_KA12_CLEANUPCHILLER/SOFFIANTE	D119
FI5_3_QF27_KA13_VALVOLEPRIMARIO	D120
FI5_4_QF28_KA14_VALVOLALOCALEPOMPE	D121
FI5_5_QF29_KA15_MOTORINOSOCOMEC	D122
FI5_6_QF34_KA17_QUADROGENERALIFANGHI	D123
FI5_7_QF35_KA18_QUADROPLC	D124
FI6_0_QF36_KA19_COMPRESSORE ARIA	D125

8.1.3. Electrical data

PLC denomination	Unit of measure	Code
POWER_Q_SALA_CONTROLLO_3_PHASE_SYSTEM_POWER_FACTOR	-	E1
POWER_Q_SALA_CONTROLLO_3_PHASE_SYSTEM_COSF	-	E2
POWER_Q_SALA_CONTROLLO_3_PHASE_SYSTEM_CURRENT	A	E3
POWER_Q_SALA_CONTROLLO_3_PHASE_SYSTEM_APPARENT_POWER	kVA	E4
POWER_Q_CABINA_ELETTRICA_3_PHASE_SYSTEM_COSF	-	E5
POWER_Q_CABINA_ELETTRICA_3_PHASE_SYSTEM_ACTIVE_POWER	kW	E6
POWER_Q_CABINA_ELETTRICA_3_PHASE_SYSTEM_APPARENT_POWER	kVA	E7
POWER_Q_CABINA_ELETTRICA_3_PHASE_SYSTEM_ACTIVE_ENERGY	MWh	E8
POWER_Q_SALA_CONTROLLO_3_PHASE_SYSTEM_ACTIVE_POWER	kW	E9
POWER_Q_CABINA_ELETTRICA_3_PHASE_SYSTEM_CURRENT	A	E10
POWER_Q_CABINA_ELETTRICA_3_PHASE_SYSTEM_POWER_FACTOR	-	E11
POWER_Q_SALA_CONTROLLO_3_PHASE_SYSTEM_ACTIVE_ENERGY	MWh	E12
POWER_Q_CABINA_ELETTRICA_FREQUENCY	Hz	E13
POWER_Q_CABINA_ELETTRICA_3_PHASE_SYSTEM_VOLTAGE	V	E14
POWER_Q_CABINA_ELETTRICA_3_PHASE_SYSTEM_REACTIVE_POWER	kVAR	E15
POWER_Q_CABINA_ELETTRICA_3_PHASE_SYSTEM_REACTIVE_ENERGY	MVARh	E16
POWER_Q_SALA_CONTROLLO_3_PHASE_SYSTEM_REACTIVE_ENERGY	MVARh	E17
POWER_Q_SALA_CONTROLLO_3_PHASE_SYSTEM_REACTIVE_POWER	kVAR	E18
POWER_Q_SALA_CONTROLLO_3_PHASE_SYSTEM_VOLTAGE	V	E19
POWER_Q_SALA_CONTROLLO_FREQUENCY	Hz	E20
MB_485_Var_POWER_METER_Q1_2_3_PHASE_SYSTEM_ACTIVE_POWER	kW	E21
MB_485_Var_POWER_METER_Q1_2_3_PHASE_SYSTEM_CURRENT	A	E22
MB_485_Var_POWER_METER_Q1_2_3_PHASE_SYSTEM_VOLTAGE	V	E23
MB_485_Var_POWER_METER_Q1_3_3_PHASE_SYSTEM_ACTIVE_POWER	kW	E24
MB_485_Var_POWER_METER_Q1_3_3_PHASE_SYSTEM_CURRENT	A	E25
MB_485_Var_POWER_METER_Q1_3_3_PHASE_SYSTEM_VOLTAGE	V	E26
MB_485_Var_POWER_METER_Q2_1_3_PHASE_SYSTEM_ACTIVE_POWER	kW	E27
MB_485_Var_POWER_METER_Q2_1_3_PHASE_SYSTEM_CURRENT	A	E28
MB_485_Var_POWER_METER_Q2_1_3_PHASE_SYSTEM_VOLTAGE	V	E29

8.1.4. Gas composition data

PLC denomination	Code
CH4_S0	G1
CO_S0	G2
H2S_S0	G3
O2_S0	G4
Si_S0	G5
CH4_S1	G6
CO_S1	G7
H2S_S1	G8
O2_S1	G9
Si_S1	G10

PLC denomination	Code
CH4_S4	G21
CO_S4	G22
H2S_S4	G23
O2_S4	G24
Si_S4	G25
CH4_S5	G26
CO_S5	G27
H2S_S5	G28
O2_S5	G29
Si_S5	G30

CH4_S2	G11		CH4_S6	G31
CO_S2	G12		CO_S6	G32
H2S_S2	G13		H2S_S6	G33
O2_S2	G14		O2_S6	G34
Si_S2	G15		Si_S6	G35
CH4_S3	G16		CH4_S7	G36
CO_S3	G17		CO_S7	G37
H2S_S3	G18		H2S_S7	G38
O2_S3	G19		O2_S7	G39
Si_S3	G20		Si_S7	G40

8.2. List of auxiliary equipment

Equipment	**Switch**
Light UPS room	QF2
Lighht dashboard room	QF3
External light	QF4
Spare	QF5
Router socket dashboard room	QF6
Wall socket control room	QF7
Air conditioning UPS room	QF8
Air conditioning dashboard room	QF9
Air conditioning control room	QF10
Spare	QF11
Spare	QF12
Dashboard PLC transformer room	QF13
Pump DePR101A	QF14
Pump DePR101B	QF15
Pump DePR201A	QF16
Pump DePR201B	QF17
Pump DePR301A	QF18
Pump DePR301B	QF19
Pump DePR01A	QF20
Pump DePR01B	QF21
Spare	QF22
Extractor fan of dashboard room	Extra
DeP01 sludge pump	QF23
Bio-Komp container	QF24
External service socket	QF25
Bio-Komp chiller and blower	QF26
Glycol water line valves	QF27
Sludge line valves	QF28
Socomec motor	QF29
Spare	QF30
Compressed air dessicator	QF31
Convion operator panel	QF32
Sludge macerator	QF33
PLC dashboard sludge room	QF34
PLC dashboard control room	QF35
Air compressor	QF36
Heater	QF37
Heater	QF38
Spare	QF39
Spare	QF40
Switches dashboard auxiliaries	QF41

8.3. VBA code

8.3.1. Import procedure

```
Public Ovw, ex As Boolean

Public Sub Callback1(control As IRibbonControl)
Dim ToolkitName As String
ToolkitName = "DEMOSOFC - Data analsysis toolkit.xlsm"
Dim ofcount As Integer
Dim oWB As Excel.Workbook
Dim CSVsh As Worksheet
Dim CSVName As String

'Opening File dialog box
With Application.FileDialog(msoFileDialogOpen)
    .AllowMultiSelect = True
    .Filters.Clear

    If .show = True Then
        For ofcount = 1 To .SelectedItems.Count
            'Opening selected file
            Set oWB = Workbooks.Open(Filename:=.SelectedItems(ofcount), _
            AddToMRU:=False, Local:=True)
            CSVName = Mid(oWB.Name, InStrRev(oWB.Name, "\") + 1)
            Call CSV_Import(CSVName)
        Next

        'Order sheets and confirm data writing
        Dim ShCount As Long
        Windows(ToolkitName).Activate
        ShCount = Sheets.Count
        For i = 1 To ShCount - 1
            For j = i + 1 To ShCount
                If UCase(Sheets(j).Name) < UCase(Sheets(i).Name) Then _
                Sheets(j).Move before:=Sheets(i)
            Next
        Next

        MsgBox "Import procedure completed.", vbInformation, "Done"
    Else
        MsgBox "No file selected.", vbInformation, "Sorry!"
        Exit Sub
    End If

End With
End Sub
```

```
Public Sub CSV_Import(CSVName As String)
oldStatusBar = Application.DisplayStatusBar
Application.DisplayStatusBar = True

'Assign workbooks names to variables
Dim ToolkitName As String
ToolkitName = "DEMOSOFC - Data analsysis toolkit.xlsm"
Dim CSVsh As Worksheet
If Workbooks(CSVName).Sheets(1).Type = xlWorksheet Then
    Set CSVsh = Workbooks(CSVName).Sheets(1)
Else
    GoTo InputErrorHandle
End If

Windows(ToolkitName).Activate
Application.ScreenUpdating = False

'Clean log errors
Dim del1 As String: del1 = "$RT_OFF$"
Dim del2 As String: del2 = "$RT_DIS$"
Dim del3 As String: del3 = "$RT_COUNT$"
Call CleanLog(del1, CSVsh)
Call CleanLog(del2, CSVsh)
Call CleanLog(del3, CSVsh)

'Find data type and number
Dim CSVType, SheetType As String
Dim ProcessNumber, LogVarNumber, LogVarSize, LogVarNumberClassic As Long
Dim rng As Range
Dim notstand As Boolean

With CSVsh
    CSVType = .Range("A2").Value
    notstand = False
    Select Case CSVType
        Case "FIW_512_AP01_PRESSURE_DeEPT003"
            SheetType = "Analogic"
            LogVarNumberClassic = 91
        Case "FDIO.FI0_0_SC01_EMERGENCY_SHUTDOWN"
            SheetType = "Digital"
            LogVarNumberClassic = 125
        Case "POWER_Q_SALA_CONTROLLO_3_PHASE_SYSTEM_POWER_FACTOR"
            SheetType = "Electric"
            LogVarNumberClassic = 29
        Case Else
            If CSVType Like "CH4_S#" Then
                SheetType = "Gas"
                LogVarNumberClassic = 5
            Else
                Set rng = Workbooks(ToolkitName).Worksheets(_
                "2019 - Analogic").Range("A1").EntireRow.Find (_
                CSVType, , xlValues, xlWhole)
                If Not rng Is Nothing Then
                    SheetType = "Analogic"
                    LogVarNumberClassic = 91
                    notstand = True
                Else
                    Set rng = Workbooks(ToolkitName).Worksheets(_
                    "2019 - Digital").Range("A1").EntireRow.Find(_
                    CSVType, , xlValues, xlWhole)
```

```
                    If Not rng Is Nothing Then
                        SheetType = "Digital"
                        LogVarNumberClassic = 125
                        notstand = True
                    Else
                        Set rng = Workbooks(ToolkitName).Worksheets(_
                        "2019 - Electric").Range("A1").EntireRow.Find(_
                        CSVType, , xlValues, xlWhole)
                        If Not rng Is Nothing Then
                            SheetType = "Electric"
                            LogVarNumberClassic = 29
                            notstand = True
                        Else
                            Set rng = Workbooks(ToolkitName).Worksheets(_
                            "2019 - Gas").Range("A1").EntireRow.Find(_
                            CSVType, , xlValues, xlWhole)
                            If Not rng Is Nothing Then
                                SheetType = "Gas"
                                LogVarNumberClassic = 5
                                notstand = True
                            Else
InputErrorHandle:
                                MsgBox "File " & CSVName & _
                                " not recognized as Analogic, Digital, " &_
                                "Electrical or Gas data log." & Chr(13) & _
                                "The file will be closed.", vbInformation
                                GoTo CloseLine
                            End If
                        End If
                    End If
                End If
            End If
    End Select
    ProcessNumber = WorksheetFunction.CountIf(.Range("A:A"), CSVType)
    LogVarSize = (CSVsh.Range("A1").End(xlDown).Row - 1)
    LogVarNumber = WorksheetFunction.Ceiling_Precise(_
                   LogVarSize / ProcessNumber)
End With

'Variables setting
Dim shfound, miss, force, dstbool As Boolean
Dim ws, TKsh As Worksheet
Dim id1, id2, rng1, rng2, rng3 As Range
Dim unusedrow, i, j, k, col, year As Long
year = 0
col = 2
Dim dstind As String: dstind = "0"
dstbool = False
ex = False
UF_Duplicate.CB_Check.Value = False

'Preemptive exit for empty Gas logs
If SheetType = "Gas" Then
    col = col + 5 * Right(CSVsh.Range("A2"), 1)
    If WorksheetFunction.Sum(CSVsh.Range("C:C")) = 0 Then GoTo CloseLine
End If
```

```
'Warnings for forcing correct detection of odd logs
Set id2 = CSVsh.Range("A2")
If notstand Then
    force = True
    result = MsgBox("The data outputs appear to be in a different order " & _
             "with respect to the most recent setup." & Chr(13) & Chr(13) & _
             "Select OK for forcing automatic detection of correct data " & _
             "output (This will require more processing time)." & Chr(13) & _
             Chr(13) & "Select CANCEL to abort the import procedure.", _
             vbOKCancel + vbExclamation, "Warning")
    Select Case result
        Case vbOK
            force = True
        Case vbCancel
            GoTo CloseLine
    End Select
Else
    If LogVarNumber = LogVarNumberClassic Then
        force = False
    Else
        result = MsgBox("The number of identified data outputs (" & _
              LogVarNumber & ") is different from the most recent setup (" & _
              & LogVarNumberClassic & ") for the " & SheetType & _
              " data type." & Chr(13) & Chr(13) & _
              "Select YES for forcing automatic detection of correct data" & _
              "output (This will require more processing time)." & Chr(13) & _
              Chr(13) & "Select CANCEL to abort the import procedure.", _
              vbOKCancel + vbExclamation, "Warning")
        Select Case result
            Case vbOK
                force = True
            Case vbCancel
                GoTo CloseLine
        End Select
    End If
End If



'Data identification, copy and paste, ordering, sheet creations
For i = 1 To ProcessNumber
    Application.StatusBar = ("Processing " & CSVName & ", data " & i & _
    " of " & ProcessNumber & " (" & Round(i / ProcessNumber * 100) & "%)")
    Set id1 = id2
    If i = ProcessNumber Then
        Set id2 = CSVsh.Range("A" & LogVarSize + 2)
    Else
        Set id2 = CSVsh.Range("A:A").Find(CSVType, id1, xlValues, xlWhole)
    End If
    If id2.Row - id1.Row = LogVarNumber Then
        miss = False
    Else
        miss = True
    End If
    Set rng1 = CSVsh.Range("B" & id1.Row)

    'Check for October daylight saving time change
    If Val(Mid(rng1, 4, 2)) = 10 And (Not i = 1) Then
        If dstbool = False Then
            Set rng2 = CSVsh.Range("B" & id1.Row - 1)
            dt = deltat(rng1, rng2)
            If dt < 0 Then
```

```
                rng1.Value = rng1.Value & " DST"
                dstind = rng2.Value
                dstbool = True
            End If
        Else
            dt = deltat(rng1, dstind)
            If dt < 0.003 Then
                rng1.Value = rng1.Value & " DST"
            Else
                dstbool = False
            End If
        End If
    End If

    'Sheet ordering if year changes
    If Not year = 0 Then
        If Not Val(Mid(rng1, 7, 4)) = year Then
            Call SheetOrder(year, ToolkitName, SheetType)
            Application.StatusBar = ("Processing data " & (i + 1) & " of " & _
            ProcessNumber & " (" & Round((i + 1) / (ProcessNumber) * 100) & _
            "%)")
        End If
    End If

    'Find or create correct worksheet for data addition
    year = Val(Mid(rng1, 7, 4))
    shfound = False
    With Workbooks(ToolkitName)
        For j = 1 To .Worksheets.Count
            If .Worksheets(j).Name = (year & " - " & SheetType) Then
                shfound = True
                Exit For
            End If
        Next
        If Not shfound = True Then
            Set ws = .Worksheets.Add(Type:=xlWorksheet)
            ws.Name = (year & " - " & SheetType)
            .Worksheets(2019 & " - " & SheetType).Range("A1").EntireRow.Copy _
            ws.Range("A1")
            If SheetType = "Analogic" Then
               .Worksheets(year - 1 & " - " & SheetType).Range("A1").End(_
               xlDown).Copy ws.Range("A2")
            End If
        End If

        'Search for pre-existing date
        Set TKsh = .Worksheets(year & " - " & SheetType)
        Set rng2 = TKsh.Range("A:A").Find(rng1.Value, , xlValues, xlWhole)
        If rng2 Is Nothing Then
            'Check for Gas similar data
            If SheetType = "Gas" Then
                Set rng3 = TKsh.Range("A:A").Find(Mid(rng1.Value, 1, 16), , _
                           xlValues, xlPart)
                If Not rng3 Is Nothing Then
                    If Not ((rng3.Value Like "*DST") Xor _
                    (rng1.Value Like "*DST")) Then
                        Set rng2 = rng3
                        GoTo GasBypass1
                    Else
```

```
                        If rng1.Value Like "*DST" Then
                           Set rng3 = _
                               TKsh.Range("A" & rng3.Row + 1 & ":A" & _
                               rng3.Row + 12).Find(Mid(rng1.Value, 1, 16), , _
                               xlValues, xlPart)
                            If Not rng3 Is Nothing Then
                                Set rng2 = rng3
                                GoTo GasBypass1
                            End If
                        End If
                    End If
                End If
            End If

            'No duplicate: copy and paste date and data on last line
            unusedrow = _
            TKsh.Cells.SpecialCells(xlCellTypeLastCell).Offset(1, 0).Row
            CSVsh.Range("B" & id1.Row).Copy _
            TKsh.Range("A" & unusedrow)
            If miss Or force Then
                For j = id1.Row To id2.Row - 1
                    Set rng3 = _
                    TKsh.Range("A1").EntireRow.Find(CSVsh.Range("A" & _
                    j).Value, , xlValues, xlWhole)
                    If Not rng3 Is Nothing Then
                        CSVsh.Range("C" & j).Copy _
                        TKsh.Cells(unusedrow, rng3.Column)
                    Else
                        CSVsh.Range("A" & j).Copy _
                        TKsh.Range("A1").End(xlToRight).Offset(0, 1)
                        CSVsh.Range("C" & j).Copy _
                        TKsh.Cells(unusedrow, TKsh.Range(_
                        "A1").End(xlToRight).Column)
                    End If
                Next
            Else
                CSVsh.Range("C" & id1.Row & ":" & "C" & id2.Row - 1).Copy
                TKsh.Cells(unusedrow, col).PasteSpecial Paste:=xlPasteAll, _
                Operation:=xlNone, _
                SkipBlanks:=False, Transpose:=True
            End If

        Else
GasBypass1:
            If SheetType = "Gas" And WorksheetFunction.CountA(_
            TKsh.Range(TKsh.Cells(rng2.Row, col), TKsh.Cells(_
            rng2.Row, col + 4))) = 0 Then GoTo GasBypass2

            'Duplicate: open userform, overwrite or ignore
            If UF_Duplicate.CB_Check.Value = False Then
                UF_Duplicate.OVLabel.Caption = _
                ("Data for the date " & Chr(13) & rng1.Value & Chr(13) & _
                " is already present." & Chr(13) & _
                "Do you want to overwrite it or ignore it?")
                Call CloseButtonSettings(UF_Duplicate, False)
                UF_Duplicate.show
            Else
            End If
```

```
            If ex = True Then GoTo CloseLine
            If Ovw = True Then
                unusedrow = rng2.Row
                If SheetType = "Gas" Then
                    TKsh.Range(TKsh.Cells(unusedrow, col), TKsh.Cells(_
                    unusedrow, col + 4)).ClearContents
                Else
                    TKsh.Rows(unusedrow).ClearContents
                    CSVsh.Range("B" & id1.Row).Copy _
                    TKsh.Range("A" & unusedrow)
                End If
                If miss Or force Then
                    For j = id1.Row To id2.Row - 1
                        Set rng3 = _
                        TKsh.Range("A1").EntireRow.Find(_
                        CSVsh.Range("A" & j).Value, , _
                        xlValues, xlWhole)

                        If Not rng3 Is Nothing Then
                            CSVsh.Range("C" & j).Copy _
                            TKsh.Cells(unusedrow, rng3.Column)
                            Else
                            CSVsh.Range("A" & j).Copy _
                            TKsh.Range("A1").End(xlToRight).Offset(0, 1)
                            CSVsh.Range("C" & j).Copy _
                            TKsh.Cells(unusedrow, TKsh.Range(_
                            "A1").End(xlToRight).Column)
                        End If
                    Next
                Else
GasBypass2:
                    unusedrow = rng2.Row
                    CSVsh.Range("C" & id1.Row & ":" & "C" & id2.Row - 1).Copy
                    TKsh.Cells(unusedrow, col).PasteSpecial _
                    Paste:=xlPasteAll, Operation:=xlNone, _
                    SkipBlanks:=False, Transpose:=True
                End If

            End If
        End If
    End With

Next

'Sheet ordering
Call SheetOrder(year, ToolkitName, SheetType)

Application.StatusBar = False
Application.DisplayStatusBar = oldStatusBar
'MsgBox "Data correctly loaded from selected file." & Chr(13) & _
"The file will now be closed.", vbInformation, CSVName & _
" file import completed"

'Close CSV
CloseLine:
Windows(CSVName).Close savechanges:=False
Application.ScreenUpdating = True
Application.StatusBar = False
Application.DisplayStatusBar = oldStatusBar

End Sub
```

```
Public Sub SheetOrder(year As Long, ToolkitName, SheetType As String)
'Imported sheet ordering
Application.StatusBar = "Ordering data for the worksheet '" & year & " - " &
SheetType & "'"
Dim i, j, k, p As Long
Dim x As Boolean
x = False

With Workbooks(ToolkitName).Worksheets(year & " - " & SheetType)
    .Range("B:B").Insert
    .Range("B1") = "Month"
    For k = 2 To .Range("A1").End(xlDown).Row
        .Range("B" & k) = Mid(.Range("A" & k), 4, 2)
    Next
    .Sort.SortFields.Clear
    .Sort.SortFields.Add Key:=.Range("B1"), SortOn:=xlSortOnValues,_
     Order:=xlAscending, DataOption:=xlSortNormal
    .Sort.SortFields.Add Key:=.Range("A1"), SortOn:=xlSortOnValues, _
     Order:=xlAscending, DataOption:=xlSortNormal
    .Sort.SetRange .UsedRange
    .Sort.Header = xlYes
    .Sort.MatchCase = False
    .Sort.Orientation = xlTopToBottom
    .Sort.Apply
    .Range("B:B").Delete

    Set rngd = .Range("A:A").Find("DST", , xlValues, xlPart, , xlPrevious)
    If Not rngd Is Nothing Then
        Set rngc = .Range("A" & rngd.Row + 1)
        dt = deltat(rngd, rngc)
        If Abs(dt) < 0.003 Then
            .Range("A" & rngd.Row).EntireRow.Cut
            .Range("A" & rngd.Row + 2).EntireRow.Insert shift:=xlShiftDown
            Set rngd = _
            .Range("A:A").Find("DST", , xlValues, xlPart, , xlPrevious)
        End If
        j = 0
        For i = rngd.Row - 1 To rngd.Row - 12 Step -1
            If .Range("A" & i) Like "*" & "DST" Then
                If Not i = rngd.Row - 1 - j Then
                   .Range("A" & i).EntireRow.Cut
                   .Range("A" & rngd.Row - j).EntireRow.Insert _
                   shift:=xlShiftDown
                End If
                j = j + 1
            End If
        Next
    End If

    Set rngd = .Range("A:A").Find(year - 1, , xlValues, xlPart)
    .Range("A" & rngd.Row).EntireRow.Cut
    .Range("A" & 2).EntireRow.Insert shift:=xlShiftDown

End With
End Sub
```

8.3.2. Key Performance Indicators evaluation

```
Public Sub KPIsub(year As Long)

UF_KPI.Hide
oldStatusBar = Application.DisplayStatusBar
Application.DisplayStatusBar = True
Application.ScreenUpdating = False


'Identify Digital and Electric worksheets names for the same year
Dim ToolkitName As String
ToolkitName = "DEMOSOFC - Data analsysis toolkit.xlsm"
Dim dig, ele, gas As String
dig = year & " - Digital"
ele = year & " - Electric"
gas = year & " - Gas"
With Workbooks(ToolkitName).Worksheets(year & " - Analogic")
Application.StatusBar = _
"Evaluating key performance indexes for the year " & year

'Identify last row for analogic worksheet
Dim lastrow, i, j, x, mw, f1, f2, f3, fp, Tdiff, module, b, eps, z, si As Long
Dim rng As Range

lastrow = .Range("A2").End(xlDown).Row

For i = 3 To lastrow
Application.StatusBar = "Evaluating KPIs for the year " & year & " - " & _
Round(i / lastrow * 100) & "%"
'Evaluate elapsed time
If UF_KPI.OptionButton1.Value = True Or IsEmpty(.Range("DA" & i)) Then
    .Range("DA" & i) = deltat(.Range("A" & i), .Range("A" & i - 1))
    If .Range("DA" & i) < 0 Or _
        (Round(.Range("DA" & i), 3) = 1.167 And _
        (Mid(.Range("A" & i), 5, 1) = 3 Or Mid(.Range("A" & i), 5, 1) = 4)) _
        Then .Range("DA" & i) = 1 / 6

    'Thermal info
    Set rng = Workbooks(ToolkitName).Worksheets(dig).Range("A:A").Find(_
    .Range("A" & i), , xlValues, xlWhole)
    If Not rng Is Nothing Then
        j = rng.Row
        Workbooks(ToolkitName).Worksheets(dig).Range("J" & j & ", AL" & _
        j & ", BJ" & j).Copy
        .Range("FA" & i).PasteSpecial Paste:=xlPasteAll, Operation:=xlNone, _
        SkipBlanks:=False
        'Info on thermal mod.1
        If IsEmpty(.Range("F" & i)) Or IsEmpty(.Range("E" & i)) Or _
        IsEmpty(.Range("S" & i)) Then
            .Range("DQ" & i).Clear
            .Range("DX" & i).Clear
            .Range("EA" & i).Clear
        Else
            Tdiff = .Range("F" & i) - .Range("E" & i)
            If Tdiff > 0 And Not .Range("S" & i) = 0.3 Then
                .Range("DQ" & i) = Round(1.071 * .Range("S" & i) * Tdiff, 3)
            Else
                .Range("DQ" & i) = 0
            End If
        End If
```

```
'Info on thermal mod.2
If IsEmpty(.Range("I" & i)) Or IsEmpty(.Range("H" & i)) Or _
IsEmpty(.Range("T" & i)) Then
    .Range("DR" & i).Clear
Else
    Tdiff = .Range("I" & i) - .Range("H" & i)
    If Tdiff > 0 And Not .Range("T" & i) = 0.3 Then
        .Range("DR" & i) = Round(1.071 * .Range("T" & i) * Tdiff, 3)
    Else
        .Range("DR" & i) = 0
    End If
End If

'Info on thermal mod.3
If IsEmpty(.Range("K" & i)) Or IsEmpty(.Range("J" & i)) Or _
IsEmpty(.Range("U" & i)) Then
    .Range("DS" & i).Clear
Else
    Tdiff = .Range("K" & i) - .Range("J" & i)
    If Tdiff > 0 And Not .Range("U" & i) = 0.3 Then
        .Range("DS" & i) = Round(1.071 * .Range("U" & i) * Tdiff, 3)
    Else
        .Range("DS" & i) = 0
    End If
End If

'Info on sludge heat exchanger (glycol side)
If IsEmpty(.Range("AC" & i)) Or IsEmpty(.Range("AD" & i)) Or _
IsEmpty(.Range("AI" & i)) Then
.Range("DU" & i).Clear
Else
    Tdiff = .Range("AC" & i) - .Range("AD" & i)
    If Tdiff > 0 And _
    (Workbooks(ToolkitName).Worksheets(dig).Range("DB" & j) = -1 Or _
     Workbooks(ToolkitName).Worksheets(dig).Range("DC" & j) = -1) Then
         .Range("DU" & i) = Round(1.071 * .Range("AI" & i) * Tdiff, 3)
    Else
         .Range("DU" & i) = 0
    End If
End If

'Info on sludge heat exchanger (sludge side)
If IsEmpty(.Range("AF" & i)) Or IsEmpty(.Range("AE" & i)) Or _
IsEmpty(.Range("AJ" & i)) Then
     .Range("DV" & i).Clear
Else
    Tdiff = .Range("AF" & i) - .Range("AE" & i)
    If Tdiff > 0 And _
    (Workbooks(ToolkitName).Worksheets(dig).Range("DB" & j) = -1 Or _
     Workbooks(ToolkitName).Worksheets(dig).Range("DC" & j) = -1) Then
         .Range("DV" & i) = Round(1.156 * .Range("AJ" & i) * Tdiff, 3)
    Else
         .Range("DV" & i) = 0
    End If
End If
```

```
        'Total thermal power removed
        If IsEmpty(.Range("DQ" & i)) Or IsEmpty(.Range("DR" & i)) Or _
        IsEmpty(.Range("DS" & i)) Then
             .Range("DT" & i).Clear
        Else
             .Range("DT" & i) = _
             WorksheetFunction.Sum(.Range("DQ" & i & ":DS" & i))
        End If

        'Thermal efficiency of heat exchange system
        If IsEmpty(.Range("DU" & i)) Or IsEmpty(.Range("DT" & i)) Then
            .Range("DW" & i).Clear
        Else
            If .Range("DT" & i) = 0 Then
                .Range("DW" & i) = 0
            Else
                Call eff(year, i, "DU", "DT", "DW")
            End If
        End If
    Else
        .Range("DQ" & i & ":DW" & i).Clear
    End If

    'Info on biogas methane content
    If IsEmpty(.Range("CN" & i)) And IsEmpty(.Range("AA" & i)) Then
        .Range("DB" & i).Clear
    Else
        If IsEmpty(.Range("CN" & i)) Then
            x = .Range("AA" & i) - 1.5
            If x < 0 Then x = 0
        Else
            x = .Range("CN" & i)
        End If
        .Range("DB" & i) = (x / 100) * 16.04 + (1 - (x / 100)) * 44.01
    End If
    If IsEmpty(.Range("DB" & i)) Then
        .Range("DB" & i & ":DK" & i).Clear
        .Range("DN" & i & ":DP" & i).Clear
        .Range("DX" & i & ":EC" & i).Clear
        .Range("EF" & i & ":EH" & i).Clear
    Else
        mw = .Range("DB" & i)
        'Info on flow rate to mod.1
        If IsEmpty(.Range("BI" & i)) Then
            .Range("DC" & i).Clear
            .Range("DF" & i).Clear
            .Range("DI" & i).Clear
            .Range("DN" & i).Clear
            .Range("DX" & i).Clear
            .Range("EA" & i).Clear
        Else
            f1 = .Range("BI" & i)
            .Range("DC" & i) = Round(f1 * 22.414 / mw, 3)
            .Range("DF" & i) = Round(f1 * (x / 100) * 16.04 / mw, 3)
            .Range("DI" & i) = Round((50 / 3.6) * .Range("DF" & i), 3)

            'Info on power from mod.1
            If f1 = 0 Then
                .Range("DN" & i) = 0
                .Range("DX" & i) = 0
                .Range("EA" & i) = 0
            Else
```

```
                If IsEmpty(.Range("AK" & i)) Then
                   .Range("DN" & i).Clear
                   .Range("EA" & i).Clear
                Else
                   Call eff(year, i, "AK", "DI", "DN")
                End If
                If IsEmpty(.Range("DQ" & i)) Then
                   .Range("DX" & i).Clear
                   .Range("EA" & i).Clear
                Else
                   Call eff(year, i, "DQ", "DI", "DX")
                End If
                If IsEmpty(.Range("DN" & i)) Or IsEmpty(.Range("DX" & i)) Then
                    .Range("EA" & i).Clear
                Else
                    .Range("EA" & i) = .Range("DN" & i) + .Range("DX" & i)
                    If .Range("EA" & i) > 100 Then .Range("EA" & i) = 100
                End If
            End If
        End If

        'Info on flow rate to mod.2
        If IsEmpty(.Range("BJ" & i)) Then
            .Range("DD" & i).Clear
            .Range("DG" & i).Clear
            .Range("DJ" & i).Clear
            .Range("DO" & i).Clear
            .Range("DY" & i).Clear
            .Range("EB" & i).Clear
        Else
            f2 = .Range("BJ" & i)
            .Range("DD" & i) = Round(f2 * 22.414 / mw, 3)
            .Range("DG" & i) = Round(f2 * (x / 100) * 16.04 / mw, 3)
            .Range("DJ" & i) = Round((50 / 3.6) * .Range("DG" & i), 3)
            'Info on power from mod.2
            If f2 = 0 Then
                .Range("DO" & i) = 0
                .Range("DY" & i) = 0
                .Range("EB" & i) = 0
            Else
                If IsEmpty(.Range("AL" & i)) Then
                   .Range("DO" & i).Clear
                   .Range("EB" & i).Clear
                Else
                   Call eff(year, i, "AL", "DJ", "DO")
                End If
                If IsEmpty(.Range("DR" & i)) Then
                   .Range("DY" & i).Clear
                   .Range("EB" & i).Clear
                Else
                   Call eff(year, i, "DR", "DJ", "DY")
                End If
                If IsEmpty(.Range("DO" & i)) Or IsEmpty(.Range("DY" & i)) Then
                    .Range("EB" & i).Clear
                Else
                    .Range("EB" & i) = .Range("DO" & i) + .Range("DY" & i)
                    If .Range("EB" & i) > 100 Then .Range("EB" & i) = 100
                End If
            End If
        End If
```

```
        'Info on flow rate to mod.3
       If IsEmpty(.Range("BK" & i)) Then
           .Range("DE" & i).Clear
           .Range("DH" & i).Clear
           .Range("DK" & i).Clear
           .Range("DP" & i).Clear
           .Range("DZ" & i).Clear
           .Range("EC" & i).Clear
       Else
           f3 = .Range("BK" & i)
           .Range("DE" & i) = Round(f3 * 22.414 / mw, 3)
           .Range("DH" & i) = Round(f3 * (x / 100) * 16.04 / mw, 3)
           .Range("DK" & i) = Round((50 / 3.6) * .Range("DH" & i), 3)
           'Info on power from mod.3
           If f3 = 0 Then
               .Range("DP" & i) = 0
               .Range("DZ" & i) = 0
               .Range("EC" & i) = 0
           Else
               If IsEmpty(.Range("AM" & i)) Then
                  .Range("DP" & i).Clear
                  .Range("EC" & i).Clear
               Else
                  Call eff(year, i, "AM", "DK", "DP")
               End If
               If IsEmpty(.Range("DS" & i)) Then
                  .Range("DZ" & i).Clear
                  .Range("EC" & i).Clear
               Else
                  Call eff(year, i, "DS", "DK", "DZ")
               End If
               If IsEmpty(.Range("DP" & i)) Or IsEmpty(.Range("DZ" & i)) Then
                   .Range("EC" & i).Clear
               Else
                   .Range("EC" & i) = .Range("DP" & i) + .Range("DZ" & i)
                   If .Range("EC" & i) > 100 Then .Range("EC" & i) = 100
               End If
           End If
       End If

       'Total biogas volumetric flow rate
       If Not (IsEmpty(.Range("DC" & i)) Or IsEmpty(.Range("DD" & i)) Or _
       IsEmpty(.Range("DE" & i))) Then
           .Range("DB" & i) = _
           WorksheetFunction.Sum(.Range("DC" & i & ":DE" & i))
           pf = WorksheetFunction.Sum(.Range("DI" & i & ":DK" & i))
       Else
           .Range("DB" & i).Clear
       End If
   End If
```

```
    'Global powers, auxiliaries, modules current and voltage, efficiencies
    If IsEmpty(.Range("AK" & i)) Or IsEmpty(.Range("AL" & i)) Or _
    IsEmpty(.Range("AM" & i)) Then
        .Range("DL" & i).Clear
        .Range("DM" & i).Clear
    Else
        .Range("DL" & i) = _
         WorksheetFunction.SumIf(.Range("AK" & i & ":AM" & i), ">0")
        .Range("DM" & i) = _
         WorksheetFunction.Sum(.Range("AK" & i & ":AM" & i))
    End If

    Set rng = Workbooks(ToolkitName).Worksheets(ele).Range("A:A").Find(_
    .Range("A" & i), , xlValues, xlWhole)
    If Not rng Is Nothing Then
       j = rng.Row
       If IsEmpty(Workbooks(ToolkitName).Worksheets(ele).Range("AC" & j)) Or _
       IsEmpty(Workbooks(ToolkitName).Worksheets(ele).Range("AD" & j)) Then
            .Range("ED" & i & ":EF" & i).Clear
       Else
            .Range("ED" & i) = Round(Sqr(3) * _
            Workbooks(ToolkitName).Worksheets(ele).Range("AC" & j) * _
            Workbooks(ToolkitName).Worksheets(ele).Range("AD" & j) / 1000, 3)
            If IsEmpty(.Range("DM" & i)) Then
                .Range("EE" & i & ":EF" & i).Clear
            Else
                .Range("EE" & i) = .Range("DM" & i) - .Range("ED" & i)
            End If
       End If
        .Range("EI" & i) = _
        Workbooks(ToolkitName).Worksheets(ele).Range("D" & j)
        .Range("EJ" & i) = _
        Workbooks(ToolkitName).Worksheets(ele).Range("T" & j)
    Else
        .Range("ED" & i & ":EF" & i).Clear
        .Range("EI" & i & ":EJ" & i).Clear
    End If

    If IsEmpty(.Range("DB" & i)) Then
        .Range("EF" & i & ":EH" & i).Clear
    Else
        If pf = 0 Then
            .Range("EF" & i & ":EH" & i) = 0
        Else
            If IsEmpty(.Range("EE" & i)) Then
               .Range("EF" & i).Clear
            Else
               Call eff2(year, i, "EE", pf, "EF")
            End If
            If IsEmpty(.Range("DU" & i)) Then
               .Range("EG" & i).Clear
            Else
               Call eff2(year, i, "DU", pf, "EG")
            End If
            If IsEmpty(.Range("EF" & i)) Or IsEmpty(.Range("EG" & i)) Then
               .Range("EH" & i).Clear
            Else
               .Range("EH" & i) = .Range("EF" & i) + .Range("EG" & i)
               If .Range("EH" & i) > 100 Then .Range("EH" & i) = 100
            End If
        End If
    End If
```

```
    'Loading of clean-up reactors
    If Len(.Range("A" & i)) = 19 Then
        Set rng = Workbooks(ToolkitName).Worksheets(gas).Range("A:A").Find(_
        Mid(.Range("A" & i), 1, 15), , xlValues, xlPart)
    Else
        v = Application.Match(Mid(.Range("A" & i), 1, 15) & "*" & "DST", _
        Workbooks(ToolkitName).Worksheets(gas).Range("A:A"), 0)
        If Not IsError(v) Then
            Set rng = Workbooks(ToolkitName).Worksheets(gas).Range("A" & v)
        Else
            Set rng = Nothing
        End If
    End If

    If (Not rng Is Nothing) And (Not IsEmpty(.Range("DB" & i))) Then
        j = rng.Row
        z = 0
        si = 0
        x = 0
        For k = 0 To 7
            If Not WorksheetFunction.CountBlank( _
            Workbooks(ToolkitName).Worksheets(gas).Range( _
            Workbooks(ToolkitName).Worksheets(gas).Cells(j, 2 + k * 5), _
            Workbooks(ToolkitName).Worksheets(gas).Cells(j, 6 + k * 5))) = 5 _
            Then
               z = Workbooks(ToolkitName).Worksheets(gas).Cells(j, 4 + k * 5)
               si = Workbooks(ToolkitName).Worksheets(gas).Cells(j, 6 + k * 5)
               Exit For
            End If
        Next
        If z > 0 Then
            .Range("EN" & i) = z * 1.43058 * .Range("DB" & i)
        Else
            .Range("EN" & i).Clear
        End If
        If si > 0 Then
            .Range("EO" & i) = si * 1.21763 * .Range("DB" & i)
        Else
            .Range("EO" & i).Clear
        End If
    Else
        .Range("EN" & i & ":" & "EO" & i).Clear
    End If


End If
Next
```

```
'Evaluation of operation mode
'Common variables
tim = "DA"
comp = "FC"
isl = "FA"
fuel = "FB"
b = 0

For module = 1 To 3
    Select Case module
      Case 1
          pow = "AK"
          temp = "AN"
          air_n = "AT"
          air_c = "AW"
          op = "EK"
      Case 2
          pow = "AL"
          temp = "AO"
          air_n = "AU"
          air_c = "AX"
          op = "EL"
      Case 3
          pow = "AM"
          temp = "AP"
          air_n = "AV"
          air_c = "AY"
          op = "EM"
    End Select

    For i = 3 To lastrow
        If UF_KPI.OptionButton1.Value = True Or IsEmpty(.Range(op & i)) Then
            Application.StatusBar = "Evaluating module " & module & _
            " operation mode for the year " & year & " - " & _
            Round(i / lastrow * 100) & "%"
            If IsEmpty(.Range(op & i - 1)) Or .Range(tim & i) > 0.17 Then

                'Static assumption: mode can be ON, OFF, START UP, SHUT DOWN
                If .Range(pow & i) > 0 Then
                    If .Range(air_c & i) < 1 Then
                        .Range(op & i) = "ON"
                    Else
                        Call op_start_up(year, i, tim, comp, isl, fuel, _
                        pow, temp, air_n, air_c, op, b, lastrow)
                    End If
                Else
                    If .Range(pow & i) <= 0 And .Range(pow & i) > -1 Then
                      .Range(op & i) = "OFF"
                    Else
                        Call op_temp_grad(year, i, tim, comp, isl, fuel, _
                        pow, temp, air_n, air_c, op, b, lastrow)
                        If b < 0 Then
                            Call op_shut_down(year, i, tim, comp, isl, fuel, _
                            pow, temp, air_n, air_c, op, b, lastrow)
                        Else
                            Call op_start_up(year, i, tim, comp, isl, fuel, _
                            pow, temp, air_n, air_c, op, b, lastrow)
                        End If
                    End If
                End If
            Else
```

```
            'Dynamic assumption: based on previous timestep
            Select Case .Range(op & i - 1)
            Case "OFF"
                If .Range(pow & i) < -1 Then
                    Call op_start_up(year, i, tim, comp, isl, fuel, _
                    pow, temp, air_n, air_c, op, b, lastrow)
                Else
                    .Range(op & i) = "OFF"
                End If

            Case "ON"
                If ((.Range(pow & i - 1) - .Range(pow & i)) / _
                .Range(pow & i - 1)) > 0.25 Then
                    Call op_temp_grad(year, i, tim, comp, isl, fuel, _
                    pow, temp, air_n, air_c, op, b, lastrow)
                    If b < -1 Then
                        Call op_shut_down(year, i, tim, comp, isl, fuel, _
                        pow, temp, air_n, air_c, op, b, lastrow)
                    Else
                       If .Range(pow & i) > 0 Then
                          Call op_island(year, i, tim, comp, isl, fuel, _
                          pow, temp, air_n, air_c, op, b, lastrow)
                       Else
                          Call op_hot_int(year, i, tim, comp, isl, fuel, _
                        pow, temp, air_n, air_c, op, b, lastrow)
                       End If
                    End If
                Else
                    .Range(op & i) = "ON"
                End If

            Case "ISL", "HOT", "INT"
            If .Range(pow & i) > 0 Then
                If WorksheetFunction.CountIf(_
                .Range(air_c & i & ":" & air_c & i + 2), ">1") = 3 Then
                    Call op_island(year, i, tim, comp, isl, fuel, _
                    pow, temp, air_n, air_c, op, b, lastrow)
                Else
                    .Range(op & i) = "ON"
                End If
            Else
                    Call op_hot_int(year, i, tim, comp, isl, fuel, _
                    pow, temp, air_n, air_c, op, b, lastrow)
            End If

            Case "SHUT"
                .Range(op & i) = "OFF"

            Case "START"
                .Range(op & i) = "ON"

            End Select
        End If
    End If
Next
```

```
    'Copy operation mode of last record to start of next year
    For j = 1 To Workbooks(ToolkitName).Worksheets.Count
        If Workbooks(ToolkitName).Worksheets(j).Name = _
        (year + 1 & " - " & Analogic) Then
            Set nws = _
            Workbooks(ToolkitName).Worksheets(year + 1 & " - Analogic")
            .Range(op & lastrow).Copy nws.Range(op & 2)
            Exit For
        End If
    Next
Next

'Delete copied digital values
.Range("FA:FC").Delete
End With

'Create summary
With Workbooks(ToolkitName).Worksheets("-INFO-")
x = year - 2018
.Range("E1:H48").Copy .Cells(1, 5 + x * 6)
.Cells(1, 6 + x * 6) = year
End With

'Restore interface
Application.ScreenUpdating = True
Application.StatusBar = False
Application.DisplayStatusBar = oldStatusBar
MsgBox "Evaluation of KPIs and modules operation mode has been carried out for
the selected year.", vbOKOnly, "Done"
End Sub
```

```
Sub eff(y, i, nom, denom, cell)
ToolkitName = "DEMOSOFC - Data analsysis toolkit.xlsm"
With Workbooks(ToolkitName).Worksheets(y & " - Analogic")
    If .Range(denom & i) = 0 Then
        .Range(cell & i).Clear
    Else
        eps = .Range(nom & i) / .Range(denom & i)
        If eps < 0 Or eps > 1 Then
            .Range(cell & i).Clear
        Else
            .Range(cell & i) = Round(eps * 100, 2)
        End If
    End If
End With
End Sub
```

```
Sub eff2(y, i, nom, denom, cell)
ToolkitName = "DEMOSOFC - Data analsysis toolkit.xlsm"
With Workbooks(ToolkitName).Worksheets(y & " - Analogic")
    eps = .Range(nom & i) / denom
    If eps < 0 Or eps > 1 Then
        .Range(cell & i).Clear
    Else
        .Range(cell & i) = Round(eps * 100, 2)
    End If
End With
End Sub
```

8.3.3. Operation mode identification sub routines

```
'--- ISLAND MODE ---
Sub op_island(year, i, tim, comp, isl, fuel, _
pow, temp, air_n, air_c, op, b, lastrow)
ToolkitName = "DEMOSOFC - Data analsysis toolkit.xlsm"
With Workbooks(ToolkitName).Worksheets(year & " - Analogic")
    j = i + 1
    Do While .Range(tim & j) < 0.5 And j <= lastrow And .Range(pow & j) > 0
        If (.Range(isl & j - 1) = -1 And .Range(isl & j) = 0) Or _
            .Range(air_c & j) < 1 Then
            Exit Do
        End If
        j = j + 1
    Loop
    .Range(op & i & ":" & op & j - 1) = "ISL"
    i = j - 1
End With
End Sub
```

```
'--- HOT STANDBY or INTERRUPTION ---
Sub op_hot_int(year, i, tim, comp, isl, fuel, _
pow, temp, air_n, air_c, op, b, lastrow)
ToolkitName = "DEMOSOFC - Data analsysis toolkit.xlsm"
With Workbooks(ToolkitName).Worksheets(year & " - Analogic")
    j = i + 1
    Do While .Range(tim & j) < 0.5 And j <= lastrow
        If .Range(pow & j) > 0 And _
            (.Range(comp & j) = -1 Or .Range(air_c & j) < 1) Then
            Exit Do
        End If
        j = j + 1
    Loop
    If Not WorksheetFunction.Sum(_
    .Range(fuel & i & ":" & fuel & j)) = -(j - i + 1) Then
        .Range(op & i & ":" & op & j - 1) = "HOT"
    Else
        .Range(op & i & ":" & op & j - 1) = "INT"
    End If
    i = j - 1
End With
End Sub
```

```
'--- SHUT DOWN ---
Sub op_shut_down(year, i, tim, comp, isl, fuel, _
pow, temp, air_n, air_c, op, b, lastrow)
ToolkitName = "DEMOSOFC - Data analsysis toolkit.xlsm"
With Workbooks(ToolkitName).Worksheets(year & " - Analogic")
    lapse = 1 / 6
    j = i + 1
    Do While .Range(tim & j) < 0.5 And lapse < 360 And j <= lastrow
        If (.Range(pow & j) <= 0 And .Range(pow & j) > -1) And _
            .Range(air_n & j) < 10 And _
            (.Range(temp & j) < 50 And Not .Range(temp & j) = 0) Then
            Exit Do
        End If
        lapse = lapse + .Range(tim & j)
        j = j + 1
    Loop
    .Range(op & i & ":" & op & j - 1) = "SHUT":    i = j - 1
End With
End Sub
```

```
'--- START UP or TEST (OFF) ---
Sub op_start_up(year, i, tim, comp, isl, fuel, _
pow, temp, air_n, air_c, op, b, lastrow)
ToolkitName = "DEMOSOFC - Data analsysis toolkit.xlsm"
With Workbooks(ToolkitName).Worksheets(year & " - Analogic")
    found = False
    lapse = 1 / 6
    j = i + 1
    Do While .Range(tim & j) < 0.5 And lapse < 100 And j <= lastrow
        If .Range(pow & j) > 0 And .Range(air_c & j) < 1 And _
        .Range(temp & j) > 700 Then
            found = True
            Exit Do
        End If
        lapse = lapse + .Range(tim & j)
        j = j + 1
    Loop
    If found Then
        .Range(op & i & ":" & op & j - 1) = "START"
        i = j - 1
    Else
        j = i + 1
        Do While .Range(tim & j) < 0.5 And j <= lastrow
            If (.Range(pow & j) <= 0 And .Range(pow & j) > -1) And _
                .Range(air_n & j) < 10 And _
                (.Range(temp & j) < 50 And Not .Range(temp & j) = 0) Then
                Exit Do
            End If
            j = j + 1
        Loop
        .Range(op & i & ":" & op & j - 1) = "OFF"
        i = j - 1
    End If
End With
End Sub
```

```
'--- TEMPERATURE GRADIENT (SIMPLE LINEAR REGRESSION)---
Sub op_temp_grad(year, i, tim, comp, isl, fuel, pow, temp, air_n, air_c, op,
b, lastrow)
ToolkitName = "DEMOSOFC - Data analsysis toolkit.xlsm"
With Workbooks(ToolkitName).Worksheets(year & " - Analogic")
    lapse = 0
    j = i
    x = 0
    xx = 0
    xy = 0
    y_m = 0
    x_m = 0
    n = 0
    b = 0
    j = j + 1
    Do While .Range(tim & j) < 0.5 And lapse < 24 And j <= lastrow
        x = x + 1
        If Not .Range(temp & j) = 0 Then
            xx = xx + x * x
            xy = xy + x * .Range(temp & j)
            y_m = y_m + .Range(temp & j)
            x_m = x_m + x
            n = n + 1
        End If
        lapse = lapse + .Range(tim & j)
        j = j + 1
    Loop
    x_m = x_m / n
    y_m = y_m / n
    b = (xy - n * x_m * y_m) / (xx - n * x_m * x_m)

End With
End Sub
```

8.3.4. Data visualization

```
Public Sub Callback3(control As IRibbonControl)
'Set years for dates selection
UF_Graph_2.ComboBox1.Style = fmStyleDropDownCombo
UF_Graph_2.ComboBox1.Clear
With Workbooks("DEMOSOFC - Data analsysis toolkit.xlsm")
        For j = 1 To .Worksheets.Count
            For i = 0 To UF_Graph_2.ComboBox1.ListCount
                If Val((Mid(.Worksheets(j).Name, 1, 4))) > 2000 And _
                Val((Mid(.Worksheets(j).Name, 1, 4))) < 2100 Then
                    If i = UF_Graph_2.ComboBox1.ListCount Then
                        UF_Graph_2.ComboBox1.AddItem _
                        ((Mid(.Worksheets(j).Name, 1, 4)))
                        Exit For
                    End If
                    If UF_Graph_2.ComboBox1.list(i) = _
                    Mid(.Worksheets(j).Name, 1, 4) Then
                        Exit For
                    End If
                End If

            Next
        Next
End With
UF_Graph_2.ComboBox1.ListIndex = -1
UF_Graph_2.ComboBox1.Style = fmStyleDropDownList

'Set existing graphics list
UF_Graph_1.TextBox1.Text = ""
UF_Graph_1.ComboBox1.Style = fmStyleDropDownCombo
UF_Graph_1.ComboBox1.Clear
With Workbooks("DEMOSOFC - Data analsysis toolkit.xlsm")
        For j = 1 To .Charts.Count
            UF_Graph_1.ComboBox1.AddItem .Charts(j).Name
        Next
End With
UF_Graph_1.ComboBox1.ListIndex = -1
UF_Graph_1.ComboBox1.Style = fmStyleDropDownList
UF_Graph_1.show

End Sub
```

```
'Build graphic button
Sub CommandButton1_Click()
Dim ToolkitName As String: ToolkitName = "DEMOSOFC - Data analsysis
toolkit.xlsm"
Dim ws As Worksheet
Dim row1, row2 As Long
Dim rng As Range
Dim kpist(18), t1(8), t2(8), t3(8), avex As Variant
Dim rng1, rng2, rng3 As Range
Dim a, b, c, i, j, ax1, ax2, year As Long
Dim rarea, rcell As Range
Dim boolout, booldate As Boolean
boolout = True
booldate = False

'Check for input content (output variables and dates)
For i = 10 To 17
    If Not Controls("ComboBox" & i).Value = "" Then
        boolout = False
        Exit For
    End If
Next
For i = 1 To 7
    If Controls("ComboBox" & i).Value = "" Then
        booldate = True
        Exit For
    End If
Next
If boolout Then
    MsgBox "Select at least one variable to plot.", vbOKOnly, _
    "Error in output variables"
Else
    If booldate Then
        MsgBox "Fill every date input before plotting.", vbOKOnly, _
        "Error in date input"
    Else
        date_start = ComboBox3.Text & "/" & ComboBox2.Text & "/" & _
        ComboBox1.Text & " " & ComboBox4.Text
        date_end = ComboBox6.Text & "/" & ComboBox5.Text & "/" & _
        ComboBox1.Text & " " & ComboBox7.Text
        year = ComboBox1.Value
        Set ws = Workbooks(ToolkitName).Worksheets(year & " - Analogic")
        If (24 * (DateValue(Mid(date_end, 1, 10)) - _
        DateValue(Mid(date_start, 1, 10))) + _
        Val(ComboBox7.Text) - Val(ComboBox4.Text)) <= 0 Then
            MsgBox "The ending date must be set after the starting date.", _
            vbOKOnly, "Error in date input"
        Else

            'First search attempt: exact match
            row1 = 1
            row2 = 1
            date_start_c = Mid(date_start, 1, 10)
            date_end_c = Mid(date_end, 1, 10)
            Set rng1 = ws.Range("A:A").Find _
            (todot(date_start), , xlValues, xlPart)




           If Not rng1 Is Nothing Then
```

```
                row1 = rng1.Row
                GoTo End_Check
            Else
                'Second search attempt: day by day
                Do Until date_start_c = date_end_c
                    Set rng1 = ws.Range("A:A").Find _
                    (todot(date_start_c), , xlValues, xlPart)
                    If Not rng1 Is Nothing Then
                        row1 = rng1.Row
                        GoTo End_Check
                    Else
                        date_start_c = _
                        Format(DateValue(date_start_c) + 1, "dd/mm/yyyy")
                    End If
                Loop
            End If
End_Check:
            'First search attempt: exact match
            Set rng2 = ws.Range("A:A").Find _
            (todot(date_end), , xlValues, xlPart, , xlPrevious)
            If Not rng2 Is Nothing Then
                row2 = rng2.Row
            Else
                'Second search attempt: day by day
                Do Until date_start_c = date_end_c
                    Set rng2 = ws.Range("A:A").Find _
                    (todot(date_end_c), , xlValues, xlPart, , xlPrevious)
                    If Not rng2 Is Nothing Then
                        row2 = rng2.Row
                        Exit Do
                    Else
                        date_end_c = _
                        Format(DateValue(date_end_c) - 1, "dd/mm/yyyy")
                    End If
                Loop
            End If

            If row1 = row2 Then
                MsgBox "There is no available data for the time period" & _
                " selected.", vbOKOnly, "No data available"
            Else
                If WorksheetFunction.CountBlank(_
                ws.Range("DA" & row1 & ":DA" & row2)) > 0 Then
                   MsgBox "KPIs have not been yet evaluated for some " & _
                   of the dates selected." & Chr(13) & _
                   "Use the dedicated tool before plotting.", vbOKOnly, _
                   "Missing KPIs for the reqested time period"
                Else


                    'Series addition and overview evaluation
                    Application.ScreenUpdating = False
                    UF_Graph_2.MultiPage1.Pages(2).Enabled = True
                    UF_Graph_2.MultiPage1.Pages(3).Enabled = True
                    UF_Graph_2.CommandButton2.Enabled = True
                    UF_Graph_2.MultiPage1.Value = 2
                    UF_Graph_2.ListBoxM.Clear
                    chartn = ActiveChart.Name
                    Application.DisplayAlerts = False
                    Charts(chartn).Delete
                    Application.DisplayAlerts = True
                    Set nc = Workbooks(ToolkitName).Sheets.Add(Type:=xlChart)
```

```
                    nc.ChartArea.ClearContents
                    nc.Name = chartn
                    nc.Activate
                    pvs = vars(Controls("ComboBox" & 9).ListIndex, 1)
                    ax1 = 0
                    ax2 = 0
                    j = 1
                    For i = 10 To 17
                        If Not Controls("ComboBox" & i) = "" Then
                            'New series
                            If Controls(_
                            "OptionButton" & ((i - 9) * 2)).Value = True Then
                                ax1 = ax1 + 1
                            Else
                                ax2 = ax2 + 1
                            End If
                            col = vars(_
                            Controls("ComboBox" & i).ListIndex + 2, 1)

                            Set ns = Charts(chartn).SeriesCollection.NewSeries
                            ns.Name = Controls("ComboBox" & i).Value

                            'Complete range selection and
                            'overview data evaluation
                            Set rng1 = ws.Range(col & row1 & ":" & col & row2)
                            Set rng2 = ws.Range(pvs & row1 & ":" & pvs & row2)
                            If j = 1 Then Set rng3 = _
                            ws.Range("DA" & row1 & ":" & "DA" & row2)
                            Charts(chartn).SeriesCollection(j).Values = rng1
                            Charts(chartn).SeriesCollection(j).XValues = rng2
                            avex = WorksheetFunction.IfError(_
                            WorksheetFunction.AverageIf(rng1, "<>"), "/")
                            maxx = WorksheetFunction.Max(rng1)
                            minx = WorksheetFunction.Min(rng1)

                            'Final evaluation and writing of:
                            'Overview data
                            UF_Graph_2.MultiPage3.Pages(j - 1).Enabled = True
                            Controls("Frame" & j & 0).Caption = _
                            Controls("ComboBox" & i).Value
                            Controls("ListBox" & j & 1).Clear
                            Controls("ListBox" & j & 1).AddItem Round(maxx, 2)
                            Controls("ListBox" & j & 2).Clear
                            Controls("ListBox" & j & 2).AddItem Round(minx, 2)
                            Controls("ListBox" & j & 3).Clear
                            If Not avex = "/" Then _
                            Controls("ListBox" & j & 3).AddItem Round(avex, 2)

                            If j = 1 Then
                                'Operation mode lengths
                                For b = 0 To 8
                                    t1(b) = 0
                                    t2(b) = 0
                                    t3(b) = 0
                                Next





                                For a = 1 To 7
```

```
        Select Case a
            Case 1
            w = "OFF"
            Case 2
            w = "ON"
            Case 3
            w = "ISL"
            Case 4
            w = "HOT"
            Case 5
            w = "START"
            Case 6
            w = "SHUT"
            Case 7
            w = "INT"
        End Select
        t1(a) = _
        WorksheetFunction.CountIf(ws.Range(_
        "EK" & row1 & ":" & "EK" & row2), w) / 6
        t2(a) = _
        WorksheetFunction.CountIf(ws.Range(_
        "EL" & row1 & ":" & "EL" & row2), w) / 6
        t3(a) = _
        WorksheetFunction.CountIf(ws.Range(_
        "EM" & row1 & ":" & "EM" & row2), w) / 6
    Next
    t1(8) = WorksheetFunction.SumIf(ws.Range(_
    "DA" & row1 & ":" & "DA" & row2), ">0.17") - _
    WorksheetFunction.CountIf(ws.Range(_
    "DA" & row1 & ":" & "DA" & row2), ">0.17") / 6
    t2(8) = t1(8)
    t3(8) = t1(8)
    t1(0) = WorksheetFunction.Sum(t1)
    t2(0) = WorksheetFunction.Sum(t2)
    t3(0) = WorksheetFunction.Sum(t3)

    For b = 1 To 9
        Controls("ListBox" & 11 & b).Clear
        Controls("ListBox" & 11 & b).AddItem _
        Round(t1(b - 1), 2)
        Controls("ListBox" & 12 & b).Clear
        Controls("ListBox" & 12 & b).AddItem _
        Round(t1(b - 1) / t1(0) * 100, 2)

        Controls("ListBox" & 21 & b).Clear
        Controls("ListBox" & 21 & b).AddItem _
        Round(t2(b - 1), 2)
        Controls("ListBox" & 22 & b).Clear
        Controls("ListBox" & 22 & b).AddItem _
        Round(t2(b - 1) / t2(0) * 100, 2)

        Controls("ListBox" & 31 & b).Clear
        Controls("ListBox" & 31 & b).AddItem _
        Round(t3(b - 1), 2)
        Controls("ListBox" & 32 & b).Clear
        Controls("ListBox" & 32 & b).AddItem _
        Round(t3(b - 1) / t3(0) * 100, 2)
    Next
```

```
'Time dependant KPIs
For a = 0 To 18
    kpist(a) = 0
Next
kpist(0) = WorksheetFunction.SumIf(ws.Range(_
"AK" & row1 & ":" & "AK" & row2), ">0") / 6

kpist(1) = WorksheetFunction.SumIf(ws.Range(_
"AL" & row1 & ":" & "AL" & row2), ">0") / 6

kpist(2) = WorksheetFunction.SumIf(ws.Range(_
"AM" & row1 & ":" & "AM" & row2), ">0") / 6

kpist(3) = kpist(0) + kpist(1) + kpist(2)

kpist(4) = WorksheetFunction.SumIf(ws.Range(_
"EE" & row1 & ":" & "EE" & row2), ">0") / 6

kpist(5) = WorksheetFunction.SumIfs(ws.Range(_
"AK" & row1 & ":" & "AK" & row2), _
ws.Range("AK" & row1 & ":" & "AK" & row2), _
">0", ws.Range(_
"EK" & row1 & ":" & "EK" & row2), "ON") / 6

kpist(6) = WorksheetFunction.SumIfs(ws.Range(_
"AL" & row1 & ":" & "AL" & row2), ws.Range(_
"AL" & row1 & ":" & "AL" & row2), ">0", _
ws.Range(_
"EL" & row1 & ":" & "EL" & row2), "ON") / 6

kpist(7) = WorksheetFunction.SumIfs(ws.Range(_
"AM" & row1 & ":" & "AM" & row2), _
ws.Range(_
"AM" & row1 & ":" & "AM" & row2), ">0", _
ws.Range(_
"EM" & row1 & ":" & "EM" & row2), "ON") / 6

If (t1(2) + t2(2) + t3(2)) = 0 Then
    kpist(8) = "/"
Else
    kpist(8) = _
    (kpist(5) + kpist(6) + kpist(7)) / _
    (58 * (t1(2) + t3(2)) + 45 * t2(2)) * 100
End If

kpist(9) = WorksheetFunction.SumIf(ws.Range(_
"DQ" & row1 & ":" & "DQ" & row2), ">0") / 6

kpist(10) = WorksheetFunction.SumIf(ws.Range(_
"DR" & row1 & ":" & "DR" & row2), ">0") / 6

kpist(11) = WorksheetFunction.SumIf(ws.Range(_
"DS" & row1 & ":" & "DS" & row2), ">0") / 6

kpist(12) = kpist(9) + kpist(10) + kpist(11)

kpist(13) = WorksheetFunction.SumIf(ws.Range(_
"DV" & row1 & ":" & "DV" & row2), ">0") / 6
```

```
                    kpist(14) = WorksheetFunction.SumIf(ws.Range(_
                    "DB" & row1 & ":" & "DB" & row2), ">0") / _
                    6 * 44.01 / 22.414

                    kpist(15) = WorksheetFunction.Sum(ws.Range(_
                    "EN" & row1 & ":" & "EN" & row2)) / 6

                    kpist(16) = WorksheetFunction.Sum(ws.Range(_
                    "EO" & row1 & ":" & "EO" & row2)) / 6

                    If kpist(15) = 0 Then
                        kpist(15) = "/"
                        kpist(17) = "/"
                    Else
                        kpist(17) = kpist(15) / _
                        (80 * 250000) * 100
                    End If

                    If kpist(16) = 0 Then
                        kpist(16) = "/"
                        kpist(18) = "/"
                    Else
                        kpist(18) = kpist(16) / _
                        (140 * 250000) * 100
                    End If

                    If t1(2) = 0 Then
                        kpist(5) = "/"
                    Else
                        kpist(5) = kpist(5) / (58 * t1(2)) * 100
                    End If

                    If t2(2) = 0 Then
                        kpist(6) = "/"
                    Else
                        kpist(6) = kpist(6) / (45 * t2(2)) * 100
                    End If

                    If t3(2) = 0 Then
                        kpist(7) = "/"
                    Else
                        kpist(7) = kpist(7) / (58 * t3(2)) * 100
                    End If

                    For c = 1 To 19
                        Controls("ListBoxK" & c).Clear
                        If Not kpist(c - 1) = "/" Then _
                        kpist(c - 1) = Round(kpist(c - 1), 2)
                        Controls("ListBoxK" & c).AddItem _
                        kpist(c - 1)
                    Next

                    For a = 1 To 3
                        Controls("ListBoxMOD" & a).Clear
                        Call oplog(UF_Graph_2, ws, row1, row2, _
                        a, "ListBoxMOD" & a, "ListBoxM")
                    Next
                End If
                j = j + 1
            End If
        Next
```

```
                If ax1 = 0 Or ax2 = 0 Then
                    'All on main axis, one title only
                    Charts(chartn).Axes(xlValue, xlPrimary).HasTitle = _
                    False
                    Charts(chartn).SetElement (_
                    msoElementPrimaryValueAxisTitleAdjacentToAxis)
                    Charts(chartn).Axes(xlValue, xlPrimary).HasTitle = _
                    True
                    Charts(chartn).Axes(xlValue, _
                    xlPrimary).AxisTitle.Text = "AX1"
                    Charts(chartn).Axes(xlValue, xlPrimary) _
                    .AxisTitle.Format.TextFrame2.TextRange.Font.Size = 12
                    Charts(chartn).Axes(xlValue, _
                    xlPrimary).TickLabels.Font.Size = 12
                    j = 1

                    For i = 10 To 17
                        If Not Controls("ComboBox" & i) = "" Then
                            Charts(chartn).SeriesCollection(j)_
                            .AxisGroup = 1
                            uni = Mid(Charts(chartn).SeriesCollection(j)_
                            .Name, InStr(Charts(chartn) _
                            .SeriesCollection(j).Name, "["))
                            If Charts(chartn).Axes(xlValue, xlPrimary)_
                            .AxisTitle.Text = "AX1" Then
                                Charts(chartn).Axes(xlValue, xlPrimary) _
                                .AxisTitle.Text = uni
                            Else
                                If InStr(Charts(chartn).Axes(xlValue, _
                                xlPrimary).AxisTitle.Text, uni) = 0 Then
                                    Charts(chartn).Axes(xlValue, _
                                    xlPrimary).AxisTitle.Text = _
                                    Charts(chartn).Axes(xlValue, _
                                    xlPrimary).AxisTitle.Text & ", " & uni
                                End If
                            End If
                            j = j + 1
                        End If
                    Next
                Else

                    'Move on right axis, then add two titles
                    j = 1
                    For i = 10 To 17
                        If (Not Controls("ComboBox" & i) = "") Then
                            If Controls("OptionButton" & _
                            ((i - 9) * 2)).Value = True Then
                                Charts(chartn).SeriesCollection(j)_
                                .AxisGroup = 1
                            Else
                                Charts(chartn).SeriesCollection(j)_
                                .AxisGroup = 2
                            End If
                            j = j + 1
                        End If
                    Next
                    Charts(chartn).Axes(xlValue, xlPrimary)_
                    .HasTitle = False
                    Charts(chartn).Axes(xlValue, xlSecondary)_
                    .HasTitle = False
                    Charts(chartn).SetElement (_
                    msoElementPrimaryValueAxisTitleAdjacentToAxis)
```

```
                    Charts(chartn).SetElement (_
                    msoElementSecondaryValueAxisTitleAdjacentToAxis)
                    Charts(chartn).Axes(xlValue, xlPrimary)_
                    .HasTitle = True
                    Charts(chartn).Axes(xlValue, xlSecondary)_
                    .HasTitle = True
                    Charts(chartn).Axes(xlValue, xlPrimary)_
                    .AxisTitle.Text = "AX1"
                    Charts(chartn).Axes(xlValue, xlSecondary)_
                    .AxisTitle.Text = "AX2"
                    Charts(chartn).Axes(xlValue, xlPrimary)_
                    .AxisTitle.Format.TextFrame2.TextRange.Font.Size = 12
                    Charts(chartn).Axes(xlValue, xlSecondary)_
                    .AxisTitle.Format.TextFrame2.TextRange.Font.Size = 12
                    Charts(chartn).Axes(xlValue, xlPrimary)_
                    .TickLabels.Font.Size = 12
                    Charts(chartn).Axes(xlValue, xlSecondary)_
                    .TickLabels.Font.Size = 12
                    j = 1

                    For i = 10 To 17
                        If Not Controls("ComboBox" & i) = "" Then
                            uni = Mid(Charts(chartn).SeriesCollection(j)_
                            .Name, InStr(Charts(chartn) _
                            .SeriesCollection(j).Name, "["))
                            If Charts(chartn).SeriesCollection(j)_
                            .AxisGroup = 1 Then
                                If Charts(chartn).Axes(xlValue, _
                                xlPrimary).AxisTitle.Text = "AX1" Then
                                    Charts(chartn).Axes(xlValue, _
                                    xlPrimary).AxisTitle.Text = uni
                                Else
                                  If InStr(Charts(chartn).Axes(xlValue, _
                                  xlPrimary).AxisTitle.Text, uni) = 0 Then
                                        Charts(chartn).Axes(xlValue, _
                                        xlPrimary).AxisTitle.Text = _
                                        Charts(chartn).Axes(xlValue, _
                                        xlPrimary).AxisTitle.Text & _
                                        ", " & uni
                                  End If
                                End If
                            Else
                                If Charts(chartn).Axes(xlValue, _
                                xlSecondary).AxisTitle.Text = "AX2" Then
                                    Charts(chartn).Axes(xlValue, _
                                    xlSecondary).AxisTitle.Text = uni
                                Else
                                    If InStr(Charts(chartn).Axes(_
                                    xlValue, xlSecondary).AxisTitle_
                                    .Text, uni) = 0 Then
                                        Charts(chartn).Axes(xlValue, _
                                        xlSecondary).AxisTitle.Text = _
                                        Charts(chartn).Axes(xlValue, _
                                        xlSecondary).AxisTitle.Text & _
                                        ", " & uni
                                    End If
                                End If
                            End If
                            j = j + 1
                        End If
                    Next
                End If
```

```
'Graphical changes based on X axis: X axis title, series
'type, missing data periods
If Not Controls("ComboBox9").Value = "Time" Then
    Charts(chartn).Axes(xlCategory).HasTitle = True
    Charts(chartn).SetElement (_
    msoElementPrimaryCategoryAxisTitleAdjacentToAxis)
    Charts(chartn).Axes(xlCategory).AxisTitle.Text = _
    Controls("ComboBox9").Value
    For i = 1 To j - 1
        Charts(chartn).SeriesCollection(i).ChartType = _
        xlXYScatter
        Charts(chartn).SeriesCollection(i).MarkerSize = 3
    Next
Else
    Charts(chartn).Axes(xlCategory).HasTitle = False
    For i = 1 To j - 1
        Charts(chartn).SeriesCollection(i).ChartType = _
        xlLine
        Charts(chartn).SeriesCollection(i) _
        .Format.Line.Weight = 1.5
    Next

    'Add empty data series for each missing data period
    Charts(chartn).Axes(xlValue, xlPrimary)_
    .MaximumScaleIsAuto = True
    Charts(chartn).Axes(xlValue, xlPrimary)_
    .MinimumScaleIsAuto = True
    maxx = Charts(chartn).Axes(xlValue, xlPrimary)_
    .MaximumScale
    minx = Charts(chartn).Axes(xlValue, xlPrimary)_
    .MinimumScale
    If Abs(maxx) > Abs(minx) Then
        avex = maxx
    Else
        avex = minx
    End If
    sizx = rng2.Count
    counter = 0

    For Each rcell In rng3
        counter = counter + 1
        If rcell.Value > 0.17 And Not counter = 1 Then
            For c = 1 To j - 1
                Charts(chartn).SeriesCollection(c)_
                .Points(counter).Format.Line.Visible = _
                msoFalse
            Next

            If rcell.Value > 1 Then
                Set ns = Charts(chartn)_
                .SeriesCollection.NewSeries
                With Workbooks(ToolkitName)_
                .Worksheets("-INFO-")
                ns.Values = Union(.Range(.Cells(1, 1), _
                .Cells(counter - 1, 1)), .Range("E49"), _
                .Range(.Cells(counter + 1, 1), _
                Cells(sizx, 1)))
                End With
                ns.XValues = rng2
                ns.ChartType = xlXYScatter
                ns.ErrorBar Direction:=xlY, _
```

```
                        Include:=xlBoth, Type:=xlFixedValue, _
                        Amount:=Abs(avex)
                        ns.ErrorBars.EndStyle = xlNoCap
                        ns.MarkerStyle = -4115
                        ns.ErrorBars.Format.Line.DashStyle = _
                        msoLineSysDash
                        ns.ErrorBars.Format.Line.Weight = 1.5
                        ns.ErrorBars.Format.Line.ForeColor.RGB = _
                        RGB(0, 0, 0)
                        ns.Format.Fill.ForeColor.RGB = _
                        RGB(0, 0, 0)
                        ns.Format.Fill.Solid
                        ns.Format.Line.ForeColor.RGB = _
                        RGB(0, 0, 0)

                        ns.Name = "Missing: " & _
                        Mid(ws.Range(_
                        "A" & rcell.Row - 1), 1, 5) & _
                        Mid(ws.Range(
                        "A" & rcell.Row - 1), 11, 6) & _
                        Mid(ws.Range(_
                        "A" & rcell.Row - 1), 20, 4) & " - " & _
                        Mid(ws.Range("A" & rcell.Row), 1, 5) & _
                        Mid(ws.Range("A" & rcell.Row), 11, 6) & _
                        Mid(ws.Range("A" & rcell.Row), 20, 4)

                        ns.AxisGroup = 1
                    End If
                End If
            Next

            If sizx > 12 Then
                Charts(chartn).Axes(xlCategory).TickLabelSpacing _
                = WorksheetFunction.RoundDown(sizx / 12, 0)
                Charts(chartn).Axes(xlCategory).TickMarkSpacing _
                = WorksheetFunction.RoundDown(sizx / 12, 0)
            End If
            Charts(chartn).Axes(xlCategory).TickLabels_
            .Orientation = xlHorizontal
            Charts(chartn).Axes(xlCategory).TickLabels_
            .Orientation = -45

            'Restore axis
            Charts(chartn).Axes(xlValue, 1).MaximumScale = maxx
            Charts(chartn).Axes(xlValue, 1).MinimumScale = minx
        End If
```

```
                    'Common graphical design
                    Charts(chartn).SetElement (msoElementChartTitleAboveChart)
                    Charts(chartn).ChartTitle.Text = "Data from " & _
                    ws.Range("A" & row1) & " to " & ws.Range("A" & row2)
                    Charts(chartn).ChartTitle.Format.TextFrame2_
                    .TextRange.Font.Size = 14
                    Charts(chartn).SetElement (msoElementLegendBottom)
                    Charts(chartn).Legend.Format.TextFrame2_
                    .TextRange.Font.Size = 12
                    Charts(chartn).Axes(xlCategory).TickLabelPosition = xlLow
                    Charts(chartn).Axes(xlCategory).TickLabels.Font.Size = 12
                    Charts(chartn).SetElement (_
                    msoElementPrimaryCategoryGridLinesMajor)

                    ActiveWindow.Zoom = 100
                    Application.ScreenUpdating = True
                End If
            End If

        End If
    End If
End If
End Sub
```

8.3.5. Operation log

```
Private Sub ComboBox1_Change()
Dim ToolkitName As String
ToolkitName = "DEMOSOFC - Data analsysis toolkit.xlsm"
Dim ws, wsinf As Worksheet
Set wsinf = Workbooks(ToolkitName).Worksheets("-INFO-")
Dim a, b, c, x, dt, t, lastrow, y As Long
Dim str, t1, t2 As String
Dim v(13, 4) As Variant

ListBox1.Clear
ListBoxM.Clear
UF_OpLog.MultiPage1.Pages(5).Enabled = True
If ComboBox1.Value = "Lifetime" Then

    'Clear
    For a = 1 To 3
        Controls("ListBoxOP" & a).Clear
        For b = 2 To 9
            Controls("ListBox" & a & 1 & b).Clear
            Controls("ListBox" & a & 2 & b).Clear
        Next
        For b = 1 To 9
            Controls("ListBoxK" & a & b).Clear
        Next
        For b = 1 To 3
            Controls("ListBoxKO" & a & b).Clear
        Next
    Next
    For b = 1 To 12
        Controls("ListBoxP" & b).Clear
    Next
    For a = 1 To 4
        For b = 1 To 13
            v(b, a) = 0
        Next
    Next

    'Evaluate
    x = 0
    Do While (Not IsEmpty(wsinf.Cells(1, 6 + x * 6)))
        y = 2018 + x
        Set ws = Workbooks(ToolkitName).Worksheets(y & " - Analogic")
        For a = 1 To 3
            col = 5 + a + x * 6
            'Time
            For b = 1 To 7
                v(b, a) = v(b, a) + wsinf.Cells(b + 5, col)
            Next

            'KPIs
            For b = 8 To 13
                If b = 8 Or b = 10 Then
                    v(b, a) = v(b, a) + wsinf.Cells(b + 21, col)
                Else
                    v(b, a) = v(b, a) + wsinf.Cells(b + 21, col) * _
                    wsinf.Cells(7, col)
                End If
            Next
            Call oplog(UF_OpLog, ws, 3, ws.Range("A1").End(xlDown).Row, _
            a, "ListBoxOP" & a, "ListBoxM")
        Next
```

```
    'Plant
    For b = 1 To 12
        If b = 3 Then
            v(b, 4) = v(b, 4) + wsinf.Cells(b + 36, 7 + x * 6) * _
            ((58 * wsinf.Cells(7, 6 + x * 6)) + _
            (45 * wsinf.Cells(7, 7 + x * 6)) + _
            (58 * wsinf.Cells(7, 8 + x * 6)))
        Else
            v(b, 4) = v(b, 4) + wsinf.Cells(b + 36, 7 + x * 6)
        End If
    Next
    x = x + 1
Loop
t1 = _
Workbooks(ToolkitName).Worksheets("2018 - Analogic").Range("A2").Value
t2 = ws.Range("A1").End(xlDown).Value
ListBox1.AddItem "From " & t1 & " to " & t2
dt = deltat(t2, t1)
y = dt
For b = 1 To 7
    y = y - v(b, 1)
Next

'Assign
For a = 1 To 3
    'Time
    For b = 2 To 8
        Controls("ListBox" & a & 1 & b).AddItem Round(v(b - 1, a), 2)
        Controls("ListBox" & a & 2 & b).AddItem _
        Round(v(b - 1, a) / dt * 100, 2)
    Next
    Controls("ListBox" & a & 19).AddItem Round(y, 2)
    Controls("ListBox" & a & 29).AddItem Round(y / dt * 100, 2)

    'KPIs
    For b = 1 To 3
        If b = 2 Then
            If Not v(2, a) = 0 Then
                Controls("ListBoxK" & a & b).AddItem _
                Round(v(b + 7, a) / v(2, a), 2)
            Else
                Controls("ListBoxK" & a & b).AddItem 0
            End If
        Else
            Controls("ListBoxK" & a & b).AddItem Round(v(b + 7, a), 2)
        End If
        If Not v(2, a) = 0 Then
            Controls("ListBoxKO" & a & b).AddItem _
            Round(v(b + 10, a) / v(2, a), 2)
        Else
            Controls("ListBoxKO" & a & b).AddItem 0
        End If
    Next
Next
```

```
    'Plant
    For b = 1 To 12
        If b = 3 Then
            Controls("ListBoxP" & b).AddItem _
            Round(v(b, 4) / ((58 * v(2, 1)) + (45 * v(2, 2)) + _
            (58 * v(2, 3))), 2)
        Else
            Controls("ListBoxP" & b).AddItem Round(v(b, 4), 2)
        End If
    Next


Else

    Set ws = Workbooks(ToolkitName).Worksheets(_
    ComboBox1.Value & " - Analogic")
    t1 = ws.Range("A2").Value
    t2 = ws.Range("A1").End(xlDown).Value
    ListBox1.AddItem "From " & t1 & " to " & t2
    x = ComboBox1.Value - 2018
    For a = 1 To 3
        col = 5 + a + x * 6
        'Time
        For b = 2 To 9
            Controls("ListBox" & a & 1 & b).Clear
            Controls("ListBox" & a & 1 & b).AddItem _
            Round(wsinf.Cells(b + 4, col).Value, 2)
            Controls("ListBox" & a & 2 & b).Clear
            Controls("ListBox" & a & 2 & b).AddItem _
            Round(wsinf.Cells(b + 16, col).Value, 2)
        Next

        'KPIs
        For b = 1 To 3
            Controls("ListBoxK" & a & b).Clear
            Controls("ListBoxK" & a & b).AddItem _
            Round(wsinf.Cells(b + 28, col).Value, 2)
            Controls("ListBoxKO" & a & b).Clear
            Controls("ListBoxKO" & a & b).AddItem _
            Round(wsinf.Cells(b + 31, col).Value, 2)
        Next
        For b = 4 To 9
            Controls("ListBoxK" & a & b).Clear
        Next

        Controls("ListBoxOP" & a).Clear
        Call oplog(UF_OpLog, ws, 3, ws.Range("A1").End(xlDown).Row, _
        a, "ListBoxOP" & a, "ListBoxM")
    Next

    'Plant
    For b = 1 To 12
        Controls("ListBoxP" & b).Clear
        Controls("ListBoxP" & b).AddItem _
        Round(wsinf.Cells(b + 36, 7 + x * 6), 2)
    Next

End If

End Sub
```

```
Sub ON_details(a As Long)
Dim ToolkitName As String
ToolkitName = "DEMOSOFC - Data analsysis toolkit.xlsm"
Dim ws As Worksheet
Dim rng1, rng2 As Range
For i = 0 To Controls("ListBoxOP" & a).ListCount - 1
    If Controls("ListBoxOP" & a).Selected(i) And _
    Mid(Controls("ListBoxOP" & a).list(i), 1, 2) = "ON" Then
        t1 = Mid(Controls("ListBoxOP" & a).list(i), _
        InStr(Controls("ListBoxOP" & a).list(i), "From ") + 5, 19)
        t2 = Mid(Controls("ListBoxOP" & a).list(i), _
        InStr(Controls("ListBoxOP" & a).list(i), "to ") + 3, 19)
        dt = deltat(t2, t1)
        y = Mid(t2, 7, 4)
        Set ws = Workbooks(ToolkitName).Worksheets(y & " - Analogic")
        Set rng1 = ws.Range("A:A").Find(t1, , xlValues, xlWhole)
        Set rng2 = ws.Range("A:A").Find(t2, , xlValues, xlWhole)
        If ws.Range("DA" & rng1.Row) < 0.17 And Not rng1.Row = 2 Then
            Set rng1 = ws.Range("A" & rng1.Row + 1)
        End If
        Controls("ListBoxK" & a & 4).AddItem _
        Round(WorksheetFunction.Average(_
        ws.Range(ws.Cells(rng1.Row, a + 36), _
        ws.Cells(rng2.Row, a + 36))), 2)

        Controls("ListBoxK" & a & 5).AddItem _
        Round(WorksheetFunction.Average(_
        ws.Range(ws.Cells(rng1.Row, a + 120), _
        ws.Cells(rng2.Row, a + 120))), 2)

        Controls("ListBoxK" & a & 6).AddItem _
        Round(WorksheetFunction.Average(_
        ws.Range(ws.Cells(rng1.Row, a + 117), _
        ws.Cells(rng2.Row, a + 117))), 2)

        Controls("ListBoxK" & a & 7).AddItem _
        Round(WorksheetFunction.Average(_
        ws.Range(ws.Cells(rng1.Row, a + 127), _
        ws.Cells(rng2.Row, a + 127))), 2)

        Controls("ListBoxK" & a & 8).AddItem _
        Round(WorksheetFunction.Average(_
        ws.Range(ws.Cells(rng1.Row, a + 130), _
        ws.Cells(rng2.Row, a + 130))), 2)
        Call degradation(a, rng1, rng2, dt, y)

        Exit For
    Else
        If Not Controls("ListBoxK" & a & 4).ListCount = 0 Then
            For b = 4 To 9
                Controls("ListBoxK" & a & b).Clear
            Next
        End If
    End If
Next
End Sub
```

```
Private Sub CommandButton1_Click()
x = TextBoxR.Text
If x = "" Then
    MsgBox "Please insert a name for the Word report.", _
    vbOKOnly, "Invalid name!"
Else
If InStr(x, "\") = 0 And InStr(x, "/") = 0 And InStr(x, "?") = 0 And _
InStr(x, "*") = 0 And InStr(x, ":") = 0 And InStr(x, ".") = 0 And _
InStr(x, ">") = 0 And InStr(x, "<") = 0 And InStr(x, "|") = 0 Then

        'Export overview data
        ChDrive ActiveWorkbook.Path
        ChDir ActiveWorkbook.Path

        Dim objWord As Object
        Dim wdname As String
        Set objWord = CreateObject("Word.Application")
        objWord.Visible = True
        wdname = ActiveWorkbook.Path & "\DEMOSOFC_ReportDefault2.docx"
        Set wd = objWord.Documents.Open(wdname)

        'Time period
        t = Controls("ListBox1").list(0)
        Call findandreplace(t, wd)

        'Elapsed time
        t1 = Mid(t, InStr(t, "From ") + 5, 19)
        t2 = Mid(t, InStr(t, "to ") + 3, 19)
        dt = deltat(t2, t1)
        Call findandreplace(Round(dt, 2) & " [h]", wd)

        'Missing time periods
        If UF_OpLog.ListBoxM.ListCount = 0 Then
            Call findandreplace("None.", wd)
        Else
            Call findandreplace(UF_OpLog.ListBoxM.list(0) & Chr(11), wd)
            For i = 1 To UF_OpLog.ListBoxM.ListCount - 1
                wd.Application.Selection = UF_OpLog.ListBoxM.list(i) & Chr(11)
                wd.Application.Selection.Collapse wdCollapseEnd
            Next
        End If

        'Modules
        For a = 1 To 3
            For i = 1 To 3
                Call findandreplace(Controls("ListBoxK" & a & i).list(0), wd)
            Next
            For i = 1 To 3
                Call findandreplace(Controls("ListBoxKO" & a & i).list(0), wd)
            Next

            For i = 2 To 9
                Call findandreplace(Controls(_
                "ListBox" & a & 1 & i).list(0), wd)
                Call findandreplace(Controls(_
                "ListBox" & a & 2 & i).list(0), wd)
            Next

            Call findandreplace(Controls(_
            "ListBoxOP" & a).list(0) & Chr(11), wd)
```

```
            For i = 1 To Controls("ListBoxOP" & a).ListCount - 1
                wd.Application.Selection = Controls(_
                "ListBoxOP" & a).list(i) & Chr(11)
                wd.Application.Selection.Collapse wdCollapseEnd
            Next
        Next

        'Plant KPIs
        For i = 11 To 12
            Call findandreplace(Controls("ListBoxP" & i).list(0), wd)
        Next
        For i = 1 To 10
            Call findandreplace(Controls("ListBoxP" & i).list(0), wd)
        Next

        'Pie charts
        a = 1
        For Each shp In wd.InlineShapes
          If shp.HasChart Then
            Set ws = shp.Chart.ChartData.Workbook.Sheets(1)
            For i = 2 To 9
                ws.Cells(i, 2).Value = Val(Controls(_
                "ListBox" & a & 1 & i).list(0))
            Next
            shp.Chart.SetSourceData Source:="='Foglio1'!$A$1:$B$9"
            a = a + 1
          End If
        Next

        wd.SaveAs (ActiveWorkbook.Path & "\" & x & ".docx")
        wd.Close
        objWord.Quit
        MsgBox "Export completed.", vbInformation, "Done."
    Else
        MsgBox "Special characters as \, /, ?, *, :, |, ., <, >  can" & _
        " not be used.", vbOKOnly, "Invalid name!"
    End If
End If
End Sub
```

```
Sub degradation(a, rng1, rng2, dt, year)
Dim ToolkitName As String
ToolkitName = "DEMOSOFC - Data analsysis toolkit.xlsm"
Set ws = Workbooks(ToolkitName).Worksheets(year & " - Analogic")
Dim x, y, d As Variant
Dim m, b As Long
y = Application.Transpose(ws.Range(ws.Cells(rng1.Row, a + 36),
ws.Cells(rng2.Row, a + 36)))
ReDim x(1 To UBound(y))
For k = 1 To UBound(y)
    x(k) = k
Next
m = WorksheetFunction.slope(y, x)
d = (m * (1 - x(UBound(x))) * 100 * 1000) / (WorksheetFunction.Max(y) * dt)
If d < 0 Then d = 0
Controls("ListBoxK" & a & 9).AddItem d
End Sub
```

8.3.6. General

```
Sub oplog(uf, ws, r1, r2, a, listb, listm)
'Operation log
With uf
For c = r1 To r2
    s = ws.Cells(c, 140 + a).Value
    If ws.Range("DA" & c) > 0.17 Then
        t1 = ws.Range("A" & c)
    Else
        t1 = ws.Range("A" & c - 1)
    End If
    c = c + 1
    Do While ws.Range("DA" & c) < 0.17 And ws.Cells(c, 140 + a).Value = s _
    And c <= r2
        c = c + 1
    Loop
    t2 = ws.Range("A" & c - 1)
    dt = deltat(t2, t1)
    If dt < 1 Then
        .Controls(listb).AddItem (s & " | " & Round(dt * 60) & _
        " [min] | From " & t1 & " to " & t2)
    Else
        .Controls(listb).AddItem (s & " | " & Round(dt) & _
        " [h] | From " & t1 & " to " & t2)
    End If
    If ws.Range("DA" & c) > 0.17 And a = 1 Then
        'Missing data
        t1 = ws.Range("A" & c - 1)
        t2 = ws.Range("A" & c)
        dt = ws.Range("DA" & c) - 1 / 6
        If dt < 1 Then
            .Controls(listm).AddItem ("MISS | " & Round(dt * 60) & _
            " [min] | From " & t1 & " to " & t2)
        Else
            .Controls(listm).AddItem ("MISS | " & Round(dt) & _
            " [h] | From " & t1 & " to " & t2)
        End If
    End If
    c = c - 1
Next
End With
End Sub
```

```
Public Sub CloseButtonSettings(frm As Object, show As Boolean)

    Dim windowHandle As Long
    Dim menuHandle As Long
    windowHandle = FindWindowA(vbNullString, frm.Caption)

    If show = True Then

        menuHandle = GetSystemMenu(windowHandle, 1)
    Else
        menuHandle = GetSystemMenu(windowHandle, 0)
        DeleteMenu menuHandle, SC_CLOSE, 0&
    End If
End Sub
```

```
Public Sub CleanLog(del As String, sh As Worksheet)
Dim rng As Range
With sh
    Set rng = .Range("A:A").Find(del, , xlValues, xlWhole)
    Do Until rng Is Nothing
    rng.EntireRow.Delete
    Set rng = .Range("A:A").Find(del, , xlValues, xlWhole)
    Loop
End With
End Sub
```

```
Public Function deltat(rng1, rng2)
deltat = 24 * _
(DateValue(Mid(rng1, 1, 2) & "/" & Mid(rng1, 4, 2) & "/" & Mid(rng1, 7, 4)) _
- DateValue(Mid(rng2, 1, 2) & "/" & Mid(rng2, 4, 2) & "/" & Mid(rng2, 7, 4)) _
+ TimeValue(Mid(rng1, 12, 8)) - TimeValue(Mid(rng2, 12, 8)))
End Function
```

```
Sub findandreplace(t, wd)
wd.Application.Selection.Find.Text = "XYZ"
wd.Application.Selection.Find.Execute
wd.Application.Selection = t
wd.Application.Selection.Collapse wdCollapseEnd
End Sub
```