



Prescient building Operation utilizing Real Time data for Energy Dynamic Optimization

WP3 – Interoperable dynamic module integration in multi-simulation dataspace

D3.2 – Dynamic FRM module Version 2

Issue date:	31/05/2022 (29/09/2023 updated after RP2 review))
Author(s):	Giacomo Chiesa, Francesca Fasano, Paolo Grasso (Politecnico di Torino)
Editor:	Giacomo Chiesa (Politecnico di Torino)
Lead Beneficiary:	Partner 04 – Politecnico di Torino
Dissemination level:	Public
Type:	other
Reviewers:	Peter Klanatsky (FB), Hicham Johra (AAU)



EXPLANATIONS FOR FRONT PAGE

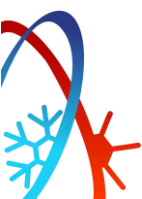
Author(s): Name(s) of the person(s) having generated the Foreground respectively having written the content of the report/document. In case the report is a summary of Foreground generated by other individuals, the latter have to be indicated by name and partner whose employees he/she is. List them alphabetically.

Editor: Only one. As formal editorial name only one main author as responsible quality manager in case of written reports: Name the person and the name of the partner whose employee the Editor is. For the avoidance of doubt, editing only does not qualify for generating Foreground; however, an individual may be an Author – if he has generated the Foreground – as well as an Editor – if he also edits the report on its own Foreground.

Lead Beneficiary of Deliverable: Only one. Identifies name of the partner that is responsible for the Deliverable according to the PRELUDE Description of Work (DOW). The lead beneficiary partner should be listed on the front page as Authors and Partner. If not, that would require an explanation.

Internal Reviewers: They should not belong to the authors. They should be any employees of the remaining partners of the consortium, not directly involved in that deliverable, but should be competent in reviewing the content of the deliverable. Typically, this review includes identifying typos, identifying syntax and other grammatical errors, altering content, adding or deleting content.

Task 3.2 also includes works by BUL, AAU, CORE, STAM, LIBRA, TREE, USMART (UNIPV) whose results are described in other deliverables being interconnected with other tasks.



PRELUDE KEY FACTS

Project Title	Prescient building Operation utilizing Real Time data for Energy Dynamic Optimization
Starting date	01/12/2020
Duration in months	42
Call (part) identifier	H2020-NMBP-ST-IND-2020-singlestage
Topic	LC-EEB-07-2020 Smart Operation of Proactive Residential Buildings (IA)
Fixed EC Keywords	-
Free Keywords	Free running, model based predicted control, dynamic building simulation, demand side flexibility, proactive buildings, predictive maintenance, occupancy models, smartness assessment
Consortium	21 organisations

PRELUDE CONSORTIUM PARTNERS

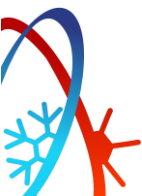
	Participant organisation name	Country
1	AALBORG UNIVERSITET	DK
2	TAMPEREEN KORKEAKOULUSAATIO SR	FI
3	ASOCIACIÓN DE INVESTIGACIÓN METALÚRGICA DEL NOROESTE	ES
4	POLITECNICO DI TORINO	IT
5	FORSCHUNG BURGENLAND GMBH	AT
6	UNISMAST - FONDAZIONE UNIVERSITÀ DEGLI STUDI DI PADOVA	IT
7	BRUNEL UNIVERSITY LONDON	UK
8	EMTECH DIASTIMIKI MONOPROSOPI IDIOTIKI ETAIREIA	EL
9	CORE INNOVATION AND TECHNOLOGY OE	EL
10	ESTIA SA	CH
11	EUROCORE CONSULTING	BE
12	IREN SMART SOLUTIONS SPA	IT
13	LIBRA AI TECHNOLOGIES PRIVATE IDIOTIKI KEFALAIOUCHIKI ETAIREIA	EL
14	STAM SRL	IT
15	LA SIA SRL	IT
16	TREE TECHNOLOGY SA	ES
17	1A INGENIEROS S.L.P	ES
18	DIMOS ATHINAION EPICHEIRISI MICHANOGRAFISIS	EL
19	BLOK ARCHITEKCI SPOLKA Z OGRANICZONA ODPOWIEDZIALNOSCIA	PL
20	CAISSE DE PREVOYANCE DE L'ETAT DE GENEVE	CH
21	INNOVACION Y CONSULTING TECNOLOGICOSL	ES

DISCLAIMER

Copyright © 2020 – 2024 by PRELUDE consortium

Use of any knowledge, information or data contained in this document shall be at the user's sole risk. Neither the PRELUDE Consortium nor any of its members, their officers, employees or agents shall be liable or responsible, in negligence or otherwise, for any loss, damage or expense whatever sustained by any person as a result of the use, in any manner or form, of any knowledge, information or data contained in this document, or due to any inaccuracy, omission or error therein contained. If you notice information in this publication that you believe should be corrected or updated, please get in contact with the project coordinator.

The authors intended not to use any copyrighted material for the publication or, if not possible, to indicate the copyright of the respective object. The copyright for any material created by the authors is reserved. Any duplication or use of objects such as diagrams, sounds or texts in other electronic or printed publications is not permitted without the author's agreement.



EXECUTIVE SUMMARY

PRELUDE Task T3.2 – *Dynamic and free running building energy simulation* is part of WP3 – *Interoperable dynamic module integration in multisimulation dataspace* and it is devoted to support the definition of a methodology to calculate building climate-related KPIs (Key Performance Indicators) and the local free-running potential. Additionally, the task includes the development of an approach to compare real building behaviours and model building behaviours under the same weather conditions. An approach to suggest optimized free-running usage (shading and ventilation for cooling) is also defined. All these aspects are taken into account for the development of a new python library able to manage dynamic building simulations via EnergyPlus. The new tool is also able to manage other type of data, like monitored data. The library, which is under development at POLITO, is based on two developing actions: the 'PRE' action, which is funded by PRELUDE and whose main usages are described in this Deliverable, and the 'DYCE' action that received funds under another project and that is mainly focussed on defining the initial tool architecture and basic functionalities for standardised building analyses.

The library is structured to support automatic changes in simulation inputs, managing multiple simulations (which can be run on a devoted server under request) and calculation of specific outputs and KPIs in line with PRELUDE objectives and following project tasks. The tool also includes the possibility to perform climate analyses to define the local potentials of free-running technologies. Additionally, different pre-defined scenarios of usage are also provided, including the development of a new PRELUDE-devoted scenario able to support a 24h forecasting of optimised shading and ventilation strategies for maximising thermal comfort during the next day by means of self-user actuating and/or automatic control systems. Scenarios may be run via REST (REpresentational State Transfer), allowing server-to-server communication. The simulation tool may thus be managed at the consortium level for specific scopes via the project middleware.

The current deliverable describes the main tool functionalities focussing on the 'PRE' developing action. Among these main functionalities, it is possible to mention the ability to support climate analyses to verify the local potential of different free-running and low-energy technologies (e.g., ventilative cooling, direct evaporative cooling, ground cooling) and to include the same technologies in the EnergyPlus simulations to support the design and operational suggestions. Additionally, the tool can include, under request, EMS (Energy Management System) functionalities, and to run a new scenario of usage of the proposed tool named "24h forecasting" to support shading and ventilation optimisation during the next day for exploiting potential for free-cooling. Additionally, comparisons between monitored and simulated data are available and may be used to verify the adoption of 24h forecasting suggestions. Finally, the deliverable includes some examples of applications to give consistency to the technical description of the tool development action.

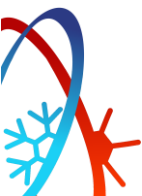
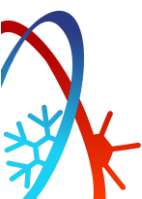


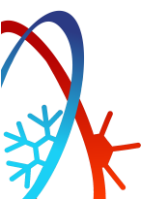
TABLE OF CONTENTS

EXPLANATIONS FOR FRONT PAGE	2
PRELUDE KEY FACTS.....	3
PRELUDE CONSORTIUM PARTNERS.....	3
EXECUTIVE SUMMARY	4
TABLE OF CONTENTS.....	5
ABBREVIATIONS.....	6
1. INTRODUCTION.....	7
1.1 Background.....	7
1.2 Task objectives	9
2. PREDYCE – Dynamic simulation platform.....	10
2.1 PREDYCE general architecture	10
2.2 The ‘PRE’ development action	15
2.3 Sensitivity analysis scenario for free-running	23
2.3.I Climate analyses	23
2.3.II Building analyses	33
2.4 24h forecasting scenario	38
2.5 Verification scenario to support pre-calibration actions.....	40
2.6 Real vs simulated KPIs for free-running	41
2.7 Weather input coding	43
3. Application samples	45
3.1 Climate, bioclimatic and free-running related KPIs.....	45
3.2 Testing free-running control solutions – shading system	47
3.3 Suggestions for optimising free-running usage.....	47
3.4 Verification scenario.....	50
4. CONCLUSIONS.....	53
APPENDIX A.....	55
BIBLIOGRAPHY.....	55



ABBREVIATIONS

ACH	Air Change per Hour (Air Change Rate)
ACM	Adaptive Comfort Model
CDD	Cooling Degree Day
CDH	Cooling Degree Hour
CCP	Climate Cooling Potential
CDD _{res}	residual CDD
CDH _{res}	residual CDH
CIDH	Cooling Internal Degree hour
COP	Coefficient of Performance
DEC	Direct evaporative cooling
EAHX	Earth-to-air heat exchanger
EAHE	equal to EAHX
EER	Energy Efficiency Ratio
EPBD	Energy Performance of Buildings Directive
FR	Free Running
GUI	Graphical User Interface
HDD	Heating Degree Day
HDH	Heating Degree Hour
HDD _{res}	residual HDD
HDH _{res}	residual HDH
HIDH	Heating Internal Degree Hour
ICT	Information and Communication Technologies
KPI	Key Performance Indicator
PDEC	Passive draught evaporative cooling/Passive direct evaporative cooling
PMV	Predicted Mean Vote
PPD	Predicted Percentage of Dissatisfied
TMY	Typical Meteorological Year



1. INTRODUCTION

1.1 Background

Buildings are responsible for 40% of EU energy consumption and 36% of greenhouse gas emissions related to energy. They thus are the largest consumer of energy in the EU [1]. About 80% of this energy use is dedicated to space heating, cooling and domestic hot water production. 35% of the European building stock is composed of buildings built before the 70's and about 75% of buildings in the EU are energy inefficient [2]. These issues underline how the building sector is a crucial point in the EU's directives and energy and environmental policies to support European objectives. The EPBD (Energy Performance of Buildings Directive) is at the base of the EU's legislative framework in the energy and building context together with the Energy Efficiency Directive. The importance of these policies, which are periodically upgraded and revised, is also underlined by the European Green Deal supporting a new revision proposal for EPBD in 2021 [3]. Among innovative proposed aspects, it is possible to mention the interest to support a higher modernisation and renovation rate of the building stock to increase resilience and accessibility together with minimum energy performance standards. Additionally, the proposed amendment will also support a larger interest in additional aspects such as the digitalisation of energy systems and buildings and the improvement of indoor air quality in buildings. Finally, the proposed revision for the EPBD supports sustainable mobility choices and facilitates specific financial investments in buildings to fight energy poverty in line with the UN's development goals.

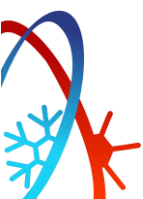
Additionally, it is also possible to mention two other great challenges: i. the spread of IT (Information Technologies) and ICT (Information and Communication Technologies) in the building sector with correlated potentialities, and ii. the need to face climate changes and additional microclimate impacts affecting buildings and energy needs, developing a new resilient dimension in the performance-based design and operational approaches. IT and ICT support the possibility to drastically increase the knowledge of current, past and presciently future building conditions supporting a smart and efficient control of building operational choices and potentially design definitions for renovations. Similarly, they also support the possibility to develop new intercorrelated platforms able to manage real and simulated data suggesting new scenarios for building optimisations. Typical approaches to building dynamic simulations, based on manual analyses, may be supported by new interfaces in which simulation inputs are automatically managed supporting massive parametric and optimisation studies using white, grey or black box modelling techniques. In this task, a new simulation platform is introduced supporting mainly white box modelling, even if the basic approach may be translated into future developments to feed surrogate models or other techniques. Concerning the climate dimension, it starts to be recognized in recent years that traditional climate analyses to support building design simulations and studies and regulations are based on not-up-to-date typical weather files not considering the dynamic dimension of climate conditions due to underlined changes. The approach proposed in this task is open to different climate data sources, including monitored weather conditions, even in real-time, forecasted weather, and other data sources, like the ones that will be elaborated for building purposes in T8.5 starting from the reanalysis databases of Copernicus and data based on future weather scenarios.

For PRELUDE application, it is possible to define a free-running (FR) building as a building, or building zone, that works, in a given time period, without system activation. This condition can be reached in both buildings in which no mechanical systems are installed (for example, traditional residential buildings without cooling system) or in which they are turned off. Therefore, FR buildings will be analysed mainly for comfort/discomfort purposes, although also the impact of FR technologies in mitigating the heating, and/or cooling, and/or ventilation energy needs are also studied in PRELUDE. The latter include the analysis not only of free-running buildings, but also hybrid usages, in which mechanical systems (when present) are activated only in specific conditions for which FR technologies alone are not able to cover cooling or

heating demand peaks. For the purpose of PRELUDE, we may define FR technologies as technologies able to allow the usages of natural and passive heating and cooling sources. Focussing on the heating point of view, this is mainly correlated to heat gain collection and heat loss reduction supporting, for example, the adoption of sunspaces and/or the choice of a proper insulation layer. From the cooling point of view, although, technologies and strategies for both thermal control and dissipation are considered, in particular, heat gain prevention technologies (e.g., shading devices), heat gain modulation solutions (e.g., thermal masses) and heat gain dissipation technologies and strategies (e.g., passive/hybrid cooling based on heat sinks).

In addition to the need to reduce energy use in the building sector by increasing the efficiency of both envelopes and systems, renewable and alternative solutions must be emphasized. Focussing on the latter, it can be mentioned that, in current regulations, the possibility to valorise passive and natural solutions is still very limited with the risk to not incentivise the adoption of bioclimatic and free-running technologies¹ for climates in which they may be sufficient to cover the majority of the heating and cooling needs [8,9]. This is particularly true for cooling solutions supporting mainly solutions based on high discrepancies between indoor and outdoor environments with high insulation and airtightness. Nevertheless, the majority of traditional buildings, even in the Mediterranean and hot climates, does not have a mechanical cooling system, but inhabitants have thermal comfort conditions supporting free-running building management choices and taking advantages of low-energy technologies and indoor-outdoor adaptations and together with activation of personal devices during hottest hours. The possibility to support, especially for the summer season, these free-running solutions devoted to overheating prevention, heat gain mitigation and heat gain dissipation – see also [10–13] – is also part of the topics treated in this task. It focusses, for example, on optimising shading and ventilation for cooling strategies when they are able to positively impact thermal comfort. The attention to the summer season is correlated to three main aspects. Firstly, building cooling needs and correlated consumptions are growing fast due to climate changes and urban heat island phenomena, the international style of building that lacks in bio-regionalism – see a detailed description in [11,14] –, the changes in the culture of comfort and the increase of living standards, and the growth of internal gains – see [14]. This phenomenon is supported by the constant global growth in the number of installed air-conditioning units [15]. Regarding the climate, the growth in cooling indicators, such as the cooling degree days, at average European level is underlined in several studies [16,17], supporting the analyses that underlined how global consumption for cooling reached 1.25 PWh in 2010 [18] and is expected to overpass global heating needs for building space treatments in Europe between 2050-2100 [19,20]. The possibility to valorise and optimise all bioclimatic and free-running potentials to reduce energy needs is hence essential and an urgent topic. Secondly, the number of bioclimatic and free-running technologies able to be applied to the cooling season is considerably larger in respect to the winter ones in which solar gains and, eventually, ground-source heating are the dominant passive sources. Free-running cooling solutions are not only based on thermal control considering the prevention of solar and internal gains, and their modulation, e.g. by using thermal masses, but also on dissipating heat gains to reduce internal temperatures, e.g. by thermal sinks (air, water, ground, sky), supporting natural and hybrid cooling solutions. Thirdly, natural and hybrid low-energy cooling solutions have very climate-specific applicability, meaning that, for example, the direct evaporative cooling local potential may vary between locations, even in the same region or at the same latitude since it is dependent of the local wet bulb. The variation in local applicability is a peculiarity of several cooling systems and differ in respect to solar-driven

¹ For a definition of Bioclimatic architecture please refer to the following references [4–7]. Differently, we consider free-running technologies, those technologies that are able to increase comfort conditions in a building in which mechanical (heating/cooling) systems are not installed or are turned off – see also the definition of free-running (FR) building for PRELUDE given above.



passive heating techniques that are more correlated to the distribution of irradiations following a clearer geographical distribution.

This deliverable supports the PRELUDE free-running approach to exploit local potentials of low-energy technologies, including passive cooling sources. The work presented here is based on the development of a new python tool to support simulations and will be further developed during next project steps in WP3 and in all correlated WPs' actions, including the large work on climate resilience modelling of T8.5.

1.2 Task objectives

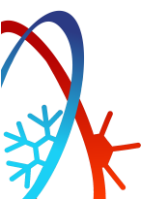
The main objectives of this deliverable are here reported, in line with the general description of task 2.3 actions.

- Define a methodology and a tool to calculate climate and building KPIs for the free-running potential under typical and real weather conditions supported by EnergyPlus simulation.
- Allow to simulate the reference model under real conditions.
- Include in the simulation approach the possibility to compare real building behaviours and model building behaviours under real weather conditions to verify free-running operation schemes and support users' information.
- Define free-running operation schemes suggesting optimised free-running actions during building operation.
- Allow to test specific free-running technologies supporting, for example, heat gain preventions in summer (e.g., shading systems) to optimise design and operational decisions.

To answer all of the above-mentioned objectives, the deliverable is based on a new tool which is under development at POLITO. It allows the calculation of climate and building-related KPIs and acts as a new EnergyPlus automation interface. The tool, named PREDYCE (Python semi-Realtime Energy DYNamics and Climate Evaluation) is under implementation thanks to two development actions: the "PRE" action, funded by the PRELUDE project, and the "DYCE" action. The "PRE" development action includes several new functionalities, such as the possibility to calculate climate-related KPIs, extra free-running simulation actions (e.g. shading system and wind/stack driven ventilation) and technologies (e.g. direct evaporative cooling and earth-to-air heat exchangers), new FR visualisations and instruments (e.g. bioclimatic charts), a new forecasting scenario suggesting optimised free-running actions to users (e.g. shading and ventilation) named "24h forecast", and the possibility to use the tool to test simple and advanced control rules by integrating EnergyPlus EMS (energy management system) functionalities (e.g. shading control based on the previous hour(s) of adaptive thermal comfort values).

The tool includes an input module and a specific output module to analyse simulation and climate analyses' results considering specific FR KPIs. The tool is able not only to analyse the mitigation potential of FR in reducing the intensity of energy needs in hybrid modes, but also to define the FR impact on comfort/discomfort number of hours during fully free-running periods of operation. Additionally, the tool includes a running module to manage simulations.

The PREDYCE tool can calculate both climate-related and building-related KPIs. Climate-KPIs based on the sole elaboration of weather/climate data, defined by given databases using the EPW file extension (EnergyPlus Weather), considering the "virtual" impact of technologies and buildings without assuming a specific building configuration and without running building simulations. These climate-correlated analyses are mainly supported for early-design purposes – see also Section 2.3.I – and/or to support policy and territorial analyses. The building-KPIs are based on dynamic simulation results assuming a specific building model and given input condition(s). This second approach can support specific and advanced design suggestions, parametric analyses to improve building characteristics or operational choices and all actions in which technologies and buildings need to be characterized in a "real" or "realistic" way.



The tool is detailed with main contents, actions, and KPIs in Section 2, while methodological example of applications reported in Section 3.

During the remaining WP3 months (M18-M24), activities of T3.2 will focus on the development of extra functionalities to be added to the tool, including the possibility to identify indoor/outdoor correlations based on the methodology developed in T3.4 to support inter-module integration. Additionally, the work will continue to support integration aspects considering PRELUDE WP4 and WP5 requirements and the testing of the tool in WP6 and WP7. Finally, the tool will be adopted, and partially implemented, also during the activities of T8.5 about climate change resilience studies.

2. PREDYCE – Dynamic simulation platform

2.1 PREDYCE general architecture

The core of the deliverable has been the co-development of a Python library named PREDYCE (Python semi-Realtime Energy DYNamics and Climate Evaluation). It is composed of three main modules: the IDF editor, the runner module and the KPIs calculator. Each module is responsible for different tasks: i) the IDF editor module allows automatic editing of EnergyPlus building models in IDF format through pre-built actions which are able to act on several IDF parts such as building activities, envelope, technologies, HVAC system; ii) the runner module can execute multiple parallel EnergyPlus simulations asynchronously exploiting available CPU capabilities; and iii) the KPIs calculator module computes several building and climate-related indicators on both simulations results and monitored data, and generates clear analysis graphs. The three modules have been thought to work together, allowing the definition and testing of different usage scenarios and methodologies. They can also be used independently if needed and leave flexibility and freedom of development for future needs.

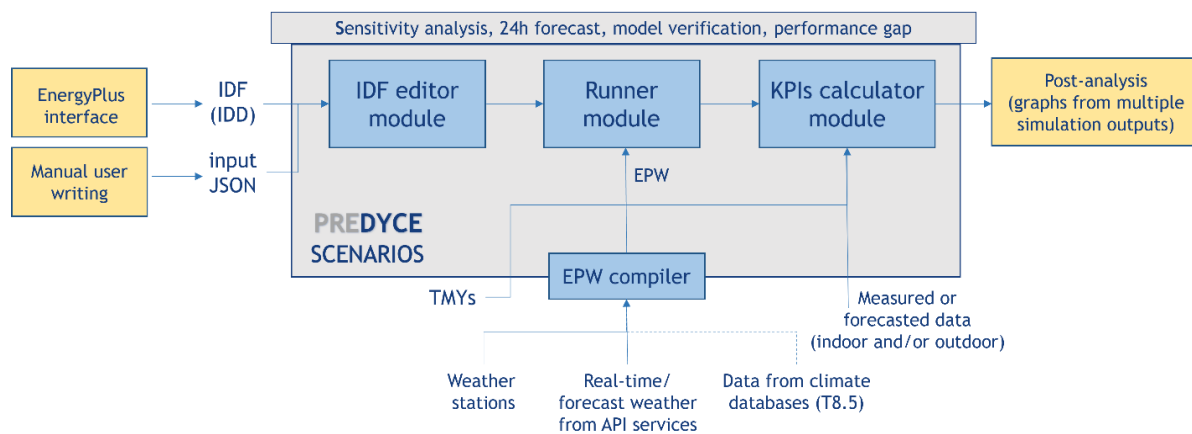
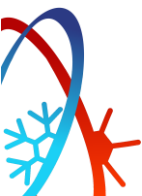


Fig. 1 – Overview of PREDYCE modular structure

Figure 1 shows the library structure: the three main modules are combined inside different automatic or semi-automatic task-oriented usages, such as sensitivity analysis (which is the base scenario on which the others are built on), model verification, 24h forecast and performance gap scenarios, which allow to fully respond to PRELUDE goals. The different scenarios will be described in the following sections together with related application examples. Each scenario consists in a Python script exploiting the functionalities of the library and it can be launched both by command line and through a dedicated REST API service. The latter allows for an easy server-to-server communication with the FusiX platform acting as project middleware – see PRELUDE T5.5. Besides the main library modules, another module called EPW compiler has been



developed and integrated, to allow the generation of EPW files, which are the weather files needed for EnergyPlus simulations. These weather file can be real or forecasted weather data coming from different sources. Figure 1 also highlights which steps in a full PREDYCE scenario workflow still needs manual intervention and have not been automatized. Particularly, the generation of the building model has to be performed through an EnergyPlus CAD interface allowing the export of IDF files (such as DesignBuilder [21] or OpenStudio [22]) for geometry and base building settings. Moreover, post-analyses requiring working on results from multiple simulations have to be performed in a second step, i.e. by analysing the various outputs through devoted scripts or manually, e.g. via spreadsheet tools. Finally, each scenario requires a JSON input file that has to be compiled manually. It contains personalized parametric settings for building simulations and lists KPIs that has to be computed on each performed simulation.

Before detailing PREDYCE use for PRELUDE, some insights on the library internal structure and main technical aspects are given hereafter to highlight the potentialities given by its structural modularity and flexibility. Figure 2 describes the input/output workflow of a generic PREDYCE scenario. As previously mentioned, each PREDYCE scenario is executable from command line, following the same structure used by EnergyPlus software made of several options and of an IDF model (e.g., *-w* option is followed by the EPW weather file, *-i* by the IDD version). It thus eases the use of the tool for users already familiar with EnergyPlus. The main mandatory input files for any generic script are: i) the building model in IDF format, ii) the weather file in EPW format (*-w* option), iii) a JSON file (*-f* option), structured to contain user personalized requests.

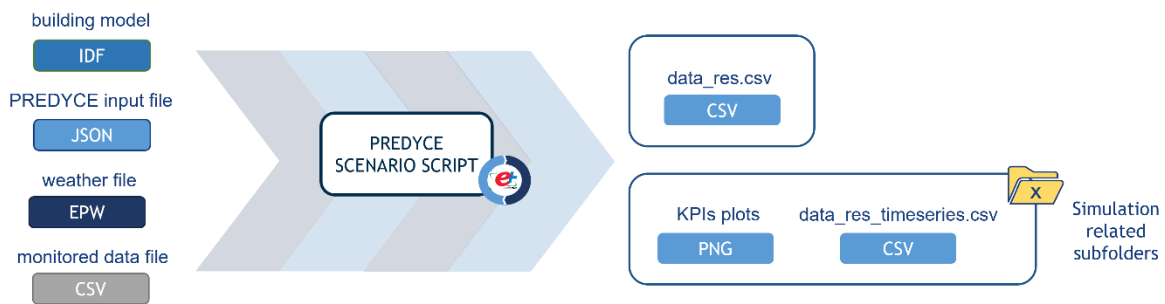


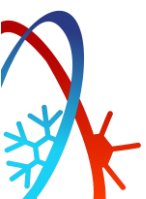
Fig. 2 – PREDYCE I/O workflow

Besides mandatory inputs, many other arguments can be specified, e.g. desired directories for outputs (*-d*) and plots (*-p*), whether to include or not the base building model with no modifications in the permutation of parameters (*-o*), the number of CPUs to be used (*-j*), which are otherwise automatically set to the maximum allowed by the machine. Among the arguments, it can be also specified the path of monitored data file (*-m*) if the scenario needs it. Instead, if a PREDYCE script is executed through the REST API service, a reduced number of options (including almost only mandatory inputs) can be set through the POST request, since directories and CPU usage is handled automatically inside the developed web service.

The main PREDYCE outputs are:

- a CSV file named *data_res.csv* containing aggregated KPIs in the considered run period for all the performed simulations,
- a subfolder containing timeseries KPIs (*data_res_timeseries.csv*) and plots related to each simulation through *data_res.csv* indexing numbers.

As previously mentioned, each PREDYCE script needs a JSON input file which allows to personalize the user request and differentiate the analyses. Figure 3 shows the structure of such a file made of keys, which are recognized as keywords inside the library modules, and values, which are interpreted and used by the different developed actions. For example, the *building name* is the name of the main block of the IDF. It is



utilized by the tool to know which zone elements to edit and perform calculations on. The *preliminary actions* field lists all the building editing actions which are executed only once before running the simulations for the parametric analyses. Consequently, all simulated buildings have in common the same modifications listed in the preliminary action section. The *actions* field contains the parametric modifications that will be applied to the building. All actions and their parameters are combined such that each simulated building is the result of a specific combination executed on the initial building (after the preliminary actions have been applied). The *kpi* field includes the indicators that are computed at the end of each simulation. Other keys in the input JSON file can be used for example to define different spatial aggregations on which the listed KPIs are computed. The *scenario* key can be used in the future to understand the requested task and select inside a general application which PREDYCE script to run, making PREDYCE work as a task-oriented simulation platform interfacing EnergyPlus.

```
{
  "building_name": "MainBlock",
  "preliminary_actions": {
    "change_runperiod": {
      "start": "01-05",
      "end": "30-09",
      "fmt": "%d-%m"}},
  "actions": {
    "change_ach": {
      "ach": [0, 2.5, 5],
      "ach_type": ["ventilation"]},
    "add_overhang_simple": {
      "extension": [0.1, 0.2, ...]}
    "other_actions": "..."},
  "kpi": {
    "cdhi": {},
    "fict_cool": {}}
```

Fig. 3 – Example of input JSON file for PREDYCE

PREDYCE library operates with two internal JSON databases hidden to the external user. They are used inside the tool to define IDF editing actions and KPIs default values and to store the information necessary to modify IDF objects (e.g., materials composition, schedules). The former, as shown in Figure 4, is called *database of actions* and was inspired by the concept of measures in OpenStudio – see also [23]. It is not fully implemented inside PREDYCE, but it has been however initialized to lay the foundations to a future possible development of a GUI (Graphical User Interface) allowing to compile in a more user-friendly way the JSON input file, choosing actions and values by pre-defined lists. The latter is called *database of objects* and contains dictionary structures of basic IDF objects which are retrieved by the IDF editing functions, e.g., it allows to retrieve the characteristic of a blind element given its name. This kind of database is fundamental to any EnergyPlus interface (e.g., both DesignBuilder and OpenStudio have a database structured like this), since any IDF editing action must follow rules for object structure (and interaction) which are described in detail in the IDD file associated to each software release.

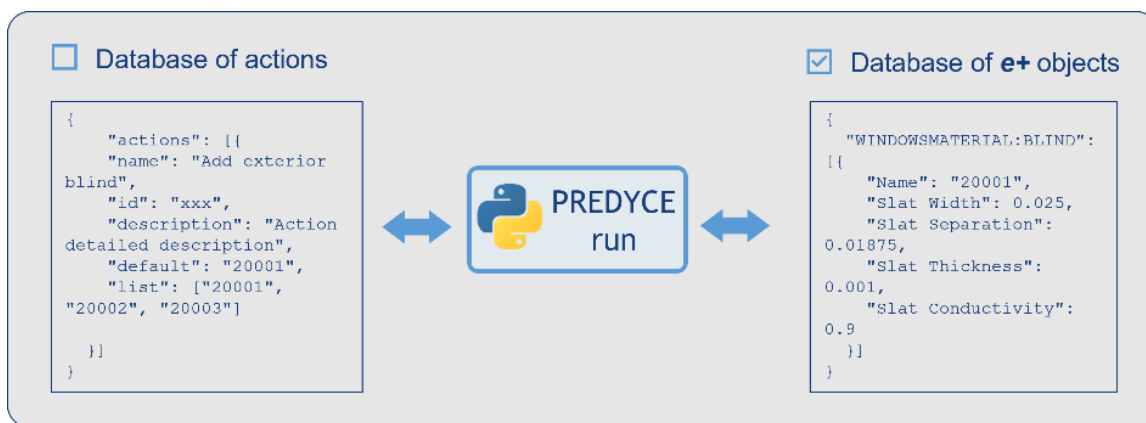


Fig. 4 – Example of PREDYCE databases content

The core of the PREDYCE library is the runner module which allows to perform multiple EnergyPlus simulations and KPIs computation at once. It thus reduces the time needed for running parametric analyses. The runner module oversees the creation of a pool of simulations according to parameters listed in the JSON input file. It then handles its execution in an asynchronous way by multiple instances of EnergyPlus on the same machine according to the number of cores of the current CPU (parameter that can be both specified by the user or set automatically to the maximum available).

Figure 5 shows the structure of the data frame which is created inside the runner module whenever a PREDYCE script is executed. Each row represents a building setting that will be simulated. Each column contains one modification that is applied to the building. After each simulation run, all KPIs are calculated and appended in the new data frame columns. In the end, a table containing all simulated buildings results is saved on a CSV (*data_res.csv*) file for further analysis.

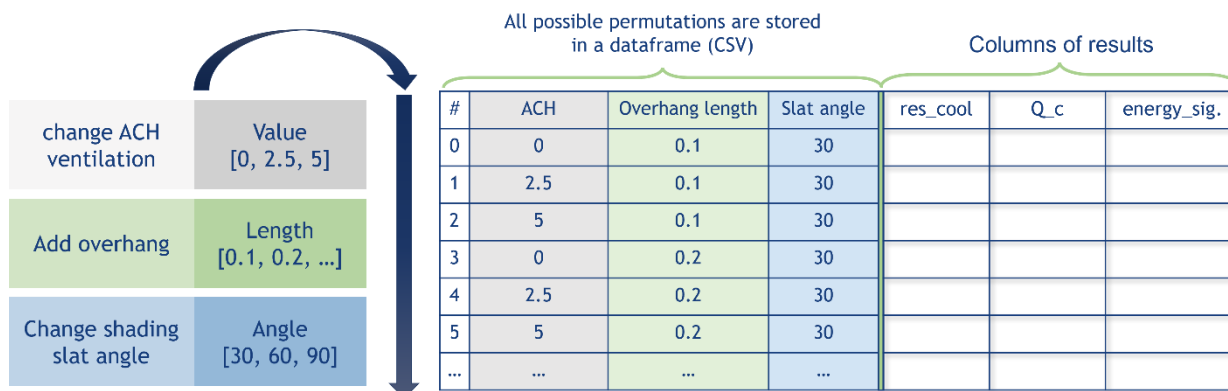
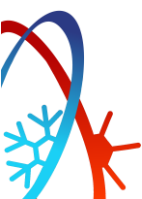


Fig. 5 – Runner module workflow and creation of aggregated output

The generated CSV file contains aggregated KPIs results over the simulated time period. Nevertheless, in order to deepen temporal trends, as shown in Figure 6, the runner module is also in charge of creating subfolders for each simulation. These subfolders contain KPIs timeseries results (*data_res_timeseries.csv*) and plots. These outputs are matched to aggregated results contained in the *data_res.csv* file by naming each folder with the same row index of *data_res.csv* file corresponding to a specific run. This is done to keep in memory more detailed information about performed simulations for further analysis but allowing a much lighter storage usage rather than keeping all EnergyPlus output files for each simulation. Timeseries results are by default saved with hourly resolution, but the data resolution can be also personalized through the input JSON. Plots generation is controlled by a Boolean flag which is passed to KPI methods.



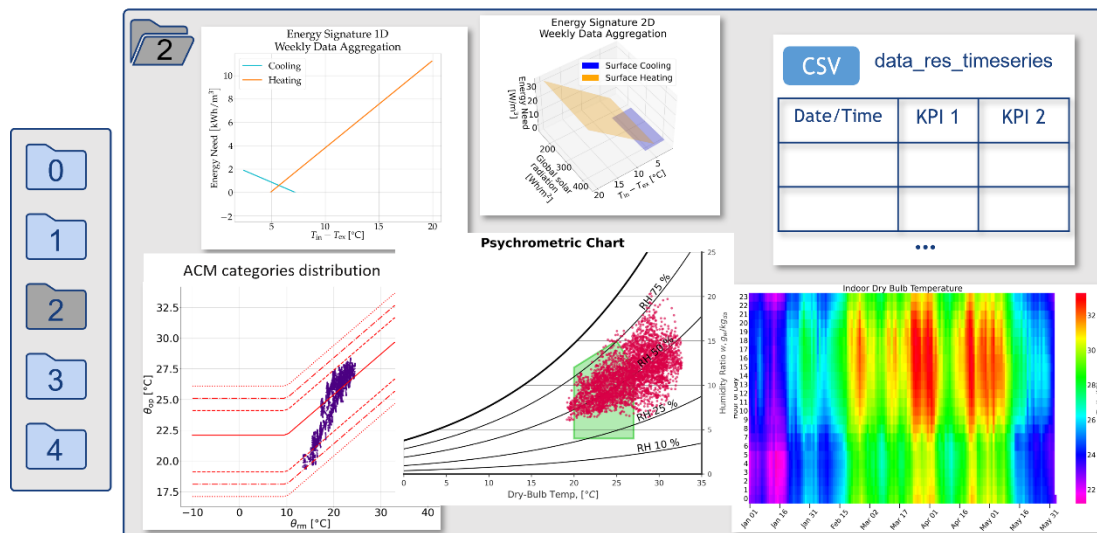
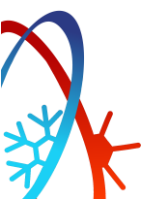


Fig. 6 – Simulation specific subfolder output

Besides the runner module, the IDF editor module and the KPIs calculator modules allows to manage simulation inputs and outputs. In particular, the IDF editor module contains a collection of methods able to modify EnergyPlus building models for different purposes thanks to the Eppy scripting language [24]. Since IDF objects' structure and interconnections are linked to the EnergyPlus version release, the IDF editor was developed having in mind a specific software version, the 8.9. So, in order to be compatible with more recent releases some functions may require an update. The IDF editor module allows to overcome several difficulties that could occur during the IDF editing procedures. For example, it allows to manage the integration of some objects available in EnergyPlus software but that cannot be, at the present, used inside the main CAD interfaces for simulation interface, such as DesignBuilder (e.g., PDEC objects or wind and stack effect ventilation objects). It allows much more complex editing actions than base editor provided together with EnergyPlus software, which only allows to change values associated to specific fields. The user does not have to study IDF objects composition and complex interconnections, which is instead an indispensable knowledge for a manual IDF file editing action. Hence, the IDF editor module is built according to a parametric and automatic vision, allowing to exploit the potentialities of a massive sensitivity analysis approach.

The KPIs calculator module instead is in charge of performing indicators calculations, analyses and plots generation treating both simulated and monitored data and considering buildings and climate based KPIs. Developed methods usually consist in resampling output data and applying formulas to compute indicators based on European standards. Figure 7 shows how this module is built reporting a sample part extracted by the PREDYCE code (so units of measures and other information on internal variables are not visible). Inside it there are two main Python classes containing methods with the same names: *EnergyPlusKPI* class is thought to work on EnergyPlus outputs, accessing its output files (mainly eplusout.csv) and pre-processing them (e.g., standardizing nomenclature, averaging values on aggregation of thermal zones, changing unit of measures) before actual KPIs computation made by parent *KPI* class; instead, *KPI* class is thought to work both on structured monitored data and on simulation results and it performs the indicators calculation and plots generation. This structure made of parent and child classes allows for very high flexibility of data sources (even possibly coming from other simulation tools), since it just requires common variables nomenclature to be recognised by the parent *KPI* class, for example, as shown in Figure 7, "T_db_o[C]" is the variable name for the outdoor dry-bulb temperature in degree Celsius. Among developed KPIs methods, some return a number, while others return a dictionary of values (e.g., describing a distribution) allowing for more complex post-analyses. Moreover, timeseries KPIs (recognized by the keyword *timeseries*) return a table with values associated to each timestep, according to the defined time resolution.



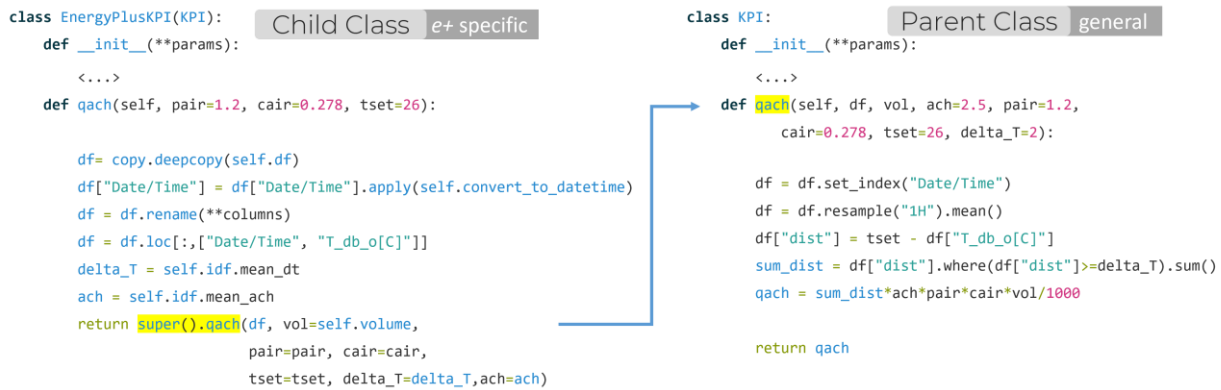


Fig. 7 – Example of KPIs calculator module structure

2.2 The ‘PRE’ development action

The PREDYCE library is currently under co-implementation thanks to two main development actions: i) the “PRE” action, funded by the PRELUDE project, and ii) the “DYCE” action, funded by the E-DYCE² project. The core structure of the library, which makes possible the development of different task-oriented scenarios, is common to the two projects together with some main IDF editing methods and KPIs. Although, the initial architecture of the tool refers to the “DYCE” development action. Instead specific scenarios such as the 24h forecast and several IDF editing methods and KPIs, e.g. climate ones, have been included specifically to answer PRELUDE needs. In particular, PRELUDE project is focused on free-running buildings simulations, passive cooling (and heating) solutions and their local potential and climate resilience analyses. In the following sections, KPIs methods and scenarios of use developed for PRELUDE are discussed, while this section is focused on IDF editing methods. In fact, being able to modify the building model controlling all desired building aspects is the base for meaningful scenarios applicability. Several pieces of PREDYCE manual will be reported in the following to explain the methods working and their input/output workflow.

Among PRELUDE main goals, there are the estimation of local climate potentials, the optimisation of buildings free-running mode and exploitation of passive and low-energy heating and cooling technologies to reduce both time and amount of energy from HVAC systems usage. Consequently, it was very important to integrate inside IDF editor module methods able to manage these aspects. Several functions, for example, have been developed to manage overhangs and shadings, considering the different kinds of objects existing inside EnergyPlus software: blinds (both interior and exterior) can be added to windows based on their orientation and managed according to defined schedules or base thresholds strategies; also, overhang objects can be included inside the model (as shown in the following methods) being able to manage freely both “simple” and “complex” objects parameters. Moreover, simple EMS (Energy Management System) programs [25,26] written in Erl (EnergyPlus Runtime Language) can be included inside the IDF file to control the shading devices with user defined actuators . It enables the definition and testing of more complex and flexible thresholds strategies than the basic ones integrated as keywords in the shading control objects. In fact, several shading control methods are already included in the basic EnergyPlus shading control object, allowing to simulate actuators based on temperature, solar radiation or cooling consumption thresholds. However, when using EMS programs, it is possible to also consider temporal trends, illuminance level in the room, or any other desired variables combination. Also, EMS programs added to the IDF file could be varied through other IDF editor actions, allowing to test parametrically different thresholds combinations. Although, several EMS programs may be added to the

² E-DYCE (Energy flexible DYnamic building Certification), which has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 893945.

PREDYCE library in the future to answer to specific-usage requests, allowing to control e.g., mechanical ventilation systems. An example of EMS programs applied through PREDYCE is detailed in Section 3.

predyce.idf_editor.add_overhangs_simple(*idf*, *extension=1*, *tilt=90*, *shift=0.04*)

Add simple overhang.

Parameters

- **idf** (class:*predyce.IDF_class.IDF*) – IDF object
- **extension** (*int*, *optional*) – Extension of the overhang, defaults to 1
- **tilt** (*int*, *optional*) – Tilt of the overhang, defaults to 90
- **shift** (*float*, *optional*) – Shift of the overhang, defaults to 0.04

predyce.idf_editor.add_overhangs_complex(*idf*, *depth=<class 'float'>*, *tilt=90*, *orientation=None*, *transmittance_schedule=')***

Set the overhang shading on all external windows by creating a SHADING:ZONE:DETAILED object which can be viewed using *geomeppy*'s *view_model()* function.

Note: Depth must be greater than 0

Parameters

- **idf** (class:*predyce.IDF_class.IDF*) – IDF object
- **depth** (*float*) – The depth in meters of the overhang.
- **tilt** (*int*) – Tilt Angle from Window/Door {deg}.
- **orientation** (*int*) – One of "north", "east", "south", "west". Walls within 45 degrees will be affected.
- **transmittance_schedule** (*str*, *optional*) – Transmittance Schedule Name, defaults to ""

Another set of IDF editor methods is devoted to control the ventilation rate inside the thermal zones, supporting different computation methodologies and technologies. In particular, the *change_ach* method allows to simulate a specified fixed amount of air changes per hour (ACH) in a thermal zone due to natural ventilation effects. This can be made according to a defined schedule, applicable to modulate the amount of ACH in different hours and days or restricted to certain indoor/outdoor temperature ranges and differences.

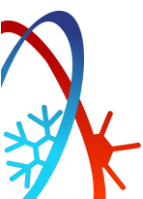
predyce.idf_editor.change_ach(*idf*, *ach*, *ach_type='ventilation'*, *filter_by=''*, *relative=False*, *fields*)**

Change ACH for all zones of a specific block of the building.

Parameters

- **idf** (class:*predyce.IDF_class.IDF*) – IDF object
- **ach** (*int or dict*) – New ACH (air changes per hour) value or dictionary of ACH values for each zone.
- **ach_type** (*str*) – ACH type, between *Infiltration* and *Ventilation*
- **filter_by** (*str*, *optional*) – Filter zones by name. Can be the block name or specific zone name, defaults to "", ignored if *ach* is a *dict*
- **relative** (*bool*, *optional*) – Specify if new value will be assigned as a percentage increment from the old value, defaults to *False*
- **fields** – Additional fields to be set to the *DesignFlowRate* object, passed as keyword arguments

In order to consider a more realistic ventilation effect which takes into consideration wind and stack effects with basic formulas, an IDF editor function allows to substitute in each zone the basic *ZoneVentilation:DesignFlowRate* EnergyPlus object, which is inserted by main CAD interfaces like *DesignBuilder* when activating the natural ventilation inside the building, with the *ZoneVentilation:WindandStackOpenArea* object and manage its parameters. In particular, opening area and windows height parameters can be automatically derived from an analysis of the thermal zone windows, considering different strategies (e.g., selecting which window is usually opened, averaging windows dimensions), while managing the schedule allows to act on the opening area fraction with hourly



resolution and control the effective windows opening. This change allows to include powerful functionalities to scheduled zone ventilation analyses allowing to define flows on the base of stack-driven and wind-driven air movements without considerably increase the complexity of the EnergyPlus ventilation calculation engine. With specific considerations, it can be possible to adapt well-known wind and stack driven expressions used for designing controlled natural ventilation flows – e.g. see [27–30] – to define the EnergyPlus required opening area fraction. At present, the simplest case in which a single window is opened inside a thermal zone has been tested, adapting EnergyPlus formulas [31] to those described in BS 5925:1991 [32] for single side natural ventilation, and supported also by IEA EBC Annex 62 [29] and other sources [28]. However, in the future, also the effects of multiple openings can be considered through additional formulas adaptations. In particular, the following single-side ventilation formula for one large opening was adopted for the computations considering the stack effect:

$$q_s = c_d \cdot \frac{A}{3} \cdot \sqrt{\frac{\Delta T \cdot g \cdot h}{T_{av}}} \quad [m^3/s]$$

Where:

c_d is the discharge coefficient of the openings (set to 0.6 for external openings [27]), A the window opening area, h the window height [m], ΔT the absolute difference between indoor and outdoor temperatures [K], and finally T_{av} the indoor temperature [K].

EnergyPlus formula was adapted to the above-mentioned formula by forcing the c_d parameter and by imposing the schedule multiplier factor for windows opening to 0.33 when open and to 0 when close.

predyce.idf_editor.add_windandstack_nat_vent(*idf*, *value=None*, *filter_by=""*, *schedule=None*, *params={}*, *func=<built-in function max>*)

Add scheduled natural ventilation exploiting stack and wind effects formulas to desired zones.

Parameters

- **idf** (class:*predyce.IDF_class.IDF*) – IDF object
- **filter_by** (*str*, *optional*) – Filter zone by name. Can be the block name or specific zone name, defaults to ""
- **schedule** (*str or dict*, *optional*) – Name of the Schedule:Compact construction object to be added in IDF or the Schedule:Compact object in a dictionary format, defaults to None
- **params** (*dict*, *optional*) – Additional scheduled natural ventilation parameters for the ZoneVentilation:WindandStackOpenArea object, defaults to {}

Still concerning ventilation related objects, some functions have been developed to support also the calculated natural ventilation objects, based on the EnergyPlus airflow network [31]. This kind of ventilation model – still simplified if compared to detailed ones, such as CFD-based approaches – takes into consideration airflow components and a set of nodes supporting flow distribution inside the entire building by evaluating external and internal condition elements. PREDYCE is not able to recreate from scratch all necessary nodes to build-up the calculated ventilation scenario, but it has the possibility to edit some parameters such as opening factor and discharge coefficient for window openings and modify the arrays of pressure coefficients for the external airflow given by the wind angle. Hence, these editing actions can be used only in case the calculated ventilation scenario has been previously added to the model through the chosen EnergyPlus initial interface, but they are useful to extend parametric analyses on building. An issue that occurs when dealing with this specific calculated natural ventilation is the limited possibility to perform controlled parametric analyses, for example, windows opening factor is based on the probability that someone in the room wants to open the window and cannot be directly controlled by a scheduled timetable as with other ventilation typologies.

Moreover, other IDF editor methods allow to manage a mechanical ventilation system eventually installed in the building. The possibility to manage also a mechanical system was added because of the interest in

testing some PRELUDE methodologies for optimal free-running mode building evaluation also inside buildings already equipped with more advanced systems, such the case of some of the PRELUDE demos. In particular, the adoption of optimized scheduled patterns, which takes into consideration also the use of shading devices, could lead to energy consumption reduction of other installed systems, such as the cooling system.

predyce.idf_editor.add_mechanical_ventilation(*idf*, *filter_by*='', *schedule*=None)

Add mechanical ventilation.

Parameters

- **idf** (class:*predyce.IDF_class.IDF*) – IDF object
- **filter_by** (*str*, *optional*) – Filter zone by name. Can be the block name or specific zone name, defaults to ""
- **schedule** (*dict or str*, *optional*) – [description], defaults to None

For this purpose, an IDF editor function was also developed to handle inside EnergyPlus simulations hourly detailed and variable scheduling strategies through external CSV files. This kind of function can be used for several purposes, considering for example mechanical ventilation systems, but also controllable shading devices, or HVAC setpoints. The same mechanism can be also used to suggest best strategies for windows opening and shading controls in building without actuators, then providing to the user a scheduling suggestion to be adopted with manual intervention. Also, this action could result useful in model verification task to set indoor temperatures in desired boundary zones.

predyce.idf_editor.edit_csv_schedule(*idf*, *csv*, *hours*, *columns*, *value*, *sep*=';', *index_col*=0, *save*=True, *fix_paths*=True)

Edit a CSV representing a Schedule:File object of EnergyPlus.

Parameters

- **idf** (class:*predyce.IDF_class.IDF*) – IDF object
- **csv** (*str* or class:*pandas.core.frame.DataFrame*) – CSV file
- **hours** (*int or list*) – Hours of CSV file to be changed
- **columns** (*list*) – Columns of the CSV file to be changed
- **value** (*float*) – Value to be assigned
- **sep** (*str*, *optional*) – Separator of the CSV file, defaults to ";"
- **index_col** (*int*, *optional*) – Index column number, defaults to 0
- **save** (*bool*, *optional*) – Save modified CSV to file, defaults to True
- **fix_paths** (*bool*, *optional*) – Convert CSV file paths in IDF to point at the new file, defaults to True

Other developed methods are focused instead on HVAC system handling. In fact, in anticipation of the PRELUDE goal linked to buildings climate resilience analysis that will be deepened in T8.5, an IDF editor method able to perform sizing of simple EnergyPlus HVAC system (not considering the detailed HVAC version) has been developed. The function has been thought to work in two main ways: first, if system technical details (capacity and flow rate limits) are known, they could be given in input in substitution to simulation auto-sizing; secondly, the simulation auto-sizing, normally performed on the design days defined in the EPW weather file, is substituted by maximum system hourly values found performing a yearly simulation without plants limits. This second option will result useful in case typical buildings HVAC system resilience is studied with respect to future climate changes. This method is more complex than others since it requires to perform an initial simulation from which extracts outputs to modify the IDF, instead of directly acting on IDF objects. Consequently, it is thought to be used mainly as a preliminary action inside the input JSON file and it is made by two main functions defined with the same name *simple_hvac_sizing*, one inside the IDF editor module and the other inside the runner module. The runner module method oversees performing an initial simulation without plants limits, in order to extract for each thermal zone maximum yearly capacity values (Watts): this simulation is performed also if system details are given in input, since global numbers should be split to zonal values according to found percentages of use. Then, the IDF editor

method is in charge of modifying the original IDF with the found values, removing auto-sizing related objects. Moreover, a scaling factor can be passed in input to the function to give more flexibility to maximum find values (e.g., if a 1.25 factor is passed, maximum values will be increased of 25%, e.g. such as suggested in ASHRAE standards).

predyce.runner.simple_hvac_sizing(*idf, epw, temp_dir, values*)

HVAC sizing according to yearly heating and cooling zonal max values, in substitution to autosize defined values: a yearly simulation is run and max found hourly capacity values are passed to `idf_editor` function. Eventually user-defined max values for capacity are distributed according to zonal system use in percentage and then passed to `idf_editor` function.

Parameters

- **idf** (class:*predyce.IDF_class.IDF*) – IDF object
- **epw** (*Path or str*) – EPW directory
- **temp_dir** (*Path or str*) – Output directory of EnergyPlus simulations.
- **values** (*dict*) – parameters associated to `simple_hvac_sizing` `idf_editor` function

predyce.idf_editor.simple_hvac_sizing(*idf, zone_heat_dict, zone_cool_dict, factor=1, **fields*)

HVAC sizing according to yearly heating and cooling zonal max values, in substitution to autosize defined values. The function is supposed to work automatically together with the homonymous function in runner module. The user can also provide the known maximum plant capacity [W] with keys 'Maximum_Sensible_Heating_Capacity' and 'Maximum_Total_Cooling_Capacity' and eventually the known maximum flow rate values [m³/s] with keys 'Maximum_Heating_Air_Flow_Rate' and 'Maximum_Cooling_Air_Flow_Rate'.

Parameters

- **idf** (class:*predyce.IDF_class.IDF*) – IDF object
- **zone_heat_dict** (*dict*) – dictionary containing maximum heating capacity for each zone. It's automatically generated inside the runner module.
- **zone_cool_dict** (*dict*) – dictionary containing maximum cooling capacity for each zone. It's automatically generated inside the runner module.
- **factor** (*float, optional*) – factor to multiply found max capacity zonal values, in order to be flexible to different climate conditions, defaults to 1

Other developed methods allow to include in the IDF additional passive and low-energy cooling (and heating) technologies and related EnergyPlus outputs, in particular PDEC (passive downdraught evaporative cooling) and EAHX (earth-to-air heat exchangers) systems – the latter may be also referred in literature as EHX or EAHE. While EAHX systems could be currently included also in the original model generated by e.g., DesignBuilder, even if with some limitations – e.g., outputs of the system are not originally plotted and soil surface parameters are not automatically retrieved –, DEC (direct evaporative cooling) systems, despite foreseen by EnergyPlus software, are still not included in the majority of CAD interfaces. Both EAHX and DEC solutions act as passive heat gain dissipation techniques for space cooling purposes, while EAHX may also be used as pre-heating technology for reducing ventilation heat losses in winter and/or to reduce HVAC energy needs for space heating in air-based systems. Complete descriptions on DEC applications for passive and low-energy cooling purposes may be retrieved in literature – e.g. see [10,12,13,33–39] – together with references on basic physical and design principles of EAHX systems – e.g. see [10,12,13,40–45].

The addition of both technologies inside the IDF file requires the settings of many parameters, as shown in the following PREDYCE manual descriptions. Particularly, the EAHX addition also requires the exploitation of an auxiliary program for EnergyPlus [46] called *CalcSoilSurfTemp.exe* to compute three fundamental soil properties from EPW file data and then fill correlated earth tube object parameters to calculate the soil temperature at the give depth and moment of the year. These three parameters are: the yearly average soil surface temperature, its amplitude and its phase. The soil condition and surface condition parameters in input to the PREDYCE `add_eahx` method are in fact used in input to the *CalcSoilSurfTemp* program, which

is executed internally to PREDYCE method hidden to users, for its internal computation that automatically fills *av_soil_surf_temp*, *ampl_soil_surf_temp* and *phase_soil_surf_temp* parameters if not given by the user through the input JSON. Soil properties computed through this program are also saved inside PREDYCE internal building object, in order to be re-used if needed inside EAHX related KPIs without executing the program again. In addition to inputs correlated to soil conditions, which is used by EnergyPlus to define the soil thermal diffusivity and conductivity, and soil surface parameters, the EAHX calculation also requires to define design-parameters correlated to the buried pipe technology, including the pipe diameter, length, depth, material and thickness, the designed airflow and fan-correlated properties [9,13,40,42,47,48]. If those inputs are not defined, default suggested values are used in PREDYCE for preliminary analyses.

Considering instead the addition of DEC objects, it requires quite complex cascade changes inside the IDF file, such originating a quite complex editing action *add_pdec* which could take in input many parameters. In particular, at present two of the five typologies of evaporative cooler units available inside EnergyPlus software are considered inside PREDYCE: the *EvaporativeCooler:Direct:CelDekPad* object, which is set by default and requires in input e.g., the pad area and depth from which it is internally computed the unit efficiency, or the *EvaporativeCooler:Direct:ResearchSpecial* object, which can take in input the unit efficiency. At present, the developed method manages the use of *Fan:ConstantVolume* object only, as fan associated to the PDEC unit, but others could be included for future needs. Also, the integration of optional Availability Manager objects associated to the units, which allow to override the adopted schedule if particular conditions occur, could be included in the future through parallel actions.

```
predyce.idf_editor.add_eahx(idf, filter_by="", schedule=None, dfr=None, min_temp_cool=24, max_temp_heat=20,
delta_temp=2, earthtube_type="Intake", fan_pressure_rise=None, fan_total_efficiency=None, pipe_r=0.125, pipe_
t=0.004, pipe_l=75, pipe_cond=0.4, pipe_p=2.5, soil_condition=2, soil_surf_cond=2, av_soil_surf_temp=None, amp
l_soil_surf_temp=None, phase_soil_surf_temp=None, params={})
```

Add Earth tubes objects in each zone allowed by filter_by parameter.

Parameters

- **idf** (class:predyce.IDF_class.IDF) – IDF object
- **filter_by** (str, optional) – Filter zone by name. Can be the block name or specific zone name, defaults to ""
- **schedule** (str or dict, optional) – Name of the Schedule:Compact construction object to be added in IDF or the Schedule:Compact object in a dictionary format, defaults to None
- **dfr** (float, optional) – design flow rate of earth tubes, defaults to 0.139
- **min_temp_cool** (float, optional) – minimum indoor temperature for cooling activation, defaults to 24
- **max_temp_heat** (float, optional) – maximum indoor temperature for heating activation, defaults to 20
- **delta_temp** (float, optional) – delta temperature for system activation, defaults to 2
- **earthtube_type** (str, optional) – it can be Natural, Exhaust, or Intake; defaults to "Intake"
- **fan_pressure_rise** (float, optional) – pressure rise experienced across the eventual fan in Pascals (N/m²), defaults to None
- **fan_total_efficiency** (float, optional) – total fan efficiency (a decimal number between 0.0 and 1.0), defaults to None
- **pipe_r** (float, optional) – pipe radius in meters, defaults to 0.125
- **pipe_t** (float, optional) – pipe thickness in meters, defaults to 0.004
- **pipe_l** (float, optional) – pipe length in meters, defaults to 75
- **pipe_cond** (float, optional) – pipe thermal conductivity in W/mK, defaults to 0.4
- **pipe_p** (float, optional) – pipe depth in meters, defaults to 2.5
- **soil_condition** (int, optional) – it can be 1. HEAVY AND SATURATED, 2. HEAVY AND DUMP, 3. HEAVY AND DRY or 4. LIGHT AND DRY, defaults to 2
- **soil_surf_cond** (int, optional) – it can be 1. BARE AND WET, 2. BARE AND MOIST, 3. BARE AND ARID, 4. BARE AND DRY, 5. COVERED AND WET, 6. COVERED AND MOIST, 7. COVERED AND ARID and 8. COVERED AND DRY, defaults to 2
- **av_soil_surf_temp** (float, optional) – average soil surface temperature, defaults to None
- **ampl_soil_surf_temp** (float, optional) – amplitude of soil surface temperature, defaults to None
- **phase_soil_surf_temp** (float, optional) – phase of soil surface temperature, defaults to None
- **params** (dict, optional) – Additional scheduled natural ventilation parameters for the ZoneVentilation:DesignFlowRate object, defaults to {}

```
predyce.idf_editor.add_pdec(idf, filter_by="", schedule=None, dfr=None, ev_cool_typology='EvaporativeCooler:Direct:CelDekPad', direct_pad_area=None, direct_pad_depth=None, water_pump_pc=None, fan_press_rise=None, fan_dfr=None, ZoneHVAC_EvaporativeCoolerUnit_params={}, EvaporativeCooler_params={}, Fan_ConstantVolume_params={})
```

Add Evaporative Cooler units and all related objects to all zones allowed by the `filter_by` parameter.

Parameters

- **idf** (class:*predyce.IDF_class.IDF*) – IDF object
- **filter_by** (*str*, *optional*) – Filter zone by name. Can be the block name or specific zone name, defaults to ""
- **schedule** (*str or dict*, *optional*) – Name of the Schedule:Compact construction object to be added in IDF or the Schedule:Compact object in a dictionary format, defaults to None
- **dfr** (*float*, *optional*) – design flow rate of evaporative cooler units, defaults to None
- **ev_cool_typology** (*str*, *optional*) – typology of evaporative cooler unit object. It can be EvaporativeCooler:Direct:CelDekPad or EvaporativeCooler:Direct:ResearchSpecial, defaults to EvaporativeCooler:Direct:CelDekPad
- **direct_pad_area** (*float*, *optional*) – face area of the evaporative pad in meters if `ev_cool_typology` is set to EvaporativeCooler:Direct:CelDekPad, defaults to None
- **direct_pad_depth** (*float*, *optional*) – depth of the evaporative pad in meters if `ev_cool_typology` is set to EvaporativeCooler:Direct:CelDekPad, defaults to None, defaults to None
- **water_pump_pc** (*float*, *optional*) – power consumed by the evaporative cooler recirculating pump in Watts, defaults to None
- **fan_press_rise** (*float*, *optional*) – pressure rise in Pascals at full flow and standard (sea level) conditions (20°C and 101325 Pa), defaults to None
- **fan_dfr** (*float*, *optional*) – fan full load air volumetric flow rate (m³/sec) at standard temperature and pressure (dry air at 20°C drybulb), defaults to None
- **ZoneHVAC_EvaporativeCoolerUnit_params** (*dict*, *optional*) – additional parameters for ZoneHVAC:EvaporativeCoolerUnit object, defaults to {}
- **EvaporativeCooler_params** (*dict*, *optional*) – additional parameters for EvaporativeCooler unit object, defaults to {}
- **Fan_ConstantVolume_params** (*dict*, *optional*) – additional parameters for Fan:ConstantVolume object, defaults to {}

Finally, several IDF methods were developed to support building model verification with respect to monitored temperature or consumption data. Model verification is a crucial task whenever the goal is to give behavioural advice or studying best control thresholds of real buildings, especially if they are quite old buildings in which not all construction details are known and consequently included in the model. Calibration with respect to monitored data is usually a long and complex task, executed manually though the EnergyPlus CAD interfaces on few days (maximum few weeks) of monitoring. Hence, to speed up and support this task, different functions were developed to be then used as parametric building modifications around known construction values or to try understanding possible real values where unknown. Particularly, developed functions can be grouped in four categories: i) methods acting on building opaque envelope changing U-value of walls and roofs (acting on thickness of mostly insulated layer); ii) methods acting on U-value and SHGC (Solar Heat Gain Coefficient) of windows; iii) methods allowing to change internal mass and equipment gains in each thermal zone; iv) methods acting on ventilation and infiltration ACH. Moreover, other functions can be also used for model verification purposes depending on the specific building and goals: e.g., acting on occupancy, HVAC setpoints, CO₂ generation rate, ventilation schedules, shadings. Also, another IDF editor function allows to align the day of week of simulated calibration period to the actual one, such that effects due to occupancy and other activities schedules are correctly evaluated.

2.3 Sensitivity analysis scenario for free-running

Sensitivity analysis is most important PREDYCE scenario and allows the definition of other more task-oriented scenarios on top. Its main functionality is allowing to perform parametric analyses by automatically modifying the base building model (through IDF editor methods) according to combinations of all parameters listed in the input JSON file and then computing requested KPIs. However, considering PRELUDE goals related to climate potential and resilience analyses, this scenario was thought to use as simulation parameter, besides building related parameters, also the EPW file, such allowing geographical and/or temporal analyses, depending on used EPW files. In order to use the EPW file as a simulation parameter it is possible to act in several ways: it is possible to directly specify by command line options an EPW folder instead of a single EPW file, or the EPW folder can be specified inside the input JSON file together with the other parameters, or the desired folder can be specified in a configuration file (eventually together with other execution options such as plots and outputs folders) avoiding then to specify it in each run. In case EPW files are used parametrically, it is necessary to specify an additional EPW file to be used during preliminary actions like HVAC sizing by simply passing the *epw* parameter to those actions.

Other PREDYCE scenarios may exploit different logics in the choice or combination of parameters with respect to simple sensitivity approach, but at the base there is the library capability of executing parallel runs of EnergyPlus simulations and storing results in aggregated and lightweight form. At present, analyses on best values combination (e.g., to find minimum energy consumption) and plots generation requiring data from multiple simulation results (e.g., plotting geographical distribution of a climate KPI to see the climate potential of a passive cooling technology) has to be done next by looking, manually or with scripting support, at aggregated results contained in *data_res.csv*, while it is not implemented yet an optimization procedure through optimization algorithms or surrogate modelling in order to accelerate the space of parameters scanning. Anyhow, the future integration of optimization procedures, also exploiting the many already existing powerful Python libraries devoted to this task, is made possible by the tool modularity and is currently under study.

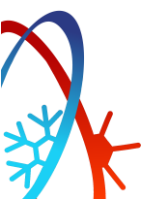
Figure 8 shows an example of *data_res.csv* where the EPW file is used as a simulation parameter: both the filename and geographical coordinates extracted from it are present in the output file, to ease further geographical plots without accessing the files anymore. Eventually, parametric actions modifying the building model can be combined with different weather files, e.g., testing different ventilation strategies or passive cooling technologies in all the different locations, allowing then to group results contained in the output file according to different logics. Section 3 will provide a more detailed example of the sensitivity analysis scenario applied to climate related studies.

#	Epw	Location	ccp	cdd	cdh
0	GRC_Athens.167160_IWEC	{"city": "ATHENS", "longitude": 23.73, "latitude": 37.9}	{"ccp": 64598.9, "ccpd": 176.9}	559.8	{"dist_18": 26828.8, "dist_21": 15846.9, "dist_24": 7792.9, "dist_26": 4276.8, "dist_28": 2061.2}
1	ITA_Torino- Caselle.160590_IGDG	{"city": "Torino- Caselle", "longitude": 7.65, "latitude": 45.18}	{"ccp": 119249.3, "ccpd": 326.7}	39.9	{"dist_18": 9350.1, "dist_21": 4301.7, "dist_24": 1558, "dist_26": 622.5, "dist_28": 167.7}
2	POL_Krakow.125660_IWE C	{"city": "KRAKOW", "longitude": 19.8, "latitude": 50.08}	{"ccp": 143366.1, "ccpd": 392.7}	6.6	{"dist_18": 4463.5, "dist_21": 1863.3, "dist_24": 657.3, "dist_26": 299.47, "dist_28": 113.1}

Fig. 8 – Example of PREDYCE *data_res.csv* with EPW file used as simulation parameter

2.3.I Climate analyses

Bioclimatic and climate-correlated design approaches start with the study of local climate conditions to analyse the potential applicability of specific low-energy, natural and passive technologies even before the definition of specific building shapes. In line with traditional and contemporary references, the possibility to include, since the building programming phase, site and microclimate analyses is a crucial aspect to



support green and sustainable design principles, with the possibility to integrate them also in building early development phases when design changes are highly possible and cost changes null or very limited with respect to later design stages. Among climate analyses, the study of local potential of passive technologies is very important to support designers and local policies in identifying low-energy heating, cooling and ventilation solutions able to guarantee space comfort conditions even without (or with very limited) system activations. Recognizing this principle, the PREDYCE tool has been adapted during the PRELUDE project to include the possibility to perform climate analyses to study the local potential of selected free-running technologies, although additional technologies and solutions may be added in future under request.

Differently by building-simulation scenarios, in which a specific building is defined, including given characteristics, climate analyses are based on the “virtual” effect of a low-energy technology – e.g., ventilation for cooling, evaporative cooling, etc. – on a “virtual” space unit, where the word “virtual” refers to a potential space which is not defined. For example, well-known climatic indicators for “virtual” space unit heating needs are the Heating Degree Days (HDD) and/or the Heating Degree Hours (HDH) that calculate the cumulative differences (or the hourly differences) between the environmental air temperatures and a given set-point (building base temperature), e.g., 20 °C. Similarly, several literature studies have defined specific climatic indicators able to define the “virtual” impact of different technologies on building energy needs or on the reduction of local heating or cooling climatic basic indices, i.e., the above-mentioned HDD, or the cooling counterparts: the Cooling Degree Days (CDD) and/or the Cooling Degree Hours (CDH).

In the following, climatic indices that have been implemented inside PREDYCE tool for PRELUDE are described in detail, supported by parts of the tool manual. They can be then used in the different scenarios of use to define at “virtual” level the local free running potential of different solutions under typical and real weather conditions.

Starting from CDD and CDH indices, they are defined in PREDYCE KPIs calculator module as following:

$$CDD = \sum m_d \cdot (\vartheta_{e,d} - 21^\circ C) \quad \begin{cases} m_d = 1 \leftarrow \vartheta_{e,d} \geq 24^\circ C \\ m_d = 0 \leftarrow \vartheta_{e,d} < 24^\circ C \end{cases}$$

Where:

$\vartheta_{e,d}$ is the outdoor dry bulb daily average temperature, and 24°C and 21°C are used respectively as first and second thresholds, as suggested by EUROSTAT [49] – see also EN ISO 15927-6:2008.

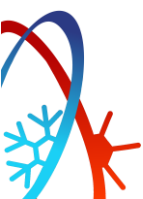
And:

$$CDH = \sum m_h \cdot (\vartheta_{e,h} - \vartheta_b) \quad \begin{cases} m_h = 1 \leftarrow \vartheta_{e,h} > \vartheta_b \\ m_h = 0 \leftarrow \vartheta_{e,h} \leq \vartheta_b \end{cases}$$

Where:

the variable m_h is a climate cooling activation check and ϑ_b the cooling base temperature, ranging by default in the domain {18°C; 21; 24; 26; 28} but customizable as an input parameter if needed, $\vartheta_{e,h}$ is the hourly dry bulb environmental temperature.

Hence, CDD output will be of type float, while CDH output will be a dictionary with results associated to each defined base temperature.



cdd(df)

Compute Cooling Degree Days.

Parameters

df (class: *pandas.core.frame.DataFrame*) – DataFrame containing at least “Date/Time” and “T_db_o[C]” columns

Returns

Cooling degree day, float

cdh(df, thresholds=[18, 21, 24, 26, 28])

Compute Cooling Degree Hours

Parameters

- **df** (class: *pandas.core.frame.DataFrame*) – DataFrame containing at least “Date/Time” and “T_db_o[C]” columns
- **thresholds** (*list, optional*) – temperature thresholds from which compute residuals, defaults to [18, 21, 24, 26, 28]

Returns

Cooling degree hour, dict

Parallely, HDD and HDH indices are defined as:

$$HDD = \sum m_d \cdot (18^\circ\text{C} - \vartheta_{e,d}) \quad \begin{cases} m_d = 1 \leftarrow \vartheta_{e,d} \leq 15^\circ\text{C} \\ m_d = 0 \leftarrow \vartheta_{e,d} > 15^\circ\text{C} \end{cases}$$

Where:

$\vartheta_{e,d}$ is the outdoor dry bulb daily average temperature, and 18°C and 15°C are used respectively as first and second thresholds, as suggested by EUROSTAT [49].

And:

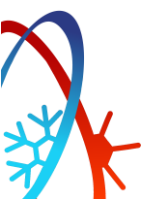
$$HDH = \sum m_h \cdot (\vartheta_b - \vartheta_{e,h}) \quad \begin{cases} m_h = 1 \leftarrow \vartheta_{e,h} \leq \vartheta_b \\ m_h = 0 \leftarrow \vartheta_{e,h} > \vartheta_b \end{cases}$$

Where:

the variable m_h is the activation check and ϑ_b the heating base temperature, ranging by default in the domain {18°C; 20; 22} but customizable as an input parameter if needed, $\vartheta_{e,h}$ is the hourly dry bulb environmental temperature.

Degree-days/hours are also used as base for climate KPIs developed to analyse the potential impact of low-energy technologies in different locations. For example, a series of indices can define the residual CDH_{res} amount of the CDH by calculating the theoretical impact of different heat gain dissipation technologies, i.e., ventilation for cooling, evaporative cooling (PDEC) and ground cooling (EAHX) – e.g. see [50]. These indices calculate the residual discomfort in terms of degree-days/hours assuming the thermal conditions of the treated airflow by the system instead of the original ones. They can be used to calculate the percent reduction in the original CDH in percentage by comparing changes in CDH induced by the system to the environment – see also [51] – and/or to define the number of climatic discomfort hours turned to comfort by a specific system/technology – see also [39].

Concerning ventilation for cooling, different indices are developed in literature, including residual CDH (or CDD assuming in this case that CDD may be calculated from CDH by dividing the latter by 24) adapted expressions considering wind-driven comfort ventilation (CDH_{res-w}) and structural ventilation (night cooling) (CDH_{res-g}) [52]. Additionally, the local potential of wind-driven environmental natural ventilation



(environmental cooling) may also be defined by calculating the sensible cooling dissipation due to internal air-exchanges with external air at lower temperature ($Q_{wind,p}$) [53].

$$CDH_{res-w} = \sum_{hrs} \left((\vartheta_{e,h} - (2.319 \cdot v_{air} + 0.4816)) - \vartheta_b \right)^+$$

Where the air velocity correlated expression is derived by regressing the ASHRAE table on perceived air temperature reduction due to air velocities in confined spaces – see the mentioned Ref. The wind velocity may be adapted to define the air velocity entering a room by limiting max velocity such as in case of “virtual” control and including a discharge coefficient effect and/or the effect of opening heights in respect to EPW data.

$$CDH_{res-gEF} = \sum_{day} \left(\sum_{nday,cd} (\vartheta_{e,h} - \vartheta_b) + EF \cdot \sum_{nday,nc} (\vartheta_{e,h} - \vartheta_b) \right)^+$$

Where:

day is the number of days in the cooling period, *nday,cd* is the number of hours in the specific day in which there is a cooling demand ($\vartheta_{e,h} > \vartheta_b$), *nday,nc* is the number of hours in the specific day with night cooling potential ($\vartheta_{e,h} < \vartheta_b$), EF is an exploitation factor considering the mass capacity of the building in exploiting day-night temperature variations (default value = 0.5; ranges between 0 to 1).

$$Q_{wind-p} = \sum (3600 (c_d \cdot v_{air} \cdot A \cdot \sqrt{\Delta c_p}) \rho_a c_A (\vartheta_{e,h} - \vartheta_b) \cdot 1h \quad [when \vartheta_{e,h} < \vartheta_b] \quad [Wh])$$

Where:

3600 [s/h] is used if the air specific capacity is expressed as 0.28 [Wh/(kg K)], *A* is the “virtual” average net opening area of opposite openings [m²], and Δc_p is the “virtual” difference in opening coefficients of pressure, which may be assumed as a fixed number or a domain for climate analyses, or defined by using wind direction correlated equations by defining the building dimensions of a “virtual” building [54]. The formula may be adapted to different calculation intervals.

In particular, the structural ventilation $CDH_{res,g}$ analysis is included in PREDYCE including a *cdd_res* indicator, while the other indices may be included during next project phases (WP3 and T8.5). Additionally, also the well-known Climate Cooling Potential index [55] is included in PREDYCE for PRELUDE purposes – see below.

cdd_res(df, tset=26, ef=0.5)

Compute Cooling Degree Days residuals.

Parameters

- **df** – DataFrame containing at least “Date/Time” and “T_db_o[C]” columns
- **tset** (*int, optional*) – Setpoint temperature, defaults 26
- **ef** (*float, optional*) – free parameter, defaults to 0.5

Returns

Cooling degree days residuals, int

CDH_{res} indicators for other-two considered passive low-energy technologies (DEC and EAHE) are computed inside PREDYCE KPIs calculator module in two different ways, depending on the value of the *with_control* Boolean parameter. The latter allows to include a “virtual” control in the system operation allowing to exclude its activation under unfavourable conditions by simulating a control action. In particular, if *with_control* is set to *False*:

$$CDH_{res,tech} = \sum m_h \cdot (\vartheta_{treat,h} - \vartheta_b) \quad \begin{cases} m_h = 1 \leftarrow \vartheta_{treat,h} > \vartheta_b \\ m_h = 0 \leftarrow \vartheta_{treat,h} \leq \vartheta_b \end{cases}$$

Instead, if *with_control* is set to *True*:

$$CDH_{res,tech} = \sum \begin{cases} (\vartheta_{treat,h} - \vartheta_b) & \text{if } \vartheta_{e,h} > \vartheta_b \wedge \vartheta_{e,h} \geq \vartheta_{treat,h} \\ (\vartheta_{e,h} - \vartheta_b) & \text{if } \vartheta_{e,h} > \vartheta_b \wedge \vartheta_{e,h} < \vartheta_{treat,h} \\ 0 & \text{if } \vartheta_{treat,h} \leq \vartheta_b \wedge \vartheta_{e,h} \leq \vartheta_b \end{cases}$$

Where:

$\vartheta_{e,h}$ is the outdoor dry bulb temperature, $\vartheta_{treat,h}$ is the outlet air of the considered system that can be used to cool or heat the building spaces, and ϑ_b is the considered base temperature.

The $\vartheta_{treat,h}$ is computed according to the considered technology:

$$\vartheta_{treat,h,PDEC} = \vartheta_{e,h} - slope \cdot (\vartheta_{e,h} - \vartheta_{WBT,h}) \quad [^{\circ}C]$$

$$\vartheta_{treat,h,EAHX} = \vartheta_{e,h} - slope \cdot (\vartheta_{e,h} - \vartheta_{soil,h}) \quad [^{\circ}C]$$

Where:

slope represents an efficiency factor of the system that depends by its configuration. Sample values are available in literature – e.g. see [10,41,56] –, $\vartheta_{WBT,h}$ is the wet bulb temperature of the airflow, representing the theoretical minimum temperature that may be reached during the adiabatic direct evaporative cooling process, and $\vartheta_{soil,h}$ is the soil temperature at the tube depth, representing the summer minimum and the winter maximum temperature that may be theoretically reached by the treated airflow.

And the soil temperature $\vartheta_{soil,h}$ for $\vartheta_{treat,h,EAHX}$ formula is computed according to the following well-known expression elaborated for hourly definition – see also Refs. [12,57]:

$$\vartheta_{soil,h} = \vartheta_{sf,av} - A_s \cdot \exp\left(-h \cdot \sqrt{\frac{\pi}{t_0 \cdot \alpha}}\right) \cdot \cos\left(\frac{2\pi}{t_0} \cdot (t - t_\varphi) - h \cdot \sqrt{\frac{\pi}{t_0 \cdot \alpha}}\right) \quad [^{\circ}C]$$

Where:

$\vartheta_{sf,av}$, A_s and t_φ are respectively the yearly soil average surface temperature, its amplitude and its phase (which is expressed in hours), when the external temperature reaches the minimum. These values are computed thanks to the EnergyPlus auxiliary program *CalcSoilSurfTemp.exe* if not given in input by the user. Then, t_0 is the year duration in hours, t the current time instant in hours from the beginning of the year, h is the soil depth at which earth tubes are located, and finally α the soil thermal diffusivity (which is given in input in m^2/s but here converted in m^2/h before the calculation).

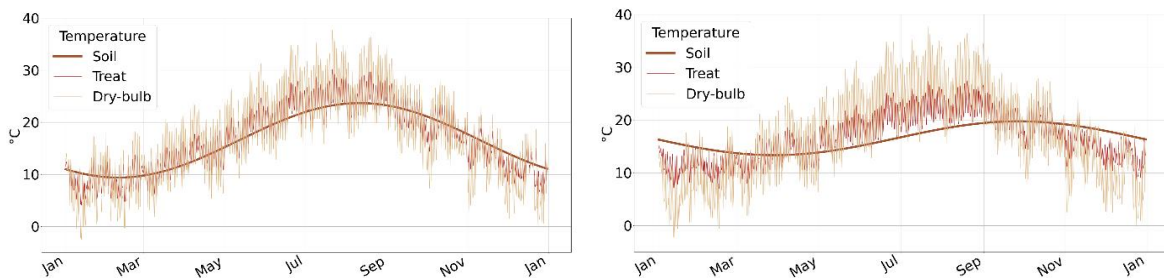
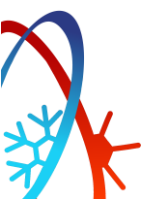


Fig. 9 – Soil temperature, EAHX outlet temperature and outdoor dry bulb temperature yearly trend on Rome TMY, considering a EAXH tube 0.5 m depth (a) and a 2.5 m depth (b)

Figure 9 shows the results of a sample application of those EAHX expressions via PREDYCE considering on the left a soil depth of 0.5 m, and on the right a soil depth of 2.5 m while all other parameters are assumed to be the suggested values given in the tool.



Eventually, EAHX technology can be also used to heat the environment. Hence, also HDH_{res} indicator can be defined, considering as base temperature e.g., 20 °C:

$$HDH_{res,eahx} = \sum \begin{cases} (\vartheta_b - \vartheta_{treat,h}) & \text{if } \vartheta_{e,h} < \vartheta_b \wedge \vartheta_{e,h} \leq \vartheta_{treat,h} \\ (\vartheta_b - \vartheta_{e,h}) & \text{if } \vartheta_{e,h} < \vartheta_b \wedge \vartheta_{e,h} > \vartheta_{treat,h} \\ 0 & \text{if } \vartheta_{treat,h} \geq \vartheta_b \wedge \vartheta_{e,h} \geq \vartheta_b \end{cases}$$

Moreover, it is also possible to divide CDH_{res}/HDH_{res} for 24 hours obtaining a simplified CDD_{res}/HDD_{res} indicator for the two technologies.

The PDEC climate action is described here below:

cdh_res_pdec(df, t_b=24, eff=0.8, with_control=True)

Compute CDH_{res} considering PDEC technology.

Parameters

- **df** (class: *pandas.core.frame.DataFrame*) – DataFrame containing at least “Date/Time”, “T_db_o[C]” and “RH_o[%]” columns
- **t_b** (*float, optional*) – default 24
- **eff** (*float, optional*) – default 0.8
- **with_control** (*bool, optional*) – default True

Returns

CDH_{res_PDEC} , float

The EAHX climate action is described here below:

cdh_res_eahx(df, var1, var2, var3, diffuse=6.177*10⁻⁷, depth = 2.5, eff=0.52, t_b=24, with_control=True)

Compute CDH_{res} considering EAHX technology.

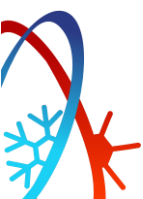
Parameters

- **df** (class: *pandas.core.frame.DataFrame*) – DataFrame containing at least “Date/Time” and “T_db_o[C]” columns
- **var1** (*float*) – average soil surface temperature
- **var2** (*float*) – amplitude of soil surface temperature
- **var3** (*float*) – phase of soil surface temperature
- **diffus** (*float, optional*) – soil thermal diffusivity in m²/s, defaults to 6.177·10⁻⁷
- **depth** (*float, optional*) – soil depth of EAHX tubes in meters, defaults to 2.5
- **eff** (*float, optional*) – EAHX efficiency, defaults to 0.52
- **t_b** (*float, optional*) – base temperature, defaults to 24
- **with_control** (*bool, optional*) – whether to sum positive values only when control condition is satisfied, defaults to *True*

Returns

CDH_{res_EAHX} , float

Among climate KPIs inside PREDYCE it can be also computed the Climate Cooling Potential (CCP). This index bases on the difference between internal and external air temperatures and it is expressed in Kelvin hour [Kh]. It can be used for analysing in the given period the ventilative cooling potential considering its potential usage over 24h or limited to specific scheduled potentialities, such as night cooling activation when daily mechanical cooling systems are turned off. The latter case is the original one described in literature - see [55,58] – and it can be applied in office and tertiary buildings in which mechanical cooling



is used during daytime to assure standard required comfort conditions, but night ventilative cooling can be activated during unoccupied hours (e.g., 19:00-7:00) to cool-down internal spaces during free-floating periods. The adopted formula assumes internal “virtual” temperature variations by defining a 24h-sinusoidal trend around 24.5°C of $\pm 2.5^\circ\text{C}$ (max temperature is set at 19:00) considering as “virtual” conditions a daily-conditioned office area. If CCP indicator has to be computed on a specific range of hours, this can be specified with the *hours_ccp* parameter by listing all selected ones, e.g. [0-1-2-3-4-5-6-19-20-21-22-23] for the 19:00-7:00 interval, otherwise it will be computed considering all 24h. Additionally, the daily averaged indices CCP_d may be defined by dividing the CCP by the number of days in the considered period.

ccp(df, delta_T=2, hours_ccp=None)

Compute Climate Cooling Potential

Parameters

- **df** (class:*pandas.core.frame.DataFrame*) – DataFrame containing at least “Date/Time” and “T_db_o[C]” columns
- **delta_T** (*int, optional*) – Delta temperature, defaults to 2
- **hours_ccp** (*list of int, optional*) – list of hours to be included in ccp computation, defaults to None

Returns

Climate Cooling Potential and daily Climate Cooling Potential in a dictionary with keys “ccp” and “ccpd”

The adopted formula inside *ccp* PREDYCE method is:

$$CCP = \sum m_h \cdot (\vartheta_{b,h} - \vartheta_{e,h}) \quad \begin{cases} m_h = 1 \leftarrow \vartheta_{b,h} - \vartheta_{e,h} \geq 2^\circ\text{C} \\ m_h = 0 \leftarrow \vartheta_{b,h} - \vartheta_{e,h} < 2^\circ\text{C} \end{cases}$$

Where:

$$\vartheta_{b,h} = \left(24.5 + 2.5 \cdot \cos\left(\frac{2\pi(h - 19)}{24}\right) \right) [^\circ\text{C}]$$

and *h* is the number of the daily hour [1-24]. A critical difference in temperature between base and environmental temperature of 2 °C is assumed by default [29], but it is a customizable input parameter.

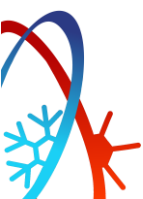
Another well-known indicator called Q_{ach} [Wh] allows to evaluate the climate heat gain dissipation potential of ventilative cooling. By knowing the airflow rate (that can be also automatically derived by IDF properties by the developed KPI method) and the air thermal properties (which are set by default, but can be also passed as input parameters) it is possible to calculate the Q_{ach} indicator with the following expression:

$$Q_{ach} = \sum m_h \cdot ACH \cdot Vol \cdot \rho_{air} \cdot c_{air} \cdot (\vartheta_{set} - \vartheta_{e,h}) \cdot 1h \quad \begin{cases} m_h = 1 \leftarrow \vartheta_{e,h} \leq (\vartheta_{set} - \Delta T)^\circ\text{C} \\ m_h = 0 \leftarrow \vartheta_{e,h} > (\vartheta_{set} - \Delta T)^\circ\text{C} \end{cases}$$

[Wh]

Where:

The variable *ACH* is the number of air change per hour, assumed by default equal to 2.5; *Vol* is the building-thermal-zone net volume [m³]; air density and heat capacity are assumed respectively to be 1.2 [kg/m³] and 0.28 [Wh/kg K], while the set point temperature ϑ_{set} is assumed by default equal to 26 °C. Finally, ΔT , assumed by default 2°C, is the minimal difference in temperature from the set point to consider a sufficient activation of ventilative cooling.



qach(df, vol, ach=2.5, pair=1.2, cair=0.278, tset=26, delta_T=2)

Compute climate heat gain dissipation potential of ventilative cooling, Qach.

Parameters

- **df** (class: *pandas.core.frame.DataFrame*) – DataFrame containing at least "Date/Time" and "T_db_o[C]" columns
- **vol** (float) – Volume of the building
- **ach** (float, optional) – Air changes per hour (ventilation), defaults to 2.5
- **pair** (float, optional) – Air density in kg³/m, defaults to 1.2
- **cair** (float, optional) – Air specific heat capacity in Wh/kg·K, defaults to 0.278
- **tset** (int, optional) – Setpoint temperature in °C, defaults to 26
- **delta_T** (int, optional) – Delta temperature in °C, defaults to 2

Returns

Qach, float

Similarly, the ventilative cooling potential referred to EAHX and PDEC technologies can be defined with a similar formula:

$$Q_{c,tech} = \sum 3600 \cdot A \cdot v_{air} \cdot \rho_{air} \cdot c_{air} \cdot (\vartheta_{e,h} - \vartheta_{treat,h}) \cdot 1h \begin{cases} m_h = 1 \leftarrow \vartheta_{treat,h} \leq (\vartheta_{e,h} - \Delta T) \text{ °C} \\ m_h = 0 \leftarrow \vartheta_{treat,h} > (\vartheta_{e,h} - \Delta T) \text{ °C} \end{cases}$$

[Wh]

Where:

Variable *A* is the area of EAHX tube or PDEC tower and *v_{air}* the airflow velocity [m/s]. In this expression 3600 [s/h] is the number of seconds in the hourly interval considering that *c_{air}* is expressed in [Wh/(kg°C)]. The analysis is conducted for each hourly interval.

Considering EAHX technology a similar formula can be used to define the system pre-heating potential:

$$Q_{h,eahx} = \sum 3600 \cdot A \cdot v_{air} \cdot \rho_{air} \cdot c_{air} \cdot (\vartheta_{treat,h} - \vartheta_{e,h}) \cdot 1h \begin{cases} m_h = 1 \leftarrow \vartheta_{e,h} \leq (\vartheta_{treat,h} - \Delta T) \text{ °C} \\ m_h = 0 \leftarrow \vartheta_{e,h} > (\vartheta_{treat,h} - \Delta T) \text{ °C} \end{cases}$$

[Wh]

Another KPI added to PREDYCE for PRELUDE allows to compute the average values of the wet bulb depression (defined as the difference between dry bulb and wet bulb temperatures) in the defined periods: the default working of the method retrieves a dictionary with monthly averages in the simulated period, however it is also possible to specify a desired period in which computing the average through *start* and *end* parameters, allowing also seasonal analyses.

wbd_averages(df, timestep='1M', start=None, end=None)

Compute timeseries of wetbulb depression according to requested time aggregation (average on the time period).

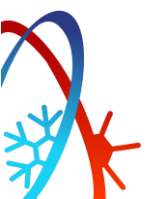
Parameters

- **df** (class: *pandas.core.frame.DataFrame*) – DataFrame which must contain at least outdoor drybulb temperature values "T_db_o[C]", relative humidity "RH_o[%]" and "Date/Time" columns.
- **timestep** (str, defaults to "1M") – time aggregation required
- **start** (str, defaults to None) – optional parameter, format "%d-%m"
- **end** (str, defaults to None) – optional parameter, format "%d-%m"

Returns

DataFrame of wetbulb depression values over time

Additionally, it has been included a KPI able to calculate the relative humidity (RH%) of an airflow when the mixing ratio (moisture content) is known. This analysis may be adopted for example to analyse if in an



EAHX a potential condensation phenomenon arrives during the cooling season. Sensible cooling is not in fact affecting the moisture content, nevertheless, when the air temperature reaches the dewpoint condensation occurs with a consequent moisture reduction. This phenomenon may be detected by verifying if outlet RH% is lower than 100%, otherwise a different mixing ration will be calculated for the treated air temperature assuming that RH% reaches 100%. This sample application is needed to represent on a psychrometric chart the EAHX treated air flow [41]. The same expression may be adopted in those databases in which humidity is described by mixing ratio.

$$RH\% = \frac{X \cdot p}{0.622 \cdot p_{vs} + X \cdot p_{vs}} \cdot 100\%$$

Where:

X is the mixing ratio [kg/kg] (e.g., computed from outdoor dry bulb temperature and relative humidity for the given EAHX application sample); p [Pa] = $p_{atm} + p_{dym}$, $p_{dym} = 0.5 \cdot \rho \cdot v^2$, $\rho = 353.118/\theta_{ab}[K]$, v is the airflow velocity and θ_{ab} the dry bulb temperature [K] (e.g., the treated temperature of EAHX tube); and finally $\ln p_{vs} = ((A \cdot \theta_{ab})/(B + \theta_{ab})) + C$ with $A = 22.376$, $B = 271.68$, $C = 6.4146$ if $-40^\circ\text{C} < T \leq 0^\circ\text{C}$ and $A = 17.438$, $B = 239.78$, $C = 6.4147$ if $0^\circ\text{C} < T < 40^\circ\text{C}$.

timeseries_rh_from_mix_ratio(df, timestep='1H')

Compute timeseries of relative humidity [%] starting from mixed ratio vapour quality [kg/kg] according to requested time aggregation (average on the time period).

Parameters

- **timestep** (*str*, defaults to "1H") – time aggregation required
- **df** (class:*pandas.core.frame.DataFrame*) – Dataframe which must contain at least drybulb temperature values "T_db[C]", vapour quality "mix_ratio[kg/kg]" and "Date/Time" columns.

Returns

DataFrame of relative humidity [%]

Another added psychrometric indicator, which is described below in this climate section, allows to compute the wet bulb temperature of an airflow giving the dry bulb temperature and the relative humidity.

Among those KPIs that are mainly devoted to graphs generation, the *carpet_plot* method can receive in input different environmental variables and adapt plot generation to the different cases, e.g., outdoor dry bulb temperature, wet bulb depression. Examples are shown in Figure 10 (a).

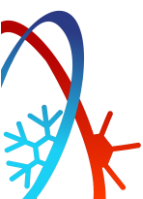
carpet_plot(df, variable, title=None)

Generate a carpet plot. Currently supported variables are: indoor and outdoor drybulb temperature "T_db_i[C]"/"T_db_o[C]", indoor and outdoor CO₂ concentration "CO₂_i[ppm]"/"CO₂_o[ppm]", indoor operative temperature "T_op_i[C]", the Predicted Percentage of Dissatisfied (PPD) "ppd", and the wetbulb depression "WBD[C]".

Parameters

- **df** (class:*pandas.core.frame.DataFrame*) – DataFrame which must contain Date/Time and variable column.
- **variable** (*str*) – variable name according to standardized nomenclature
- **title** (*str*, optional) – Title of the figure, defaults to None

Moreover, a new developed method for PRELUDE applications is in charge to generate psychrometric charts, exploiting the Psychrochart Python library functionalities [59], which is based on psychroLib [60]. Psychrometric charts are fundamental to apply bioclimatic analyses at both climate and building levels [5]. The chosen carrier-based chart is based on Givoni studies [10,61,62] to support the well-known approach to define, since early-design stages, the applicability of different natural and low-energy techniques, such as ventilative cooling, evaporative cooling, internal masses – see also [63]and [9]. An example of PREDYCE graphical output for this indicator is shown in Figure 10 (b), where both mixed ratio vapour quality and



Givoni comfort zone are highlighted. Moreover, this method can compute the number of points falling inside and outside the comfort zone and return the aggregated results over the considered time period in the *data_res.csv* file. The comfort zone can be adapted under specific request to different standards or references, including ASHRAE Handbook of fundamentals (up to 2005) or current ASHRAE Standard 55. Additionally, extended comfort zones, e.g., high thermal mass, natural ventilation, fan-forced ventilation for cooling, direct evaporative cooling and others, may be also included during future expansions of the tool, in line with different studies – e.g. see Givoni-Milne [62] or others [41]. Specific applications of these analyses are planned in PRELUDE during the activities of T8.5.

carrier_plot(range_temp_c=[0, 35], range_humidity_g_kg=[0, 25], altitude_m=0, data=None)

Generate a Carrier psychrometric chart.

Parameters

- **range_temp_c** (*list, optional*) – x-axis limits, defaults to [0, 35]
- **range_humidity_g_kg** (*list, optional*) – y-axis limits, defaults to [0, 25]
- **altitude_m** (*int, optional*) – Altitude in meters, used to perform calculations of absolute humidity, defaults 0
- **data** (*class:pandas.core.frame.DataFrame, optional*) – Points to plot on the chart, passed as a DataFrame containing dry_bulb_temperature and relative_humidity columns, defaults to None

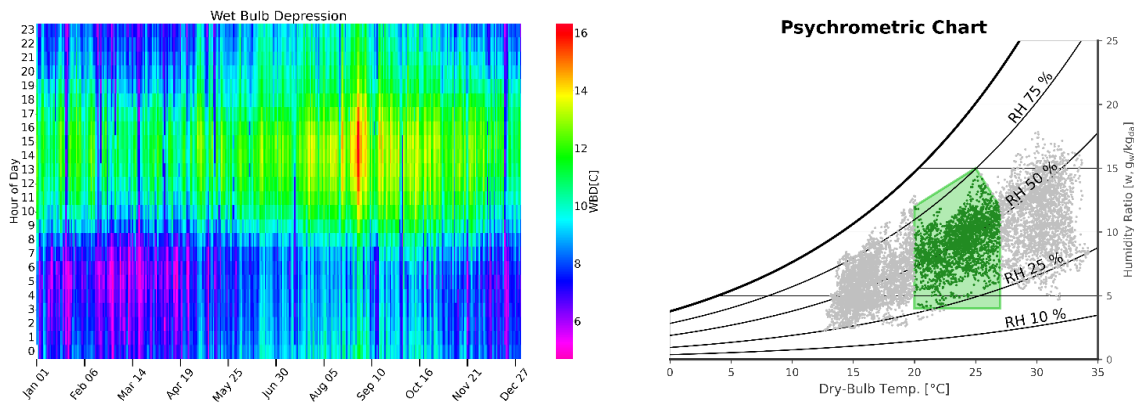


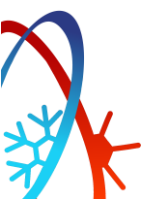
Fig. 10 – Examples of carpet plot (a) and carrier plot (b) generated in PRELUDE

Finally, several indicators can be also returned in timeseries format and stored in the *data_res_timeseries.csv* file associated to each simulation, in which each row is a timestep and each column a KPI. Examples of indicators that are currently available in this format are: dry bulb and wet bulb temperatures, wet bulb depression, relative humidity and vapour quality mixed ratio (computed starting from relative humidity and dry bulb temperature according to the well-known psychrometric expression – see also above). As an example, Figure 11 (a) shows wet bulb temperatures results computed according to the Stull formula [64] as a function of the dry bulb temperature θ_{db} [°C] and of the relative humidity rh [%] (where a value such as 50.5% is imputed as the number 50.5) inside *timeseries_wbt* method:

$$\theta_{wb} = \theta_{db} \cdot \tan^{-1}(0.151977 \cdot \sqrt{rh + 8.313659}) + \tan^{-1}(\theta_{db} + rh) - \tan^{-1}(rh - 1.676331) + 0.00391838 \cdot (rh)^{\frac{3}{2}} \cdot \tan^{-1}(0.023101 \cdot rh) - 4.686035$$

The validity range is identified following original paper images, shown in Figure 11 (b), and is defined as:

$$f(x) > \begin{cases} -\frac{70}{29}x + \frac{775}{29}, & x < 9 \\ 5, & x \geq 9 \end{cases}$$



timeseries_wbt(df, timestep='1H')

Compute timeseries of outdoor wetbulb temperature according to requested time aggregation (average on the time period). It also return a “check” column which says which points are outside the Stull validity range (1).

Parameters

- **df** (class: *pandas.core.frame.DataFrame*) – Dataframe which must contain at least outdoor drybulb temperature values “T_db_o[C]”, relative humidity “RH_o[%]” and “Date/Time” columns.
- **timestep** (*str, optional*) – time aggregation required, defaults to “1H”

Returns

DataFrame of outdoor wetbulb temperature values over time

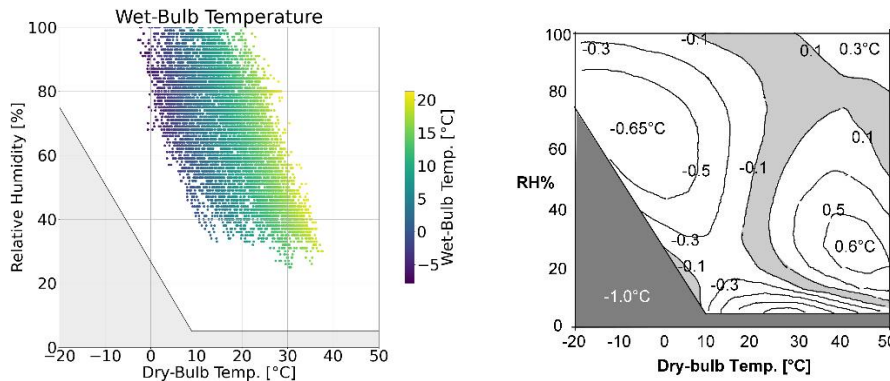


Fig. 11 – Wet Bulb Temperature computed in PREDYCE (a) and Stull validity range re-elaborated by [64] (b)

2.3.II Building analyses

Another group of KPIs computable inside KPIs calculator module concerns building related indicators. Those indicators can be used to find correlations with climate ones, understanding climate impact on indoor thermal comfort and consumption. Those building indicators refers to dynamic building simulation results using the EnergyPlus engine. Furthermore, several of them may also be applied to monitored data or to other databases to retrieve the specific KPI.

Energy consumptions and needs are at present considered in PREDYCE as outputs of a simple HVAC system, but it can be possible to integrate correction factors due to utilisation losses. The currently implemented approach adopts four consecutive correction factors: emission, regulation, distribution, and generation – see UNI-TS 11300-2 concerning heating and domestic hot water and UNI-TS 11300-3 concerning cooling – which are by default set equal to 1 in PREDYCE such returning the EnergyPlus District Heating and Cooling consumption. Although, specific values may be assigned to each of those factors. Additionally, the action allows to include an EP conversion factor – that may be left equal to 1 if not needed – in order to retrieve primary energy values.

class predyce.kpi.EnergyPlusKPI(df, idf, plot_dir, start_date=None, end_date=None, graph=False)

Q_c(em_factor=1, reg_factor=1, distr_factor=1, gen_factor=1, ep_factor=1)

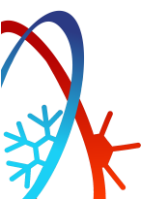
Prepare eplosout.csv to compute primary energy need for cooling in kWh/m², applying correction factors for emission, regulation, distribution and generation losses, then multiplying by primary energy factor.

Parameters

- **em_factor** (*float, optional*) – correction factor for emission losses, in range [0,1], defaults to 1.
- **reg_factor** (*float, optional*) – correction factor for regulation losses, in range [0,1], defaults to 1.
- **distr_factor** (*float, optional*) – correction factor for distribution losses, in range [0,1], defaults to 1.
- **gen_factor** (*float, optional*) – correction factor for generation losses, defaults to 1.
- **ep_factor** (*float, optional*) – primary energy conversion factor, defaults to 1.

Returns

Primary energy need for cooling, float



```
class predyce.kpi.KPI(plot_dir, graph=False)
```

```
Q_c(df, gen_factor=1, ep_factor=1)
```

Compute primary energy need for cooling in kWh/m², applying correction factors for generation losses, then multiplying by primary energy factor. Corrections for emission, regulation and distribution losses are considered previously applied.

Parameters

- **df** (class:*pandas.core.frame.DataFrame*) – Dataframe which must contain at “Q_c[kWh/m²]” and “Date/Time” columns.
- **gen_factor** (*float, optional*) – correction factor for generation losses, defaults to 1.
- **ep_factor** (*float, optional*) – primary energy conversion factor, defaults to 1.

Returns

Primary energy need for cooling, float

Besides consumption values, another indicator, *n_h_kwh*, allows to identify the number of hours in which HVAC system is on, since among PRELUDE goals there is not only the development of mitigation techniques able in reducing the intensity of the energy needs, but also the definition of their impact on comfort/discomfort number of hours considering machine on/off periods. In this case the seasonal coefficient of performances (COP and EER) of systems can be inputted.

```
n_h_kwh(df, th_list=[0, 0.6, 1], cop_cool=3, cop_heat=0.3)
```

Number of hours HVAC consumption is higher then given thresholds.

Parameters

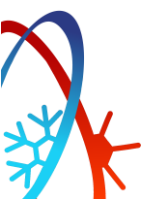
- **df** (class:*pandas.core.frame.DataFrame*) – Dataframe which must contain at least cooling and heating consumption columns “Q_c[Wh/m²]” and “Q_h[Wh/m²]” plus “Date/Time” column.
- **th_list** (*list, optional*) – List of thresholds to be compared with consumption, defaults to [0,0.6,1]
- **cop_cool** (*int, optional*) – Cop value for cooling system, defaults to 3
- **cop_heat** (*float, optional*) – Cop value for heating system, defaults to 0.3

Returns

Dictionary containing No. hours above defined thresholds for heating and cooling.

Another significative indicator correlating consumption to outdoor conditions is the energy signature, which is returned by the *energy_signature* method in dictionary format considering both the 1-dimensional and 2-dimensional signatures proposed in the literature, e.g. see [65–68] and Annex B of the EN 15603:2008, which correlates heating and cooling weekly averaged consumption with outdoor dry bulb temperature and global solar radiation. The returned dictionary contains needed points for energy signatures plots generation, but it is also possible to directly generate the graphs for each performed simulation, as shown in Figure 12. At present, for x-axis it is considered by default the difference between indoor and outdoor temperatures, but it is possible to substitute the difference in temperature with the outdoor temperature, allowing for different analyses.

```
{
  "1D": {
    "cooling": {"deltaT": list of 2 points, "cooling": list of 2 points},
    "heating": {"deltaT": list of 2 points, "heating": list of 2 points},
  },
  "2D": {
    "cooling": {"deltaT": list of 4 points, "solarRadiation": list of 4 points, "cooling": list of 4 points},
    "heating": {"deltaT": list of 4 points, "solarRadiation": list of 4 points, "heating": list of 4 points},
  }
}
```



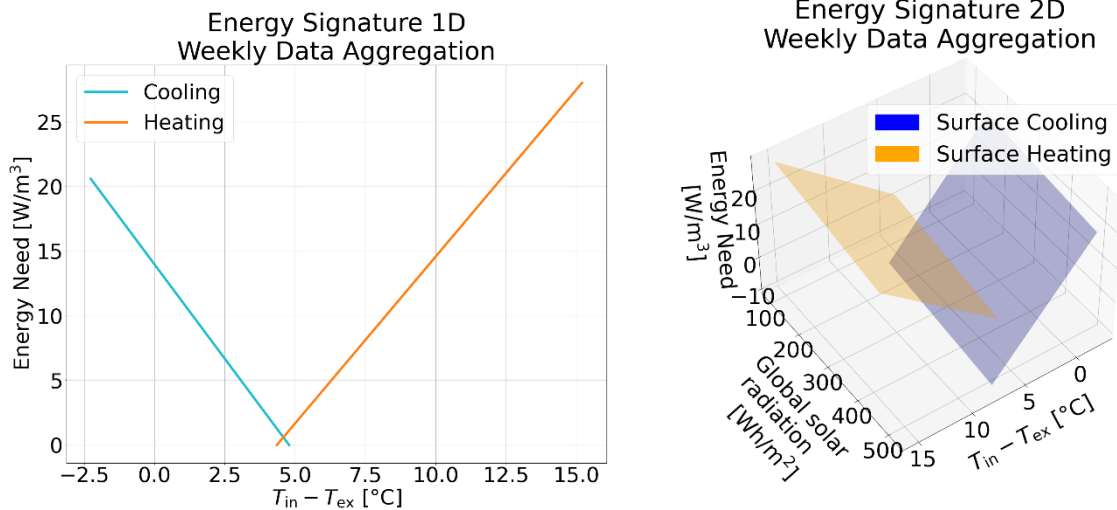


Fig. 12 – Example of 1D and 2D energy signature graphs generated inside PREDYCE

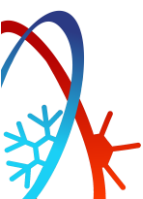
Considering instead indoor thermal comfort, Adaptive Comfort Model (ACM) categories are considered in case of a building without HVAC system installed. Both old (15251:2007) and new (16798-1:2019) European standards can be used for the computation of the running mean temperature (with the new norm set as default procedure). Hours in categories III and IV are considered for thermal discomfort in order to compute the POR (Percentage of hours Outside of Range), while for a deeper analysis all categories distribution can be retrieved. It is also possible to filter hours according to occupancy, while the field *when* can be used to reduce the analysed period with respect to full simulation period. Figure 13 shows an example of graphical ACM hourly points distribution retrievable through the PREDYCE function.

static compute_acm()

Compute running mean according to different European normatives. Filters according to occupancy or dates can be applied.

Parameters

- **df** (class: *pandas.core.frame.DataFrame*) – dataframe should contain "Date/Time" column in format 'year/month/day hour:minutes:seconds', " T_{db_o} [C]" preferably with a subhourly timestep and " T_{op_i} [C]". Optional "Occupancy" column accepting only 0/1 values.
- **eu_norm** (str, optional) – It can be set to '15251:2007' if old UE norm computation is desired, defaults to '16798-1:2019'.
- **alpha** (float, optional) – With old UE norm '15251:2007' alpha is a free parameter in range [0,1), defaults to 0.8
- **filter_by_occupancy** (int, optional) – It can be set 0 or 1, depending on whether activate occupancy filtering on thermal comfort KPIs computation or not, defaults to 0.
- **when** (dict, optional) – dictionary with 'start' and 'end' keys and values in format 'year/month/day hour:minutes:seconds'



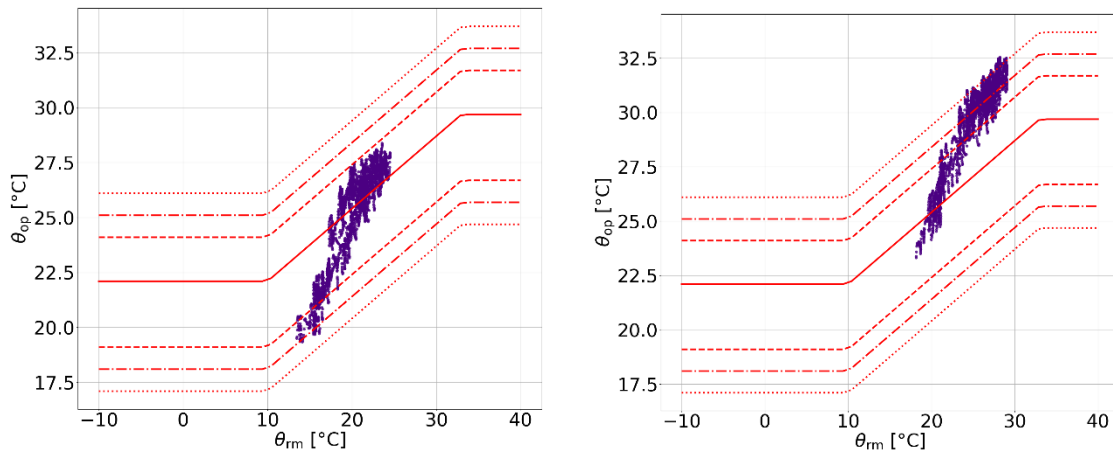


Fig. 13 – Example of Adaptive Comfort Model points distribution plots in colder (a) and hotter (b) climates

Other indicators that can be returned are called residual cooling and heating (*res_cool/res_heat*): they represent the residual amount of energy consumption needed to bring indoor temperatures inside comfort levels when a building operating in free-running mode is experiencing discomfort conditions higher than the acceptable ones. These indices represent the residual energy needs that may be potentially given by personal cooling/heating systems or domestic split units that may be turned on under request or on given controlling conditions to absorb summer and/or winter discomfort peaks. Indirectly, these indicators may be also adopted during design stages to analyse the possible intention of users to install a system, e.g. a split unit for summer comfort. The residual cooling and heating are the result of a double simulation: a building without mechanical system is simulated and ACM comfort categories evaluated for each hour, then the same building is simulated as if it had a mechanical system installed and its hourly consumption is saved as a result. The hourly energy consumption is summed up entirely if the building is in ACM discomfort, while it is summed up with a linear function of the distance from the central comfort line if it is in ACM category II, and finally nothing is summed if the building is in category I, resulting in the total residual heating or cooling energies.

res_heat(dataframe_off, dataframe_on, eu_norm='16798-1:2019', alpha=0.8)

Return residual heating computed as the amount of heating consumption needed in a building without mechanical system in order to reach category I comfort of adaptive comfort model.

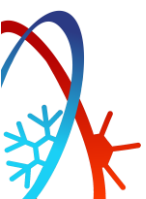
Parameters

- **dataframe_off** (class: *pandas.core.frame.DataFrame*) – dataframe containing at least "Date/Time", "T_op_i[C]", "T_db_o[C]" columns in order to compute adaptive comfort model categories.
- **dataframe_on** (class: *pandas.core.frame.DataFrame*) – dataframe containing at least "Date/Time" and "Q_h[kWh/m2]" columns, of the same building with HVAC installed and ACH ventilation set to 0.
- **eu_norm** (*str, optional*) – It can be set to '15251:2007' if old UE norm computation is desired, defaults to '16798-1:2019'.
- **alpha** (*float, optional*) – With old UE norm '15251:2007' alpha is a free parameter in range [0,1), defaults to 0.8

Returns

residual heating total [kWh/m2], float

Considering indoor comfort KPIs, the cumulative discomfort intensity for free-running buildings can be evaluated by the Cooling Internal Degree Hours (CIDH) and Heating Internal Degree Hours (HIDH) indicators [69]. These indicators are the building related counterpart of the CDH/HDH indicators described in the previous section and they have been consequently included in PREDYCE for PRELUDE analyses. Both indicators return a dictionary of values with respect to same comfort thresholds considered for CDH and HDH indicators. Adopted formulas are:



$$CIDH = \sum m_h \cdot (\vartheta_{i,h} - \vartheta_c) \quad \begin{cases} m_h = 1 \leftarrow \vartheta_{i,h} > \vartheta_c \\ m_h = 0 \leftarrow \vartheta_{i,h} \leq \vartheta_c \end{cases}$$

Where:

the variable m_h is the activation check and ϑ_c the cooling comfort temperature, assumed by default ranging in the domain {18°C; 20; 22; 24; 26; 28} such as for the CDH indicator, but it is also customizable, while $\vartheta_{i,h}$ is the hourly indoor dry bulb temperature.

cidh(df, thresholds=[18, 20, 22, 24, 26, 28])

Compute standardized CIDH (Cooling Internal Degree Hours) computation and finally return sum of residuals from defined thresholds.

Parameters

- **df** (class: *pandas.core.frame.DataFrame*) – dataframe containing at least “Date/Time” and “T_db_i[C]” columns
- **thresholds** (*list, optional*) – temperature thresholds from which compute residuals, defaults to [18, 20, 22, 24, 26, 28]

Returns

dictionary containing sum of residuals from thresholds

Similarly, the HIDH is defined as:

$$HIDH = \sum m_h \cdot (\vartheta_c - \vartheta_{i,h}) \quad \begin{cases} m_h = 1 \leftarrow \vartheta_{i,h} \leq \vartheta_c \\ m_h = 0 \leftarrow \vartheta_{i,h} > \vartheta_c \end{cases}$$

Where:

m_h is the activation check, ϑ_c the heating comfort temperature assumed ranging in the domain {18°C; 20; 22} and customizable, and $\vartheta_{i,h}$ is the hourly indoor air temperature.

PREDYCE allows also to compute the PMV (Predicted Mean Vote) and PPD (Predicted Percentage of Dissatisfied) according to Fanger model calculation workflow [70–72]. In line with ISO 7730 standard and EN 16798-1, different comfort categories are adopted based on PMV ranges, category I (± 0.2 PMV), II (± 0.5 PMV) and III (± 0.7 PMV). Hours falling in category III are considered in thermal discomfort for POR computation. Default values are provided for Fanger model parameters, but they can also be set through the input JSON file: the clothing level is assumed to be 0.5 clo, while the metabolic rate is set to 1.2 met, corresponding to standing relaxed condition or sitting activities.

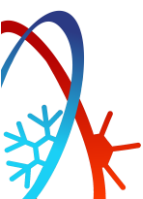
static compute_pmv_ppd()

Return Predicted Mean Vote (PMV) and Predicted Percentage of Dissatisfied (PPD) calculated in accordance to ISO 7730-2006 standard.

Parameters

- **df** (class: *pandas.core.frame.DataFrame*) – dataframe containing at least “Date/Time”, “T_db_i[C]”, “T_rad_i[C]” and “RH_i[%]” columns. Optional “Occupancy column” accepting only 0 and 1 values.
- **vel** (*float, optional*) – relative air speed, defaults 0.1
- **met** (*float, optional*) – metabolic rate, [met] defaults 1.2
- **clo** (*float, optional*) – clothing insulation, [clo] defaults 0.5
- **wme** (*float, optional*) – external work, [met] defaults 0
- **standard** (*str, optional*) – default “ISO 7730-2006”, but currently unused
- **filter_by_occupancy** (*int, optional*) – It can be set 0 or 1, depending on whether activate occupancy filtering on thermal comfort KPIs computation or not, default 0.

Moreover, several indicators can be returned in timeseries version with default hourly timestep, such as consumption values, distance from ACM central comfort line (instead of the hourly category), PPD.



Finally, in the PREDYCE tool it is possible to read all EnergyPlus outputs, specifying them in the input JSON *outputs* field, if they are included in the IDF output list. Moreover, considering the PDEC (*predyce.idf_editor.add_pdec*) and EAHX (*predyce.idf_editor.add_eahx*) actions, PREDYCE automatically includes, together with the IDF Editor action, also the EnergyPlus correlated output list in order to be able to read and analyse the specific system results, e.g., the outlet temperature of the EAHX system.

2.4 24h forecasting scenario

The following sections are focused on the description of additional PREDYCE scenarios, which, as already described, are task-oriented scripts combining all library potentialities to reach different goals. In this section it is introduced the 24h forecasting scenario that, developed specifically for PRELUDE, aims at suggesting optimized free-running strategies to users (also technical) and tenants in order to achieve best indoor comfort conditions (mainly in terms of thermal comfort evaluated through ACM) based on the weather forecasts for the next 24 hours. The scenario is mainly conceived to run on calibrated building models which have embedded realistic envelope values and schedules, e.g., for occupancy and HVAC (if present). Additionally, this scenario is organised in order to take into consideration both: i.) a fully free-running mode, which may be adopted even without actuators by supporting self-user actions, and ii.) a more complex mode in which low-energy technology systems with actuators are present in the building, e.g., mechanical ventilation systems allowing ventilative cooling and/or automatic shadings, which can be controlled locally or using remote-connections. The 24h forecasting scenario also exploits PREDYCE ability to generate EPW files from different sources, since it requires the integration of at least two weeks of monitored weather data (that could come from a weather station) for the simulation warm up period and of forecast data for the next day, e.g., retrieved by a weather forecasting service. Figure 14 shows the input/output workflow of the scenario: it takes in input the building model in IDF format, the JSON file and the EPW file, while it generates in output a CSV file containing optimized schedules for the considered “actuators” for the next 24 hours. The generated output can be then used as input to real actuators if present in the building (providing an hourly detailed schedule suggestion for mechanical systems or shading activation settings) or it can be translated into user-friendly messages to suggest manual intervention, in case in which no actuators are present in the building, e.g., by exploiting Telegram bots and/or simple graphical interfaces in a devoted app or web-service.

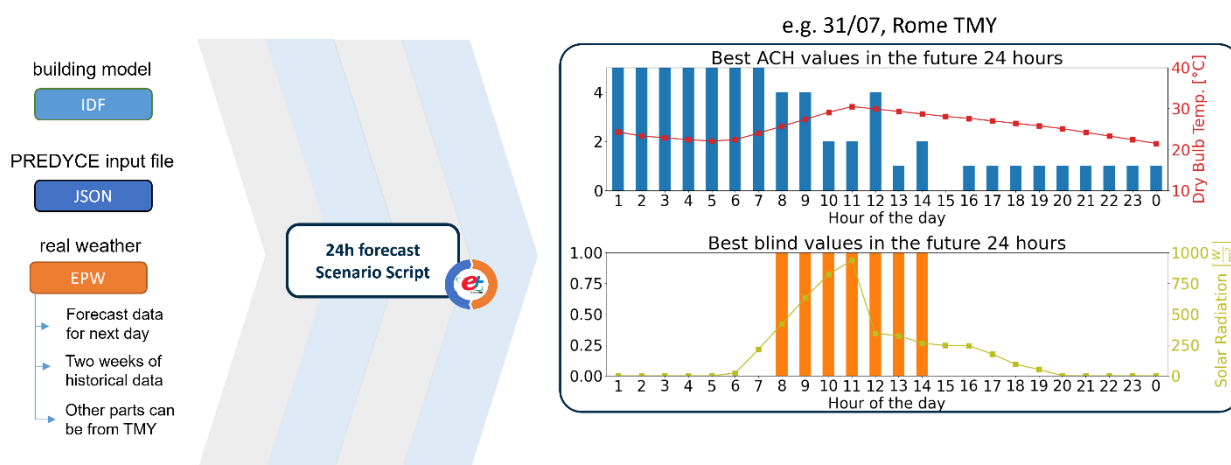
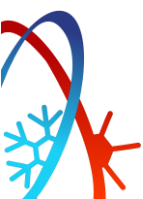


Fig. 14 – PREDYCE 24h forecast scenario workflow

In order to define plausible strategies and suggestions to be actuated when the scenario is used for manual self-actuating, the possibility of suggesting specific actions based on user availability is taken into consideration. For example, when the tenants are not at home or during night-time when people are sleeping, it is not possible that users may change the status of windows every hour. For this reason, the produced script allows to calculate optimisation values not only in a single hour interval, but also to process together specific blocks of hours that are hence optimized as a whole. For example, if tenants are expected



to not be at home in the 8:00-16:00 interval the same may be considered as a whole optimising shading and ventilation suggestions for the whole interval. This option opens the possibility to also explore how strategies like night cooling can improve or decrease comfort for the coming day. Similarly, strategies derived from IDF occupancy schedules analysis can be exploited. Through the input JSON file, it can be specified a preferred hours' organization in blocks. Also, through the input file it is possible to select the technologies to be optimized, e.g. allowing to combine the presence of an advanced scheduled system for mechanical ventilation and a manual intervention for shading control. Figure 15 shows an example of input JSON file for the 24h forecast scenario.

```

{
  "scenario": "24hf",
  "building_name": "mainblock",
  "day": "15/08",
  "hour_aggregations": [
    [0, 1, 2, 3, 4, 5, 6, 7, 8, 9],
    [10, 11, 12, 13, 14, 15, 16, 17],
    [18, 19, 20, 21],
    [22, 23]
  ],
  "forecast_actions": {
    "natural_ventilation": [1, 3, 5],
    "blind": []
  },
  "kpi": {
    "adaptive_comfort_model": {},
  }
}

```

Fig. 15 – Example of input JSON file for 24h forecast scenario

The 24h forecast scenario can be also executed from remote, e.g. exploiting the PRELUDE middleware (FusiX platform), thanks to a developed dedicate REST API. Such service can be accessed by a Web page (Figure 16) or by sending POST requests via a REST client; the body part of the request must be sent in a Multipart form that includes the building model in IDF format, the input JSON files that describes the scenario and the weather data in EPW format.

Welcome to PREDYCE API for 24h Forecast

Here you can insert building model, weather files and input file to perform a 24h forecast simulation via PREDYCE.

Please be patient during the simulation phase as it may take several minutes. Do not refresh the page during simulation phase or progress will be lost.

Please upload your files and submit.

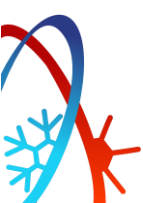
Building model in IDF v8.9 format:

Input file for 24h forecast in JSON format:

Weather data in EPW format:

Fig. 16 – 24 forecast REST API Web Page

In order to find the best strategies, the 24h forecasting script exploits a scoring system so that each building status depends both on the adaptive comfort model category that is achieved and also on bonus and malus points assigned to each strategy, for example, activating the shading will likely cool down the house (improved comfort) but may also significantly reduce the illuminance inside the rooms, so a malus will be assigned; the purpose is to find the best compromise to maximize comfort and user experience. The



currently implemented scoring system is an initial version that could be further modified and improved thanks to initial real application studies in the PRELUDE demo cases. Particularly, for each hour (or block of hours) lots of simulation are performed and the one with best impact on all following hours is chosen (e.g., open windows in the early morning can allow to reduce thermal discomfort for several hours ahead). In order to perform massive simulations in a short time (e.g., during the night for next day suggestions) the potentialities of a 128-core CPU server (256 threads), appositely adopted also for future PRELUDE tasks (climate database in T8.5 and Turin demo simulations), are exploited. Also, the server-to-server communication with PRELUDE middleware pass through this facility.

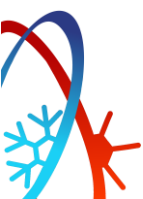
Finally, the 24h forecasting script main output is shown in Figure 17 it is a CSV file containing settings for the considered technologies for each hour. The file can be saved to support a timeline approach, by adding each day the best-found schedules by the tool in order to retrieve best values for a longer period of time. This potential longer log can be used to analyse, e.g., on a weekly or monthly base, the ability of the users in exploiting the optimised free-running potential of their living spaces by comparing the results retrieved by monitored and simulated conditions. Moreover, the CSV file is an easily translatable file into other formats, e.g., schedules for real actuators, plots or tabular form to be then exported and visualized in whatever modality, e.g., text messages, dashboard, or textual profiles to be then communicated to the users. Figure 14 (graphs on the right) shows an example of how the simulation results can be transcribed and visualized in a graphical format: in this specific case, the suggestions include the shading activation during the middle hours of day and take advantage of cool night temperatures to ventilate the building.

Date/Time	0_stack_opening_factor	90_stack_opening_factor	180_stack_opening_factor	270_stack_opening_factor
2021-08-19 01:00:00	0	0	0	0
2021-08-19 02:00:00	0	0	0	0
2021-08-19 03:00:00	0	0	0	0
2021-08-19 04:00:00	0	0	0	0
2021-08-19 05:00:00	0	0	0	0
2021-08-19 06:00:00	0	0	0	0
2021-08-19 07:00:00	0	0	0	0
2021-08-19 08:00:00	0	0	0	0
2021-08-19 09:00:00	0	0	0	0
2021-08-19 10:00:00	0	0	0	0
2021-08-19 11:00:00	0	0	0	0
2021-08-19 12:00:00	0	0	0	0
2021-08-19 13:00:00	0	0	0	0
2021-08-19 14:00:00	0	0	0	0.3
2021-08-19 15:00:00	0	0	0.3	0.7
2021-08-19 16:00:00	0	0.3	0.3	0.7
2021-08-19 17:00:00	0	0.7	0.7	0.7
2021-08-19 18:00:00	0	0.7	0.7	0.7
2021-08-19 19:00:00	0	0.3	0.3	0.7
2021-08-19 20:00:00	0	0	0.3	0.7
2021-08-19 21:00:00	0	0	0.3	0.7
2021-08-19 22:00:00	0	0	0.3	0.7
2021-08-19 23:00:00	0	0	0.3	0.7
2021-08-20 00:00:00	0	0	0.3	0.7

Fig. 17 – Example of 24h forecast CSV output: each actuator has its own suggested hourly schedule column

2.5 Verification scenario to support pre-calibration actions

Another of the developed PREDYCE scenarios of use has been conceived for model verification support. In fact, the 24h forecast scenario needs in input calibrated building models. The model verification scenario, differently from others, is a semi-automatic process, still requiring external intervention. In fact, it has been thought to support and speed up the calibration process usually performed by hand through an EnergyPlus CAD/BIM interface, but it does not completely substitute the human knowledge and expertise in the parameters' choice and eventual needed model adjustments. The calibration process exploits both PREDYCE EPW compiler module, able to generate an EPW from monitored weather data, and KPIs calculator module flexibility to work also on structured monitored data. In fact, model verification is made possible thanks to PREDYCE ability of computing same KPIs on both simulation results and indoor



monitored data. Particularly, the adopted procedure is inspired by [73] and allows to minimize a combined error measure including both RMSE (Root Mean Square Error) and MBE (Mean Bias Error) on a given variable or combination of variables, e.g., indoor dry bulb temperature in free-running conditions:

$$\text{Error}_{\text{tot}} = \sqrt{RMSE^2 + MBE^2}$$

The calibration signature described in [73] is computed according to the following equation (in the sample taking indoor dry bulb temperature as objective variable), exploiting PREDYCE potentialities.

$$\text{Calibration signature} = \frac{\text{measured } T_{\text{db}}^i - \text{simulated } T_{\text{db}}^i}{\text{max measured } T_{\text{db}}^i} \cdot 100\%$$

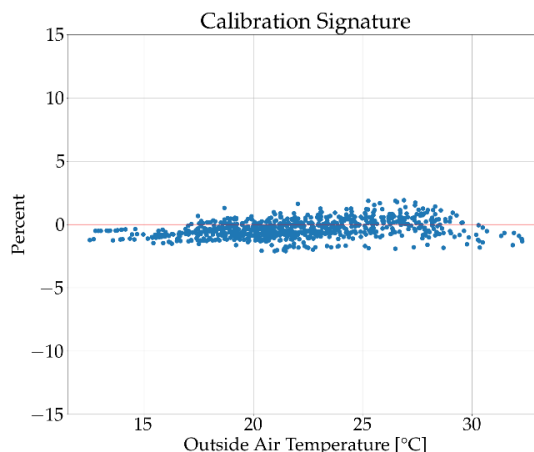
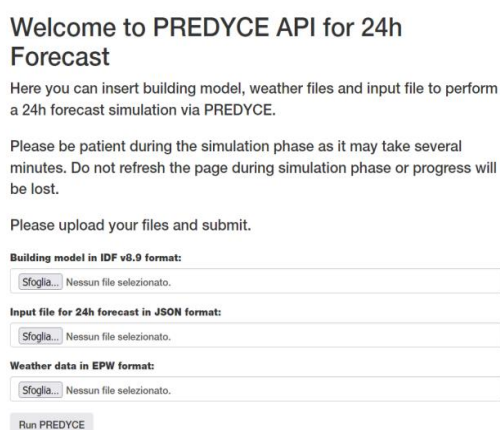


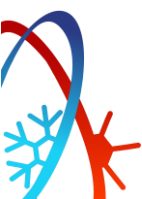
Fig. 18 – Example of calibration signatures generated inside PREDYCE. On the left the initial model results, on the right the model verified results.

Different implemented IDF editing actions are then applied parametrically to the initial building model to act on the obtained calibration signature curve shape allowing to reach a flat line inside 5% error range, as shown in Figure 18, in line with reference suggestions for calibration criteria, e.g. see ASHRAE Guideline 14-2014. Particularly, the main editing actions currently available inside this scenario are change U-value of walls and roof (acting on thickness of mostly insulated layer); change U-value and SHGC (Solar Heat Gain Coefficient) of windows; change internal masses and equipment gains in each thermal zone; change ACH ventilation and infiltration; but others can be also applied depending on the specific building (e.g., change zone area, occupancy profiles, windows area or visible transmission factor).

Which parameters to change and their changing ranges (e.g., change windows U-value in range 20% with respect to original model value) can be defined through the input JSON file considering the knowledge of each demo case. However, the semi-automatization consists in defining a subset of parameters to be tried and adjust them (both in range and typology) through several scripts executions by looking at calibration signatures and temporal trend, in order to minimize the number of simulations to be performed. This procedure may be furtherly automatized in the future developing a fully automated scenario, but the potentiality to test many building parameters simultaneously and automatically editing the model already provides an improvement in terms of effort and time with respect to traditional manual procedures. An example of the applicability of this PREDYCE scenario is detailed in Section 3.

2.6 Real vs simulated KPIs for free-running

Another PREDYCE scenario of use has been defined to allow comparisons of KPIs between simulated building models, after they have been calibrated, and actual building behaviours, leading to the identification of potential performance gaps. In line with PRELUDE goals, this scenario can be used to validate suggestions given by the 24h forecast scenario over time, comparing KPIs results under real



weather (not anymore forecasted) of the calibrated building with applied optimal schedules for free-running or low-energy technologies and of the real monitored building. This specific analysis may support a check on actual usage of the local free-running potential by comparing optimized simulations with real user-managed monitored spaces. Clearly, simulation models need to be verified before in order to reduce discrepancies due to bad modelling choices and maximise the focus of the analysis on the ability to exploit the optimal free-running potential. Also, this scenario could be used to verify the potential impact of seasonal thresholds applied to actuators (e.g., best temperature or solar thresholds for shading automatic activation) considering the current building usage.

Additionally, the real-vs-simulated scenario can be used to underline discrepancies between real and expected building behaviour according to standard definition of occupancy, ventilation and other building settings defined in line with European and/or MS standards, e.g. in line with EN 16798-1:2019, to support suggestions for improvements in building/user behaviours – e.g., “you have consumed more than a standard user”; “you have worse IAQ in respect to standard usage”; “your self-actuator strategy is working better than simulated automatic scenario”.

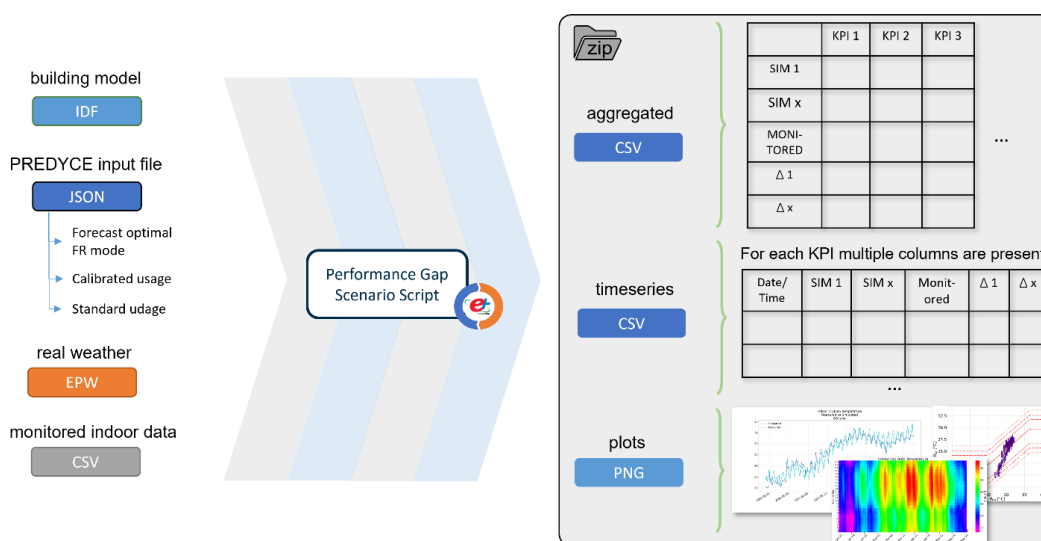
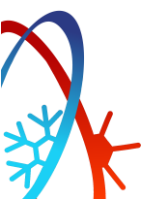


Fig. 19 – PREDYCE performance gap scenario workflow

Figure 19 shows the PREDYCE performance gap scenario workflow: exploiting the *preliminary_actions* field inside the input JSON file it is possible to define different building settings that could be compared with monitored data for that building, e.g., imposing best found schedules through the 24h forecast scenario over several days or weeks, or imposing a standard building setting, under real weather conditions (at least one week of real weather data before the desired simulation period is needed for simulation warm up). Since this scenario involves comparisons between indoor monitored and simulated data, besides adopting the correct variables nomenclature for KPIs computation, it requires a correct spatial association among sensors located in the building and model thermal zones, such that spatial aggregations for KPIs analysis correspond. This could be reached both through translation codes, specific to each application, or by adopting since the beginning the nomenclature schema proposed in Figure 20 for sensors naming: the part of the sensor name preceding the sensor identifier (e.g., the MAC address) follows the IDF-model-naming structure made of three levels, i.e., building name, block name and thermal zone name. Hence, if the same three level structure is used both inside the building model (when initially creating it through the EnergyPlus CAD/BIM interface) and for sensors names, the performance gap scenario is automatically able to compute KPIs on different building spatial aggregations as specified in the *aggregations* field in the input JSON file.



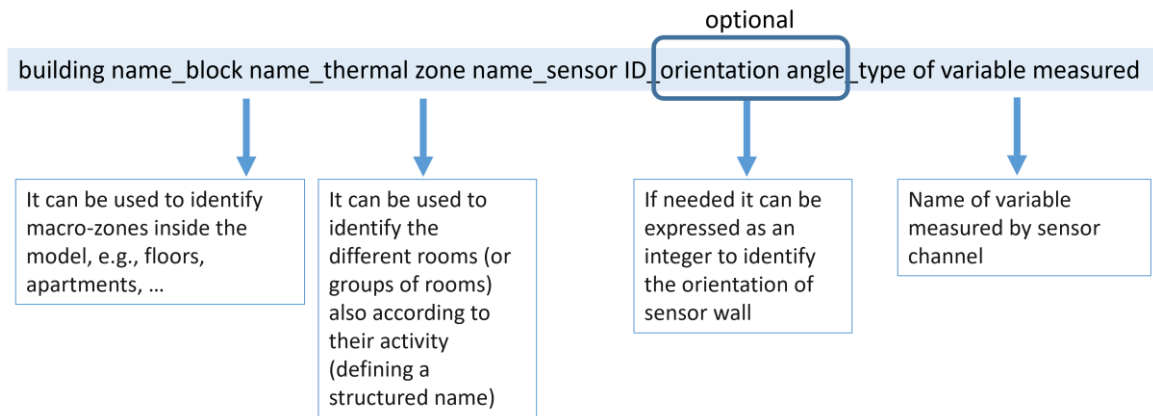


Fig. 20 – Sensor’s nomenclature scheme inside monitored buildings

Figure 19 also shows performance gap scenario output structure: differently from previously described output files, in this case both *data_res.csv* and *data_res_timeseries.csv* contain identifiers for the different simulated building settings (*simulated_x*), according to user defined order inside the input JSON, and correspondent KPIs differences with respect to monitored data (*delta_x*). Since this scenario is thought to include a limited number of simulations in each execution, a unique *data_res_timeseries.csv* file is returned, aggregating results of all simulations with the described identifiers, instead of returning multiple files. Also, it is possible to return desired plots generated inside PREDYCE KPIs methods. Post-analysis plots instead, e.g., temporal comparisons between the computed performance gaps, should be currently generated in a second moment by accessing the scenario output files.

2.7 Weather input coding³

As already stated, several PERDYCE scenarios require the use of real or forecasted weather data, to support more realistic building behaviour comparisons and support optimal free-running building mode suggestions. Hence, it was necessary to develop an EPW compiler module able to generate EPW files for EnergyPlus simulations from data coming from different sources. In particular, real-time weather data are being collected for Turin PRELUDE demo cases from a PRELUDE devoted meteorological station installed on the roof top of the highest POLITO’s building including a Thies weather station based on a Climate sensor US [74], an external CO₂ sensor and a Kipp&Zonen RAZON+ sensor [75] to monitor split solar irradiation variables. Additionally, forecast data for next days can be downloaded from a Meteoblue API service (acquiring the needed credits) given the coordinates of the specific site [76]. Historical real weather data instead could be downloaded both from Meteoblue and Weather Underground [77] API services. However, EPW files should contain several weather variables, as shown in Table 1, and some of them are rarely monitored, hence needing to be computed from other available variables and information. For example, usually the global horizontal solar irradiation (GHI) is monitored instead of its components, the direct normal (DNI) and diffuse horizontal (DIF) radiations requiring applying a splitting model.

³ This specific point is also directly correlated to POLITO’s activities conducted in Task T3.4 of PRELUDE.

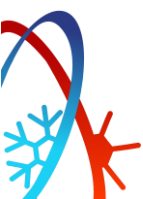


Table 1 – EPW variables insight

EPW variable name	Nomenclature inside PREDYCE	Unit of measure
Dry Bulb Temperature	T_db[C]	°C
Dew Point Temperature	T_dp[C]	°C
Relative Humidity	RH[%]	%
Atmospheric Station Pressure	P_atm[hPa]	hPa
Horizontal Infrared Radiation Intensity	HIRI[W/m2]	W/m ²
Global Horizontal Radiation	GHI[W/m2]	W/m ²
Direct Normal Radiation	DNI[W/m2]	W/m ²
Diffuse Horizontal Radiation	DIF[W/m2]	W/m ²
Wind Direction	wind_dir[o]	°
Wind Speed	v_air[m/s]	m/s
Snow Depth		cm
Liquid Precipitation Depth	rain[mm]	mm

Initial studies have been performed during the PREDYCE development to understand the impact of eventually missing EPW variables, Figure 21, and of measurement or computational errors on simulation results considering different climates. For example, it was found that the infrared horizontal radiation (HIRI) is very important for a simulation run and consequently, even if it is an almost never monitored variable, it is fundamental to be computed. Moreover, errors in global radiation components can cause quite important errors in simulation results, for what concerns both indoor temperature and consequently consumption. Hence, it is important to monitor those variables or to invest resources in the definition of accurate formulas to split them from global radiation measurements. Other EPW fields instead, like the Present Weather Codes which describes the type of precipitation, are almost never used also inside generated TMY by commercial typical weather databases, like Meteonorm [78], since they resulted to be of quite irrelevant impact inside an EnergyPlus simulation.

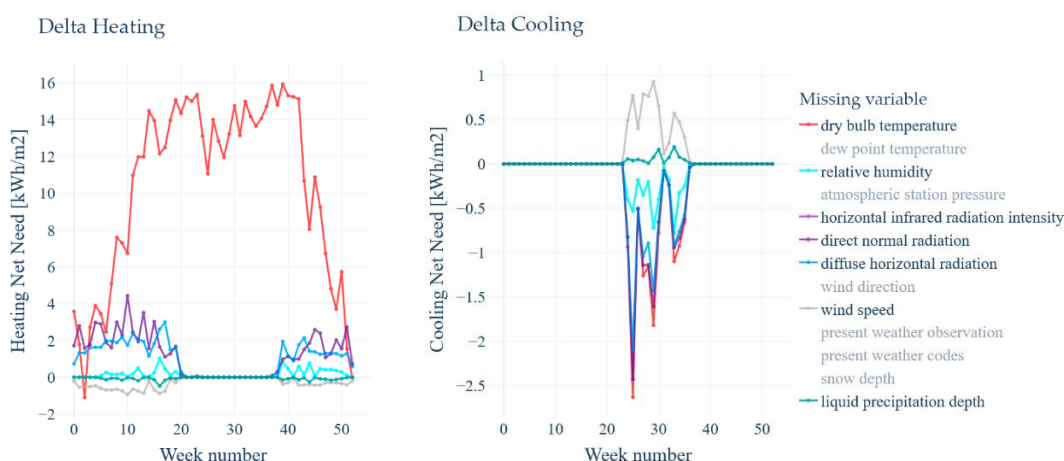
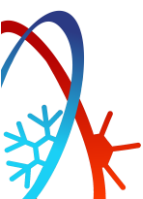


Fig. 21 – Impact of missing weather variables on heating and cooling consumption (district) with Turin TMY (shaded variables had zero impact)

Obtained conclusions lead to the development of an EPW compiler module able to download data from different sources and then compute eventually missing important variables through several weather formulas. The EPW compiler module has been consequently organized with high modularity in four independent steps (Python scripts) that can be integrated depending on the specific application needs to create a complete workflow from weather data to an EPW file, allowing also easier future integrations and



updates, like the ones expected during PRELUDE Task T8.5 works. The scripts' integration can be also organized to be executed with the needed time regularity, e.g., every evening to firstly have next day forecast and secondly run the 24h forecast scenario during night-time. In the following, the four steps are described with more details:

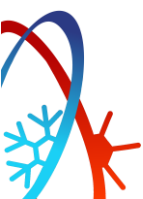
- The first step allows the creation of a CSV file containing downloaded data from the adopted data source. Hence, this script needs to be adapted to the different sources. In particular, several possibilities have been already considered and tested: the Turin weather station API, Meteoblue for next day forecast API and Weather Underground for historical weather data retrieving.
- The second step allows to clean the retrieved weather data generating a new CSV file. This step is particularly relevant if downloading data from a weather station. The adopted approach does not make use of complex cleaning algorithms, but it manages each type of data according to specific rules derived looking at timeseries, weather station manual and in accordance with EPW rules (e.g., respecting thresholds).
- The third step instead allows to generate a CSV file ready for further EPW filling and can be applied independently from the data source, respecting the defined variables nomenclature. This is obtained by hourly aggregating the data and computing missing required variables through climatological expressions from other available information. For example, the Boland-Ridley-Lauret model [79,80] is implemented for splitting diffuse and direct radiation from the monitored global if they are missing: locality more precise algorithm coefficients could be used in input instead of the average ones defined in the original research to reduce splitting errors. The developed script can be flexible to the presence/absence of different variables, adapting which formulas to use in different circumstances.
- Finally, the fourth step, integrating the Python pyepw library [81], allows to generate an EPW file and then fill it to cover initial part of the year with TMY integration (if provided in input) or implementing other filling strategies (e.g., cyclic, repetition of first date) in accordance with the specific usage purpose.

3. Application samples

In this section examples of PRELUDE scenarios applicability are reported, particularly the sensitivity analysis scenario is applied to a climate analysis and to EMS integration inside the building model, then the 24h forecast scenario is analysed, and finally sample potentialities of model verification scenario are explored.

3.1 Climate, bioclimatic and free-running related KPIs

To illustrate a PRELUDE-correlated sample of the PRELUDE sensitivity analysis scenario, some results from a project scientific publication (under publication by Springer in open access) – see [82] – will be here briefly reported and methodologically described. The main goal is to verify potential correlations between climate KPIs (e.g., CDD, CDH, CCP, Q_{ACH}) and indoor comfort KPIs (e.g., PPD-POR, ACM-POR, ACM categories distribution), with particular attention to ventilative cooling impact. For this purpose, a sample residential building flat is simulated adopting as parameters the amount of ventilative ACH in range [0, 2.5, 5], the presence/absence of mechanical cooling system (in order to also verify consumption correlations with climate KPIs), and finally the EPW file, considering the whole Italian territory TMY conditions for all municipalities (about 8000 sites). Desired parameters and KPIs were specified through the input JSON file and aggregated results for the computed KPIs saved in the *data_res.csv* file.



Exploiting the PREDYCE output file, which contains in a table-structure results of numerical KPIs with their association to EPW geographical locations, it was possible to generate several territorial plots showing both climate and building related KPIs distribution over the analysed municipalities. For example, Figure 22 shows obtained results for CDH_{18} and CCP allowing to compare a climate index analysing local expected cooling needs with the local defined night cooling ventilative potential values.

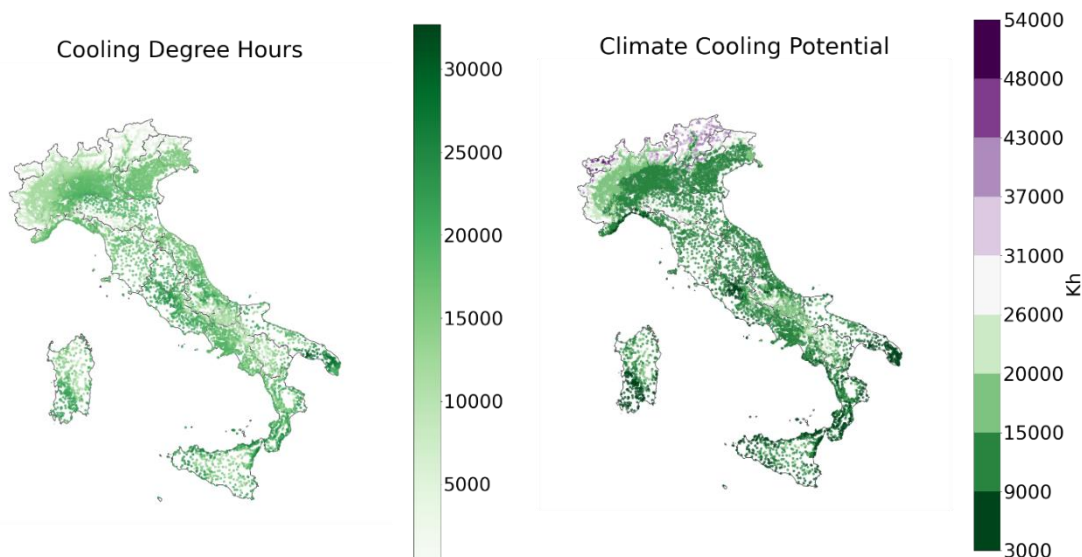


Fig. 22 – Distribution of (a - left) CDH_{18} and (b - right) CPP on the Italian territory

Moreover, analysing *data_res.csv* output file it was possible to study correlations between computed KPIs by aggregating columns according to different criteria (e.g., ACH value, presence/absence of HVAC system). Plots like Figure 23 could be generated through post-analysis scripts showing the tool potentialities in performing correlated studies and supporting further analyses expected during PRELUDE task T8.5.

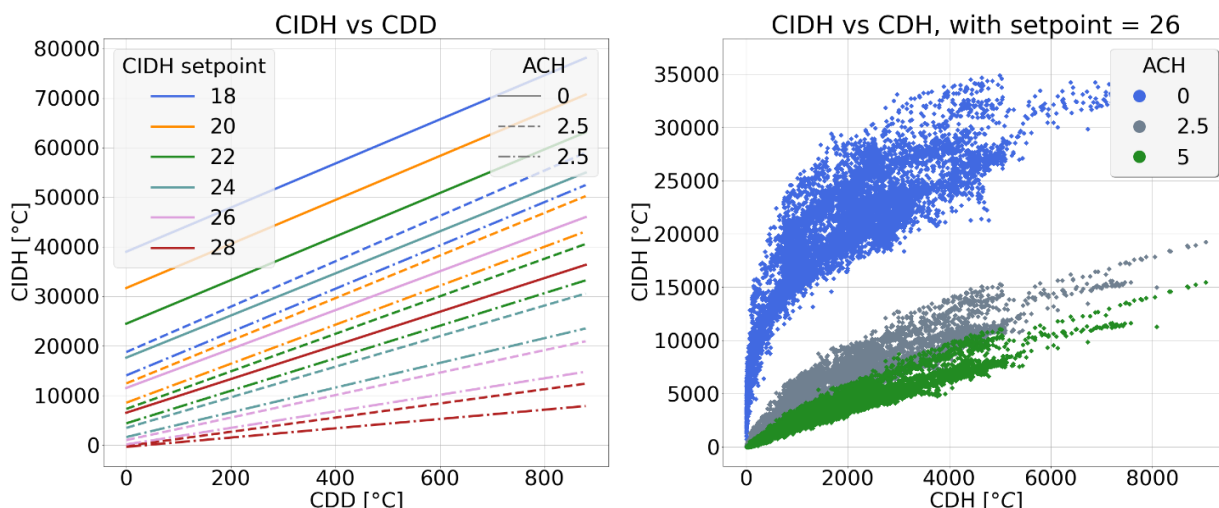
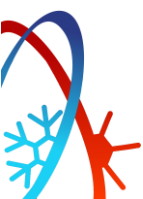


Fig. 23 – CIDH as function of local (a) CDD and (b) CDH. In (a) different CIDH base temperatures are assumed, while in (b) both KPIs have a base temperature of 26°C

Obtained results allow to identify high connections between climate and building KPIs, as shown also in Figure 24, together with the high potentiality of ventilative cooling in reducing discomfort and energy needs in the analysed territory. The adopted procedure can be applied in future studies expanding the analysed territory (Europe) and the temporal perspective (climate trend from past to now, and in a future



perspective) as requested in future works in Task 8.5 goals, but also expanding the set of analysed KPIs including other passive cooling/heating technologies (such as PDEC, EAHX) potential impact.

		CIDH					
		18	20	22	24	26	28
CDH	18	0.96	0.97	0.98	0.97	0.95	0.89
	21	0.92	0.93	0.94	0.96	0.96	0.92
	24	0.82	0.84	0.87	0.9	0.92	0.92
	26	0.73	0.75	0.79	0.83	0.87	0.89
	28	0.62	0.64	0.67	0.72	0.78	0.83

Fig. 24 – R² values combining different ACH and CIDH base temperatures for ACH=2.5 in free-running building mode

3.2 Testing free-running control solutions – shading system

Sensitivity analysis scenario of use can be also applied to more complex building analyses, allowing to simulate the presence of actuators, by exploiting the built-in Energy Management System (EMS) tool of EnergyPlus combined with the parametric simulation feature given by the tool PREDYCE. The EMS objects can be used to customise actuator strategies, e.g. the shading activation, allowing increased functionalities with respect to basic control objects, e.g., by creating specific algorithms handling more than two variables together, or by also considering variables temporal trend.

In the following example, the shading activation is performed by considering the internal temperature of each room individually, together with the outdoor temperature and the amount of solar radiation on the windows: this generates a parametric problem where the perceived comfort inside the house depends on the thresholds of the shading system given by these three parameters. By performing a sensitivity analysis on these three variables using the predicted percentage of dissatisfied as indicator, it is possible to recreate 3D representations of the curve that show how to minimize the discomfort of the tenants by selecting the best combination of thresholds to be further assigned to automatic control systems during their design phases, as shown in Figure 25.

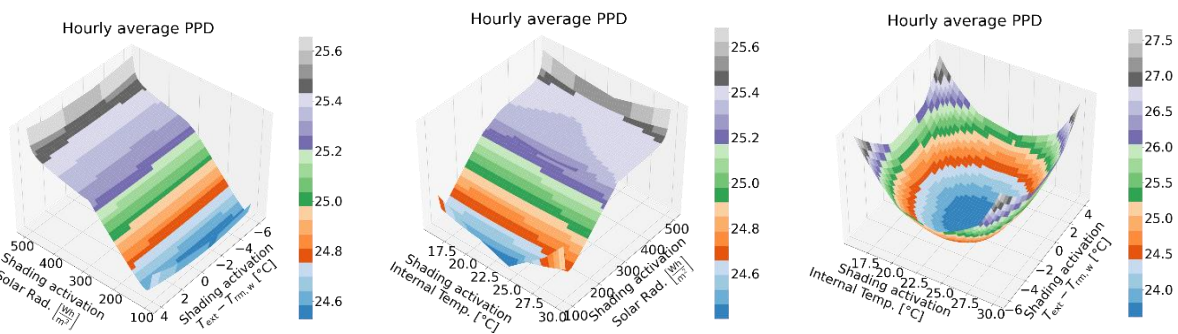
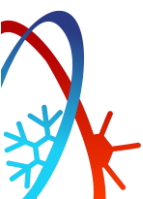


Fig. 25 – Shading activation strategies results obtained through simple EMS program

3.3 Suggestions for optimising free-running usage

To test applicability of the 24h forecasting scenario, a sample building unit generated through DesignBuilder has been considered. Figure 26 shows the sample unit structure defined in line with typical residential units derived by merging models from well-known building design supporting manuals [83]. The simulated sample is a composed by a floor with two flats. Partitions between zones in flats at the same temperature are substituted by internal masses. The building was generated without HVAC system



installed, in order to test the scenario in a full free-running mode, moreover, it features a standard scheduled natural ventilation and a shading system which is enabled by default when there is high solar radiation on windows. Concerning the EPW file, data from Athens TMY have been considered as if they were past monitored data for days before each step of the analysis and as if they were forecasted weather data for the 24h hours after each analysis step.

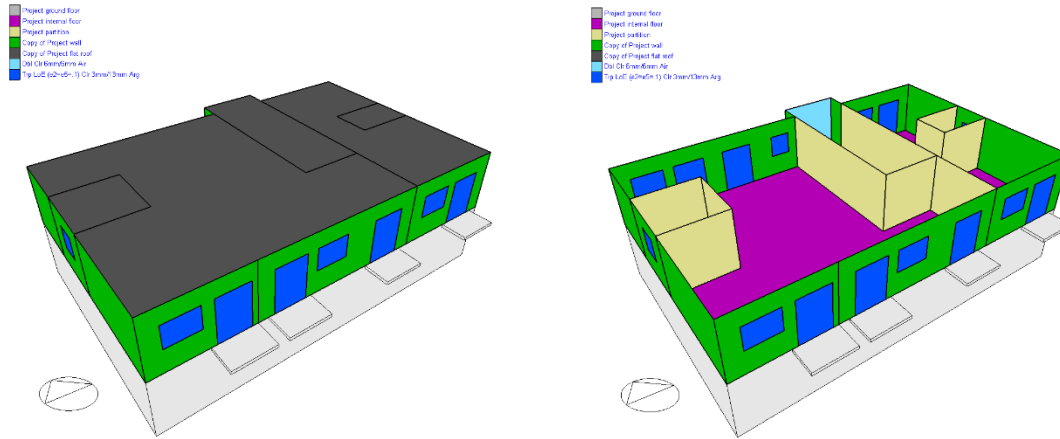


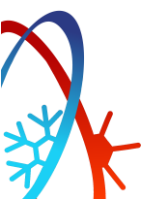
Fig. 26 – DesignBuilder model of the assumed sample building unit

To create user suggestions, a specific 24h forecast scenario has been created, using the JSON file listed in Figure 27: the purpose was to find the best strategies to ventilate the house (which can be done independently for every window orientation) and activate the shading system.

```
{
  "scenario": "24hf",
  "building_name": "block",
  "day": "20/06",
  "hour_aggregations": [
    [0, 1, 2, 3, 4, 5, 6],
    [7, 8, 9],
    [10, 11, 12, 13],
    [14, 15, 16, 17],
    [18, 19, 20, 21],
    [22, 23]
  ],
  "forecast_actions": {
    "stack_and_wind_ventilation": [0.01, 0.03, 0.7],
    "blind": []
  },
  "kpi": {
    "adaptive_comfort_model": {},
    "adaptive_residuals": {}
  }
}
```

Fig. 27 – JSON file for 24h forecast test

Since for this example the forecasted weather data were derived from a typical methodological year, it was possible to run the 24h forecast script several times in a row in order to cover a period of one week, in this way user suggestion have been stored for many days in the future, which can also be plotted like Figure 28 shows. Although each 24-h period is treated as a new application of the scenario.



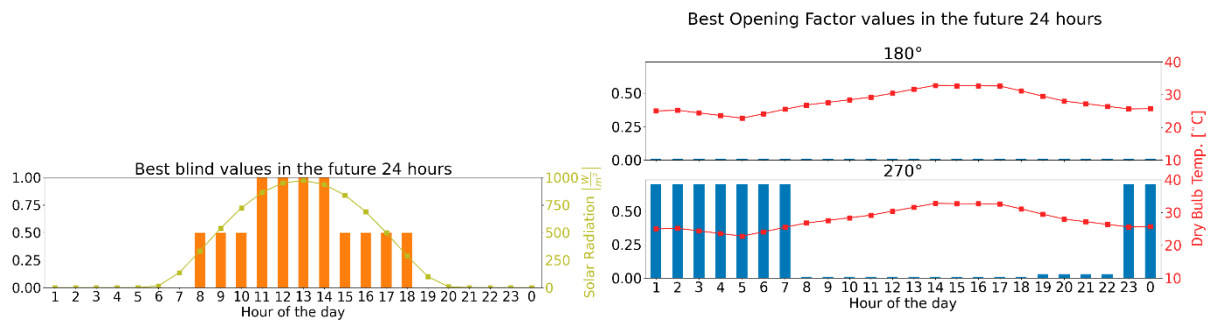


Fig. 28 – Example of user suggestions for the next day

For a final comparison, the building model has been run in two ways: the first consisted in the original building without any modifications of the original schedules, the second consisted in the building with applied suggestions given by the 24h forecast script. Finally, the percentage of dissatisfaction inside the building is computed for each case to show how the optimized suggested schedules positively impact on the building behaviour exploiting the local free-running potential.

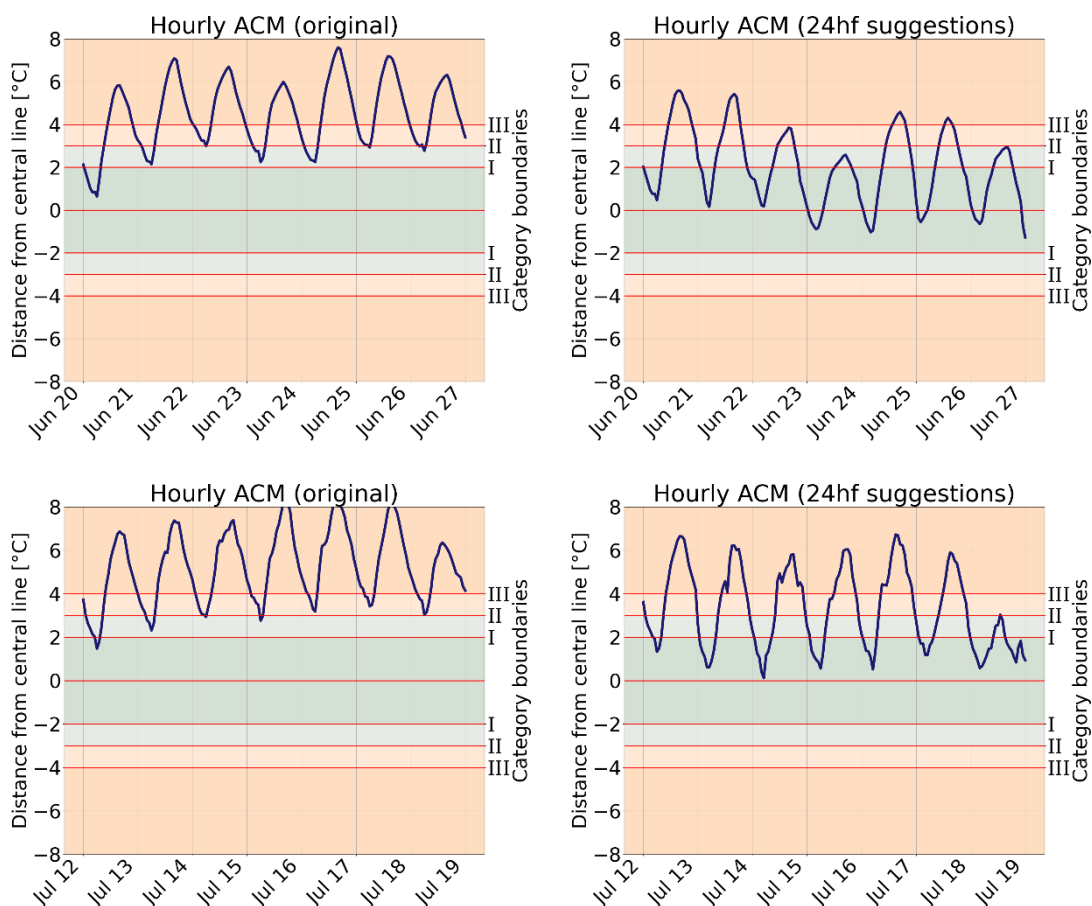
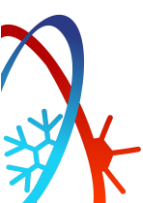


Fig. 29 – Comparison of thermal comfort dissatisfaction (based on AM categories) between the standard model and the model with forecast suggestions applied.

As shown in Figure 29, the percentage of dissatisfaction decreases when the schedules inside the model are modified using the predicted 24-h strategies. This result demonstrate that these optimized strategies are supposed to be effective since they are selected by minimizing the discomfort of the tenants for the specific kind of weather. Of course, the improvement given by the 24h forecast script strictly depends on



the accuracy of weather forecasts and the availability of users to adopt the suggested strategies when automatic actuators are not present (self-user actuating mode).

3.4 Verification scenario

This section reports an example of model verification scenario applicability. The same sample building used for the previous example, Figure 26, generated through DesignBuilder interface without HVAC system installed, is simulated in EnergyPlus with Turin TMY EPW file. In this sample, average indoor temperature results of the base model are taken as target values (as if they were monitored data) and then used inside the calibration process. The target building has been then modified through DesignBuilder interface in several aspects, particularly walls and roof U-values, windows U-value and SHGC, ventilation and infiltration ACH, internal gains, and finally a new IDF file is exported including all random modifications. The altered building model is used as starting point (base model) for the calibration process (as if there was certain degree of uncertainty about building construction values) together with temperature results obtained simulating the original model (target temperatures). Also the Turin TMY was considered as fictitious real weather inside the model verification script being not yet available a sufficient amount of monitored data to feed a full EPW file. Through the developed scenario walls, roof and windows U-values have been tested in range +/- 100% around the modified value, windows SHGC in range +/- 40%, ventilation ACH in range +/- 80%, infiltration ACH in range +/- 50%, and finally internal gains in range +/- 25%, always with a 5% step. Particularly, to avoid exponential growth of simulations to be performed, just a couple of parameters are tested at the same time, choosing each time the best value minimizing the considered error measure. The followed order is opaque envelope, windows parameters, ACH, and finally internal gains.

For the calibration period it was chosen the month of April, particularly from the 15th to the 30th, since according to Figure 30 it is one of the time periods with maximum indoor temperature discrepancies between the two models simulated under the same weather.

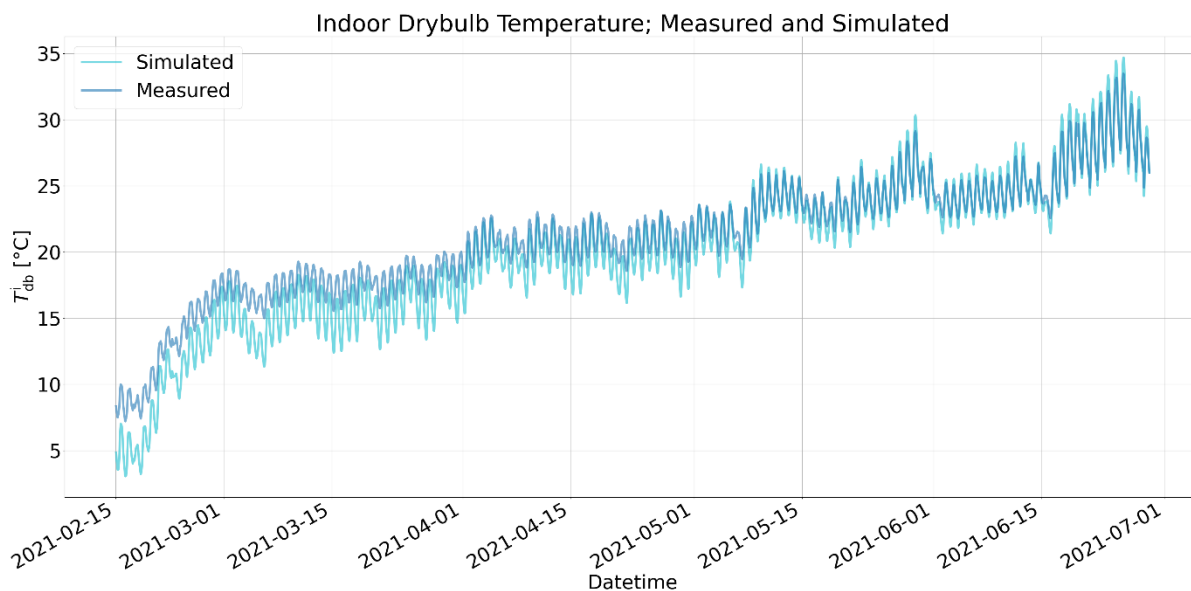
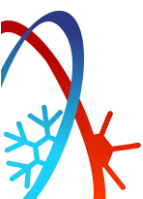


Fig. 30 – Starting point temporal trend for indoor temperature

Table 2 shows target values, together with modified values used as verification process starting point (base model) and finally found values at the end of the model calibration. U-factors end values result to be higher than true ones, particularly for windows, as if both windows U-value and SHGC tried to absorb the impact of other wrong parameters which were calibrated only in the following (ACH, internal gains). In fact, lower values of both ACH ventilation and infiltration with respect to target model were found as best ones. Internal gains are not shown in the table since they were kept the same in both target and base model and,



maybe because it was the last value to be modified in the process, zero variations were found as best. However, particularly concerning opaque envelope U-factors, the process is able to get values closer to target ones, significantly reducing the initial error between base and target models.

Table 2 – Model verification process target, base and end values

	Target model (Virtual monitored data)	Base model	Validated model
U-factor walls [W/m ² K]	0.2	0.58	0.29
U-factor roof [W/m ² K]	0.35	1	0.45
U-factor windows [W/m ² K]	0.78	0.982	1.4239
SHGC	0.474	0.474	0.5688
ACH infiltration	0.4	0.3	0.33
ACH ventilation	4	5	3.5

Figure 31 instead shows calibration signatures computed with the methodology described in Section 2: best found values allow to act on the curve reducing the error around 0.5%. Also, Figure 32 shows “virtual monitored” versus simulated indoor temperatures, before and after the model verification process. Figure 33 instead compares indoor temperature temporal trend during the calibration period in April at the beginning of the process and at the end: both daily average (which initially showed even 2°C of difference) and minimum/maximum peaks result be perfectly aligned with the original model behaviour. Best found values, even if not perfectly aligned with target ones, allows to absorb the error almost completely on the calibrated weeks, finding a different parameters equilibrium.

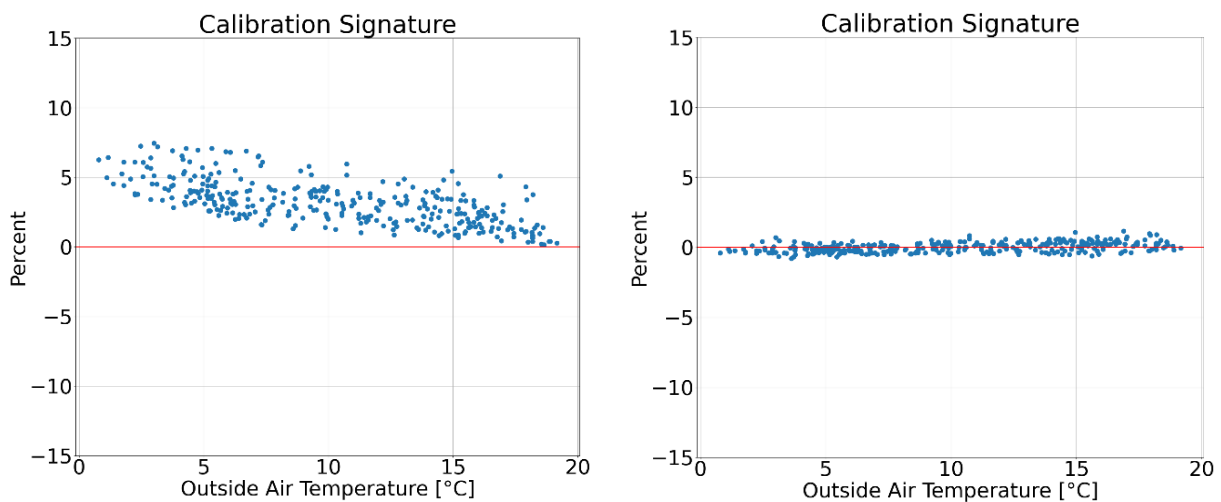
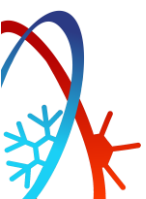


Fig. 31 – Start (a) and end (b) calibration signatures



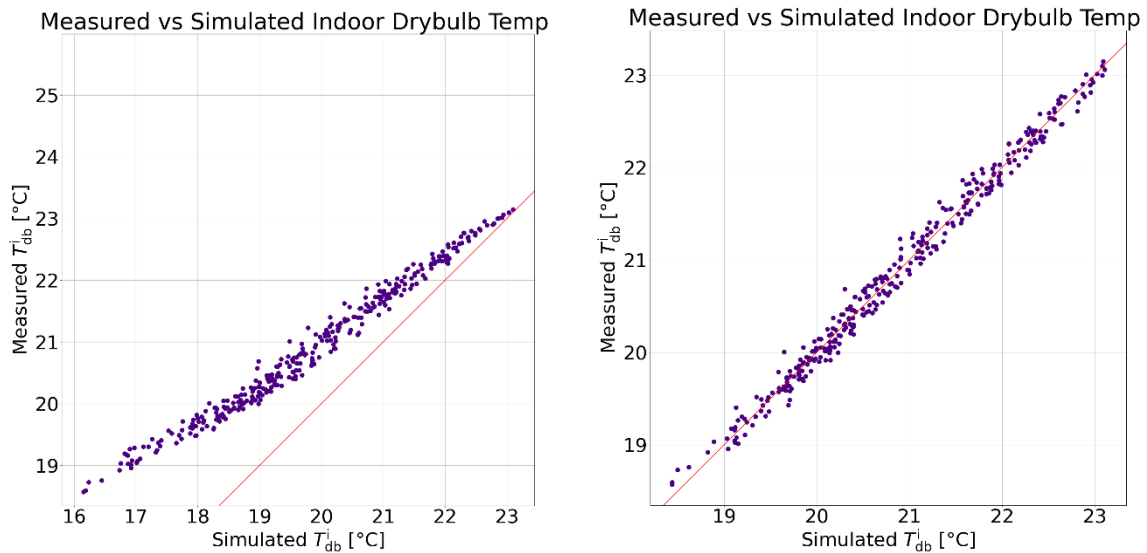


Fig. 32 – Start (a) and end (b) indoor temperatures measured versus simulated

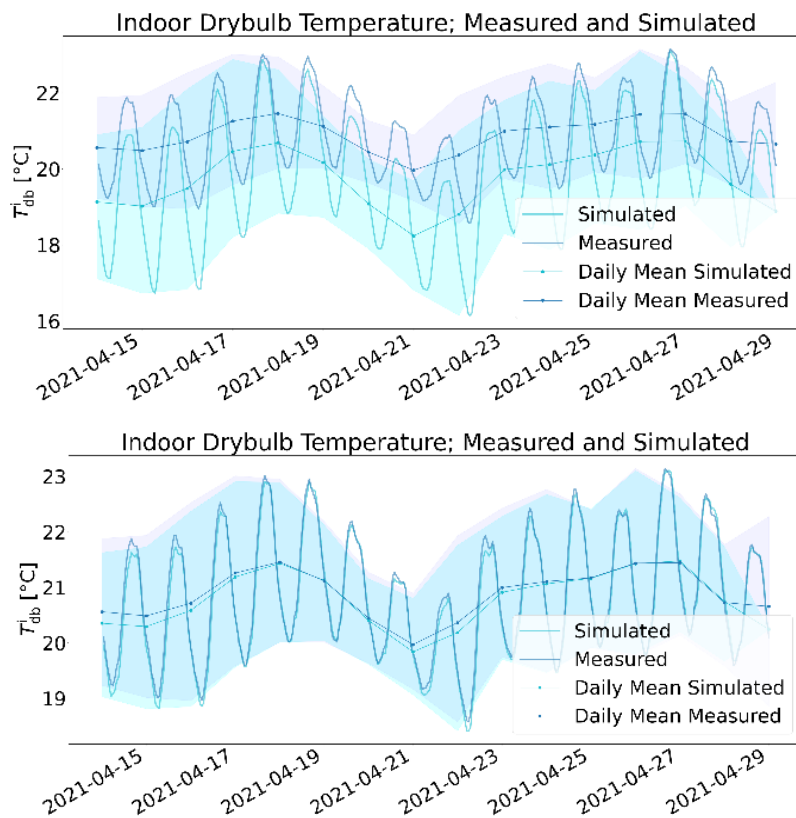
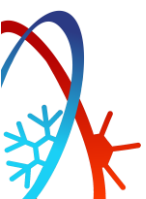


Fig. 33 – Start (a) and end (b) indoor temperature trend during the calibration period

Finally, Figure 34 shows the extended indoor temperature temporal trend of the calibrated building: even if found values were the best for the considered two weeks in April, the error is reduced over the whole year, even in February/March where it was maximum, as shown in Figure 30. Hence, best found values allowed to keep the equilibrium also in all other months (considering that the building, since simulated, has always the same utilization pattern).



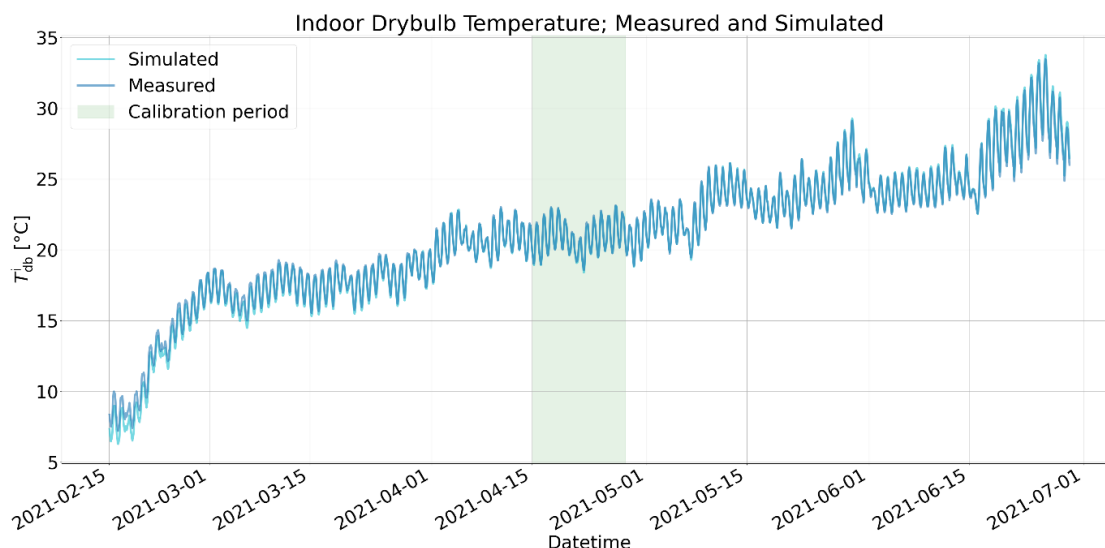


Fig. 34 – Ending point temporal trend for indoor temperature after calibration in April (highlighted in green)

The proposed semi-automatic scenario has consequently quite interesting implications in reducing model calibration time and helping in finding the different parameters impact. Also, it has to be considered that, if possible, it would be better to calibrate a model when unused, in order to remove the impact of different occupancy and usage patterns and also the impact of natural ventilation, which is intrinsically difficult to be simulated correctly: removing the impact of ACH ventilation, less important aspect would have been absorbed by other parameters, improving the quality of the results. However, it is interesting how, under same utilization conditions, also a different building parameters equilibrium can lead to a yearly aligned temporal trend with “measured” values, which is the most important thing in most applicative tasks.

4. CONCLUSIONS

Focus of D3.2 is the description of a Python library allowing automatic and parametric interfacing with EnergyPlus software. The library has been developed and adapted to answer PRELUDE goals, including:

- i. the managing inside the building model of passive and low energy technologies (e.g., controlled natural ventilative cooling, direct evaporative cooling, PDEC, and ground cooling/pre-heating systems, EAHX) and free-running mode (e.g., with particular attention to shadings and natural ventilation scheduling and activation rules).
- ii. the possibility to compute, besides building-related KPIs, also climate-related ones to analyze local potential of different solutions on virtual space units.
- iii. the development of task-oriented scenarios of use devoted to massive parametric simulation analyses considering both geographical, temporal and buildings parameters (sensitivity analysis scenario).
- iv. to forecast best building free-running mode or low energy technology scheduling for next days (24h forecast scenario).
- v. and finally, to calibrate the building model with respect to monitored data from demo cases (model verification scenario).

The developed library also allows to communicate through server-to-server communication with the PRELUDE project middleware (FusiX platform), exploiting dedicated REST services running on a server.

Main limitation of the proposed solution is the relation with specific EnergyPlus version 8.x, future work expansion can be hence devoted to re-adapt developed IDF editing methods to handle models generated with more recent software versions (v9.x). Also, the integration of optimization strategies for sensitivity

analysis and 24h forecast scenarios, exploiting the integration of already existing Python libraries devoted to this task, is under study for future library developments and expansion with potential integration of surrogate modelling approaches.

The proposed library works as an enabler for answering PRELUDE goals also looking at future project work packages and specific tasks, such as T8.5 in which studies about climate resilience of current technologies will be carried on exploiting potentialities of sensitivity analysis scenarios and KPIs and IDF editing developed methods. Future expected works in PRELUDE WP3 will focus on enabling the described contents in the project expected demonstration phases. Furthermore, an extra work is expected to potentially develop a new PREDYCE scenario able to retrieve specific KPIs polynomial regression lines in order to potentially automatize the calculation of correlations between for example indoor and outdoor variables supporting multi-task and multi-work-package integrations.

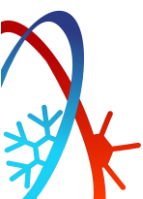
APPENDIX A**BIBLIOGRAPHY**

- [1] European Union, Buildings Factsheet, European Union, 2021. https://ec.europa.eu/commission/presscorner/detail/en/fs_21_3673.
- [2] European Commission, Energy performance of buildings directive, (2021). https://energy.ec.europa.eu/topics/energy-efficiency/energy-efficient-buildings/energy-performance-buildings-directive_en.
- [3] European Commission, Proposal for a DIRECTIVE OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL on the energy performance of buildings (recast), European Commission, 2021. <https://ec.europa.eu/energy/sites/default/files/proposal-recast-energy-performance-buildings-directive.pdf>.
- [4] B. Givoni, Man, climate, and architecture, Elsevier, Amsterdam, New York, 1969.
- [5] V. Olgay, A. Olgay, D. Lyndon, V.W. Olgay, J. Reynolds, K. Yeang, Design with climate: bioclimatic approach to architectural regionalism, New and expanded edition, Princeton University Press, Princeton, 2015.
- [6] D. Watson, K. Labs, Climatic Design. Energy-efficient building principles and practice, McGraw-Hill, New York, 1983.
- [7] G. Chiesa, ed., Bioclimatic Approaches in Urban and Building Design, Springer International Publishing, Cham, 2021. <https://doi.org/10.1007/978-3-030-59328-5>.
- [8] P. Holzer, T. Psomas, eds., Ventilative cooling sourcebook, IEA EBC Annex 62, Aalborg University, Aalborg, 2018.
- [9] G. Chiesa, M. Kolokotroni, P. Heiselberg, Innovations in ventilative cooling, 2021. <https://doi.org/10.1007/978-3-030-72385-9> (accessed July 19, 2021).
- [10] B. Givoni, Passive and low energy cooling of buildings, Van Nostrand Reinhold, New York, 1994.
- [11] G. Chiesa, M. Grosso, D. Pearlmutter, S. Ray, Editorial. Advances in adaptive comfort modelling and passive/hybrid cooling of buildings, Energy and Buildings. 148 (2017) 211–217.
- [12] J. Cook, ed., Passive Cooling, MIT press, Cambridge, 1989.
- [13] M. Santamouris, D. Asimakopoulous, eds., Passive Cooling of Buildings, James and James, London, 1996.
- [14] M. Santamouris, ed., Advances in Passive Cooling, Earthscan, London, 2007.
- [15] M. Santamouris, ed., Cooling energy solutions for buildings and cities, World Scientific, New Jersey, 2019.
- [16] EUROSTAT, Heating and cooling degree days - statistics, (2021). <https://ec.europa.eu/eurostat/statistics-explained/index.php?oldid=553136>.
- [17] EUROSTAT, dataset - Cooling and heating degree days by country - annual data, (2022). https://ec.europa.eu/eurostat/web/products-datasets/product?code=nrg_chdd_a.
- [18] M. Santamouris, Cooling the buildings – past, present and future, Energy and Buildings. 128 (2016) 617–638. <https://doi.org/10.1016/j.enbuild.2016.07.034>.
- [19] M. Isaac, D.P. van Vuuren, Modeling global residential sector energy demand for heating and air conditioning in the context of climate change, Energy Policy. 37 (2009) 507–521. <https://doi.org/10.1016/j.enpol.2008.09.051>.

- [20] G. Chiesa, Evaluating early design choices under the potential effects of climate change, *SMC - Sustainable Mediterranean Construction*. 6 (2017) 24–28.
- [21] DesignBuilder Software, DesignBuilder, DesignBuilder Software, 2020. <https://designbuilder.co.uk/>.
- [22] NREL, ANL, LBNL, ORNL, PNNL, OpenStudio, 2021. <https://www.openstudio.net/>.
- [23] Building energy modeling with openstudio: a practical guide for students and professionals, Springer Science+Business Media, New York, NY, 2018.
- [24] P. Santosh, Eppy, 2021. <https://pypi.org/project/eppy/>.
- [25] U.S. Department of Energy, Application Guide for EMS, U.S. Department of Energy, 2021. https://energyplus.net/assets/nrel_custom/pdfs/pdfs_v9.5.0/EMSAplicationGuide.pdf.
- [26] U.S. Department of Energy, Application Guide for EMS, 2018. <https://bigladdersoftware.com/epx/docs/8-9/ems-application-guide/index.html>.
- [27] M. Grosso, ed., *Il raffrescamento passivo degli edifici, IV*, Maggioli, Sant’Arcangelo di Romagna, 2017.
- [28] F. Allard, ed., *Natural ventilation in buildings: a design handbook - EC, ALTENER Progamme*, James and James (Science Publishers) Ltd, London, 1998.
- [29] M. Kolokotroni, P. Heiselberg, *IEA EBC Annex 62 - Ventilative Cooling State-of-the-Art Review*, Aalborg University, Aalborg, 2015. <https://venticool.eu/wp-content/uploads/2013/09/SOTAR-Annex-62-FINAL.pdf>.
- [30] G. Chiesa, M. Grosso, Environmental and Technological Design: a didactical experience towards a sustainable design approach, in: *WORLD HERITAGE and DISASTER Knowledge, Culture and Rapresentation Le Vie Dei Mercanti _ XV International Forum*, La scuola di Pitagora, Napoli, 2017: pp. 931–940. http://www.leviedeimercanti.it/proceedings/XV_Forum_Proceedings.pdf.
- [31] U.S. Department of Energy, EnergyPlus EngineeringReference, (2021). https://energyplus.net/assets/nrel_custom/pdfs/pdfs_v9.6.0/EngineeringReference.pdf.
- [32] British Standard, BS 5925:1991, Code of practice for Ventilation principles and designing for natural ventilation, 1991.
- [33] B. Givoni, Performance of the shower cooling tower in different climates, *Renewable Energy*. 10 (1997) 173–178.
- [34] E. Erell, Evaporative cooling, in: *Advances in Passive Cooling*, Earthscan, London, 2007: pp. 228–261.
- [35] E. Erell, D. Pearlmutter, Y. Etzion, A multi-stage down-draft evaporative cool tower for semi-enclosed spaces: aerodynamic performance, *Solar Energy*. 82 (2008) 420–429.
- [36] B. Ford, R. Schiano-Phan, E. Francis, eds., *The architecture and engineering of downdraught cooling. A design sourcebook*, PHDCpress, UK, 2010.
- [37] D. Pearlmutter, E. Erell, Y. Etzion, A multi-stage down-draft evaporative cool tower for semi-enclosed spaces: experiments with a water spraying system, *Solar Energy*. 82 (2008) 430–440.
- [38] D. Pearlmutter, E. Erell, Y. Etzion, I. Meir, H. Di, Refining the use of evaporation in an experimental down-draft cool tower, *Energy and Buildings*. 23 (1996) 191–197.
- [39] G. Chiesa, N. Huberman, D. Pearlmutter, M. Grosso, Summer Discomfort Reduction by Direct Evaporative Cooling in Southern Mediterranean Areas, *Energy Procedia*. 111 (2017) 588–598. <https://doi.org/10.1016/j.egypro.2017.03.221>.
- [40] J. Pfafferott, Evaluation of earth-to-air heat exchangers with a standardised method to calculate energy efficiency, *Energy and Buildings*. 35 (2003) 971–983.

- [41] G. Chiesa, EAHX – earth-to-air heat exchanger: simplified method and KPI for early building design phases, *Building and Environment*. 144 (2018) 142–158.
- [42] C. Peretti, A. Zarrella, M. De Carli, R. Zecchin, The design and environmental evaluation of earth-to-air heat exchangers (EAHE). A literature review, *Renewable and Sustainable Energy Reviews*. 28 (2013) 107–116.
- [43] H. Li, Y. Yu, F. Niu, M. Shafik, B. Chen, Performance of a coupled cooling system with earth-to-air heat exchanger and solar chimney, *Renewable energy*. 62 (2014) 468–477.
- [44] F. Niu, Y. Yu, D. Yu, H. Li, Heat and mass transfer performance analysis and cooling capacity prediction of earth to air heat exchanger, *Applied Energy*. 137 (2015) 211–221. <https://doi.org/10.1016/j.apenergy.2014.10.008>.
- [45] F. Ascione, L. Bellia, F. Minichiello, Earth-to-air heat exchangers for Italian climates, *Renewable Energy*. 36 (2011) 2177–2188. <https://doi.org/10.1016/j.renene.2011.01.013>.
- [46] U.S. Department of Energy, Auxiliary Programs, 2022. https://energyplus.net/assets/nrel_custom/pdfs/pdfs_v22.1.0/AuxiliaryPrograms.pdf.
- [47] G. Chiesa, Including EAHX (earth-to-air heat exchanger) in early-design phases considering local bioclimatic potential and specific technological requirements, *IOP Conf. Ser.: Mater. Sci. Eng.* 609 (2019) 032040. <https://doi.org/10.1088/1757-899X/609/3/032040>.
- [48] J. Pfafferott, S. Walker-Hertkorn, B. Sanner, Ground cooling: recent progress, in: *Advances in Passive Cooling*, Heartscan, London, 2007: pp. 190–227.
- [49] EUROSTAT, Energy statistics - cooling and heating degree days (nrg_chdd), (2021). https://ec.europa.eu/eurostat/cache/metadata/en/nrg_chdd_esms.htm.
- [50] G. Chiesa, Calculating the geo-climatic potential of different low-energy cooling techniques, *Build. Simul.* 12 (2019) 157–168. <https://doi.org/10.1007/s12273-018-0481-5>.
- [51] G. Chiesa, A. Zajch, Contrasting climate-based approaches and building simulations for the investigation of Earth-to-air heat exchanger (EAHE) cooling sensitivity to building dimensions and future climate scenarios in North America, *Energy and Buildings*. 227 (2020) 110410. <https://doi.org/10.1016/j.enbuild.2020.110410>.
- [52] G. Chiesa, M. Grosso, Geo-climatic applicability of natural ventilative cooling in the Mediterranean area, *Energy and Buildings*. 107 (2015) 376–391. <https://doi.org/10.1016/j.enbuild.2015.08.043>.
- [53] G. Chiesa, Climatic potential maps of ventilative cooling techniques in Italian climates including resilience to climate changes, *IOP Conf. Ser.: Mater. Sci. Eng.* 609 (2019) 032039. <https://doi.org/10.1088/1757-899X/609/3/032039>.
- [54] M. Grosso, CpCalc+, Politecnico di Torino, 2001.
- [55] N. Artmann, D. Gyalistras, H. Manz, P. Heiselberg, Impact of climate warming on passive night cooling potential, *Building Research & Information*. 36 (2008) 111–128. <https://doi.org/10.1080/09613210701621919>.
- [56] G. Chiesa, M. Grosso, Direct evaporative passive cooling of building. A comparison amid simplified simulation models based on experimental data, *Building and Environment*. 94 (2015) 263–272. <https://doi.org/10.1016/j.buildenv.2015.08.014>.
- [57] K. Labs, Regional analysis of ground and above-ground climate, OAK Ridge National Laboratory, New Haven, 1981.
- [58] P. Heiselberg, IEA EBC Annex 62 - Ventilative Cooling Design Guide, Aalborg University, Aalborg, 2018.

- [59] Azogue, Shenzha, Zhukovgreen, Tantecky, Psychrochart Python library, 2019. <https://github.com/azogue/psychrochart>.
- [60] D. Meyer, D. Thevenard, PsychroLib: a library of psychrometric functions to calculate thermodynamic properties of air, *JOSS*. 4 (2019) 1137. <https://doi.org/10.21105/joss.01137>.
- [61] B. Givoni, Comfort, climate analysis and building design guidelines, *Energy and Buildings*. 18 (1992) 11–23. [https://doi.org/10.1016/0378-7788\(92\)90047-K](https://doi.org/10.1016/0378-7788(92)90047-K).
- [62] M. Milne, B. Givoni, Architectural design based on climate, in: D. Watson (Ed.), *Energy Conservation through Building Design*, McGraw Hill, New York NY USA, 1979: pp. 96–113.
- [63] R. Liggett, M. Milne, *Climate Consultant 6.0*, UCLA Energy Design Tools Group, 2017.
- [64] R. Stull, Wet-Bulb Temperature from Relative Humidity and Air Temperature, *Journal of Applied Meteorology and Climatology*. 50 (2011) 2267–2269. <https://doi.org/10.1175/JAMC-D-11-0143.1>.
- [65] A. Acquaviva, D. Apiletti, A. Attanasio, E. Baralis, L. Bottaccioli, F.B. Castagnetti, T. Cerquitelli, S. Chiusano, E. Macii, D. Martellacci, E. Patti, Energy Signature Analysis: Knowledge at Your Fingertips, in: *2015 IEEE International Congress on Big Data*, IEEE, New York City, NY, USA, 2015: pp. 543–550. <https://doi.org/10.1109/BigDataCongress.2015.85>.
- [66] G. Wei, M. Lui, D. Claridge, Signature of heating and cooling energy consumption for typical AHUs, in: *Proceedings of the Eleventh Symposium of Improving Building Systems in Hot and Humid Climate*, Fort Worth, Texas, 1998.
- [67] R. Hitchin, I. Knight, Daily energy consumption signatures and control charts for air-conditioned buildings, *Energy and Buildings*. 112 (2016) 101–109. <https://doi.org/10.1016/j.enbuild.2015.11.059>.
- [68] M. Eriksson, J. Akander, B. Moshfegh, Development and validation of energy signature method – Case study on a multi-family building in Sweden before and after deep renovation, *Energy and Buildings*. 210 (2020) 109756. <https://doi.org/10.1016/j.enbuild.2020.109756>.
- [69] M. Pellegrino, M. Simonetti, G. Chiesa, Reducing thermal discomfort and energy consumption of Indian residential buildings: Model validation by in-field measurements and simulation of low-cost interventions, *Energy and Buildings*. 113 (2016) 145–158. <https://doi.org/10.1016/j.enbuild.2015.12.015>.
- [70] P. Fanger, *Thermal comfort. Analysis and applications in environmental engineering*, Danish Technical Press, Copenhagen, 1970.
- [71] P. Fanger, *Thermal Comfort: Analysis and Applications in Environmental Engineering*, McGraw-Hill, New York, 1972.
- [72] J. van Hoof, Forty years of Fanger's model of thermal comfort: comfort for all?, *Indoor Air*. 18 (2008) 182–201. <https://doi.org/10.1111/j.1600-0668.2007.00516.x>.
- [73] D. Claridge, M. Paulus, Building simulation of practical operational optimisation, in: J. Hensen, R. Lamberts (Eds.), *Building Performance Simulation for Design and Operation*, 2nd ed., Routledge, London ; New York, 2019: pp. 399–453.
- [74] Thies Clima, Thies Climate sensor US, (2022). <https://www.thiesclima.com/en/Products/Wind-Combi-sensors/?art=197>.
- [75] Kipp & Zonen, RAZON+, (2016). <https://www.kippzonen.com/Product/378/RaZON#.YoJNdXXP2Uk>.
- [76] meteoblue AG, Meteoblue weather API, (2006). <https://www.meteoblue.com/en/weather-api>.
- [77] TWC Product and Technology LLC, weather underground, (2014). <https://www.wunderground.com/>.
- [78] Meteotest, METEONORM, METEOTEST Genossenschaft, Bern, 2015. meteonorm.com.
- [79] Meteotest, *Meteteonorm handbook part I*, Meteotest, Bern, 2017.



- [80] J. Boland, J. Huang, B. Ridley, Decomposing global solar radiation into its direct and diffuse components, *Renewable and Sustainable Energy Reviews*. 28 (2013) 749–756. <https://doi.org/10.1016/j.rser.2013.08.023>.
- [81] R. Buffat, pyepw 0.1, 2014. <https://pypi.org/project/pyepw/>.
- [82] G. Chiesa, F. Fasano, P. Grasso, Thermal comfort and climatic potential of ventilative cooling in Italian climates, in: A. Sayigh (Ed.), *Achieving Building Comfort by Natural Means*, Springer, Cham, under publication: p. 24pp. <https://link.springer.com/book/9783031047138>.
- [83] E. Neufert, G.M. Di Giuda, V. Villa, A. Gottfried, P. Piantanida, *Enciclopedia pratica per progettare e costruire: manuale a uso di progettisti, costruttori, docenti e studenti: fondamentali, norme e prescrizioni per progettare, costruire, dimensionare e distribuire a misura d'uomo*, U. Hoepli, Milano, 2013.

