Ref. Ares(2020)1831639 - 30/03/2020

Project Title:

Innovative compact HYbrid electrical/thermal storage systems for low energy BUILDings

Project Acronym:

HYBUILD

Deliverable Report

Deliverable number:

D4.3

Deliverable title:

HYBUILD OPTIMISED BUILDING MANAGEMENT SYSTEM

Related task:	T4.3, T4.4
Lead beneficiary:	ENG
Authors and institutions:	Giuseppe Paternò (ENG), Alessandro Rossi (ENG), Cesar Fernández (UDL), Gabriel Zsembinszki (UDL), David Vérez (UDL), Ramón Béjar (UDL), Luisa F. Cabeza (UDL)
Due date:	31/03/2020

DISSEMINATION LEVEL		
PU	Public, fully open, e.g. web	X
CO	Confidential, restricted under conditions set out in Model Grant Agreement	
CI	Classified, information as referred to in Commission Decision 2001/844/EC.	

This project has received funding from the European Union's Horizon 2020 research and innovation programme *under grant agreement No 768824.*

The content of this document reflects only the author's view and the Commission is not responsible for any use that may be made of the information it contains.

DOCUMENT STATUS HISTORY		
Date	Description	Partner
23/03/2020	Draft	ENG
30/03/2020	Reviewed	R2M UDL COMSA
30/03/2020	Submitted	COMSA

Table of contents

Publishable executive summary

This document reports about deliverable D4.3 – "HYBUILD Optimised Building Management System" in the context of HYBUILD, an EC co-funded project which aims at developing two innovative compact hybrid electrical/thermal storage systems for stand-alone and district connected buildings, both in Mediterranean and Continental climatic conditions. In particular, this deliverable addresses the implementation of an optimised control of the HYBUILD system energy flows in the residential buildings by considering internal and external requests.

The main aim of this document is to provide a detailed description of the energy management approaches adopted for the Building Energy Management System (BEMS) proposed within the HYBUILD project. It presents the procedures and the features of the two control systems developed for addressing:

- the optimisation of the energy management for the provisioning of flexibility services to grid operators (provided by ENG);
- the minimisation of the energy operational costs (provided by UDL).

The two implemented optimisation processes adopt two different methods and pursue different objectives.

The optimiser provided by ENG relies upon a multi-objective optimisation framework able to handle two or more objectives at the same time; this has been performed by the implementation of a heuristic algorithm, the Non-dominated Sorting Genetic Algorithm II (NSGA II). The appointed objectives are: the provision of flexibility services to the grid operators, the economic management of the energy operations, and the users' comfort satisfaction.

UDL's control strategy implements a reinforcement learning technique, a Deep Learning Control (DLC) algorithm characterised by a three-layer fully connected network. This tool focuses on the internal cost management of the energy operation of each device inside the building for reducing costs.

This report shows how it is possible to adopt different approaches for addressing the same energy operations from two different standpoints. Mostly, the reduction of the costs related to the energy flows among the systems and devices inside the building is always taken into account. The comfort of the building inhabitants is always one of the most referenced constrain of the processes, as well. In this case, the solutions proposed allow also to leverage on the storage systems, in particular the electrical battery and the latent storage, not only for handling internal energy management but also for addressing requests from electric and district heating grid operators for the provision of a flexibility service. This is part of a wider framework of Demand Response (DR) implementation in the field of building energy management.

In this view, a typical DR mechanism has been envisioned within the BEMS optimiser: a grid operator sends a flexibility service request signal that triggers the external request optimisation module of the BEMS. This signal consists of a power profile to be followed by the building controlled by the BEMS whilst absorbing electricity or district heating; moreover, a reward value is provided with it, in order to encourage economically the Energy Manager of the building because this reward corresponds to the economic prize granted if the service is actually delivered. This service request drives the optimisation: the algorithm will leverage on the capabilities of the storage systems installed within the HYBUILD buildings for optimising the provision of the services requested while taking into account the energy operations costs, that are related with this service, and the building inhabitants' comfort. At the end of the process, the Energy Manager has the possibility to choose between a set of optimised solution that are put at disposal by a Decision Support System. This tool indeed selects the most convenient

solution for each aspect taken into account by the multi-objective framework and shows it in a dedicated dashboard where Energy Manager can assess the building performances and select one solution to be implemented in the building.

The delivery of the software reported in this document demonstrates how the participation to Demand Response (DR) programs could be feasible in this context, exploiting the flexibility allowed by the adoption of a HYBUILD solution for the energy management of the building.

According to the user preferences and demonstration needs, the DLC or the NSGA-II approach can be selected. In this perspective, the BEMS can be considered as the harmonization of these two different control strategies. The final version of the BEMS will integrate the final models of the building devices and its user interface. The implemented software will be tested by means of the simulation environment based on TRNSYS developed inside the same WP. This will give the possibility of testing and comparing the two control systems even before their development into the demo pilots. As expected, the results will be shown in the last deliverable of the WP (i.e. D4.4 – Report on system performance).

Acronyms and Abbreviations

BEMS	Building Energy Management System
DC	Direct Current
DHW	Domestic Hot Water
DLC	Deep Learning Control
DR	Demand Response
DSO	Distribution System Operator
MINLP	Mixed Integer Non-Linear Programming
MPC	Model Predictive Control
NSGA-II	Non-dominated Sorting Genetic Algorithm II
P&ID	Piping and Instrumentation Diagram
PCM	Phase Change Material
RBC	Rule Based Control
RPW-HEX	Refrigerant PCM Water – Heat Exchanger

1 Introduction

1.1 Aims and objectives

This report aims at introducing the two approaches adopted for energy management and implemented inside the Building Energy Management System (BEMS) developed for the HYBUILD project. The objective is to describe in detail the procedures and the main aspects of the control system developed by ENG for the provisioning of flexibility services to grid operators and the one developed by UDL for the minimisation of energy operational costs.

The implementation of the BEMS started from the definition of the requirements already presented in deliverable D4.2 – "Functional Requirements Specifications" (Paternò, et al., 2019). In order to align the two control strategies, a strong focus has been done in the initial phases on the definition of the scope of the two implementations, in order to share the same inputs and the same output. Both control strategies share also the same system block modelling, as depicted in Figure 1 and Figure 2.

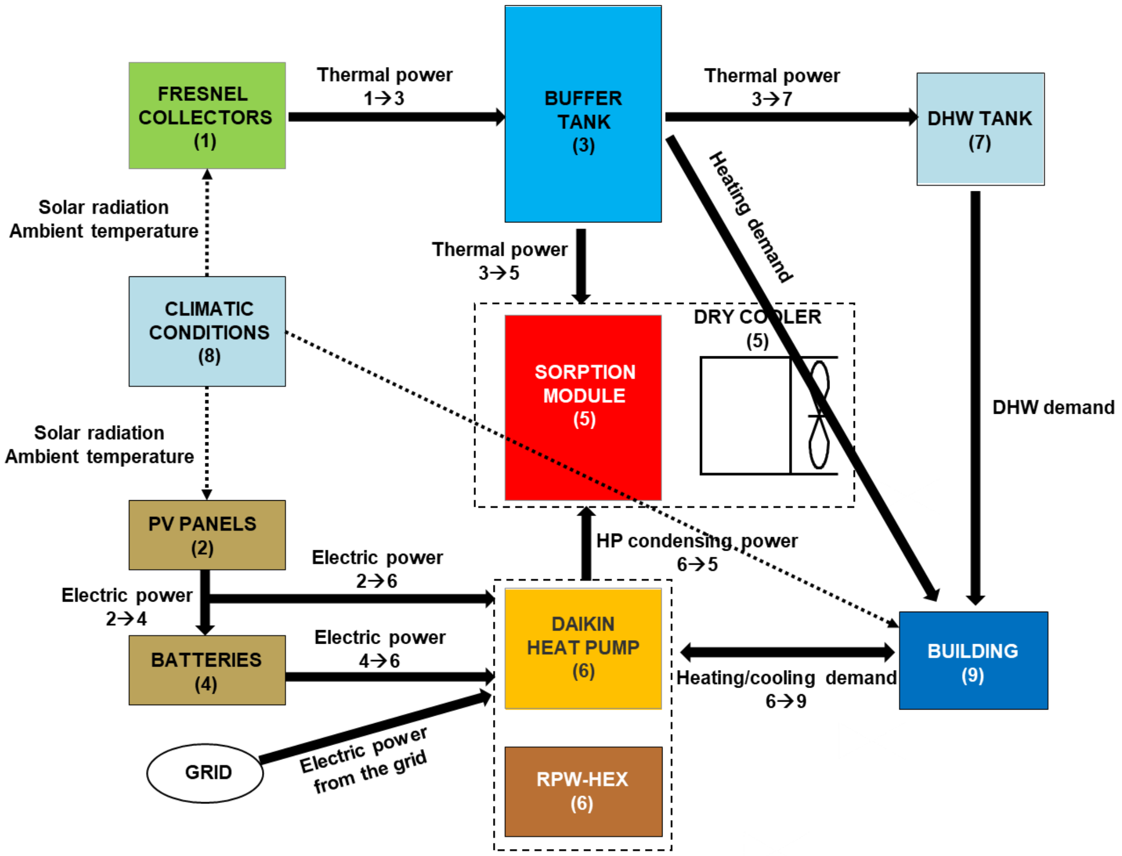

Figure 1. Representation of the Mediterranean system inside the HYBUILD BEMS

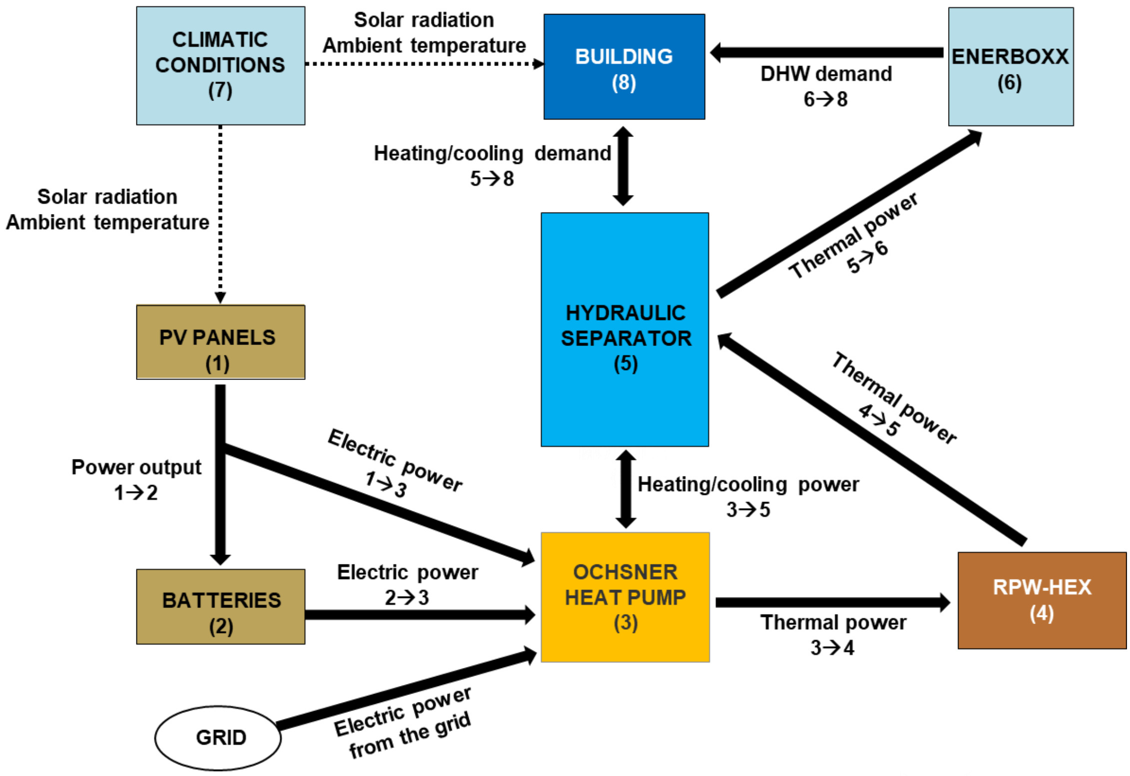

Figure 2. Representation of the Continental system inside the HYBUILD BEMS

Both in the Mediterranean and Continental cases, the overall shared building representations rely on the low-level models of the underlying building devices and subsystems, as depicted in each block. The elicitation of these models required much more time than expected from the single device developers. Most of them have been finalised by the device developer after a long-lasting production phase, and provided very close to the deadline of the current deliverable. At writing time, most of the models are ready for the Mediterranean case and the completion for the Continental one is in progress. Nevertheless, some assumptions have been done to complete the expected development task in time. In the next weeks, the final models will be integrated into the BEMS.

The document is directed to the entire consortium according to the related activities described in the following section 1.2, given the central role of the control unit within the overall system. The dissemination level of the report is "Public" for all external stakeholders potentially interested in a deeper understanding of the working mechanisms of the HYBUILD BEMS, especially from the methodological point of view and its actual implementation.

1.2 Relations to other activities in the project

The development of the BEMS has been done in the framework of Task 4.4 – "Building Energy Management System (BEMS) design" for its design and implementation, with the exploitation of the advanced control prototyping studied in Task 4.3 – "Control prototyping with hardware in the loop". It continues and will exploit the work already initiated in Task 4.1 - "Buildings model and system performance simulations", in order to test the performance with the use of the simulation environment provided in its context.

The implementation of the BEMS has been done in accordance to the monitoring strategy and operational modes defined in Task 4.2 – "Operational function design". For the aim of this report, here the working scenarios of each case will be briefly recapped.

As it happened with deliverable D4.1 – “Smart system algorithms” (Rossi, et al., 2018), this report could not be released without the collaboration with other partners from different WPs. Detailed descriptions of each component or sub-system of both the Mediterranean and Continental cases used by the BEMS control systems have been studied within Task 3.1 – “Model Based Design and Control”. Detailed studies have been carried out at technology level from the developers and a strong effort has been spent in the harmonization of these studies into a single shared definition. In this sense, the interconnection and interdependency between WP3 and WP4 has confirmed to be very high also for this deliverable, and the collaboration between the involved partners has been fostered and will be carried also beyond the end of the deliverable itself.

Other two WPs which have strong relationships with this deliverable are WP1, for the exploitation of the KPIs developed inside that WP for the definition of the correspondent objective functions, and of course WP6, since the BEMS here implemented will be customised and deployed in the demo pilots in the framework of that WP.

1.3 Report structure

The deliverable is divided into five sections.

Section 1 describes the scope of the report, its purpose, structure, contributions and relationship with the rest of the project.

Section 2 introduces the genetic algorithm used to perform the optimisation of the building energy flow considering the inclusion of external requests.

Section 3 describes the control strategy implemented for the Mediterranean case for the minimisation of the building costs.

Section 4 gives the conclusions of the report.

Section 5 provides the list of references.

1.4 Contributions of partners

ENG, as WP leader, coordinated the overall work and designed the structure of the deliverable. As responsible also of the activities for the optimisation, ENG implemented and reported about the developed system able to exploit the flexibility of building for the provisioning of DR service to the grid operators. ENG provided the description of the NSGA-II algorithm, the definition of the objectives and constraints, and a first implementation with simplified models which will be replaced with the final harmonised ones for its validation in the simulated environment.

UDL has been working on the predictive control algorithms and the detailed model of the latent storage, contributing both with a first analysis of the application of that class of algorithms and with the provision of a simplified model of the latent storage for the purposes of the deliverable. UDL has also collected the models of the other building components and subsystems and worked hard for a common harmonized definition. They developed and reported about the DLC implementation for the cost minimisation of the building and presented the first results.

2 HYBUILD building management system optimisation process

2.1 External requests optimisation of building operations

The HYBUILD BEMS optimiser proposed by ENG for the control of the energy operations inside the HYBUILD buildings aims at taking into account simultaneously both service requests from the external world, that are advanced by the electric and/or thermal grid operators, and the economic and comfort aspects of the building inhabitants, focusing on heating, cooling and Domestic Hot Water (DHW) demand.

This approach is driven by the fact that, in many modern energy applications or market contexts, the grid operators are interested in having a flexible energy behaviour from their customers. This means that they expect to ask a service to their customers, for instance to follow an absorption power profile over a well-defined time horizon, that in turn are rewarded in case they are able to respect this request and deliver a service to the grid operators, mostly intended as a flexibility service. In the scenarios described in D4.2, this interaction between grid operator, for instance a Distribution System Operator (DSO), and the energy manager brings to a reward mechanism that implies a net economic revenue.

The other aspect that drives the optimisation philosophy is the comfort of all the people that occupy the building. The energy process envisioned is based on the satisfaction of the setpoint indoor temperature and the setpoint DHW temperature, set by the users of the BEMS. These two setpoints will be at the basis of the management of the energy operations of all the components inside the buildings: coherently with the operational modes presented in the above mentioned D4.2, the request of internal energy performance is matched with a unique system configuration in terms of active components that are able to provide that amount of thermal energy.

Finally, the economic aspects of the energy process are taken into account by the optimisation as well. The main cost elements related to the building energy operation are mostly linked to the purchase of electrical energy from the grid, for supplying both the Direct Current (DC) bus, providing electricity to the heat pump system, and the electric back-up, providing thermal energy to the DHW tank. In this view, the storage systems of the HYBUILD building configurations allow to leverage on their capabilities to optimise the period of the day during which it is convenient withdrawing energy from the grid and those during which the storages efficiently contribute on energy operations of the building. This, indeed, leads to economic savings for the Energy Manager of the building.

All these different aspects of the building operations management are handled at the same time by a unique optimisation process that is able to take into account several goals simultaneously. This is achieved through the implementation of a multi-objective optimisation algorithm in charge of maximising and/or minimising different objective functions in a single process. The adopted algorithm is the Non-dominated Sorting Genetic Algorithm II (NSGA-II) (Deb, Agarwal, Pratap, & Meyarivan, 2002); the mathematical formulation and the details of the algorithm itself are be provided in the following sections. It is a heuristic algorithm, in particular a genetic one, that relies upon the replication of some biological phenomena for setting up an optimisation procedure. This typology of algorithm allows to handle complex problems that, usually, deterministic approaches fail to address, even though they deal with a lack of accuracy of the solutions in exchange for higher computational speed.

This optimiser is the core of the BEMS system and is developed in Python 3 (Python Software Foundation) exploiting many of their mathematical and scientific libraries. The entire BEMS solution is proposed as a Django (Django Software Foundation) project and all its components will be delivered as applications of the same BEMS solution, for instance, graphical user interfaces, the database, Application Programming Interface, etc.

2.2 Optimisation algorithm: Non-dominated Sorting Genetic Algorithm II

As mentioned in the introduction and presented in D4.1, the adopted algorithm is the NSGA-II. Here a complete explanation of its features and steps is provided. In the following section its adaption to the BEMS optimisation problem is addressed as well.

NSGA-II relies on the ordering criterion of the population (the set of all the individuals) following the Pareto dominance definition (see D4.1). After having properly initialised a starting population, it is ordered following this philosophy: individuals not-dominated by any other individual are grouped in a front characterised by a higher rank; individuals not-dominating any other individual are grouped in a front characterised by a lower rank. This process is performed acting on the entire population, removing all the not-dominated individuals once that this control is performed checking dominance with every other individual of the population. These not-dominated individuals are put in the front with the highest rank available (the rank is 1 if it is the first iteration). This process is performed again on the remaining set of individuals until all the individual are properly grouped in ranked fronts.

Once that this first ranking procedure is completed, another sorting criterion is implemented within each rank. This criterion is based on the concept of crowing distance, which is an index evaluating how an individual of the population is close to the neighbour individuals in the solution space. This index is indeed calculated on the basis of the values of the evaluated objective functions for that individual in the context of its front: infinite distance is assigned to boundary values of the individuals of that front; an index evaluated on the basis of the objective function values is assigned to all the other individuals. Ranking the individuals following this index allows to explore in a deeper and larger way the solution space.

At this stage, the genetic engine of algorithm starts to perform its actions. The mechanisms of Binary Tournament Selection and crowded-comparison-operator (giving priority to higher ranks and higher crowing distances) select the more valuable individuals for being the parents of the child generation. According to the analogy with the biological phenomena, crossover and mutation actions are performed on the selected individuals, through Simulated Binary Crossover and Polynomial Mutation mechanism. The former acts on two parent individuals, actually mixing them, and the latter acts on one parent individual modifying it.

The following step introduces the most innovative feature of the NSGA-II, referred as elitism, which allows to avoid possible losses of valuable solutions from the parent generations. In this view, the parent and the child population are mixed, they are ordered again following the same criteria and only the best N individuals are selected for the next generation, where N is the number of individuals within the population.

These steps are performed for a prefixed number of generations. At the end of this iteration, the algorithm provides as results an entire population of Pareto-optimal solutions.

In Figure 3, the entire NSGA-II algorithm process just described is shown through a flow chart.

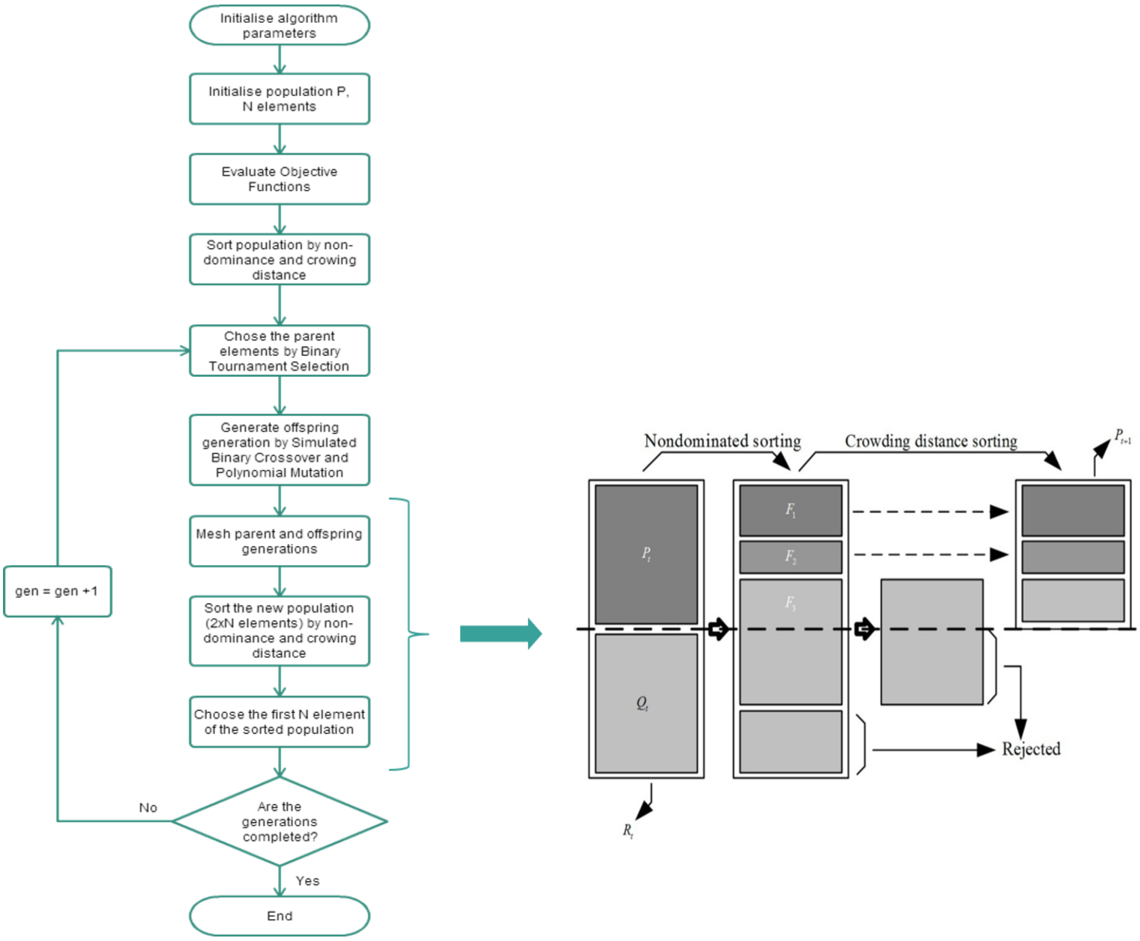

Figure 3. Flow chart of the NSGA-II algorithm and elitism block representation

2.3 Optimisation workflow

This section presents the main steps and the workflow of the optimisation process proposed by ENG for the HYBUILD BEMS solution.

The first phase consists in a set of initialisation actions aiming at collecting all the necessary information and external data for triggering the optimisation process.

The second phase refers to the optimisation process itself and the conditioning of the input data and output data of the process, in order to be coherent with the data structure adopted by the algorithm.

2.3.1 Initialisation phase

The first step towards the initialisation of the whole BEMS optimisation process is the definition of the building to be managed. The solution proposed by ENG is seamlessly able to handle both Mediterranean and Continental buildings; in this view, the first module is in charge of modelling most of the building features, in term of data about the building itself, such as surface, city, etc., and about all the systems and components installed inside the building, along with all their features. According to the different climate conditions of the city the building belongs to, the building is characterised with different systems and devices.

This is the main definition that could always be used and configured by the Energy Manager starting from a pre-set building configuration.

The second cluster of information necessary for the initialisation of the optimiser is the reception of an optimisation request that represents a trigger for the optimisation process and collects all the significant data about the optimisation itself:

- starting and ending time;
- time slot;
- objective to be optimised by the BEMS;
- optimisation status.

This trigger is always related to a single building, and it is characterised by the building itself. Whenever the optimisation process needs information on duration of the operations, days and status of the operations, etc., refers univocally to this data. The results of the optimisation are related to this data as well.

Given that, one of the most important definition performed during this phase is the proper configuration of the operational modes that can be implemented by the HYBUILD building systems in relation to the different energy needs of the Energy Manager and the Users of the BEMS; details about these roles and the operational modes are reported in deliverable D4.2. Here, some definitions are reported for the sake of clarity. The Energy Manager is the role in charge of configuring the BEMS according to the building he is responsible for, take decisions about the energy operations of the buildings and is responsible for the technical and economic performances of the BEMS operations within the building. The User is the role, mostly coinciding with building inhabitants or energy users, that exploits the benefits of the BEMS operations and can deal with the BEMS with dedicated dashboards for setting temperature and other parameters. As for the operational modes, they identify all the possible working conditions that can be actuated by the Mediterranean and Continental solutions.

These last elements are essential for the optimisation procedure, given that they define each system or device that can be activated for achieving a particular building modality. In general, these modalities are: cooling, heating, DHW, and charging (the Mediterranean system, having also a Fresnel system, can implement also a solar modality). Each of them defines which systems or devices inside that building is turned on or switched off for providing the modality it refers to.

In this phase, the optimisation process starts from the building design for assessing which is the operational mode configuration, thus the list of all the operation modes. Upon that, the possibility of having more than one operational mode at the same time, that means during the same time slot, is explored. This is due to the fact that the Energy Manager or the Users could ask for a complex energy behaviour of the building that entails more than one modality active at the same time. A typical situation could be, cooling/heating modality to be performed with DHW one, or these just mentioned associated with a charging mode. Given that, in this phase the optimisation process creates a list of possible complex operation modes that can be implemented inside the building on the basis of the compatibility of one operational mode to another. This is mostly evaluated considering the active systems for each operational mode and assessing if the systems active for two different operational mode are in contrast with each other.

Another crucial step for the definition of the optimisation environment is having the building demand in terms of thermal power to be provided to the building in order to fulfil the comfort request of its inhabitants. Relying upon the simulations performed by EURAC and reported in (EURAC, 2019)b, a tailored module of optimiser is able to retrieve the heating, cooling and DHW demand for each hour of the day over a year in accordance with the temperature setpoints provided by the Energy Manager and/or the Users of the BEMS application. These data are then properly modelled and arranged in a data structure compliant with the one adopted for the optimisation process. In this view, the definition of the thermal and DHW requests from these actors goes hand in hand with the building demand; the temperature setpoints are indeed set and chosen by Energy Managers and Users as a request. This information is then handled properly for being put in relation with the simulated data and then used for creating building demand data structures. The setpoints from the buildings actors are manipulated as inputs coming from their dedicated dashboards.

The steps just described are mostly linked with the description of the building features. The next procedure is related to ambient conditions. In order to correlate the actual energy building demand and the solar systems capability with the operation managed by the BEMS optimisation, two crucial ambient conditions are needed: ambient temperature and solar radiation. The initialisation module in charge of gathering these data accesses to an open weather service, Weatherbit (Weatherbit, 2020).

The last step of the initialisation phase concerns the conditioning of other data coming from the external world, thus the requests from grid operators for the provision of flexibility services in shape of Demand Response mechanisms. In this implementation, the BEMS considers both DSO and district heating operators (for the Continental climate) in charge of sending service request profiles to the Energy Manager. These profiles consist in a power absorption profile for a determined time horizon, a tolerance range within which this service can be considered satisfied, and a money reward signal for pushing the BEMS in pursuing the objective of providing the above-mentioned flexibility services to the grid operators. All these data are arranged in tailored structures for being handled by the BEMS optimiser.

These are the initialisation procedures that are essential for the proper configuration and the definition of all the data that are treated and used by the optimisation algorithm. Its phases are described in the next section.

2.3.2 Optimisation phase

This second phase of the optimisation process is related to the implementation of the algorithm itself, as described in section 2.2, and the data conditioning operations.

As mentioned in the introduction of this section, the adopted optimisation approach entails some genetic procedures for getting to an optimal Pareto front. Many steps are envisioned for mimicking the evolutionary biological phenomena, which are briefly presented here.

This first step of this phase performs the scheduling of the possible complex operational modes found in the initialisation phase that rules the energy flows among the building components in accordance with heating, cooling and DHW building demands. The optimisation algorithm has to know among which possible configurations is free to find a possible solution; this procedure defines the variable space of the algorithm. Once found the possible system configurations, defined by a set of complex operational modes, the heuristic engine of the algorithm seeks for the Pareto optimal solutions. This procedure is performed by assessing, time slot by time slot, the thermal demand (heating or cooling, these two modality cannot occur simultaneously) and the DHW demand. According to what is requested for heating/cooling the building and for heating the water, for each time slot, a set of complex operational modes are selected. These are able to provide the necessary thermal supply to the building, thus, time slot by slot, they ensure that all the components needed for providing the requested thermal energy are operational.

In simple words, this step allows to know in each time slot which are the possible solutions to be adopted by the BEMS for providing the requested heating/cooling and DHW demand: for instance, the algorithm could select, driven by its objectives, a configuration supplying heating demand through RPW-HEX storage or another configuration in which heating demand is satisfied by the heat pump supplied by the battery.

Once the building system configuration is defined, the proper optimisation procedure begins. The first step of the algorithm consists in creating the first population that will start the process, created randomly from the possible system configurations defined above. It is important to understand how each single individual of the population is structured: an individual will be the collection of ordered time slots from the starting time of the optimisation request trigger until its ending time. In this way, each individual will represent an energy operation configuration for the entire time horizon. This solution has been chosen because of the presence of two different energy storages upon which the optimisation can leverage for finding an optimal technical and economical solution: having a complete view on what happens during the entire time horizon in a unique solution allows to take into account the best storage management over the different periods of the day, especially focusing on the varying energy tariffs and the peaks of flexibility requests from grid operators.

Given this, each individual of a population consists of a set of decision variables equal in number to the number of time slots. Each decision variable, in turn, reports the information about the system configuration and can be considered as an array. The first information in this array reports the time slot the decision variable refers to; the other ones report the status of each system installed inside the building. This structure is depicted in Figure 4 (where: Z is the dimension of the population; K is the number of time slots; P is the number of building devices).

$$population = [individual_1 \quad individual_2 \quad \dots \quad individual_i \quad \dots \quad individual_Z]$$

$$individual_i = [decision\ variables_{i,1} \quad decision\ variables_{i,2} \quad \dots \quad decision\ variables_{i,j} \quad \dots \quad decision\ variables_{i,K}]$$

$$decision\ variables_{i,j} = [time\ stamp_{i,j} \quad device\ status^1_{i,j} \quad device\ status^2_{i,j} \quad \dots \quad device\ status^P_{i,j}]$$

Figure 4. Representation of the population individual used for the NSGA – II algorithm

The decision variable identifies a well-defined flexibility availability of the HYBUILD building system. Moreover, some of the device statuses, apart from their Boolean characterisation, could also determine the variability on another level, referred more strictly to the device itself. Indeed, some of the systems or devices installed within the building can control their energy behaviour not only depending on their status (turned on or switched off) but also setting a precise value of power absorption and/or injection.

Through the definition of the first population set aimed at starting the algorithm, it is also clear the extension of the decision variable space into which the algorithm will seek for a Pareto optimal solution. This will help also to understand how the algorithm will implement the genetic operator that are the evolutionary engine of the process.

The following step of the algorithm is actually the evolution of the population, generation by generation, towards the Pareto optimal front. As briefly explained in the dedicated section, during each generation two main genetic operators are implemented:

- Simulated Binary Crossover;
- Polynomial Mutation.

In the case of the proposed optimisation problem, an expedient is adopted for having child populations that are coherent with the parent ones. As already said, each individual of the populations consists of a set of ordered decision variables that are characterised by their time slots for allowing the description of the building energy behaviour throughout the time horizon. Due to this, the implementation of the genetic operators has to take in account this chronological order because it entails the reasoned mix of one individual (in the case of Polynomial Mutation) or two individuals (in the case of Simulated Binary Crossover) of a same parent population for creating an individual of the child population; as a matter of fact, each individual of this child population has to respect this chronological order for providing a technically feasible solution. In this view, the proposed optimisation process adopts the following criterion for implementing the genetic operators: taken one or two individuals of the parent generation, the decision variables to be handled by the operators are referred to the same time stamp; moreover, this shall be done for all the decision variables and the new born individual of the child population shall be complete, thus it has to consists of all the decision variables foreseen for the addressed time horizon (one decision variable for each time slot). Finally, for ensuring the technical feasibility of the new born individuals, each decision variable configuration has to be checked in order to be mapped over one of the possible complex operational configurations in that time stamp.

This is the method that has been adopted for having coherent solutions generation by generation and represents a customisation and an improvement of the optimisation procedure proposed by the algorithm. The other steps presented in the section 2.2, are then performed as foreseen by the algorithm.

Generation by generation, each individual of the child population is characterised by its objective functions values and it is stored in a tailored data structured for being processed by the sorting procedure. In order to enrich this new population with valuable solution and to avoid to leave behind solution that could improve the entire process, before proceeding the child population is enlarged by adding the individuals of the parent population.

As clearly reported by the algorithm name, the core of this genetic algorithm is represented by the sorting criterion that relies upon non-domination concept defined within the Pareto-optimality in multi-objective methods, see D4.1. During the sorting procedure, the individuals are arranged in fronts, properly ranked, and, within a single front, they are ordered following the crowing distance index. Once that all these techniques are implemented and completed, the child population undergoes the elitism technique and only the best solutions are selected for the next generation.

Once that all the generations are completed, the resulting data structure contains the Pareto-optimal solution of the problem. This set goes hand in hand with the Boolean and analogical setpoints that define the complete set of the optimised solution to be effectively implemented by the systems and devices inside the HYBUILD buildings.

The fact that this algorithm provides a Pareto-optimal front push for a Decision Support Layer able to select automatically the solutions among this front that allows the better performances with respect of the adopted objective functions. In this view, two solutions are proposed as the most valuable for the HYBUILD building configuration:

- the solution providing the best performance related to the economic management of the energy processes of the building;
- the solution providing the best performance related to the provision of flexibility services to the grid operator.

The Energy Manager is then provided with a tailored dashboard for assessing the results of both solutions in details and choosing one of those for being implemented by the BEMS.

2.4 Decision variables and constraints of the optimisation process

The decision variables of the optimisation process addressed by the BEMS are mostly referred to the status of the devices within the building whose energy operations have to be managed. A first description of these variables is provided in the previous section and in particular in Figure 4, in relation with the steps of the optimisation algorithm itself. Here, a more detailed definition is provided.

It is worth noting that the decision variables are different on the basis of the climate of the building to be optimised, thus on the basis of the systems and devices installed within the building itself. In this view, the envisioned decision variable for the Mediterranean building is

$$decision\ variable_{i,j}^{med} = \begin{bmatrix} time\ stamp_{i,j} \\ heat\ pump\ status_{i,j} \\ sorption\ module\ status_{i,j} \\ RPW - HEX_{i,j} \\ dhw\ tank\ status_{i,j} \\ buffer\ tank\ status_{i,j} \\ fresnel\ system\ status_{i,j} \\ dc\ bus\ status_{i,j} \\ electric\ backup\ status_{i,j} \end{bmatrix}$$

whilst, for the Continental climate building, the decision variable is:

$$decision\ variable_{i,j}^{cont} = \begin{bmatrix} time\ stamp_{i,j} \\ heat\ pump\ status_{i,j} \\ RPW - HEX_{i,j} \\ enerboxx\ status_{i,j} \\ district\ heating\ status_{i,j} \\ dc\ bus\ status_{i,j} \\ electric\ backup\ status_{i,j} \end{bmatrix}$$

As a matter of fact, each decision variable is directly linked to a complex operational mode, which is the configuration chosen by the algorithm for that individuals on that time stamp for addressing building heating, cooling and DHW demand. As explained above, once defined the status of each device, it is possible to obtain and set the power values associated with each of them.

As for the constraints of the process, the BEMS adopts the same rated values of the components data sheets. For the sake of simplicity, they are not reported here; please refer to these documents for having a complete definition of these data.

2.5 Objective functions and related KPIs

As stated many time over this section, the proposed optimisation framework provides multi-objective capabilities and allows to take into account different needs of the energy management within the HYBUILD building. The BEMS optimiser addresses simultaneously three different aspects:

- the thermal comfort of the building inhabitants;
- the economic management of the entire energy process;
- the provision of grid services to the grid operators by leveraging on building devices capabilities.

Some aspects are taken into account by the use of proper constraints for the definition of the building operational configuration. As explained in 2.3.1, comfort requests from the Energy Manager and the Users are addressed by properly initialising heating, cooling and DHW requests on the basis of the setpoint temperatures and the building thermal behaviour simulations. These pieces of information drive the entire optimisation process from the very beginning and ensure the satisfaction of the comfort objective. Since no dedicated KPI are referred in D1.3 (Barchi, et al., 2018) to this comfort aspect, apart from thermal complaints, the BEMS solution aims at addressing this aspect by calculating the average deviation of the temperature set from the monitored one. In the flowing, the definition of the Temperature Average Deviation (TAD) for both building and DHW is provided:

$$TAD_{building} = \frac{\sum_{i=1}^{n}\left(T_{i,building}^{monitored} - T_{i,building}^{setpoint}\right)}{n}$$

$$TAD_{DHW} = \frac{\sum_{i=1}^{n}\left(T_{i,DHW}^{monitored} - T_{i,DHW}^{setpoint}\right)}{n}$$

where:

- $T_{i,x}^{monitored}$ is the monitored temperature of the building/DHW in that ith time slot;
- $T_{i,x}^{setpoint}$ is the setpoint temperature of the building/DHW in that ith time slot;
- n is the number of time slots inside the time horizon.

This calculation cannot be performed at optimisation time, since it needs the monitoring temperature data of the building and DHW. It will be calculated by a BEMS dedicated module after the optimisation.

The economic aspect of the energy management inside the building is taken into account inside the optimisation process by means of a dedicated objective function. This refers to the economic balance between the expenditures for purchasing energy from outside the building, element that can be optimised by leveraging on storage systems, and the rewards provided by the appointed grid operators for the provision of flexibility services, as explained for the next objective function. This reward is evaluated time slot by time slot and it is achieved by the building if its building behaviour in terms of energy demand to these grid operators stays within a pre-defined range set by the grid operators themselves while requesting the flexibility service. In the following, the Economic Balance (EB) objective is presented:

$$EB = \sum_{i=1}^{n} (energy\ cost_i - grid\ operator\ reward_i)$$

$$energy\ cost_i = energy\ price_i \times (grid\ energy_i + back-up\ energy_i);$$

$$grid\ operator\ reward_i = reward_i \times grid\ operatorservice\ activation_i;$$

where:

- $energy\ price_i$ is the price set for purchasing $grid\ energy_i$ (electrical from distribution grid or thermal from district heating) in that ith time slot;
- $back-up\ energy_i$ is the energy for supplying in back-up mode the DHW devices in that ith time slot;
- $reward_i$ is the reward proposed by the grid operator in that ith time slot;
- $grid\ operatorservice\ activation_i$ is a Boolean indicating if the service has been provided or not in that ith time slot;
- n is the number of time slots inside the time horizon.

Finally, the third objective function is the one characterising the proposed optimisation process, since it is the only feature of the BEMS that parametrises the flexibility service to the grid operators; it is actually one of the selected KPI (KPI.6 – Flexibility) chosen in D1.3 (Barchi, et al., 2018). This objective function takes into account the provision of flexibility services to the grid operator that request a well-determined behaviour of the building in terms of energy withdrawn at the point of delivery in exchange for a reward, to be paid in case the BEMS is able to drive the energy behaviour of the building in the tolerance range defined by the grid operator. These requests are sent before the optimisation process and drive the optimisation itself. The possibility of having this mechanism of requests and response from the building energy operations entails the implementation of a Demand Response service. The objective function that parametrises this phenomenon is than labelled as Demand Response Power Tracking (DRPT):

$$DRPT = 1 - \frac{\sum_{i=1}^{n} \left| E_{Response,i} - K_{CL} E_{Demand,i} \right|}{\sum_{i=1}^{n} E_{Response,i}}$$

$$K_{CL} = \frac{\sum_{i=1}^{n} E_{Response,i}}{\sum_{i=1}^{n} E_{Demand,i}}$$

where:

- $E_{Response,i}$ is the energy behaviour of the controlled system in that ith time slot;
- $E_{Demand,i}$ is the desired energy behaviour request in that ith time slot;
- K_{CL} is the indicator of the contribution level the building may provide, calculated as normalisation factor to compare the two profiles;
- n is the number of time slots inside the time horizon.

This is the most general formulation of this objective function. In any case, as explained in section 2.3.1, according to the climate referred into the optimisation, there could be two different grid operators requesting for such services: DSO and district heating operators.

2.6 Integration of models and simulations

The process performed by the BEMS optimisation has to rely upon the modelling of both the building energy behaviour and the single system and device energy operations.

As for the building, the most important phenomena to take into account are the thermal and DHW demand for satisfying the building inhabitants comfort requests. This has been performed taking as a reference the simulation performed in (EURAC, 2019)a and (EURAC, 2019)b reporting the heating, cooling and DHW demands for each hour in a year at different temperature setpoints for the Mediterranean and Continental HYBUILD reference buildings, located in Athens and Stuttgart, respectively. As mentioned in section 2.3.1, these simulations are the basis for retrieving the building demands inside the optimisation time horizon.

Another relevant modelling feature of the BEMS is the use of the operational modes already presented in the previous sections. These are able to report in a clear and unique way the operational configuration of the building. Relying upon the description and the Piping and Instrumentation Diagram (P&ID), the BEMS configures the proper connection between all the systems and devices present in both the buildings and, consequently, the BEMS derives the possible compatibility between each operational mode, thus all the possible system configurations in terms of active systems or devices for providing a certain combination of building modality, such as heating, cooling and DHW.

Due to the high-level approach implemented by the BEMS optimisation process, focused on providing services to external actors like electric and district heating grid operators, the most important modelling process is the one just presented. This indeed allows to represent the behaviour of the building for interfacing both with the building inhabitants and the operators at the points of delivery of electricity and thermal energy.

As for the models of all the systems and devices, due the high computational effort that a complete simulation would require to the BEMS optimiser during the execution of the optimisation algorithm, the BEMS does not implement complete models of each systems. In this version, simplified models and performance maps of general equipment comparable to the systems and devices of HYBUILD reference building are adopted. In the future implementation of the BEMS inside the pilots, this software solution will be enhanced with the detailed models of the equipment installed inside each pilot building.

3 High-level control strategy for the Mediterranean system

3.1 Introduction

Beyond Rule Based Control (RBC), some control techniques exist that may improve our system performance, particularly when forecasted information is available. Model Predictive Control (MPC) is a technique that has increased its popularity for energy systems control and has proved its good performance (Rawlings & Mayne, 2009). MPC requires defining an operating time horizon of the system as an optimization problem. Usually, such a problem is encoded as a Mixed Integer Non-Linear Programming (MINLP) problem (Sahinidis, 2019) and requires of specialized solvers to find optimal solutions such as Solving Constraint Integer Programs (Gleixner, et al., 2018). However, current state-of-the-art solvers only deal with certain type of non-linearities, making, sometimes, hard or impossible to express a complex system as a quasi-linear system. Furthermore, if one can think of simplifying complex models to suitable linear expressions, its necessary accuracy remains as an open question.

A look at the description of the simplified models for the optimization of the Mediterranean system (Zsembinszki, 2019) reinforces such point of view. As an example, consider the Heat Pump+ Refrigerant PCM Water (RPW) subsystem. One of its operational modes (cooling mode 3) is modelled as an iterative function, making impossible to derive MINLP expressions to those problems. Not to mention several non-linearities found in other subsystems as the establishment of the states of charge for Phase Change Material (PCM), computation of the cooling power for different cooling modes, as well as the rules of activations for DC-bus subsystem.

Under this scenario, as mentioned in (Rossi, et al., 2018) (Section 2), our focus points to a control system based on reinforcement learning techniques.

3.2 Deep Learning Control

Our proposal of smart control for HYBUILD is based on a typical reinforcement learning paradigm as represented in Figure 5.

In this schema, the environment represents our HYBUILD system, which can be described by its corresponding models as well as its state. The agent, based on the environment state, decides an action that provokes a given reward. That reward is, actually, the optimization objective, and it is feed back to the agent in order to learn about the corresponding action and determine an optimal policy.

There are many types of agents that define a particular machine learning technique. In this case, we consider a 3-layers fully connected neural network of size $N_{inp} \times N_{hid} \times N_{out}$ with the following characteristics:

- N_{inp} is the number of inputs, defined by the system variables as well as the system state vector.
- $N_{hid,cool}$ and $N_{hid,heat}$ are the hidden layer sizes for cooling and heating modes respectively. It uses to be much larger than inputs and outputs. Actually, it is adjusted by a later hyper-parameter setting.
- $N_{out,cool}$ and $N_{out,heat}$ are the cardinality of the actions set.

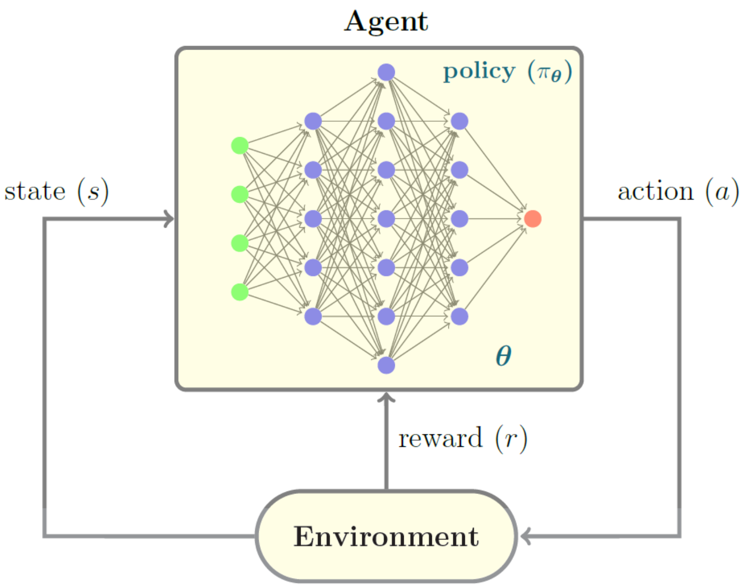

Figure 5. Reinforcement learning paradigm

3.2.1 Control model

The HYBUILD system model operates at two different slotted time scales. First, a finer slot is considered in order to numerically compute the HYBUILD system behaviour (we typically take 3 minutes). Second, a larger slot is used to manage the control system (15 and 30 minutes have been considered). We denote both slots as Δt and T_s respectively. Inside T_s, any action decided by the control system is invariant until reaching any subsystem limit. As an example, if during a given slot one decides to charge the Refrigerant PCM Water – Heat Exchanger (RPW-HEX) subsystem, the charging process will not stop until reaching the maximum state of charge. Similarly, the input system variables for the control system are considered invariant in T_s.HYBUILD control model for the Mediterranean system may be defined for cooling or heating purposes, but the heating model can be considered a subset of the cooling model because heating operations for the Mediterranean are much simpler. Actually, heating bypasses RPW-HEX and Sorption subsystems, resulting in only one operational mode for the Heat-Pump subsystem.

The system variables considered as an input to our control model are:

- Thermal energy demand for cooling/heating in the current T_s ($TE_{T_s}^{dem}$).
- Thermal energy demand for DHW in the current T_s ($TE_{T_s}^{dhw}$).
- Ambient temperature.
- Solar radiation.
- Energy price (cost) for electric demand in the current T_s (C_{T_s}).

As system status variable, we consider:

- Battery state of charge in DC-bus subsystem.
- State of charge of RPW-HEX subsystem (only for cooling model).
- Buffer tank temperature.

All input variables are standard normalized but energy price for electric demand, which has been considered binary because we deal with a binary electric tariff as a function of daytime. Consequently, $N_{inp} = 8$.

The set of actions $\mathcal{A}$ that guide control can be defined as $\mathcal{A} = (\mathcal{C}; \mathcal{S}; \mathcal{B})$, where $\mathcal{C}$ is the set of cooling/heating operation modes, $\mathcal{S}$ is the set of activation modes for the sorption subsystem and $\mathcal{B}$ is the set of battery modes in the DC-bus subsystem. As only the set $\mathcal{C}$ differs for the cooling and the heating models, one can differentiate the set of actions accordingly: $\mathcal{A}_{cool} = \{\mathcal{C}_{cool}, \mathcal{S}, \mathcal{B}\}$ and $\mathcal{A}_{heat} = \{\mathcal{C}_{heat}, \mathcal{S}, \mathcal{B}\}$.

According to (Zsembinszki, 2019) Section 6, $\mathcal{C}_{cool} = \{0, 1, 2, 3, 4\}$ where 0 corresponds to operational mode 5 (heat pump is off, cooling system is off) and the rest match with its corresponding mode. $\mathcal{S} = \{0, 1\}$ because the sorption subsystem may be on or off. For the heating modes, as sorption and RPW-HEX subsystems are bypassed, only one operational mode is considered, being $\mathcal{C}_{heat} = \{0, 1\}$.

Concerning the actions related to DC-bus subsystem, as reported in (Koch, 2019), the high-level control may determine the $E1$ and $E2$ thresholds that define the area of DC-bus operation, as well as the maximum charging/discharging power when operating in charge/discharge areas. As we only deal with discrete values for our control model, we have simplified the DC-bus control operations according the following rules:

- Charging/discharging power is set to a fixed value, namely 3 kW.
- If from control we want to force the DC-bus to operate in charging, buffer or discharging mode, we set the pair of values ($E1$, $E2$) to 3 fixed levels: (75, 90), (10, 90), and (10, 25), respectively, as a percentage of the battery state of charge.

Following these assumptions, $\mathcal{B} = \{0, 1, 2\}$, which corresponds to charging, buffer, and discharging mode, respectively.

Finally, considering that during cooling mode 2 (all energy to refrigerate from PCM) sorption mode is in mode 0, the set of possible actions are:

$$\begin{aligned}\mathcal{A}_{cool} = \{&[1, 0, 0], [1, 0, 1], [1, 0, 2], [1, 1, 0], [1, 1, 1], [1, 1, 2],\\ &[2, 0, 0], [2, 0, 1], [2, 0, 2],\\ &[3, 0, 0], [3, 0, 1], [3, 0, 2], [3, 1, 0], [3, 1, 1], [3, 1, 2],\\ &[4, 0, 0], [4, 0, 1], [4, 0, 2], [4, 1, 0], [4, 1, 1], [4, 1, 2]\}\end{aligned}$$

and $|\mathcal{A}_{cool}| = N_{out,cool} = 21$.

In heating mode, considering that sorption mode is always off, results:

$$\mathcal{A}_{heat} = \{[1, 0, 0], [1, 0, 1], [1, 0, 2]\}$$

and $|\mathcal{A}_{heat}| = N_{out,heat} = 3$.

It should be noted that all the cases where cooling/heating mode is 0 may be omitted because:

- If there is some energy demand, cooling/heating mode 0 is not an option.
- Otherwise, without energy demand, any cooling/heating mode will perform as mode 0 inside T_s.

In other words, mode 0 is adopted when energy demand is null.

3.2.2 Network description

For the purpose of the current deliverable, only a three-layer fully connected network is used. Using additional layers as well as their sizes is left for a future work where hyper-parameter optimization is planned to be analysed.

The tree layers and their corresponding characteristics are the following:

- Input layer size. N_{inp} = 8. Standard normalized.
- Hidden layer size. $N_{hid,cool}$ = 1,000. $N_{hid,heat}$= 100. Activation function ELU. Dropout rate 0.8.
- Output layer size. $N_{out,cool}$ = $|\mathcal{A}_{cool}|$ = 21. $N_{out,heat}$ = $|\mathcal{A}_{heat}|$ = 21. Outputs as softmax of logits. Action is taken as a multinomial of logarithm of outputs.

Other parameters considered and to be optimized in a later hyper-parameter optimization analysis are:

- Learning rate: 0.0005 (α).
- Discount rate: 0.99 (γ).

3.2.3 Learning algorithm

The neural network is trained by a policy gradient algorithm where the cross entropy of the multinomial outputs is minimized. Under this scenario, any objective function may be defined, being based on economic or energy reward. In our case, we consider the following objective function that depends on the neural network parameters, as:

$$J(\boldsymbol{\theta}) = \mathbb{E}_{\pi_{\boldsymbol{\theta}}}\left[\sum_{t=0}^{\mathcal{T}} r_t\right]$$

where reward r_t is defined as:

$$r_t = \left(EE_t^{fg} - 0.5 \cdot EE_t^{tg}\right) \cdot C_t + \left(TE_t^{dem} - TE_t^{hp} - TE_t^{pcm}\right) \cdot Penalty$$

being:

- $\mathcal{T}$, the set of days to be measured within an episodic reward. Training and test sets are described in section 3.3.
- EE_t^{fg}, the electrical energy bought from the grid in slot t required by different subsystems being provided from DC-bus or independently such as the DHW electric tank.
- EE_t^{tg}, the electrical energy sold to the grid in slot t by DC-bus taken from the battery. A factor of 0.5 has been considered.
- TE_t^{hp}, the thermal energy provided for cooling/heating by the heat pump subsystem in slot t.
- TE_t^{pcm}, the thermal energy provided for cooling/heating by the PCM subsystem in slot t.
- Penalty is the cost assumed for a non-covered demand. A value much higher than the energy cost is used.
- C_t and TE_t^{dem} as detailed in Subsection 3.2.1.

Note that TE_t^{dhw} is not part of the objective function because it is assumed that DHW requirements will always be fulfilled by the electrical backup heater.

When training the network, policy parameters (θ) are estimated according a gradient descent algorithm as:

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta})$$

or equivalently, according to the policy gradient theorem, as:

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \alpha \cdot \gamma^t \cdot \sum_{t'=t}^{\mathcal{T}} r_{t'} \cdot \nabla_{\boldsymbol{\theta}} \ln \pi_{\boldsymbol{\theta}}(a|s_t)$$

Being $\pi_\theta(a|s_t)$ a differentiable policy for any action (a) given a system state (s_t).

3.2.4 Implementation aspects

Both, system models as detailed in (Zsembinszki, 2019) and control models here detailed are written in Python 2 (van Rossum, 1995). Furthermore, Tensorflow libraries are used in control models (Tensorflow. an end-to-end open source machine learning platform, n.d.). The availability of a lite version of Tensorflow libraries make suitable this implementation for light hardware or micro-controller environments that may be required for control scenarios in real time.

3.3 Training the network

In this section, we describe the data set used to train and test the network. At this point, our computations are only performed for the cooling mode with Athens weather data, but it could be applied to any other location.

We also discuss the computing training time and its convergence issues.

3.3.1 Training and test data

Cooling data set spans from day 120 to 250 of the year, while heating data set spans from day 290 to 365 and from 1 to 90. Such sets are shuffled and split into smaller sets. Each set is composed of a fixed number of days ($\mathcal{T}$). Actually, its cardinality ($|\mathcal{T}|$) is a parameter. Typically, we take 3 or 6 days on each set. From the 130 available days, we take 18 for testing and the rest for training purposes.

As mentioned in Subsection 3.2.1, our inputs to the control model are: thermal demand for cooling/heating (TE_t^{dem}), thermal demand for DHW (TE_t^{dhw}), ambient temperature, solar radiation, and energy price (C_t).

Ambient temperature and solar radiation are obtained from EnergyPlus weather data Europe WMO Region 6, Greece, Athens 167160 (IWEC) (Weather Data by Location. All Regions - Europe WMO Region 6 - Greece). Because the slot time for this data is one hour, we linearly interpolate when T_s is smaller.

Thermal demand for cooling/heating and thermal demand for DHW are obtained from documents (EURAC, 2019)a and (EURAC, 2019)b respectively and multiplied by the building surface (100 m^2).

For the energy price, we take a two period tariff:

- 0.2 €/kWh from 13:00 to 23:00 hours.
- 0.1 €/kWh the rest of the day.

3.3.2 Training times

Before proceeding to describe the performance results, it is worth to mention a few aspects of the training process. As mentioned, a reward is computed during a set of days (a few days, three or six as an example), then, gradients are computed and propagated. This process is repeated for all the sets in the training set, forming an iteration. Every small number of iterations we apply the learned model to the test set in order to obtain the control system performance, always keeping the best model so far.

Figure 6 shows the reward (or cost) for the test set as a function of the iteration at two different scales, showing the learning process. During the first iterations, the network rapidly finds better strategies than the random at the beginning. It is a common behaviour to get stuck at a local minimum during some large number of iterations. At this point, the discovered strategies are quite good, but still far away from the best ones found beyond 2,000 iterations.

Note that an iteration, in a 3.3 GHz CPU takes approximately 15 seconds (for T_s = 30 min and Δt = 3 min), and consequently, the learning plot shown above results after 10 days of CPU computing.

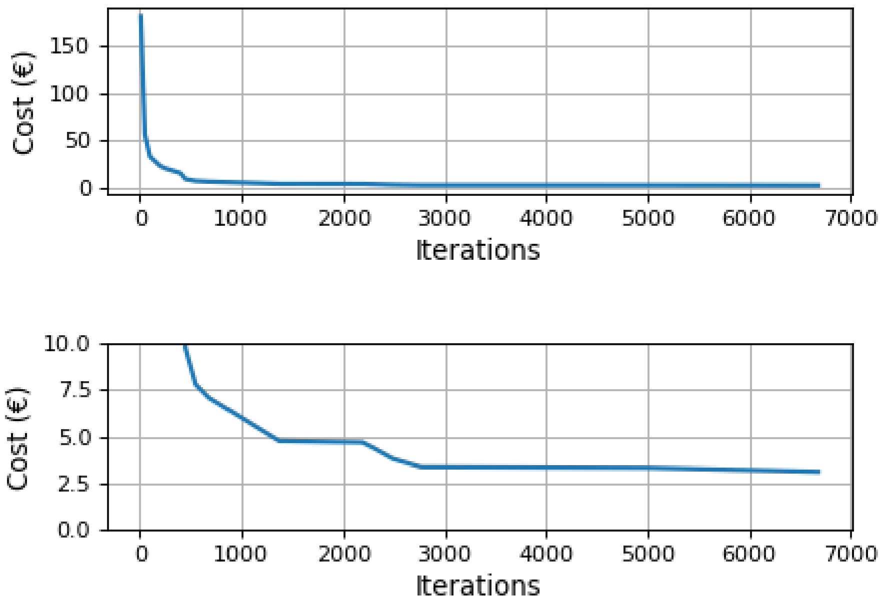

Figure 6. Reward for test set in cooling mode

3.4 Results

The training of the system is performed for the following settings:

- Maximum battery energy in DC-bus: 6 kWh.
- Maximum battery charging/discharging power in DC-bus: 3 kW.
- Photovoltaic panels surface area: 16 m^2.
- Photovoltaic panels orientation: 0°.
- Photovoltaic panels inclination: 30°.
- Photovoltaic panels latitude: 38°.
- Surface area of the Fresnel reflectors: 100 m^2.
- Energy storage capacity of the RPW-HEX subsystem: 35,000 kJ.

As settings for the control system, we take:

- Time slot: Δt = 3 min for cooling mode and Δt = 15 sec for heating mode.
- Delta time: T_s = 30 min.
- Six days for each set of data. Test set consists of three sets (18 days).

Figure 7 shows the performance of the trained network for the test set. The plots, from up to down, show:

1. The cooling and DHW demand. For the cooling demand, green and orange areas show how the demand is fulfilled; whether from heat pump or from PCM (RPW-HEX).

2. PCM and DC-bus battery state of charge, as well as the solar radiation.
3. Temperature in the buffer tank and the resulting thresholds on DC-bus subsystem and activation of sorption subsystem according to the corresponding action.
4. Cooling mode for Heat Pump, energy tariff as binary (0 corresponds to 0.1 €/kWh and 1 to 0.2 €/kWh) and cumulative operation cost.
5. Amount of energy sold and bought for every price.

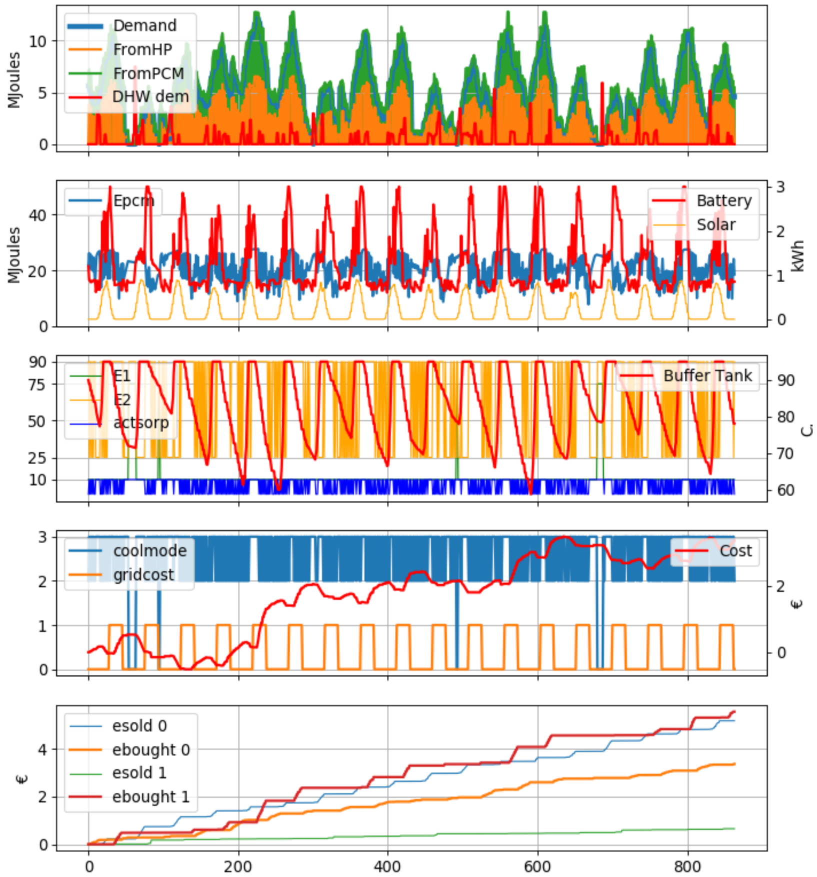

Figure 7. Reward for test set in cooling mode

Carefully inspecting Figure 7 some aspects of the control policy may be highlighted:

- The operational cost for the 18 days of the test set is very low (3.4 €). As seen below, it is far better that any of the rule based control policies tested under the same scenario (detailed in Subsection 3.4.1), indicating that our deep learning control approach is highly efficient.
- Cooling demand is always covered either from HP or from PCM in order to avoid the penalties.
- Cooling modes 1 and 4 are never (or rarely) used.
- A zoom view represented in Figure 8 shows DC-bus discharging battery at peak tariffs periods by adjusting the *E*2 threshold, putting DC-bus in discharging mode.
- Control uses PCM as a buffer and avoids its full discharge in order to avoid lack of demand cover penalties.
- As seen from the bottom plot, the amount of energy sold in tariff period 0 (low cost) exceeds the amount of energy bought during the same period. Depending of local regulations, an energy retailer may not reward consumers for the amount of energy re-injected beyond the one consumed during a certain period. In this case, even control

policy is correct, the resulting operational cost is not correct. We deal with this fact later in Section 3.5.

- No substantial differences are observed when running control with a smaller slot time (T_s = 15 min).

Heating mode, because of its simplicity, have been only run with sell/buy energy control (see Section 3.5).

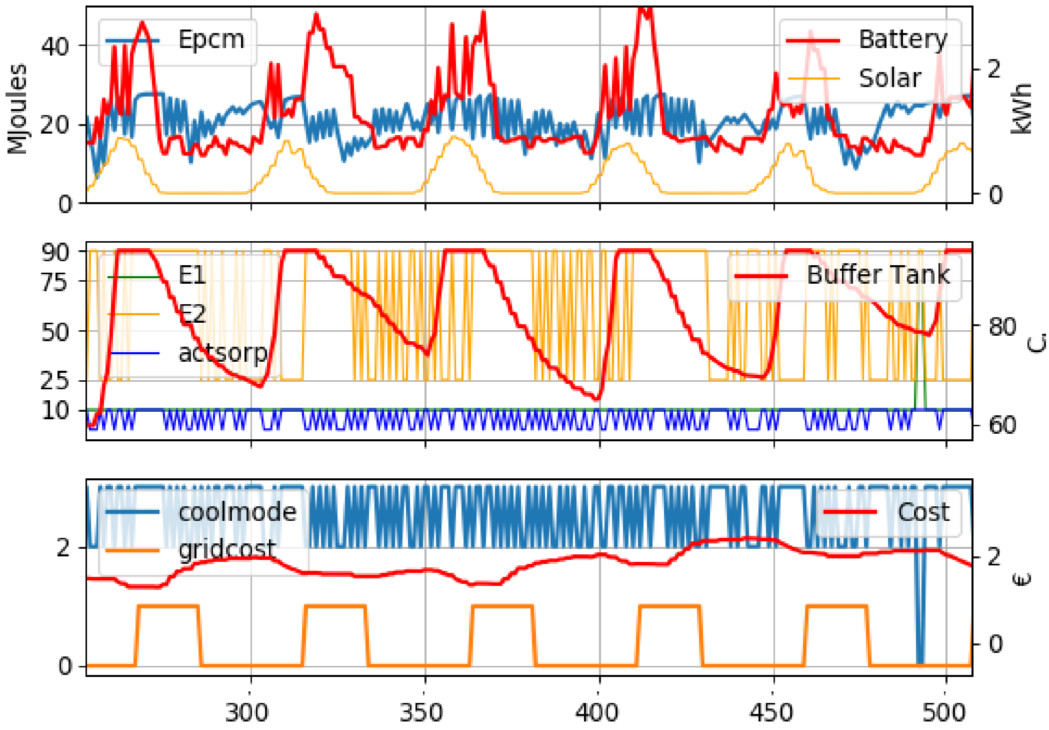

Figure 8. Control performance in cooling mode. Zoom view

3.4.1 Rule Based Control policies

Trying to evaluate the performance of the deep learning control policy, we implement three basic rule-based control policies for cooling mode. Each RBC policy is based on its own thresholds and can be described, without detail, as follows:

1. In all the RBC policies, battery mode is determined by a battery state of charge threshold. If grid cost is high, charging and buffer modes are used. Otherwise, buffer and discharging are employed. This last rule may seem counter-intuitive, but give best results than its reverse (charging/buffer with grid cost low and discharging/buffer with grid cost low). It is explained by the fact that high cost grid uses to coincide with system energy requirements, setting DC-bus to buffer, while during low cost grid less energy is required and DC-bus is set in discharging.
2. In all the RBC policies, cooling mode 1 is set if no demand exists. Otherwise, the three policies differ.
3. RBC1. Cooling mode 2 is set for a PCM state of charge threshold. Otherwise, cooling mode 3 is set. Sorption is set depending on a buffer tank temperature threshold.
4. RBC2. As RBC1 but a hysteresis control is added to the PCM threshold.
5. RBC3. Cooling mode 2 is set if PCM state of charge is above a threshold of the demand. Otherwise, cooling mode 3 or 4 are set according to a threshold of the PCM state.

For each of the RBC policies a hyper-parameter optimization is applied in order to determine the optimal thresholds. Hyperopt python library (Bergstra, Yamins, & Cox, 2013) is used employing an adaptive Tree Parzen Estimator algorithm with 400 runs over the same training test set. Table 1 (top row) shows results for Deep Learning Control (DLC) and RBC policies over the same test set. Clearly, DLC outperforms any of the tested RBC policy.

Table 1. Operational cost (€) for different control policies

	Sell/Buy Control	Policy			
		DLC	RBC1	RBC2	RBC3
Cooling mode		3.4	6.7	7.6	11.1
Cooling mode	✓	6.2	9.6	9.9	10.3
Heating mode	✓	0.0	1.6	-	-

3.5 Sell/Buy energy control

As mentioned before, energy retailers may not reward consumers for the amount of energy re-injected that surpasses the consumed energy during a certain period. To simulate this scenario, we modify our control model as follows:

- Sold energy is not rewarded if, during a reward period, it exceeds the bought energy in the corresponding tariff.
- Two inputs are added to the control network in the form Sign($BE_p - SE_p$), being BE_p and SE_p the bought and sold energy during a reward period for tariff *p* (0 or 1), and being

$$\mathrm{Sign}(x) = \begin{cases} 1 & \text{if } x \geq 0, \\ -1 & \text{otherwise.} \end{cases}$$

Figure 9 shows the control performance in this case. As expected, the performance is worse (6.2 €), but the energy sold/bought for both tariffs, now, matches the required constraint, being well balanced (bottom plot). Looking at the behaviour of threshold *E*2 (mid plot), one can observe that charge/discharge modes for DC-bus are more frequent than if Figure 7, where control relies more often in buffer mode. This behaviour allows to partially compensate performance losses by increasing the amount of energy sold in tariff 1.

Run RBC policies with the same sold/bought constraints for the reward function, gives a worse performance than our deep learning policy (Table 1).

Finally, we train heating mode with T_s = 30 min and Δt = 15 sec as shown in Figure 10. As opposite in cooling mode, control decisions are simpler. There is a unique mode for heat pump, being activated when demand exists and reward is optimized by taking correct decisions in DC-bus subsystem. We observe DC-bus discharging battery at peak tariffs periods by adjusting the *E*2 threshold, putting DC-bus in discharging mode. Actually, the system operation cost for test data is almost 0 €.

It is worth to mention that in this case, Δt is reduced to 15 sec because of the heat pump coefficient of performance, which is higher than in cooling mode, and taking Δt = 1 min would exceed in many cases the required energy from demand.

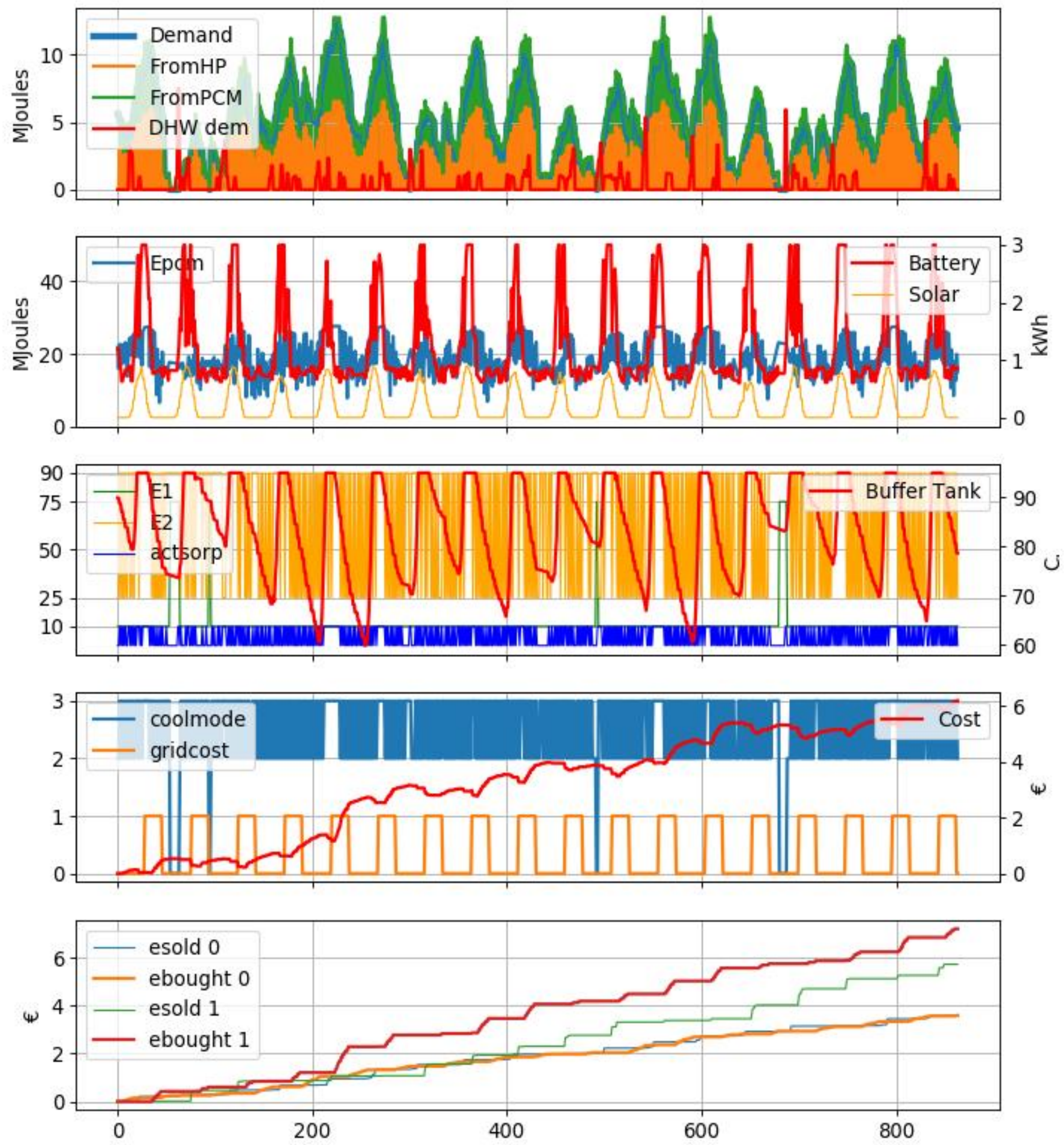

Figure 9. Control performance with control for sold/bought energy in cooling mode

As for the cooling mode, we implement a RBC policy, in this case, based on DC-bus battery state, that after being optimized their thresholds, gives a performance not too far from DLC (1.1 €) (Table 1 bottom row). This result is not surprising considering that for heating mode Mediterranean system control decisions are easier.

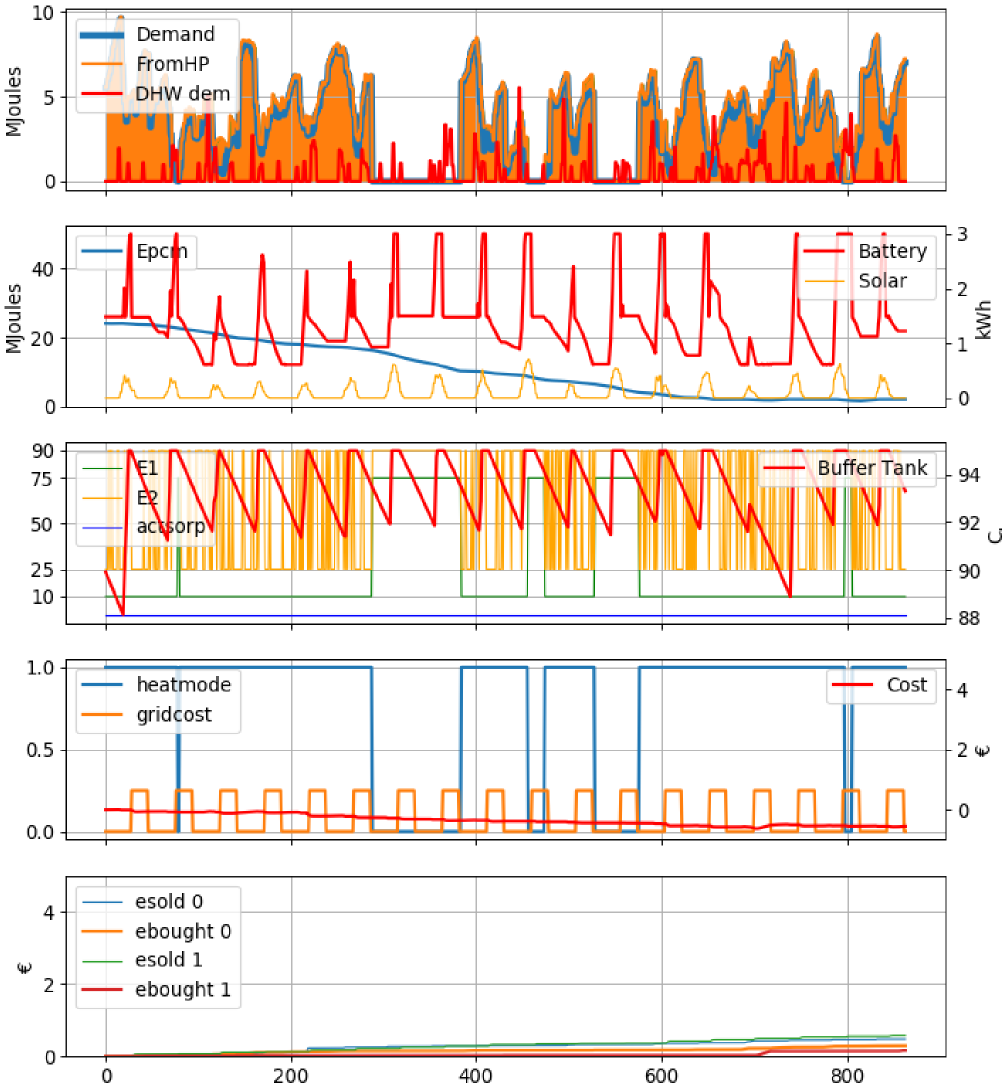

Figure 10. Control performance for T_s = 30 min and Δt = 1 min with control for sold/bought energy in heating mode

4 Conclusions

This document reports the description of the two BEMS optimisation processes implemented in the context of the thermal management of the HYBUILD buildings. These two software components allow to cover most of the energy operations performed inside the building for providing heating, cooling and DHW to ensure the comfort conditions required by the user while optimising the overall energy performances. ENG and UDL have been involved in the provision of these software tools.

The optimisation process provided by ENG focuses on optimising the internal energy resources of the building, leveraging in particular on the storage systems, in order to provide flexibility services to entities outside the building itself, such as electric and district heating grid operators. The provision of this kind of services led to an economic reward that drives the multi-objective optimisation framework proposed by ENG. This BEMS optimiser indeed pursues different objectives along with the flexibility exploitation: it takes into account also the building inhabitants' comfort and the economic management of the energy operation. This is performed by the implementation of a heuristic algorithm, the NSGA-II.

UDL's BEMS optimisation process deals with the high-level control strategy of the building aiming at minimising the expenditures afforded for providing the best comfort conditions to building users and avoid penalties. This takes into account all the appointed cost items related to the energy operation performed by the systems and devices installed inside the Mediterranean building. The optimisation process relies upon a reinforcement learning technique, in particular a DLC algorithm implementing a three-layer fully connected network.

Both the BEMS optimisation processes have been designed on the basis of simplified models of the systems under control (addressing building and its systems and devices energy behaviours) and taking into account the operational modes defined for modelling the overall energy operations of the building in a simple and atomic information. These optimisation processes could be enhanced once that more detailed models could be easily integrated on their procedures without affecting their computational performances.

These pieces of software are the core of the BEMS solution provided for the HYBUILD buildings. Their performances and the comparison on their result will be reported in the deliverable D4.4 – "Report on system performance".

5 References

Barchi, G., Dipasquale, C., Robello, P., Macciò, C., Gasia, J., Zsembinszki, G., . . . Charalambous, C. (2018). *HYBUILD D1.3 - Requirements: Key performance Indicators, system components and performance targets.*

Bergstra, J., Yamins, D., & Cox, D. (2013). Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures. In S. Dasgupta, & D. McAllester (Ed.), *Proceedings of the 30th International Conference on Machine Learning*, *28*, pp. 115-123. Atlanta, Georgia, USA.

Deb, K., Agarwal, S., Pratap, A., & Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation. 6*, pp. 182-197. IEEE. doi:10.1109/4235.996017

Django Software Foundation. (n.d.). *The Web framework for perfectionists with deadlines | Django.* Retrieved from https://www.djangoproject.com/

EURAC. (2019)a. DHW profile SFH. *20191017_MED_SFH_Reference_building_DHW_Qth_profile.xlsx*.

EURAC. (2019)b. Heating profile SFH. *20191017_simulation_summary_SFH0_3_only_HYBUILD_MED_climate.xlsx*.

Gleixner, A., Bastubbe, M., Eifler, L., Gally, T., Gamrath, G., Gottwald, R., . . . Pfetsch, M. (2018). *The SCIP Optimization Suite 6.0. ZIB-Report 18-26.* Zuse Institute Berlin, Berlin.

Koch, N. (2019). *HYBUILD - DC microgrid for continental demo site. User Manual.* CSEM.

Paternò, G., Rossi, A., Sarno, C., Macciò, C., Dipasquale, C., & Trentin, F. (2019). *HYBUILD D4.2 - Functional Requirements Specification.*

Python Software Foundation. (n.d.). *Our Documentation | Python.org.* Retrieved from https://www.python.org/doc/

Rawlings, J., & Mayne, D. (2009). *Model Predictive Control: Theory and Design.* Nob Hill Pub, L., Ed.; 1st ed.

Rossi, A., Paternò, G., Dipasquale, L., Barchi, G., Trentin, F., Palomba, V., . . . Emhofer, J. (2018). *HYBUILD D4.1 - Smart system algorithms.*

Sahinidis, N. V. (2019). Mixed-integer nonlinear programming. *Optim. Eng., 20*(2), 301–306.

Tensorflow. an end-to-end open source machine learning platform. (n.d.). Retrieved from https://www.tensorflow.org/

van Rossum, G. (1995). *Python tutorial. Technical Report CS-R9526.* Centrum voor Wiskunde en Informatica (CWI), Amsterdam.

Weather Data by Location. All Regions - Europe WMO Region 6 - Greece. (n.d.). Retrieved from https://energyplus.net/weather-location/europe_wmo_region_6/GRC//GRC_Athens.167160_IWEC

Weatherbit. (2020, January). *Weatherbit | Hourly Forecast API Documentation*. Retrieved from https://www.weatherbit.io/api/weather-forecast-120-hour

Zsembinszki, G. (2019). *Description of the simplified models for the optimisation of the Mediterranean system. HYBUILD project.*

6 Appendixes

6.1 Data model and monitoring framework

6.1.1 Data model

The Entity Relationship Diagram of data model adopted for the BEMS optimisation of the ENG tool is presented here. For the sake of clarity, it is split in four figures (Figure 11 to Figure 14). The same figure can be asked to the authors of the deliverable or the Coordinator of the Project in original format.

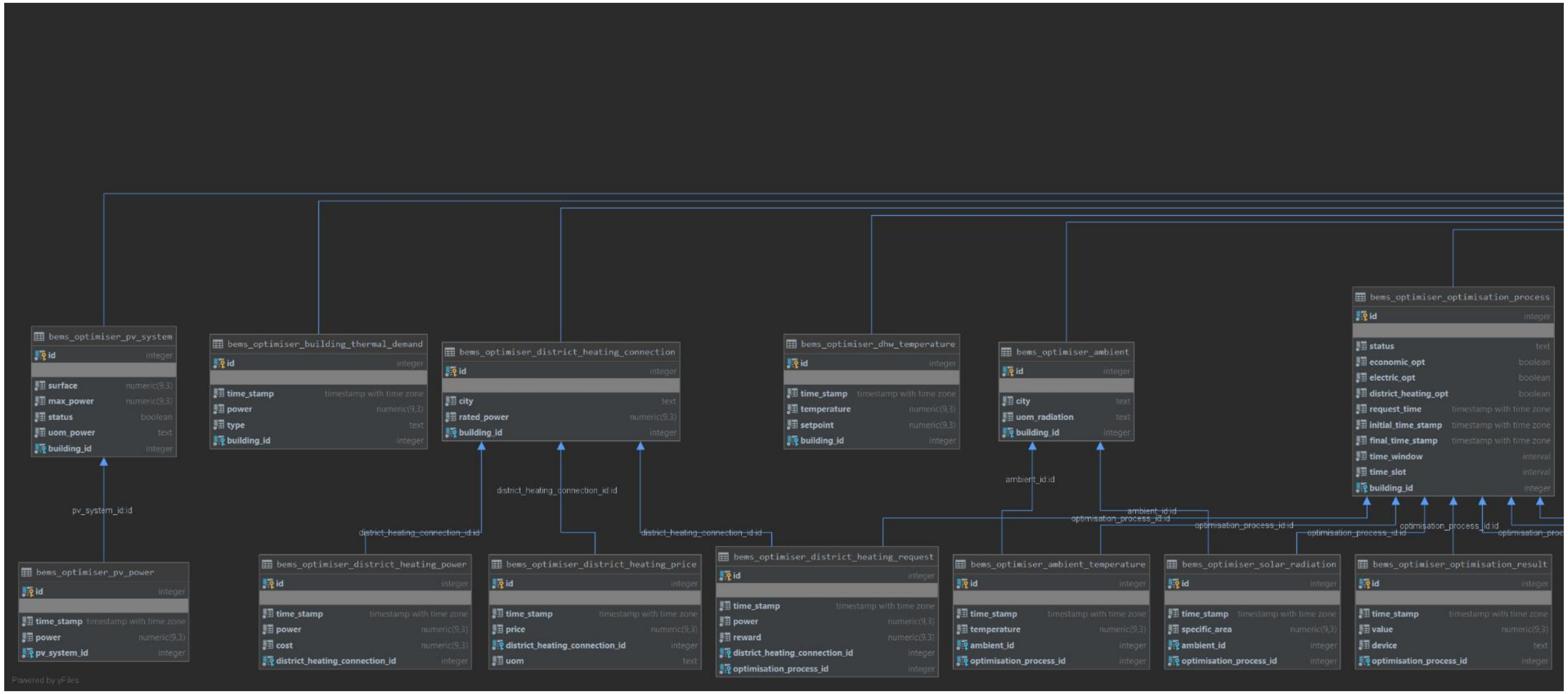

Figure 11. Entity Relationship Diagram of data model adopted for the ENG BEMS optimiser (1/4)

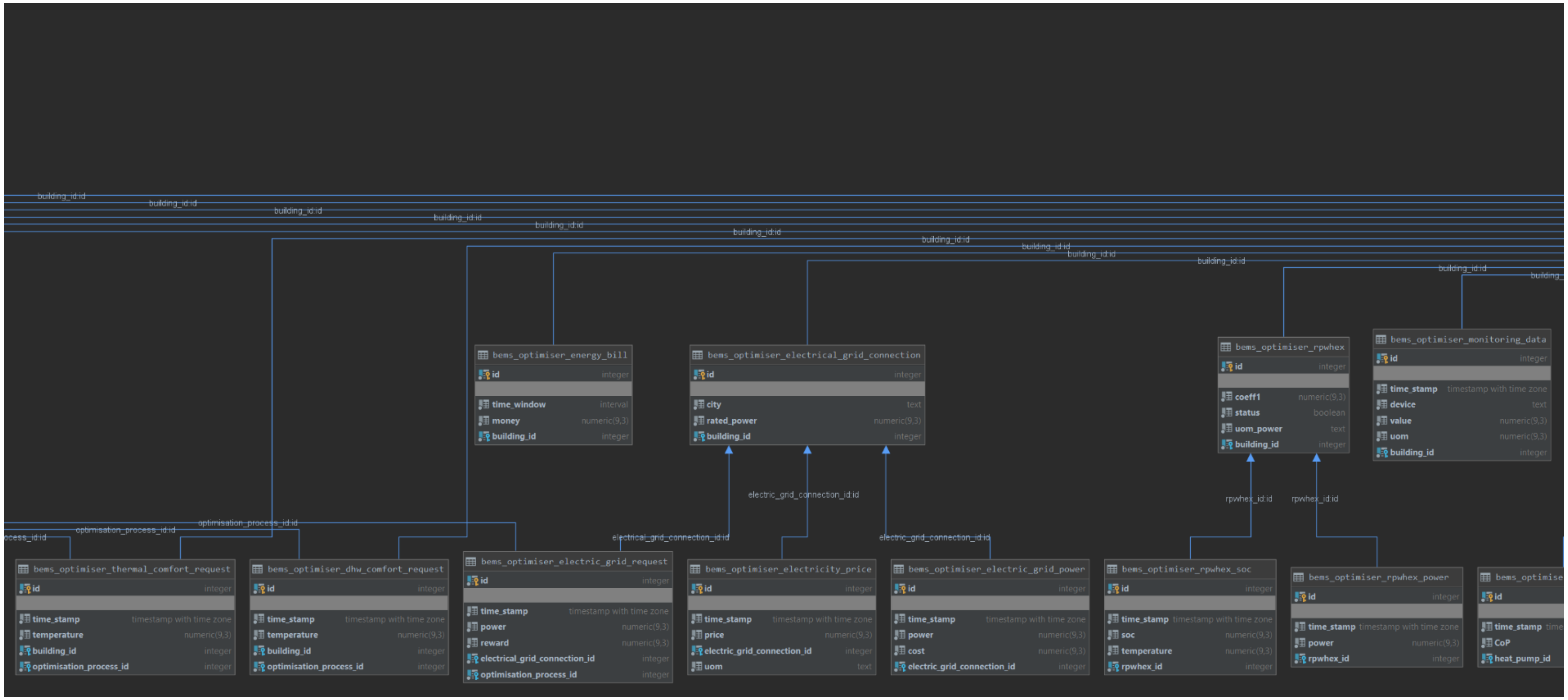

Figure 12. Entity Relationship Diagram of data model adopted for the ENG BEMS optimiser (2/4)

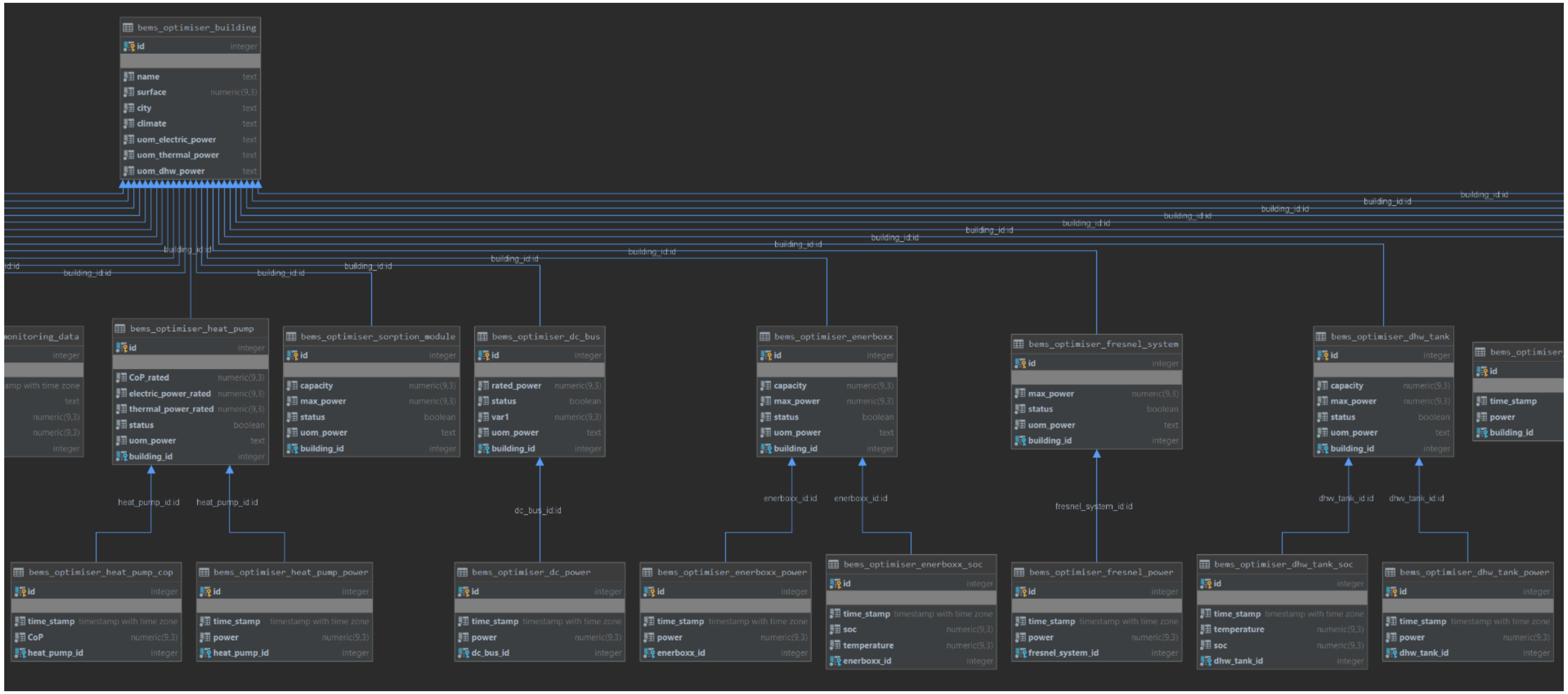

Figure 13. Entity Relationship Diagram of data model adopted for the ENG BEMS optimiser (3/4)

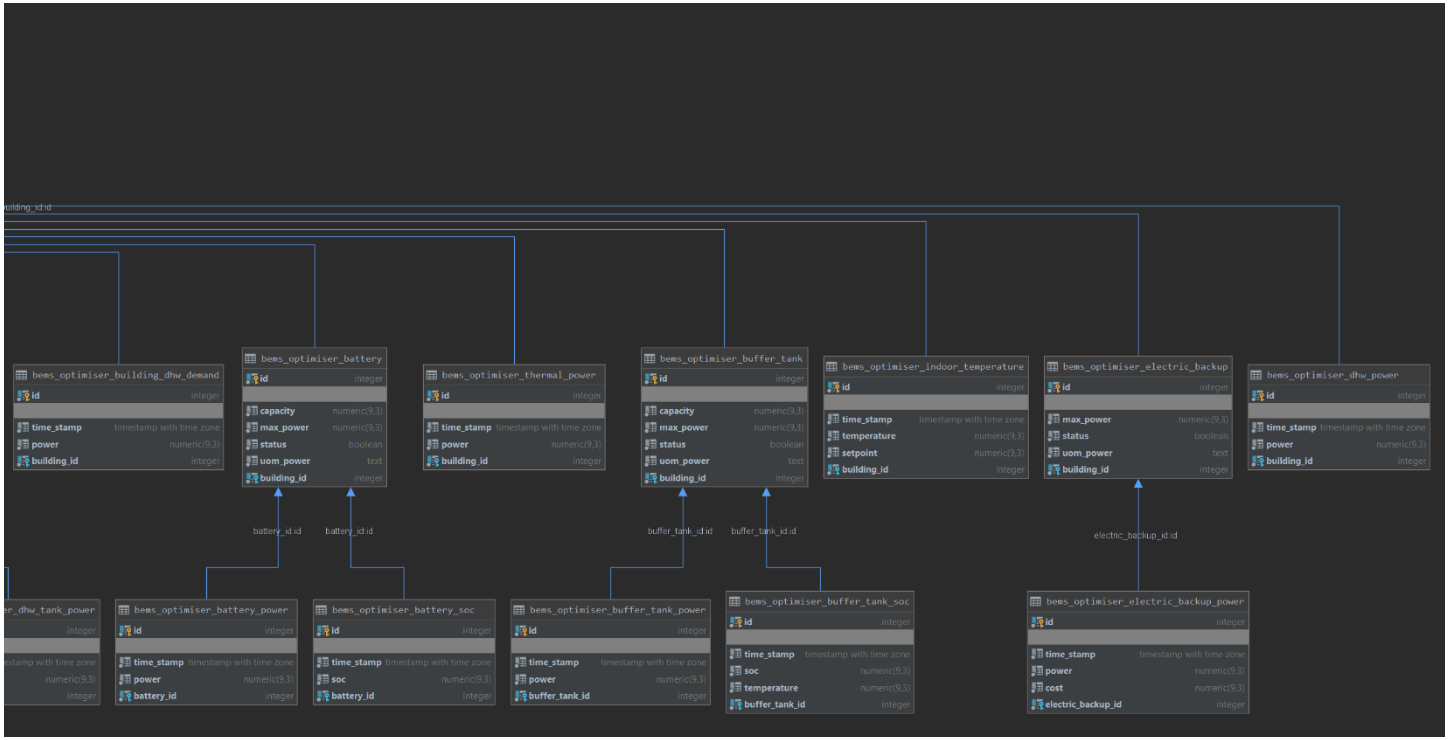

Figure 14. Entity Relationship Diagram of data model adopted for the ENG BEMS optimiser (4/4)