POLITECNICO DI TORINO

POLITECNICO DI TORINO

Corso di Laurea in Ingegneria Energetica e Nucleare
Indirizzo Progettazione Termotecnica ed Uso Razionale dell'Energia

TESI DI LAUREA MAGISTRALE

Adaptive Control Strategies for enhancing energy efficiency and comfort in buildings

Relatore
Prof. Alfonso Capozzoli
Correlatori
Ing. Silvio Brandi
Ing. Giuseppe Pinto

Candidato
Davide Coraci

Anno Accademico 2019-2020

To my family,

although it will never be enough to repay

the immense sacrifices made to get me here.

Abstract

In the field of buildings energy management, the concept of energy flexibility has become increasingly popular. It could be defined as the ability to adapt energy management to several dynamic factors, such as changing external conditions or internal comfort conditions.

The control systems have increased their importance as they have to be able to predict the continuous adjustments of dynamic factors, allowing the adaptability in the building energy management. However, this task cannot be performed by traditional control systems such as ON/OFF or PID, as they do not have any prediction capabilities.

Thus, there is a necessity to explore control strategies based on artificial intelligence, such as model-based predictive or model-free adaptive ones.

The first type has shown excellent results when applied in a real context, such as Model Predictive Control (MPC). However, it is difficult to apply this control strategy because it requires the definition of a model for its optimization, which is difficult to obtain. For this reason, interest in adaptive model-free control strategies has recently grown, especially among those based on machine learning, such as Deep Reinforcement Learning (DRL).

DRL has become increasingly popular because it can control systems in which the dynamic process is very complex, as is mainly the case with HVAC systems, because represents a learning technique that does not require an optimisation model, but a simple trial-and-error interaction with the environment to be controlled, following an action-reward process.

The current state of the art has very few, if any, real applications, but a lot of studies on the subject that are thought to become applicable in the shortest possible time.

In this thesis work, it is implemented a DRL control method for a radiant heating system, installed on a real building for office use.

In the initial phase, it was necessary to calibrate the energy model, based on the real and available temperature profiles, made available by the real Energy Management System. Through a trial-and-error approach, a calibrated model was obtained, as the metrics provided by ASHRAE were respected.

The calibrated energy model was used for the implementation of a DRL control agent, using a Soft Actor-critic (SAC) algorithm, in order to evaluate possible energy savings but above all the presence of the desired occupants comfort conditions during occupancy, compared to the baseline already present, based on the climate curve.

In the initial training phase, a sensitivity analysis was carried out on the hyperparameters in order to choose the best configuration. The best agent's configuration

allows to obtain energy savings of 5 % and at the same time to improve the internal comfort of the occupants, evaluated through the reduction of a sum of temperature violations compared to a fixed comfort range.

The agent was then used for a static deployment phase on the current radiant system.

Evaluating five different deployment scenarios, the excellent flexibility of DRL control logic concerning changes in the initial boundary conditions was proved. As a result, the comfort degree for each scenario has improved significantly, and in some cases even managed to make temperature violations very few, opposite to the huge baseline values. At the same time, the energy savings obtained varied between 7 % and 9 %.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Buildings are the place where people spend most of their time and consequently consume large amounts of energy: this depends on many factors, such as structural conditions, indoor requirements or outdoor conditions.

The energy is consumed mainly to meet the occupants comfort requirements and is therefore linked to the HVAC systems present within the edifice.

The new construction of residential and commercial facilities has meant that the energy consumption of buildings is 40% of the total worldwide with CO2 emissions up to 36% [Yang et al., 2015].

Nowadays there is a greater awareness of the need to reduce these values, especially within the European Union where, through the use of different programs (such as the *"20-20-20"*), it has tried to progress towards a horizon of greater energy efficiency in all areas, especially in the building sector. In this context, interest in energy systems that use renewable sources of energy has increased. However, this does not mean that only renewable energies should be given importance, as it becomes essential to control the same installations already present in existing buildings.

HVAC systems are the most energy consumption responsible, especially in the non-residential sector and, in the past years, significant improvements have been recorded concerning their energy efficiency.

Besides, the energy systems serving the building became increasingly complex, also facing not only simple HVAC systems, but also systems consisting of RES technologies and storage systems. Then, the control systems implemented had to face continuous changes linked, for example, to grid requirements, occupant preferences and external forcing variables. Therefore, more attention should be paid to adaptive control systems, as classical controllers would be inadequate compared to the dynamism to which new energy systems are subject, working within established limits.

Nowadays, researchers are focused on new control strategies for HVAC systems

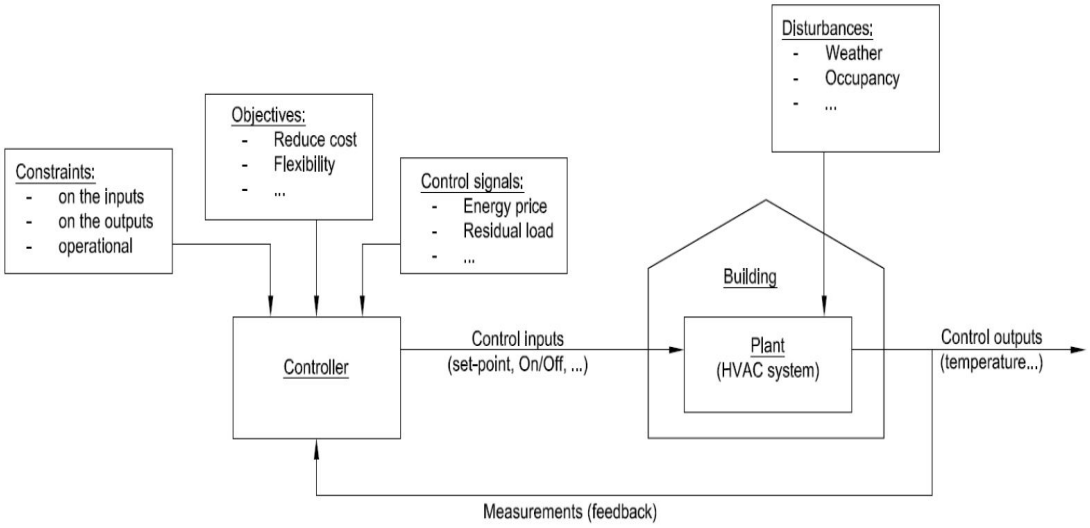

Figure 1.1. General scheme of a single-level control [Finck et al., 2017]

[Zhang et al., 2019], in order to control them while maintaining the conditions of internal comfort for the occupants and, at the same time, reducing the energy consumption linked with their operation.

One of the main stumbling blocks in the control of HVAC systems is their non-linearity since they depend on several stochastic factors, such as occupant behaviour, their interaction with the building system and external weather conditions.

In this context, the concept of building ***flexibility*** has become necessary: it represents a fundamental characteristic of the building that defines the margin within which it operates by its functional requirements [Claub et al., 2017]. Flexibility can also be seen as the ability to manage the building according to grid requirements, climatic conditions and user needs [Finck et al., 2017].

Traditional control systems are the simple **On/Off** or the more "complex" **Proportional-Integrative-Derivative Control (PID)**. These two belong to the sphere of classic rule-based controls, i.e. dependent on certain limits (such as set-point temperature ranges) within which the quantities to be controlled must remain.

The On/Off control is effortless to implement, while the PID requires adjustment of the control variables for *subsequent disturbances*. Although the latter has a better performance than the former, it is still inadequate because the performance of the PID goes down a lot when the operating conditions are different from the tuning conditions in which the constants that regulate the control are calibrated [Afram & Janabi-Sharifi, 2014]. Moreover, this process requires much computational time.

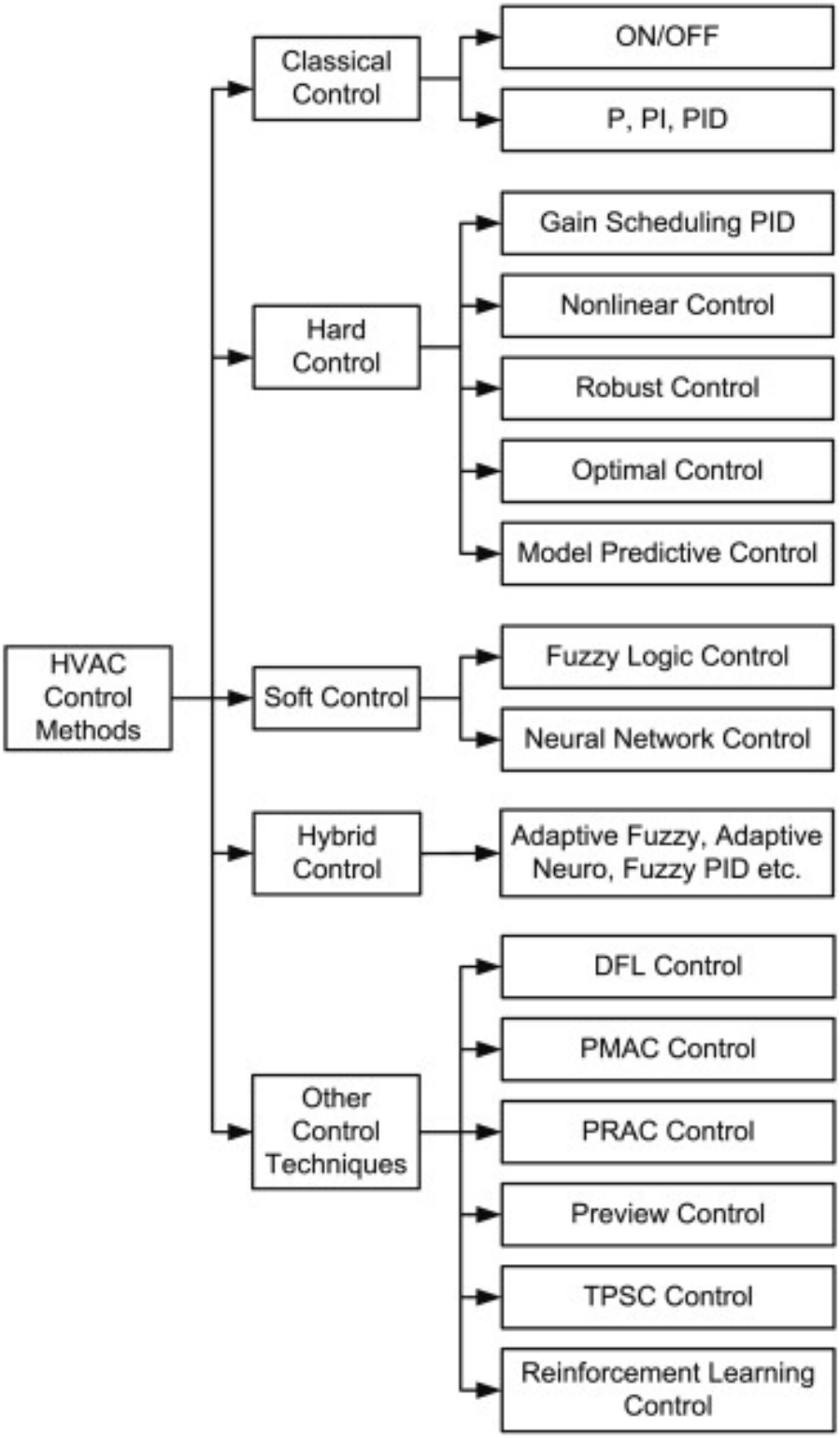

Figure 1.2. Overview of control methods for HVAC systems [Afram & Janabi-Sharifi, 2014]

Rule-based control strategies are not able to predict the dynamic variations to which HVAC systems are subjected. Then, in recent years, the application of **model-based** control strategies was explored in real contexts, such as **Model**

Predicted Control (MPC).

Their application is interesting because they can predict the future states of a system and identify the optimal control variables in order to minimize a cost function over a prediction time horizon [Yun et al., 2012].

The simulation models used in the case of MPC can be divided into:

- white box model, based on physical principles and first-principle modelling;
- black-box model, requires large datasets to train the model because entirely disregards empirical models and physics.
- grey-box model, a mix between the previsious models. In this case a set of continuous differential equations in time are derived and parameters are estimated through identification techniques.

The first and the third have an accuracy strongly dependent on the uncertainties of the simulation parameters. At the same time, the reliability of the black-box models depends on the quantity, quality and relevance of the available data [Uhn Ahn & Soo Park, 2019].

A significant problem in MPC is related to the performance dependency on the simulation model quality, and the availability of data since the availability of a basic model dictates its existence. Moreover, it is challenging to generalize its use on different types of plants and buildings, leading to define an accurate model for each of them.

Therefore, research has led to the emergence of much more complicated but at the same time more efficient methods, but above all not dependent on a model: these are **model-free control**, such as **Reinforcement Learning (RL)**.

Reinforcement Learning is a learning technique belonging to the **Machine Learning** sphere, together with *supervised* and *unsupervised control*. In recent years there has been a growing focus on this control methodology because it does not require prior knowledge of the process or environment to be controlled.

An agent-based Reinforcement Learning algorithms learns an **optimal control policy** by direct interaction with the environment, obtaining a reward based on the control action performed from a specific state.

Deep reinforcement learning (DRL) is a branch of RL that deals with continuous states and actions by using deep neural networks to approximate the policy function [Zou et al., 2020]. It is used above all in the case in which the number of states and actions available is vast and therefore, the choice of the Table Q-Learning algorithm is not optimal.

In the next section, previous works based on the use of the RL and its variants are discussed.

1.1 Previous Work of RL application in Building control system

Reinforcement Learning had increased interest in it in recent years, even if its applications remain limited, considering a control algorithm that has very different characteristics compared to traditional control systems and more frequent in state of the art. This aspect is even more amplified in the case of Deep Reinforcement Learning, a more futuristic but also a more efficient branch of these control algorithms.

The first RL application dated back to 1998, in which [Naidu & Rieger, 2011] used it for an HVAC system serving the DOE (Department of Energy) in Idaho State University, to control the water leaving boiler temperature of and the temperature in the two thermal zones served by the system. In order to understand the possible applications of RL to the world of HVAC control systems, detailed bibliographic research has been carried out, after which it was decided to create the table to make a general summary.

Among the previous work reported in the table, some deserve attention for the work to be done. For example [Zhang et al., 2019] applied an RL control type *Asynchronous Advantage Actor-Critic (A3C)* in a water-based Radiant Heating System, in which the hot water pipes are integrated into window mullions. The goal is to reduce the energy consumption of the system while respecting the internal comfort of the occupants. The control system, in this case, operated on the Mullion system supply water temperature set-point.

The same objective was achieved by [Zou et al., 2020] by applying a Deep Q-Network with Long-Short-Term-Memory (LSTM) on three AHUs system and controlling fan speed, heating valve status and damper position. Another work that focused on the same objectives is proposed by [Yoon & Moon, 2019]. A performance-based thermal comfort control using Double Deep Q-Network allowed to reduce by 32.2% and 12.4% the energy consumption associated with the VRF and humidifier system, keeping the PMV value within the comfort range. In this case, the humidifier status (On/Off), the temperature set-point and the VRF airflow rate were checked.

In [Park & Nagy, 2020] it is presented a Reinforcement Learning (RL) based Occupant-Centric Controller (OCC) for thermostats, called HVACLearn. The agent learns the unique occupant behaviour and indoor environments and monitoring indoor air temperature, occupancy, and thermal vote. It calculates adaptive thermostat set-points balancing between occupancy comfort and energy efficiency. Authors simulated HVACLearn control in a single occupant office with occupant behaviour models. Compared to a reference controller, HVACLearn reduced the

number of button presses (too hot) significantly, while consuming the same or less cooling energy.

[Yu et al., 2020] presents a Multi-Agent DRL (MADRL) called Multi-actor attention critic (MAAC), in order to minimize HVAC energy cost in a multi-zone commercial building under dynamic prices, with the consideration of random zone occupancy, thermal comfort and indoor air quality comfort in the absence of building thermal dynamics models. To be specific, air supply rate in each zone and the damper position in the air handling unit are jointly determined to minimize the long-term HVAC energy cost while maintaining comfortable temperature and CO2 concentration ranges. For encouraging exploration, Soft actor-critic (SAC) method is used. The simulation results showed the effectiveness, robustness, and scalability of the proposed algorithm.

[Nagarathinam et al., 2020] consider the optimal control problem of minimizing the building HVAC energy subject to meeting the comfort constraints by dynamically setting both the building and chiller set-points. In this frame, it is presented MARCO (Multi-Agent Reinforcement learning COntrol) for HVAC system. MARCO is based on Double Deep Q-Network algorithm and uses separate DRL agents that control both the AHUs and chillers to jointly optimize HVAC operations. Authors train and deploy the agent in real configurations and it is showed that that MARCO learned the optimal policy in a two-agent setting with single-AHU and single-chiller. MARCO not only improved comfort but also reduced the energy by 17% over a baseline that used seasonal variations in set-points.

In [Ding et al., 2020] a double deep Q-Learning named OCTOPUS, employing a novel deep reinforcement learning (DRL) framework that uses a data-driven approach to find the optimal control sequences of all building's subsystems, is used to minimize the energy used in heating/cooling coils, the electricity used in the water pumps and flow fans in the HVAC system, electricity used by the lights, and the electricity used by the motors to adjust the blinds and windows. In addition to the minimization of energy, it is requested maintaining the human comfort metrics within a particular range. Through extensive simulations it is demonstrated that OCTOPUS can achieve 14.26% and 8.1% energy savings compared with the state-of-the art rule-based method, while maintaining human comfort within a desired range.

[Qiu et al., 2020] applied the Tabular Q-Learning to improve the global COP of the cooling water systems of the HVAC system serving a subway station in Guangzhou. This work compared the results obtained with those of three other control systems (baseline controller, local feedback controller and model-based controller). In this case, it was seen as Model-free control could conserve 11% of the system energy, which is more than 7% in local feedback controller but less than

14% of model-based.

An exciting application is the one proposed by [Brandi et al., 2020], in which an algorithm based on Deep Q-Network has been used to control the supply water temperature of the boiler serving the radiant heating system installed in an office building. In this context, a static and dynamic deployment of the DRL controller is performed, and a heating energy saving ranging between 5 and 12% is obtained with enhanced indoor temperature control with both deployment.

This application is beneficial because it highlights the importance of an initial application of this control system to a simulated environment: the direct real-implementation would cause the control performance to be deficient since a DRL agent takes a long time to converge towards an acceptable control policy. Therefore, in most cases, an initial simulation phase is performed in which various tools are combined, such as EnergyPlus with deep learning libraries such as Tensorflow.

The bibliographic research has therefore shown that Reinforcement Learning, and specifically its *deep* version (DRL), represents an exciting possibility on which to base the future control systems of HVAC systems, being more efficient than current systems. The work already produced also contains a series of possible future improvements, representing a cue to improve even more reliability and performance.

1.2 Possible Contributions from this work

In this master thesis, it was decided to apply a control algorithm based on DRL to a building for office use, located in Turin in via Bazzi 4. This building is served by an HVAC system that is reduced to a heating system, consisting of one four gas-fired boiler and radiator terminals. The ventilation technique present is only natural.

The plant is currently controlled by a control system implemented by Enerbrain Srl, a company from which it was also possible to obtain some data on internal temperature and consumption: these were of fundamental importance in the calibration process adopted.

The model of the building to be studied was first built on EnergyPlus v9.2.0, and then calibrated through the available data available (as discussed in the next chapters) based on a trial-and-error approach.

Only after calibrating the model, it is implemented the new control system, joining the use of EnergyPlus and Python. This two software only linked together employing a *Building Control Virtual Test Bed (BCVTB)* and the *ExternalInterface* function of EnergyPlus.

The choice of this building is also linked to a reproducibility factor. In fact,

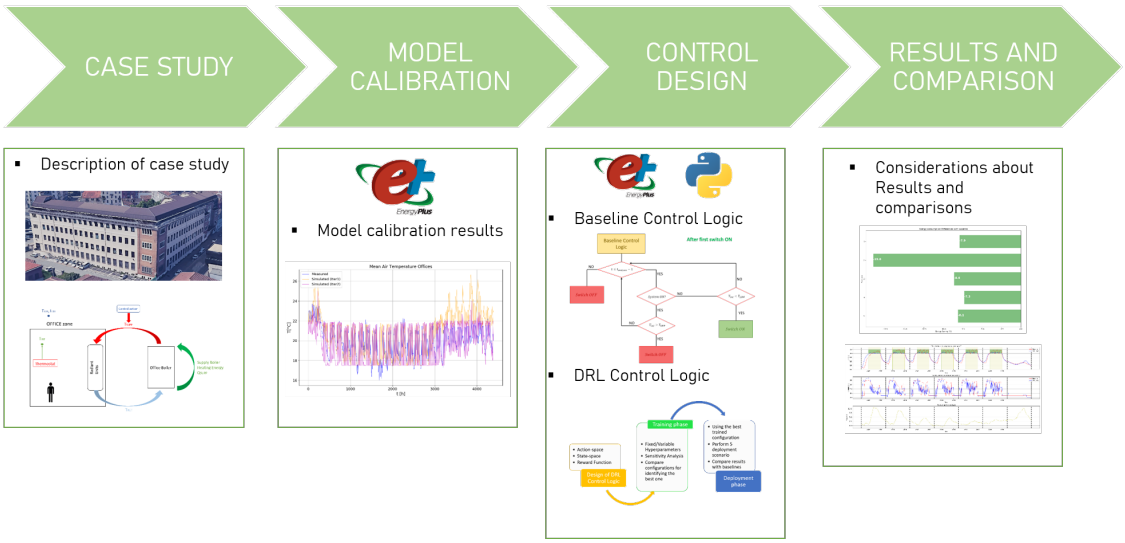

Figure 1.3. Thesis workflow: from energy model calibration to DRL control Logic implementation

in the context of Italian construction, the system solution with boilers and radiators is widespread, so this work could become a reference point for improving the energy efficiency of the buildings served by these systems, respecting the internal conditions of the comfort of the occupants. Since this is a radiator system, more specific importance will be given to internal temperature conditions.

This thesis work refers to the same building and plant of [Brandi et al., 2020], unlike which, however, Deep Q-Network will not be used, but a newly launched algorithm called Soft Actor-Critic (SAC).

The objective will be to maintain satisfactory temperature conditions inside the building during the occupation phase, looking for possible energy savings resulting from the use of a DRL control logic, through the choice of supply water temperature to heating terminal units system.

Initially, during a training phase, different configurations were tested for our agent and then choose the best one.

The training agent it is used secondarily in a deployment phase in which its adaptability to changing boundary conditions will be tested. Firstly, only the change in outdoor weather conditions is considered, considering a simulation period of two months. Subsequently, changes in indoor comfort conditions, occupancy schedules and structural conditions were considered.

The main innovative contributions that this work providing are many, both for calibration of the basic model both for DRL control logic.

- After analyzing the different procedures present in the literature on Model Calibration, the trial-and-error procedure used to vary the most influential

parameters on the temperature profile is described.

- This work would represent an opportunity to evaluate the agent's behaviour when it has to use an algorithm with the possibility of choosing an action within a continuous space, so important when it is controlled an energy system.

- The control agent will be set to guarantee an indoor temperature within a certain temperature band (between 20°C and 22°, which for us will be defined as comfort conditions, through the choice of the supply water temperature. It will be evaluated with respect to the current baseline, which provides a logic based on the climate curve. At the same time, the possibility of obtaining energy savings will be assessed.

- As in the case of Model Calibration, a phase of sensitivity analysis of the hyper-parameters that influence the controller performance will be addressed.

- The deployment phase could be useful in evaluating the adaptability of the agent concerning the change of certain conditions in the energy model (schedules, building physics) and the performance that it could assure.

The rest of this work is organized as follows. In chapter 2, it will be discussed theoretical aspects of the topics and tools covered, namely Model Calibration and Reinforcement Learning.

In chapter 3 it will be described the building, the system and the actual control logic, and then move on to the description of the calibration procedure. After this, the characteristics of the DRL control agent that will be implemented are indicated, and it is described a lot about training phase and the deployment one.

Finally, chapter 4 discusses the results obtained and will conclude by talking about possible future contributions to this work in the conclusions of chapter 5.

Table 1.1. Summary of the paper reviewed.

Reference	Energy system	Learning algorithm	Action selection	Control Objective(s)
[Zhang et al., 2019]	Water based Radiant Heating System	A3C	N/A	Energy consumption, comfort
[Zou et al., 2020]	HVAC System with 3 AHUs	Q-Learning	ϵ-greedy	Energy consumption, comfort
[Yoon & Moon, 2019]	HVAC system (VRF System and a humidifier)	Q-Learning	N/A	Energy consumption, comfort
[Park & Nagy, 2020]	HVAC system	Q-Learning	ϵ-greedy	Calculating thermostat set-points for reducing Energy consumption and optimize comfort
[Brandi et al., 2020]	Heating radiant system with radiators	Q-Learning	Boltzmann action-selection	Energy consumption, comfort
[Nagarathinam et al., 2020]	HVAC system with 12 AHUs and 3 Chillers	Q-Learning	ϵ-greedy	Energy consumption, comfort
[Ding et al., 2020]	HVAC, windows, blind and lighting system	Q-Learning	N/A	Energy consumption, comfort
[Qiu et al., 2020]	Cooling water systems served HVAC, composed of chillers, cooling towers and cooling pumps	Q-learning	ϵ-greedy	Improving the system COP
[Yu et al., 2020]	HVAC system	MAAC	ϵ-greedy	Minimizing HVAC energy cost, optimize thermal and indoor air quality comfort

Chapter 2

Calibration Process and Reinforcement Learning Methodology

In this chapter, the theoretical aspects underlying the calibration of the energy model and the RL control system to be implemented in this case study are discussed.

2.1 Fundamentals of Calibration Process

The energy calibration of buildings is a necessary step because buildings do not return the same performance between real and simulated condition. In the field of buildings the concept of simulated calibration (CS) is used, corresponding to the process of calibration and tuning of the input parameters to the simulation model: so, its results are as close as possible to those found with the measurements. The IPMVP protocol officially recognizes this methodology, but reference can be found in the FEMP too. When talking about CS it is crucial to make many clarifications regarding disturbance effects on it:

- CS is strictly dependent on the available monitored data and their frequency of recording (sub-hours, hours, daily, monthly);

- CS changes if you study a single portion of the building or the whole building itself.

The theory fundamentals were taken mainly from *[Fabrizio et al., 2015]*.

2.1.1 Typical characteristics of the calibration process

The calibration process belongs to the category of undeterminate problems: it leads to a non-unique solution! Usually *trial and error* is used as a calibration method. In essence, this process requires a series of hypotheses based on user experience, especially to avoid high computational times or even worse not to return correct solution. It follows that the impact of the hypotheses made is so relevant on the result. Data collection is essential to start the calibration process, as it is the first level to start from and therefore, the minimum requirement. In essence, data are often obtained directly from meters if possible or through bills, the important thing is that it is an annual series based, at least, on one year. Then, to improve the calibration, it is necessary to increase the number of information perceived: firstly, thorough inspection to verify the data collected and to obtain new data, after it is recommended performing detailed audit and short or long time monitoring. [Coakley et al., 2014] showed what could be the leading CS influencing parameters:

- **Standardization** in CS is a purely subjective process, based on user experience: it is of paramount importance to minimize errors and calibration costs.
- **Calibration costs** are proportional to the difficulty encountered in the process and the level of detail to be achieved.
- **Complexity and accuracy of the model** is influenced by the number of parameters in input and the type of simulation, growing from "steady" to transient models.
- It is crucial to choose the most **influential parameters** for CS because there are several of them: it is necessary to define the level of accuracy for each of them.
- **Finding the causes of the discrepancy** between the real and simulated model is of paramount importance, especially to eliminate them and evaluate the improvements deduced from the model.
- It is preferable to use **automated** rather than manual methods.

2.1.2 Criteria for assessing the goodness of the calibrated model

Statistical indices are used to assess the accuracy of the calibration: they are essential in defining how well the simulated model represents the real one. For these, lower and upper limits are set by ASHRAE, IPMPV and FEMP standards, within

which a model is acceptably calibrated. It is necessary to know the simulated data set of our model and the real one measured to calculate those statistical parameters. The data present are multiple; therefore, it is necessary to select the most important parameters to find a correlation between the simulated and measured energy consumption. MBE (Mean Bias Error) and $C_v(RMSE)$ (Coefficient of Variation of the Root Mean Square Error) are commonly used to measure respectively how much the simulated data are similar to those monitored and to define the goodness of the model, as well as the variability between the two data sets. MBE and $C_v(RMSE)$ are estimated with the following equations:

$$MBE(\%) = 100\% \frac{\Sigma_{period}(S-M)_{interval}}{\Sigma_{period} M_{interval}} \tag{2.1}$$

$$RMSE_{period} = \sqrt{\frac{\Sigma(S-M)^2_{interval}}{N_{interval}}} \tag{2.2}$$

$$A_{period} = \frac{\Sigma_{period} M_{interval}}{N_{interval}} \tag{2.3}$$

$$C_v(RMSE_{period}) = 100 \frac{RMSE_{period}}{A_{period}} \tag{2.4}$$

These equations also include $RMSE_{period}$ and A_{period}. The former avoids offsetting positive and negative terms and is, therefore, a measure of the actual deviation between the two data sets. $C_v(RMSE)$ derives from the combination of these two terms, of which A_{period} is a *"normalizer"* concerning the number of observations in the various time intervals. It is the indicator of the overall uncertainty that results in the forecast of real consumption. It is always a positive value, which the lower it is, the better it is an indicator of good calibration. Depending on calibration time and protocol considered, is is obtained a specific evaluation for considering well the model. In the case of falling within them, there is sufficient evidence that the simulated model represents the real case. These limits, indicated in Table 2.1, are, however, the first guide in the calibration of the model since does not take into account the uncertainties linked to the various parameters impacting.

Table 2.1. Threshold limits of statistical criteria for calibration

Statistical Indices	Monthly Calibration			Hourly Calibration		
	St.14	IPMVP	FEMP	St.14	IPMVP	FEMP
MBE[%]	±5	±20	±5	±10	±5	±10
$C_v(RMSE)$[%]	15	-	15	30	20	30

2.1.3 Calibration methodologies for energy simulation of buildings

There are four principal methodologies for calibrating buildings:

- **Manual** calibration based on an iterative approach;
- **Graphical** calibration method;
- Calibration based on **individual tests and procedural analysis**;
- **Automated techniques** based on analytical/mathematical approaches.

It is possible to use these in synergy, for example, the second and fourth to improve model calibration.

Manual calibration is a subjective approach and does not use a systematic procedure but refers to the experience of users and their judgement. It includes a *"trial and error"* approach, based on the iterative process of tuning the model input parameters, altered according to one's knowledge of the building.

Usually, it is coupled the *graphic calibration* technique with the manual one: it consists of time-based graphs of the measured data, simulated data and the comparison between them. There are two possibilities, either comparative 3D graphs (they allow to identify small differences between the two data sets) or calibration and characteristic signature. In the case of calibration, it is possible making a normalized plot of the differences between expected and simulated consumptions according to T_{out}. An horizontal line represents a perfectly calibrated model.

$$Residual = S - M \tag{2.5}$$

$$CalibrationSignature = \frac{-Residual}{M_{maximum}} 100\% \tag{2.6}$$

In the case of the characteristic signature, it is possible to compare data from two different simulations to take it as a baseline for the measured values. It is calculated on a daily average and has different trends depending on climate and system considered.

$$CharacteristicSignature = \frac{Changeinenergyconsumption}{M_{maximum}} 100\% \tag{2.7}$$

$$ERROR_{TOT} = (RMSE^2_{CLG} + RMSE^2_{HTG}) + (MBE^2_{CLG} + MBE^2_{HTG}) \tag{2.8}$$

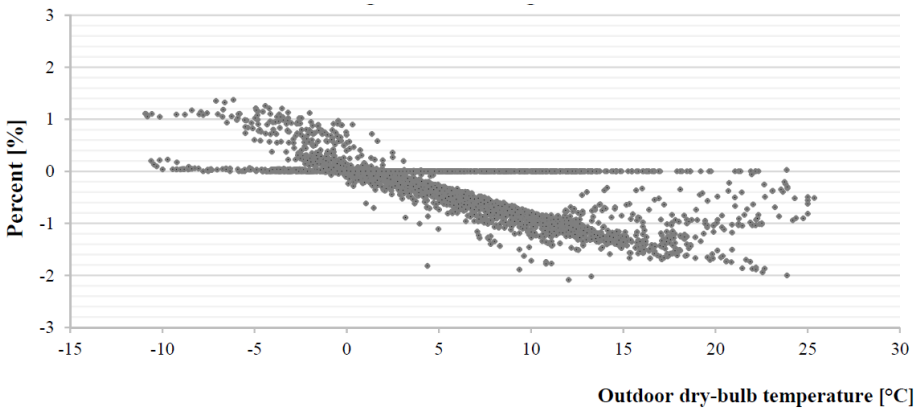

Figure 2.1. Example of heating calibration signature[Fabrizio et al., 2015]

After drawing the two signatures, the difference between them allows you to find possible errors in inputs to simulation model to improve the calibration. It is necessary to reach an acceptable $ERROR_{TOT}$ value, which represents the deviation between the two signatures when one or more input parameters change. The process of calibration ending when the minimum $ERROR_{TOT}$ value is reached. (HTG and CLG refer to heating and cooling time intervals).

Calibration with analytical procedures is based on short/long-term analysis, testing and monitoring, but this is not always possible due to the occupancy presence.

Automated techniques include all approaches that cannot be considered user experience-driven as they are based on mathematical or analytical procedures. These techniques include ***Bayesian Calibration***, ***Meta-Modeling*** and ***Optimization-Based Method***. The last one computes objective functions based on the difference between measured and simulated data. They are considered the most important as they take into account sensitivity and uncertainty analysis, which is generally not incorporated into the model calibration study.

Bayesian Analysis represents a statistical/probabilistic method useful to find, from the data observed in the field, a distribution for an unknown parameter. It also directly incorporates the uncertainties within the calibration process and formulates a set of values for the unknown parameter to match the measured data. Three different sources of uncertainty are studied and reported in the following formulation:

$$y(x) = \eta(x, \theta) + \delta(x) + \epsilon(x) \tag{2.9}$$

It is possible to calculate observed value $y(x)$ through the sum of the simulation result η with known parameter x, unknown parameter θ, observation error δ and discrepancy ϵ.

A ***Meta-Model*** is a mathematical function whose coefficients are determined by limited combinations of input and output parameters. It groups several categories, including polynomial regression (PR) and neural networks (NN). A meta-model is a model of a model, or better, a surrogate model used to reduce the complexity of the source model. The benefit in its use is the reduced computational time.

2.1.4 Model uncertainties

Calibration does not take into account uncertainties, but they are an essential tool to improve quality. In this analysis, you will find strands:

- *Uncertainty analysis (**UA**)* which helps to quantify the variability of the output;
- *Sensitivity analysis (**SA**)* that allows the understanding of how the uncertainty of the outputs can be divided proportionally between the different sources of uncertainty in input to the model.

Uncertainties come from different sources, so [Heo, 2011] in one of his studies identified many categories, shown in Table 2.2.

Table 2.2. Source of uncertainty in building energy models

Category	Factors
Scenario Uncertainty	Outdoor weather conditions Building usage/occupancy schedule
Building Physical/operational uncertainty	Building envelope properties Internal Gains HVAC Systems Operation and control settings
Model Inadequacy	Modeling assumptions Simplification in the model algorithm Ignored phenomena in the algorithm
Observation error	Metered data accuracy

There are different methods to apply UA and SA analysis. First of all, it is necessary to distinguish two main approaches, i.e. external and internal methods. The so-called *"internal"* methods include mathematical approaches with equations. The other methods, on the other hand, take into account tools to help assess the variation of outputs. Within this category, there are local and global methods.

Local methods include **OAT** (One At a Time) methods, i.e. those where, changing only one parameter at a time and the others are kept constant, evaluate the variation of the model results. They find some methods within them, including:

- *Sensitivity Index*;
- *Differential Sensitivity Analysis (DSA)*;
- *Morris Method (or Elementary Effects).*

Sensitivity Index take the name from the index estimated to judge the sensitivity of each parameter. It corresponds to the percentage difference between the extreme measured values of the parameter considered (maximum and minimum): the parameters is influential when changes considerably.

$$SI = 100\% \frac{E_{max} - E_{min}}{E_{max}} \tag{2.10}$$

The DSA method modifies the parameters one by one over time, and estimates a **coefficient of influence (IC)** to evaluate the variation of the inputs on the model output:

$$SI = \frac{\frac{\Delta OP}{OP_{bc}}}{\frac{\Delta IP}{IP_{bc}}} \tag{2.11}$$

where OP is the output value, input IP and bc indicates the value referred to the baseline model.

Often the DSA model can be used in conjunction with the Morris Method, the most common screening technique because it is the most effective because of its global approach, although it is a local method. The model sensitivity is evaluated by measuring the average value and the standard deviation of the EE indicator for the parameter considered: through it classifies various inputs according to their influence on the output.

$$EE_i = \frac{Y(x_1, x_{i-1}, x_i + \Delta_{i+1}, ..., x_k) - Y(x_1, ..., x_k)}{\Delta_i} \tag{2.12}$$

where Y is output value of evaluated system, before and after the variation of the parameter considered, and the delta is the variation of the parameter. For each parameter different *"trajectories"* must be simulated; therefore the average value and the standard deviation for each of them must be calculated, following the following formulations

$$\mu_i = \frac{\sum_{j=1}^{r} EE_i(X^j)}{r} \tag{2.13}$$

$$\sigma_i = \sqrt{\frac{\sum_{j=1}^{r}[EE_i(X^j) - \mu_i]^2}{r}} \tag{2.14}$$

The results obtained should be plotted on the graph proposed by Morris, to compare the average value of each parameter with the standard sweat deviation: in this way you can identify where the points fall on the Morris plane to understand their influence on calibration. For example, parameters with high μ and σ values are the most critical for calibration, those with high σ but small μ influence calibration.

Global methods are based on the variation of several parameters at the same time, considering the influence of input uncertainty on the whole system. Among these, it is possible to find several, such as *Regression Analysis, the Variance-Based Method and the Monte Carlo Method.*

It is helpful using the **Regression analysis** to consider different scenarios and their impact on the energy consumption of the building, valuable for the reduction of computational time.

Variance-based method helps to decompose the uncertainty of the outputs between the various inputs. This technique uses two different sensitivity measurements:

- *First-order index*, which represents the outcome of the input parameter on the variation of the output;
- *Total order index*, which measures the effect of the parameter only and the sensitivity of its interaction with the others.

This methodology makes use of **non-linear and non-monotonous models** including:

- ***Variance Analysis (ANOVA)***, useful to divide the output variance between the various input parameters, and
- ***Fourier Amplitude Sensitivity Test (FAST)*** used to calculate only the first-order sensitivity index to estimate both first and total sensitivity index. In essence, it calculates the individual contribution of each input factor to the output variance.

The ***Monte Carlo Method*** is the most common and used, and it makes use of a repeated number of simulations with a random distribution of the input parameters to the model: the contribution of each of them is evaluated through probability distributions. This method allows for estimating the overall uncertainty of the model based on the uncertainties of the input parameters.

2.2 Reinforcement Learning

Reinforcement Learning is a branch of machine learning, together with supervised and unsupervised learning. Compared to the other two, it returns not only the outputs (depending on the input entered) but also a score for this output [Yaser, 2012].

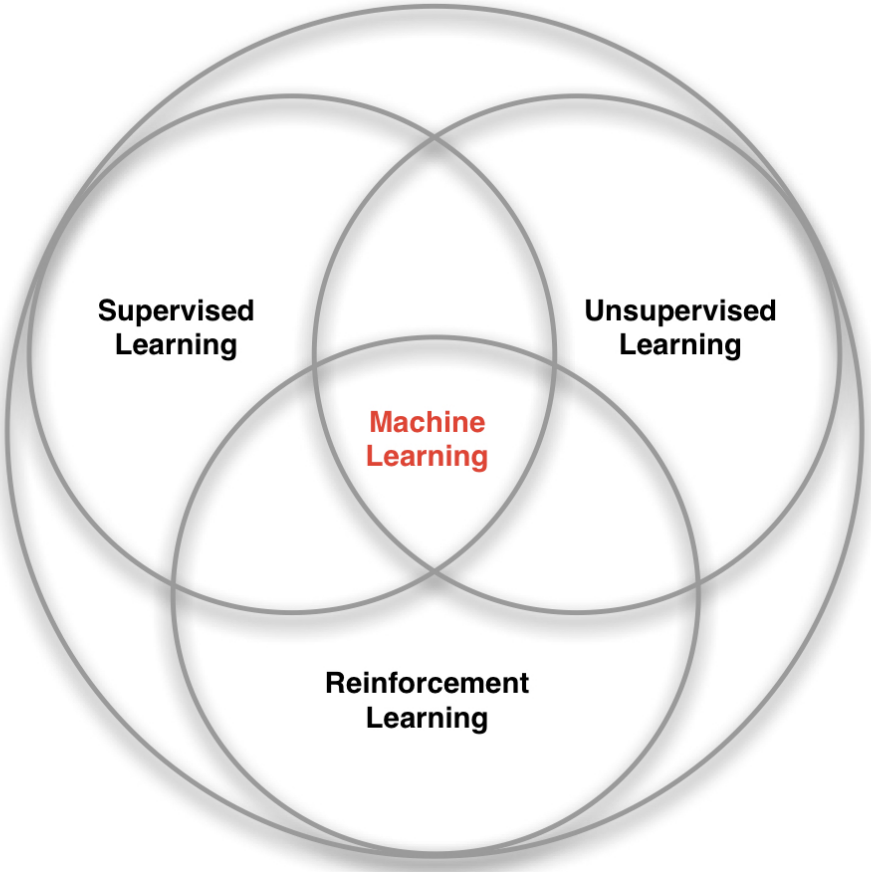

Figure 2.2. Machine learning branches [Silver, 2015]

It is a learning technique that aims at realizing agents able to choose actions to be carried out in order to achieve certain objectives, through interaction with the environment in which they operate. For this reason, it does not need a priori a model and is defined **model-free**.

It deals with problems of sequential decisions, in which the action to be taken depends on the current state of the system and determines the future one.

[Dalamagkidis et al., 2007] declare that RL is applied to problems that can be divided into two categories:

- *Episodic problems*, that have one or more terminal states. One episode is repeated much time during the agent's training phase in order to investigate all possible combination of states and rewards. When an agent reaches a

specific state, the episode ends, and the environment is reset to the initial state. Then, a new episode starts.

- *Continual problems* do not end, and they continue indefinitely.

RL refers to the model established by the **Markov Decision Process (MDP)**, according to which both the reward and the probability of transition between the previous and the next state depends only on the current state and the action chosen. MDP predicts the next state and the expected reward using only the current information available and not all the information accumulated in the past.

MDP formalizes the interaction between agent and environment mathematically, indicating the fundamental elements that this RL depend on ([Wang et al., 2017]):

- **State-space ($\mathbf{s} \epsilon \mathbf{S}$)**, i.e. all possible states of the environment considered. It is fundamental to choose well the states because if some of them are omitted relevant for the control of the environment, the agent cannot reach an optimal policy. Otherwise, if an unnecessary state is chosen, the RL agent suffers from the curse of dimensionality [Wang & Hong, 2020].
- **Action space ($\mathbf{a} \epsilon \mathbf{A}$)**, the set of possible actions that can be selected by the agent at each timestep.
- **Reward (r)**, a scalar value that is emitted from the environment after seeing the action sent by the control agent.
- **Policy (π)**, that formally linked states and the probability of each action of being selected. The agent's objective is precisely to acquire an optimal policy.
- **Transition probability distribution**, which specifies the probability that the environment can emit a specific reward and go to the next s' state, following the action due to the s' state.

For each time step, the agent will perform a particular action and receive from the environment, both information regarding his status and the reward. At the same time, the environment to be monitored will receive the action and will then issue both the scalar reward and the remarks to the agent in an instant following the action received.

Sometimes, the concept of **tuple** can be recurrent. It is a vector that contains within it four elements: state, action and reward at the current timestep and state at the next timestep.

Also, two value functions are defined, **state-value** and **action-value**, which are of fundamental importance in determining the optimal policy.

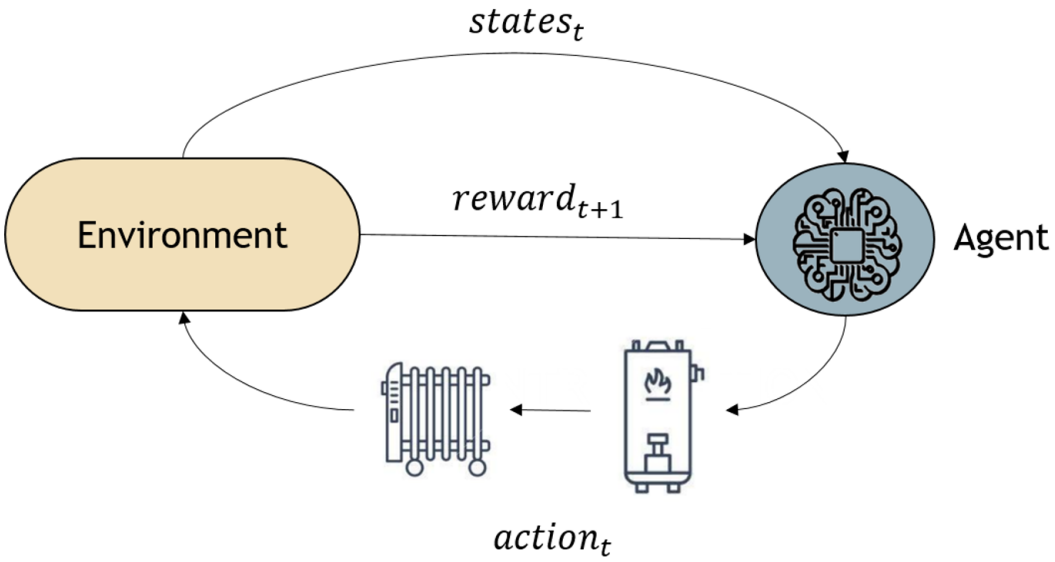

Figure 2.3. Typical control loop based on RL.

The state-value function represents the expected reward given by the agent when starting from a state s, following a specific control policy pigreco. The following equation expresses it:

$$v_\pi(s) = E[r_{t+1} + \gamma v_\pi(s')|S_t = s, S_{t+1} = s'] \tag{2.15}$$

The action-value function represents the expected reward given by the agent when he acts on the environment, starting from a state s, following a specific control policy π. The following equation expresses it:

$$q_\pi(s, a) = E[r_{t+1} + \gamma q_\pi(s', a')|S_t = s, A_t = a] \tag{2.16}$$

These two functions are updated online during the training phase of the agent, so they depend heavily on the experience gained.

The RL agent is trained through a *trial-and-error approach*: this means that it tries different trajectories (i.e. policies) and, after evaluating their performance, tries to improve them. In this case, it is spoken about **on-policy learning**, i.e. the policy output of the controller is being carried out by the environment. There is also the counterpart, **off-policy learning**, where the agent learns from other policies already created for other interests.

Some value-based algorithms use this methodology, especially for its greater flexibility than on-policy learning. However, off-policy learning has a significant disadvantage compared to its counterpart, which is a lower propensity to explore action space.

To overcome this problem, a large amount of measured data should be available, but using only measured data may be inadequate [Wang & Hong, 2020]. So, as in this case, simulated virtual environments can be created and used to train the RL agent. To do this, it is advisable interface energy simulation and control platforms, such as EnergyPlus and Python.

2.2.1 Q-Learning

Within the RL, the most widely applied model-free approach is **Q-Learning**. It belongs to the **Temporal Difference (TD) methods**, and it is used in case of incomplete information models. The TD learning methods are the most used because it has been proven that they converge towards an optimal policy faster than the other two RL methods, namely Monte-Carlo (MC) and Dynamic Programming (DP) [Dalamagkidis et al., 2007].

Q-Learning uses lookup tables called **Q-Table**, with expected returns called **Q-values** and obtained following a specific action from a specific state are stored [Uhn Ahn & Soo Park, 2019]

The Q-values are updated during the learning because of the agent's experience accumulated: this process is reported in mathematical form thanks to ***Bellman's equation***.

$$Q(s,a) = Q(s,a) + \alpha[r_t + \gamma max_{a'} Q(s',a') - Q(s,a)] \tag{2.17}$$

With α [0,1] is the **learning rate** e γ [0,1] is the **discount factor** for future rewards.

If $\gamma = 0$, the agent will give greater importance to currently reward, neglecting future reward. The opposite case happened for $\gamma = 1$. α determines with which extension new knowledge overrides old knowledge. $\alpha = 1$ means that new knowledge completely overrides the old one.

A distinctive feature of reinforcement learning is ***action-selection***. It represents the trade-off between exploration and exploitation.

Exploration is defined as that phase in which the agent finds himself exploring within a new and massive set of actions (not yet selected), neglecting his real goal of maximizing the reward.

By **exploitation**, it is intended that phase in which the agent chooses within the previously selected actions, the one that allows him to get as close as possible to his objective.

A right control agent must try to optimize the compromise between these two stages. This process must be represented in mathematical form. In this case, there are two supporting approaches:

- ϵ-greedy approach.
- Soft-max approach (alias Boltzmann action-selection approach).

The **ϵ-greedy approach** selects a known and considered 'excellent' action in the exploitation phase with a probability of $1 - \epsilon$. In contrast, it selects a random action in the exploration phase with a probability of ϵ. ϵ represents the coefficient of the rate of exploration [Zou et al., 2020]. This method is the most widely used as it is much simpler and affects control performance less [Wang & Hong, 2020]. It is clear that a specific mathematical law can be set for ϵ so that, as time goes by and the learning agent proceeds, ϵ can decrease in order to favour the exploitation phase.

The **Soft-max approach** selects the action based on the action's performance and exploits more when the majority of action space has been explored already. This strategy could be easily implemented by reducing ϵ or τ [Wang & Hong, 2020].

The agent selects a particular action with a probability ϵ and with probability referred to his Q-values [Brandi et al., 2020]:

$$Pr(s|a) = \frac{e^{\frac{Q(s,a)}{\tau}}}{\Sigma e^{\frac{Q(s,a)}{\tau}}} \tag{2.18}$$

2.2.2 Deep Q-Learning

In some situations, Q-Learning, as described above, may be inadequate if the space of actions or states is ample, as the memory storage and computation time required to update the Q-table [Sutton & Barto, 1998].

In this case, as an alternative to the tabular form of Q-Learning, it may be useful to use a **Deep Neural Networks (DNNs)** as function approximator.

The main element of a neural system is the neuron, composed of a cellular body and an axon that sends the output response to the next layer. There is a dendritic tree structure that connects it to other neighbouring neurons.

The topology of a DNN is based on multiple layers of neurons. Typically, a neuron is a non-linear transformation of a linear sum of its inputs. DNNs are composed of **input and output layers**, but between them are **hidden layers** that take input from the previous layer.

By inserting DNNs into the Q-Learning results **Deep Q-Learning** (also called as Deep Q-Network). The Q-values will be indicated with the following formula, taken from [Nair et al., 2015]:

$$Q(s, a) = Q(s, a; \theta) \tag{2.19}$$

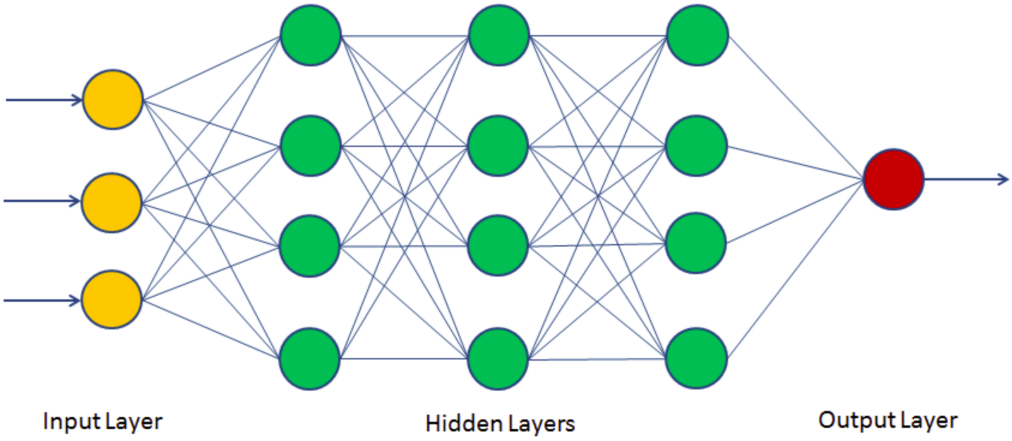

Figure 2.4. Example of Neural Network with three hidden layer [Datacamp, 2020]

The equation represents the Q-network, in which is present the term θ which parameterizes the Q-value function: it indicates the weights of the network. The number of neurons in the input layer is equal to the number of variables that make up the state space, while the number of neurons in the output layer corresponds to the size of the action space.

The process that takes place in the DQN can be schematically illustrated effortlessly:

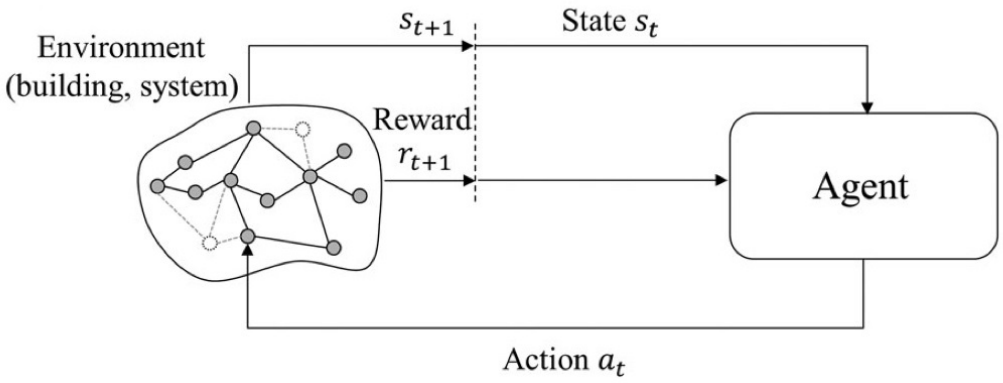

Figure 2.5. Reinforcement Learning Deep Q-Network [Uhn Ahn & Soo Park, 2019]

The network is used to represent the value functions and to find the optimal policy, i.e. the relationship, for each action, between states and Q-value. It is good to remember that this last parameter is not known a priori and is obtained during the training process as already explained in the previous paragraph on Q-Learning, and updated according to *Bellman's equation*.

The DQN could, however, be improved by introducing another NN indicated as the **target network**, which is an exact copy of the first one, called **online network**. The main characteristic of this technique is the presence of two DQNs to counteract the overestimation of the Q-values that may lead to a non-optimal outcome when using a single DQN.

This additional DQN is an exact copy of the other one. However, it is only synchronized for every τ steps (an arbitrary number), and it is used to calculate the target Q-values for expectation [Hasselt et al., 2016].

Instead, the online network is the one used to interact with the environment and updated with Bellman's equation.

Moreover, [Brandi et al., 2020] proposed the introduction of a **replay memory**, useful to store inside it the tuples referred to the previous experiences of the agent: this allows, if necessary, to reuse them and go beyond the problem of related observations. At the same time, the optimization process is carried out.

2.2.3 Soft Actor-Critic

Model-free Deep Reinforcement Learning algorithms face problems related to high sample complexity, because simple tasks could require a huge number of data collection steps: this leads to poor sample efficiency due to on-policy learning. It is necessary to try to switch to off-policy algorithm. Moreover, they suffer from dependence on the chosen values of hyperparameters, like discount factor, learning rates, exploration constants and other. These two obstacles make it challenging to apply these control algorithms to real-cases.

To try to overcome these obstacles, the **Soft Actor-Critic (SAC)**, an off-policy algorithm based on the maximum entropy RL framework, was recently introduced by [Haarnoja et al., 2018]. While most existing model-free works make use of discrete action space, SAC uses a continuous one.

This algorithm aims to maximize a **target function** composed not only of the term **expected reward** but also of an **entropy term**. This last term, is what expresses the attitude of our agent in the choice of random actions. It also has dual importance, as it **ensures that the agent is explicitly pushed towards the exploration of new policies and at the same time avoids that it transposes lousy policy**.

This algorithm uses a particular **Actor-Critic architecture** that emoloyees two different deep neural networks for approximating, respectively, action-value function and state-value function.

Finally, it allows the use of **continuous action-space**, which is essential when controlling the parameters of energy systems.

The current state of the art sees applications like those in the field of robotics, and recently it is increased its application in energy building context.

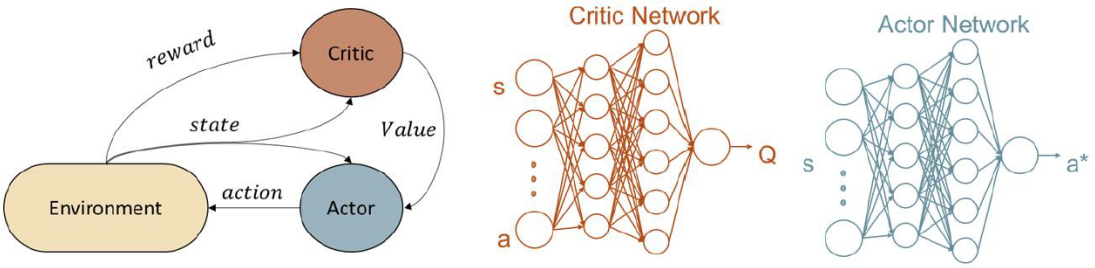

Figure 2.6. Typical control loop based on RL [Pinto et al., 2020]

The SAC presented around the last months of 2018 suffered from dependence on hyperparameter temperature, therefore in the latest version proposed in [Haarnoja et al., 2018], it is devised an automatic gradient-based temperature tuning method that adjusts the expected entropy over the visited states to match a target value.

Soft actor-critic is based on the maximum entropy reinforcement learning framework, in which the objective is maximize both expected reward both entropy. It could be seen as an extension of standard RL objective.

The maximum entropy objective requires an optimal policy like this:

$$\pi^* = argmax_\pi \sum_t \gamma([E_{(s_t,a_t)}[r(s_t,a_t) + \alpha H(\pi(\cdot|s_t))]]) \tag{2.20}$$

with α temperature parameter, that indicates the importance of the entropy term compared to reward one, also indicates the stochasticity of the optimal policy. Generally α is zero when considering conventional reinforcement learning algorithms.

It is convenient introducing a discount factor γ to ensure that the sum of expected reward and entropies is finite. The SAC is derived from a variant of the maximum entropy framework, called Soft Policy Iteration, which is not presented here.

In the first version of SAC, the temperature parameter was fixed and then considered as an hyperparameter, so its choice had an important influence on the agent's behaviour. To avoid this problem, in the next SAC update was introduced the possibility of making *alpha* as an update-able parameter. In particular, it is updated by taking the gradient of the Objective function below:

$$J_\alpha = E[-\alpha \ln \pi_t(a_t|s_t;\alpha) - \alpha\bar{H}] \tag{2.21}$$

where $\bar{H}$ represents the desired minimum entropy, set to a zero vector.

This SAC latest version improves both performances both the stability of the algorithm, and it is decided to use this one for the thesis work.

Soft actor-critic maximizes this objective by parameterizing a Gaussian policy and a Q-function with a neural network, and optimizing them using approximate dynamic programming [Haarnoja et al., 2018].

In conclusion, this algorithm is particularly useful under a changing environment or when agent's knowledge of the environment changes [Pinto et al., 2020].

Chapter 3

Case Study

The previous chapter describes the tools that were used in this case study, starting with the model calibration and then applying the DRL-based control algorithm.

In this work is taken into account an office real-building served by a radiant heating system. Initially, the building case study will be described, then the description of the simulated environment used and the calibration and control procedures applied will be discussed.

Finally, it will be described the setting up of the two phases of training and deployment, for which the results will be reported in the next chapter.

3.1 Building and HVAC System description

The building under observation is located in Turin, Italy, and consists of five heated floors and a basement, with a net heated surface of about $9300m^2$, shown in Figure 3.1. The five heated floors are divided into two heating zones for convenience:

- the ground floor is dedicated to the *"Vigili"* area and the caretaker's room;
- the remaining four floors are for offices (about $7000m^2$).

The occupancy schedules are defined knowing the actual offices opening and closing times. Every day, except Sundays and holidays, the office is occupied from 7 AM to 7 PM.

The two thermal zones are served by a single hot water circuit, consisting of two loops connected by a heat exchanger. There is a four gas-fired boiler in the first loops, serving a collector from which different pumps extracting water to serve the radiator type terminals.

In real cases, there are two possibilities to regulate the supply water temperature:

- three-way valves use, with constant speed pumps;

Figure 3.1. Building Case Study located in Via Bazzi 4, Turin

- two-way valves use, with variable speed pumps.

In this application, the first solution is employed.

The installed system is much more complicated than the implemented one in EnergyPlus. In the implementation, it was decided to build two different circuits serving each thermal zone, consisting of a single gas-fired boiler that will supply hot water to the radiators employing a constant speed pump.

The Supply Water Temperature is managed through the External Interface in EnergyPlus, receiving input directly from Python.

Since there were radiator terminals, this study was focused only on the thermal zone internal temperature for two reasons:

- because this type of system allows only its control, being able to operate only on the sensitive thermal load. A different type of system should be used to influence the other internal properties of the building;

- because the other comfort variables are not monitored.

This case study presents a control problem that mainly focuses on the heating phase in the office area. It will study the behaviour of the control agent from 1 AM until the occupants' presence, obtained from the actual occupation schedules.

The main objective of the implemented control policy will be to obtain the desired temperature conditions during occupancy and also to energy saving in the

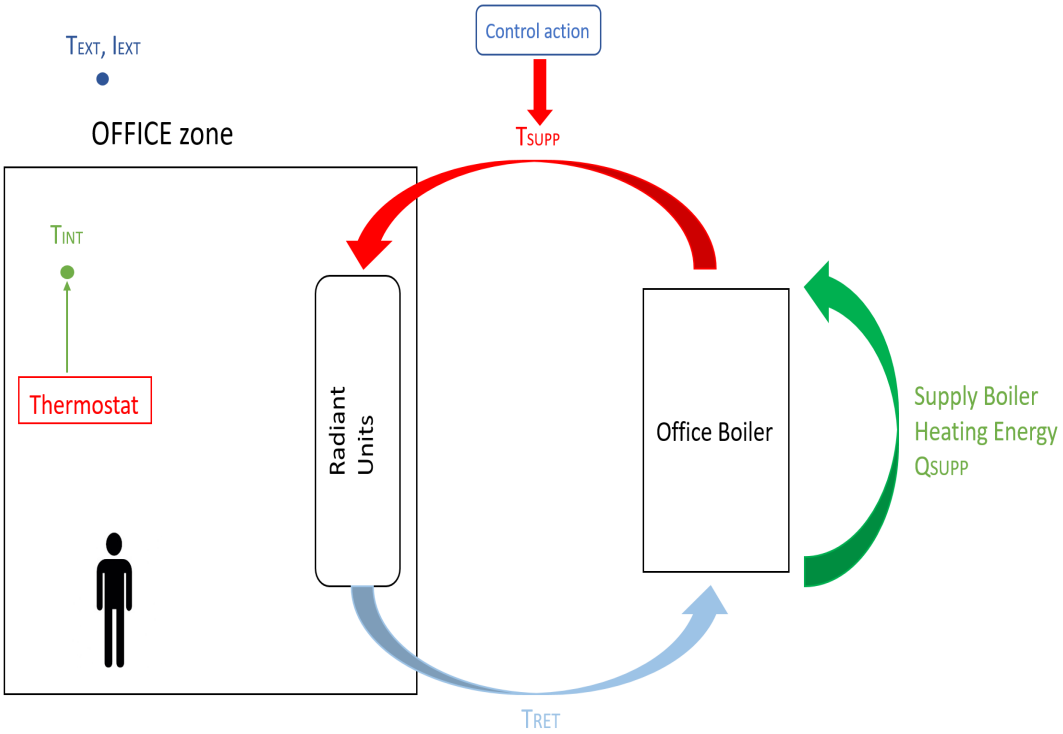

Figure 3.2. Scheme of case study heating system

heating phase, through the regulation of the supply water temperature to heating terminal units of the office zone.

In order to tighten the controller, it is chosen a temperature range within which it has an acceptable behaviour, outside which it receives a comfort penalty, which will be added to the energy-term reward linked to the consumption.

If the average temperature falls within the range of lower and upper limits, the indoor temperature comfort requirements shall be respected. In this work, the range of acceptability is defined between $[-1, 1]°C$ from the desired internal temperature value of $21°C$. Also, the focus will be on the boiler energy supplied to heat the water-carrier fluid to be sent to the terminals.

3.1.1 Baseline Control Logic

The plant control system currently implemented refers to the algorithm proposing a logic based on the combination of rule-based and climatic-based for the control of the supply water temperature.

It is a control type based on the climate curve, i.e. strictly dependent on the outside temperature. The supply water temperature varies linearly within a range from $40°C$ if outside there are more than $12°C$ and $70°C$ when the outside temperature drops below $-5°C$.

These values were chosen because they correspond to those implemented in the real building Energy Management System (EMS) for the supply water temperature control logic.

The time in which the boiler system is switched on is based on the internal temperature value when the occupants arrive. This control strategy is in operation up to one hour before the occupants leave the building.

The agent controls the switching on of the boiler when the following situations appear:

- if the temperature difference between the lower limit of the proposed range and measured one at the occupants' arrival is greater than $3°C$, the switch on occurs between four and three hours preceding the arrival of the occupants;
- if the temperature difference between the lower limit of the proposed range and measured one at the occupants' arrival is greater than $2°C$, the switch on occurs between three and two hours before the arrival of the occupants.
- if the temperature difference between the lower limit of the proposed range and measured one at the occupants' arrival is greater than 0, the switch on occurs up to two hours earlier to the arrival of the occupants.

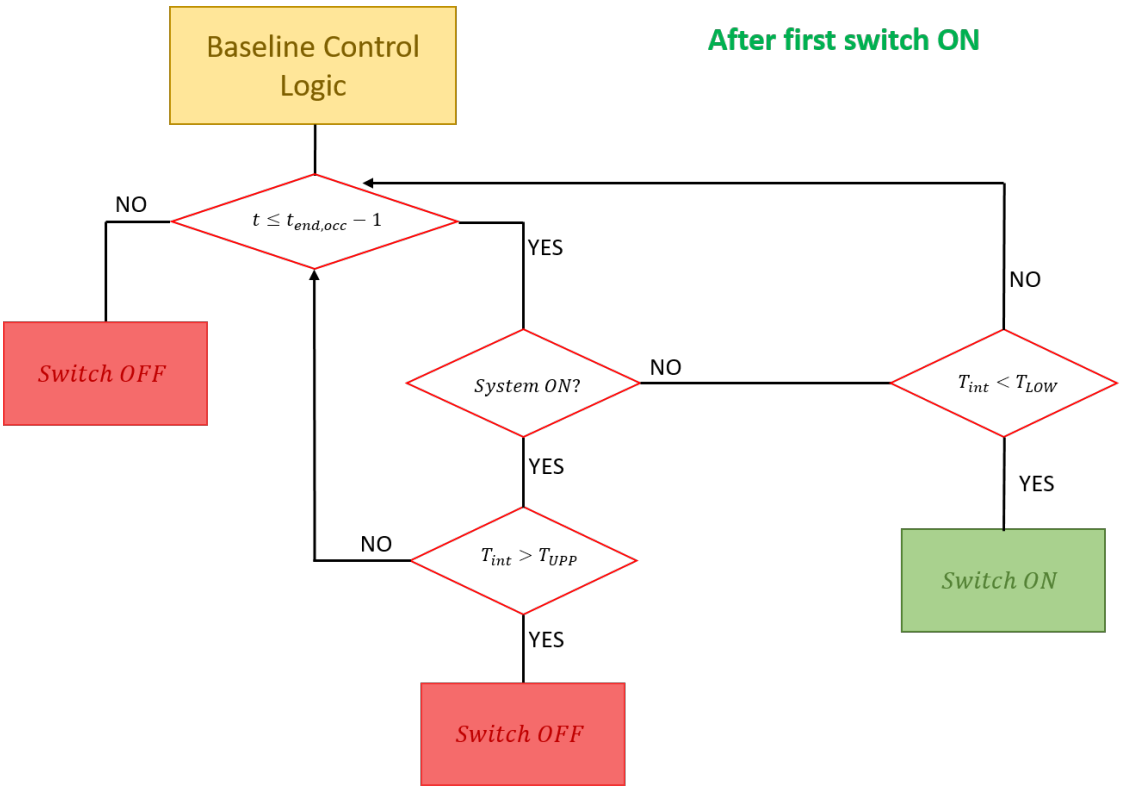

Figure 3.3. Baseline Control Logic after first switch ON.

On the other hand, shutdown occurs at any time when the internal thermal zone temperature is higher than the upper threshold limit temperature, equal to $21.8°C$. If the occupants are present and the temperature drops below the low threshold limit temperature, equal to $20°C$, the system is switched on again.

The baseline switch ON/OFF range $[-1, 0.8]$ is different on the upper limit concerning the SAC control logic. This voluntary choice was made in order to exploit the thermal inertia of the building better.

Nevertheless, for comparison with SAC control, our agent will be penalised accordingly to the requirements of subsection 3.4.3.

The boiler will remain switched off even on Sundays, when there are no occupants.

3.2 Simulation Environment

In the first part of this work, the calibration of the energy model created in the early phase is executed and, for the simulation, EnergyPlus v9.2.0 is required, with subsequent simulations according to the trial-and-error method.

Subsequently, for the control phase, an **external interface** had to be used to implement the algorithm within the EnergyPlus simulation. The interaction between the agent and the simulated building takes place through a simulation environment also composed of EnergyPlus and Python, the latter based on *OpenAI Gym*. Moreover, it is necessary to use Python libraries such as *Tensorflow*, which allows the desired interaction.

The Building Control Virtual Test Bed (BCVTB) and the ExternalInterface-Ptolemy server command from EnergyPlus were used to connect the two software.

Reinforcement Learning control requires four essential functions in Python:

- init(), a function that initializes the simulation;
- step(), a function to which a specific action is passed, which is then implemented in the simulated building and returns four objects: next state, reward, done (True/False) and info;
- reset(), a function that is called at the beginning of each episode, to repeat the simulation several times;
- render(), a function that renders one frame of the environment.

In reset() it is expressed something about the concept of **episode**: it represents a certain period on which the simulation in EnergyPlus takes place. An episode can correspond to one simulation and can be repeated several times during the training phase to get good results from the exploration phase.

The information exchange between the two software is as shown in the Figure 3.4.

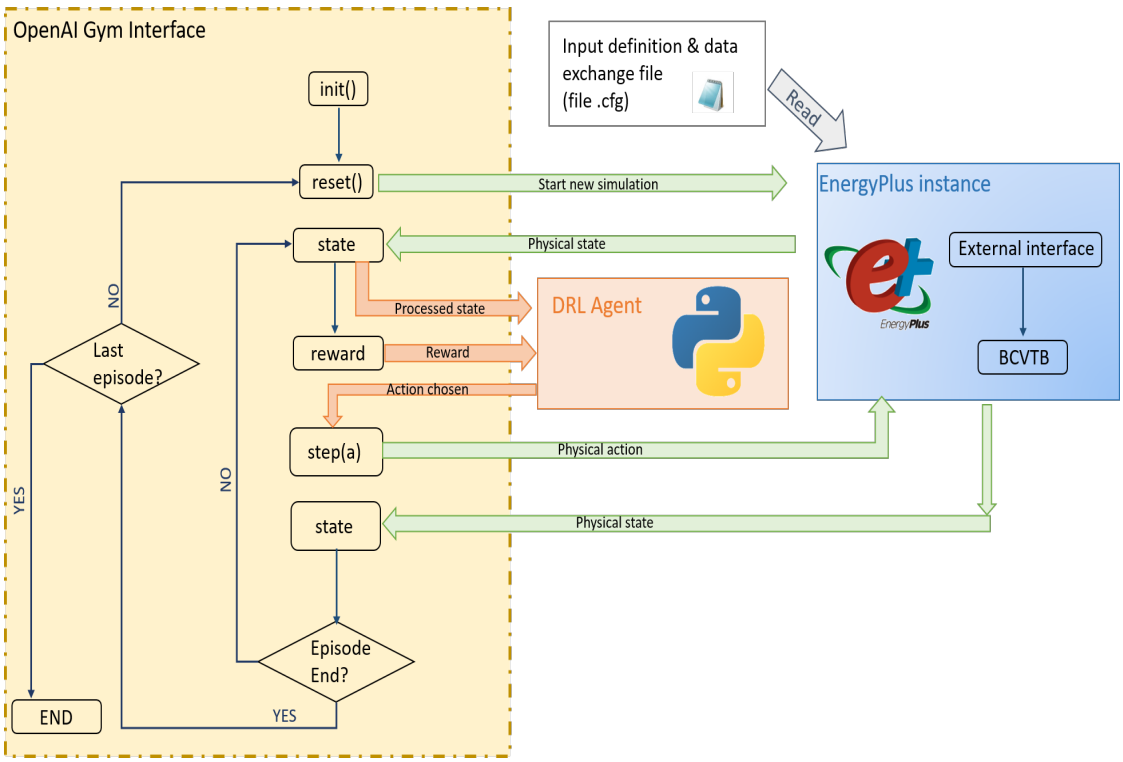

Figure 3.4. Simulation environment for DRL-SAC control

The process takes place in this way:

- the OpenAI gym interface object is initiated by calling the **init() function**, then a server socket for the communication between EnergyPlus and Python is created;
- the **reset() function** is called up by the control agent immediately afterwards: an instance of EnergyPlus is immediately created using the IDF file format and the CFG extension file that allows data exchange;
- the OpenAI Gym object creates a TCP connection with EnergyPlus, in which ExternalInterface incorporates features that are inputs from Python. The ExternalInterface using a BCVTB for performing as a client.
- the TCP connection is used to read and return the simulation output from EnergyPlus to OpenAI Gym. Then observations are processed by DRL agent for extracting state and reward;
- the DRL agent calls the **step(a) function** at each control steps and sends the action **a** to Energyplus and read the results after this control action. Then the observations are returned in order to obtain new state and reward;

- it is necessary to check if the simulation is at the end of the episode: if it happens, the process moves on to the next check, otherwise, it is repeated the above process starting from the observations obtained again by EnergyPlus;
- if the processed episode is the last one, then the process ends here. Otherwise, it starts again from the point where the **reset() function** is called.

It is also important to remember that time-steps can differ between control and simulation, as sometimes the required action can be performed over a longer time horizon than the simulation (usually simulation time-step is longer than control).

In this work, the control step will be greater than the simulation step: the simulation step is 5 minutes, while the control step is 15.

3.3 Via Bazzi Model Calibration

In this section, the calibration process applied to our case study is discussed. The building is located in Via Bazzi 4, Turin.

After the model creation, it was requested to refine the calibration to obtain an almost similar behaviour between the simulated building and the real one in the case of internal temperature and consumption profiles.

As far as natural gas consumption is concerned, consumption data from 15 October to 6 January are available, while for the internal temperature data recorded by Energy Management System are used for both zones.

It was chosen to operate as proposed by *[Davin et al., 2015]* with a ***trial-and-error*** approach.

In this thesis work, starting from the model provided, a series of parameters were adjusted, based on Energetic Diagnosis document provided by Iren to Enerbrain Srl [Iren, 2016]. In this way, it will obtain what has been called *"iteration 1"*.

Using *[Davin et al., 2015]* as a reference, some parameters influencing the behaviour of the building have been modified between an upper and a lower limit, to obtain the best result. It is essential, as indicated in Table 2.1, that the values of **MBE and Cv(RMSE)** respect the limits provided by the *ASHRAE standard*, taken as reference.

EnergyPlus v9.2 with simulation timestep 15 min was used for energy modelling, while PyCharm was used to read the database produced by the simulation.

In order to obtain a reliable result, it was necessary to create the weather file for the heating season in question, 2018-2019. EnergyPlus makes the weather files available in a section of its website for several locations around the world, including Torino Caselle: the problem is that this weather file represents the weather conditions, not of the period of our interest, so it was necessary to construct the weather

file. The weather data needed for 2018 & 2019 were requested to ARPA Piemonte and then processed, in order to create a *csv* file, compatibly with the requirements of the EP Launch *Weather tool*, then converted with the same in *epw* format.

3.3.1 Detail of results after the first iteration

After having modified some building parameters and the lights/occupancy schedule (thanks to Enerbrain WebApp and website of the city of Turin), without proceeding through a *trial-and-error* approach, the iteration 1 results were shown into following figures.

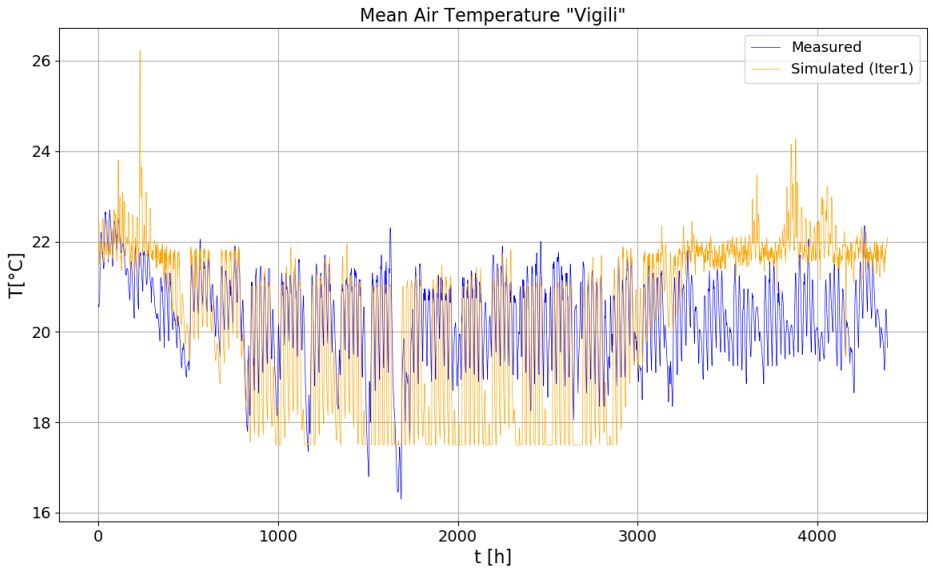

Figure 3.5. "Vigili" zone indoor Temperature trend after first iteration

It is possible to see that the model discreetly simulates the real behaviour of the building in the period between November and mid-February, as the recorded temperature profile is very similar to the simulated one. The differences are considered in the first part of the heating season and in the period after 15 February: this could be due to the external temperature and direct solar radiation profiles that reach values higher than the seasonal standards, as can be seen in the following figure.

In order to improve the calibration process, it was chosen analyzing in detail the loads for ventilation, infiltration, loads through the opaque and transparent casing. The improvements will be evaluated based on calibration metrics (MBE and $Cv(RMSE)$), evaluated on an hourly basis. At the end of the first iteration, the results obtained for internal temperature and consumption are reported in

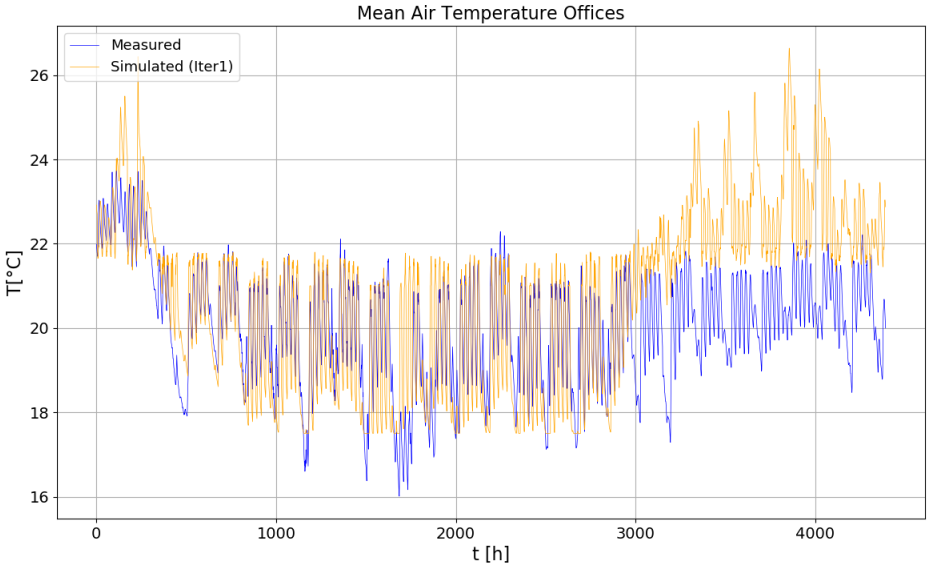

Figure 3.6. Office zone indoor Temperature trend after first iteration

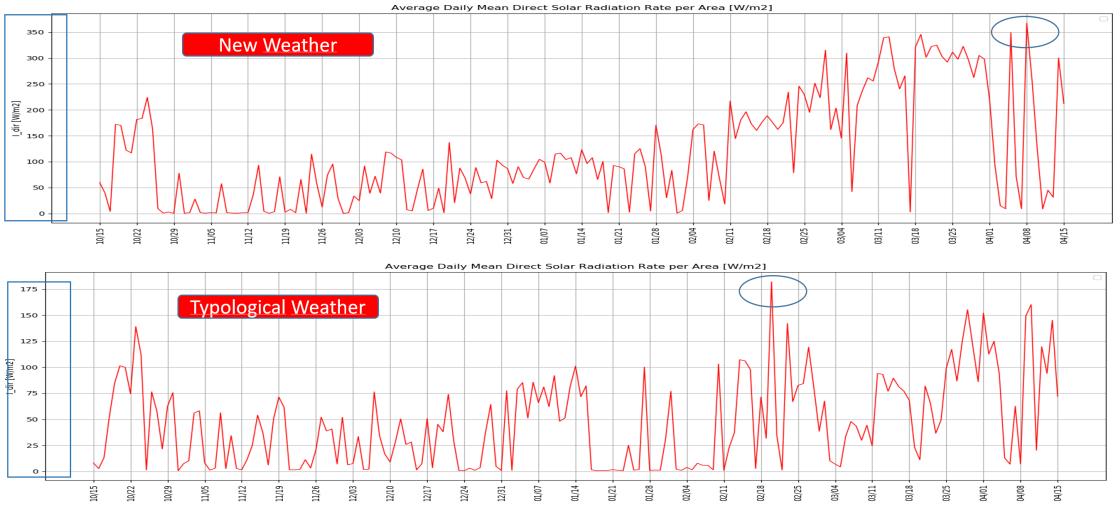

Figure 3.7. Direct Solar Radiation rate: Comparison between new weather and typological one

Table 3.1:

The temperature metric values are within limits set by the ASHRAE standard on an hourly basis. At the same time, for consumption they are very high: precisely for this reason, it was decided to evaluate at the end of the trial-and-error process those on a daily and monthly basis.

The following figure shows the *Energy Signature* for the period in which the measured consumption were available (1 November - 6 January), excluding holidays

Table 3.1. Calibration metrics results after first iteration

Variable	$MBE[\%]$	$Cv_{RMSE}[\%]$
Temperature ("Vigili")	0.15	7.09
Temperature (Office)	3.63	8.57
Consumption	-85.40	175.61

and holidays. The consumption is reported in kWh but was measured in Sm^3, so it was necessary to multiply by the LHV of natural gas, assumed to be:

$$LHV_{GN} = 9.4 \frac{kWh}{Sm^3} \tag{3.1}$$

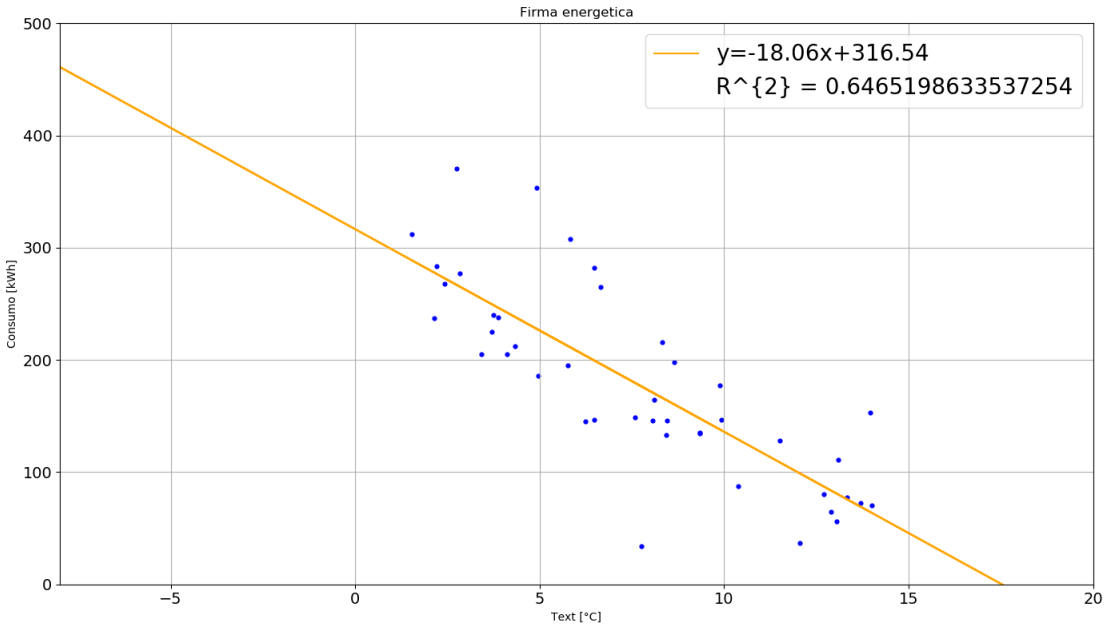

Figure 3.8. Energy Signature for consumption from 1 November to 6 January

The consumption value at the outside design temperature for Turin ($T_{ext} = -8°C$) and the zero-gas consumption outside temperature were obtained from regression equation.

$$E_{gas} = -18.06 * T_{out} + 316.54 \tag{3.2}$$

$$E_{gas,T_{ext}=-8°C} = 461kWh \tag{3.3}$$

$$T_{ext,E_{gas}=0} = 17.53°C \tag{3.4}$$

The conclusion is, however, a particular affinity among outside temperature and natural gas consumption trends.

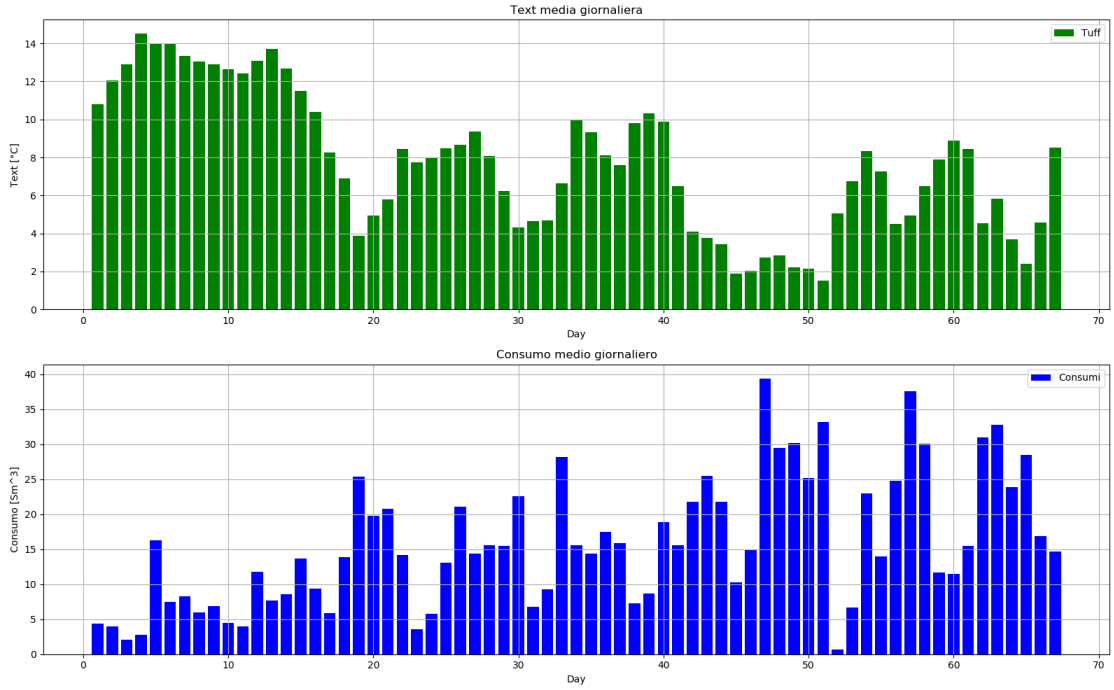

Figure 3.9. Comparison between gas consumption & outdoor temperature

As far as the temperature trend inside our building is concerned, it appears from Figure 3.9 that when the consumption drops, the average temperature also decreases. This could be due to the presence of Sundays or holidays (where the building is not occupied in the office area, and therefore the boiler does not have to supply hot water to the radiators) or to the lower demand for thermal power linked perhaps to an increase in internal gain.

Now, the building loads are analyzed, including the power supplied by the HVAC system (with boiler and radiators), the ventilation/infiltration loads and the opaque and windows heat gains.

HVAC Power Supply

The power supplied by the HVAC system to reach the temperature set-point initially set at 21 °C, should increase in the intermediate phase of the heating season, with lower values during the mid-season.

The following Figure 3.10 shows how assumptions are respected. It can be noted that the power required is higher on the fourth level because it is the one with the highest WTW (Windows to Wall Ratio) throughout the building and the most dispersant (the peak value was around $40kW$).

For brevity, it is shown the overall trend in the building and the fourth floor one only, because behaviour was almost equal everywhere.

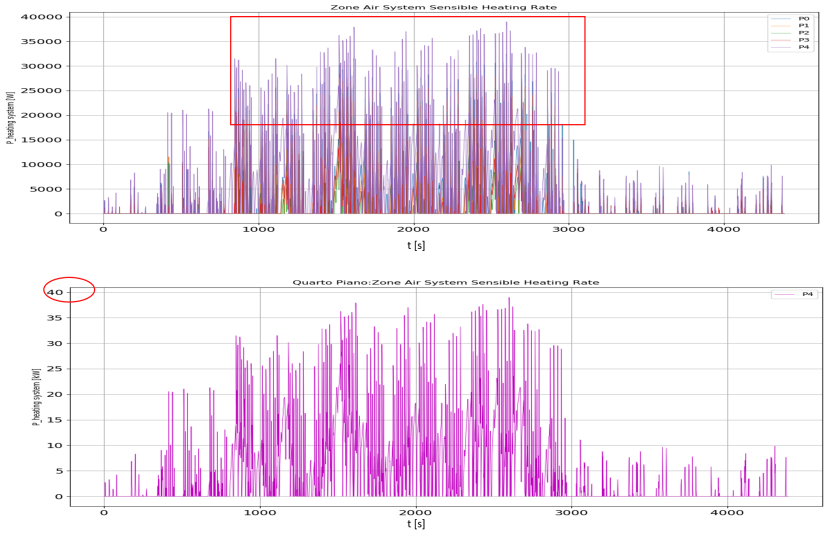

Figure 3.10. HVAC power supply: All floor requests and detail on fourth floor

Windows Heat Gain

In all building floors, both windows both opaque heat gain recognize an increase in heat gain during the mid-season, with a gradual decrease during the colder months, mainly from November to January.

In addition to showing the trend over the entire heating season, for opaque and windows heat gains it was also chosen to show a weekly detail, choosing two weeks within the heating season considered as typical. The choice fell on the following two weeks:

- From 10th to 17 December;
- From 18th to 25 March.

Windows heat gain is related to the amount of solar radiation coming from outside, putting itself in our building as a dominant load from the power input values inside the building.

Since the fourth floor is the most glazed, the values are expected to be quite high for windows heat gains: the Figure 3.11 shows just this evidence, making a comparison daily between the third and fourth floor.

The trend was the same for all floors, with peak values reached around the end of the heating season that are almost three times higher for the fourth floor than the third.

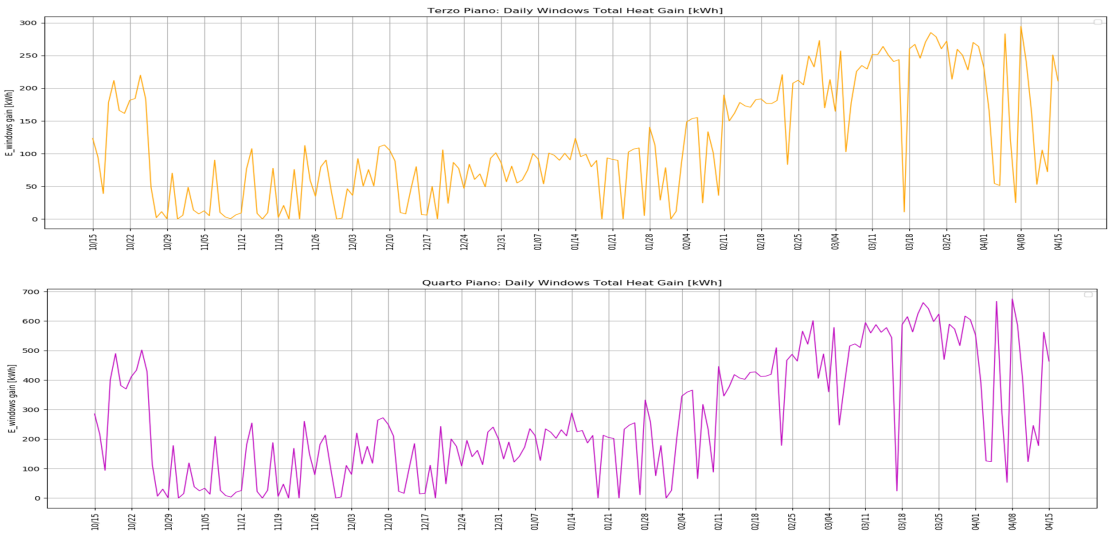

Figure 3.11. Windows Heat Gain: Comparison between third floor and fourth one

It may also be exciting note that when comparing the trends of windows heat gain and direct solar radiation, they appear identical (Figure 3.12).

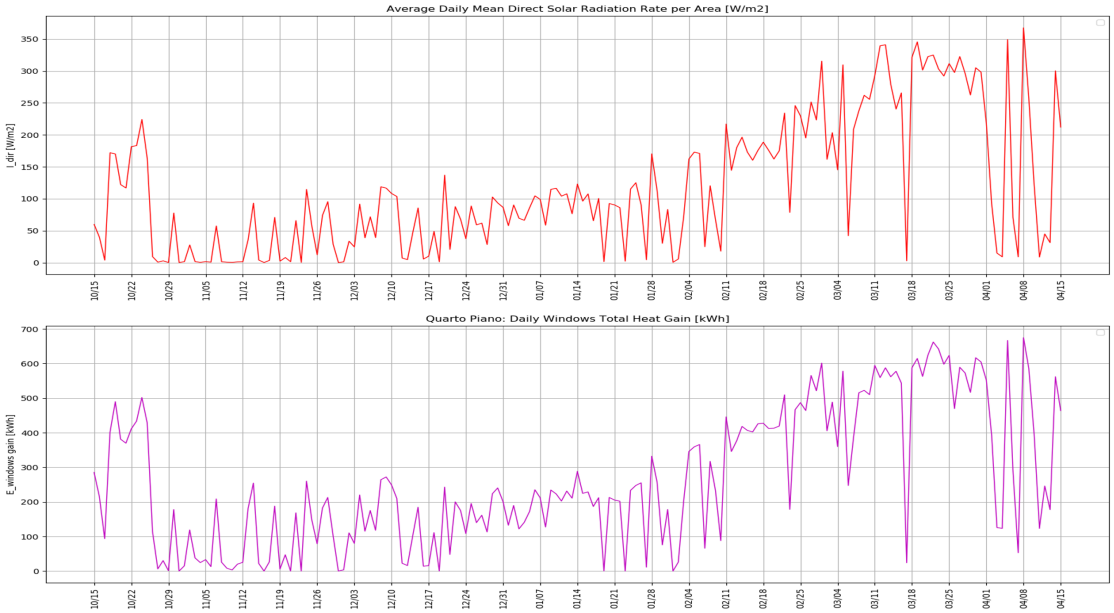

Figure 3.12. Windows Heat Gain: Comparison between Direct solar Radiation and fourth floor heat gain

Trends are also shown weekly in Figure 3.13 for these internal contributions, evaluating the differences.

In this case, it was possible seeing how the windows heat gains reach peaks around 60kW on the fourth floor and lower values for the other levels, confirming what already highlighted before.

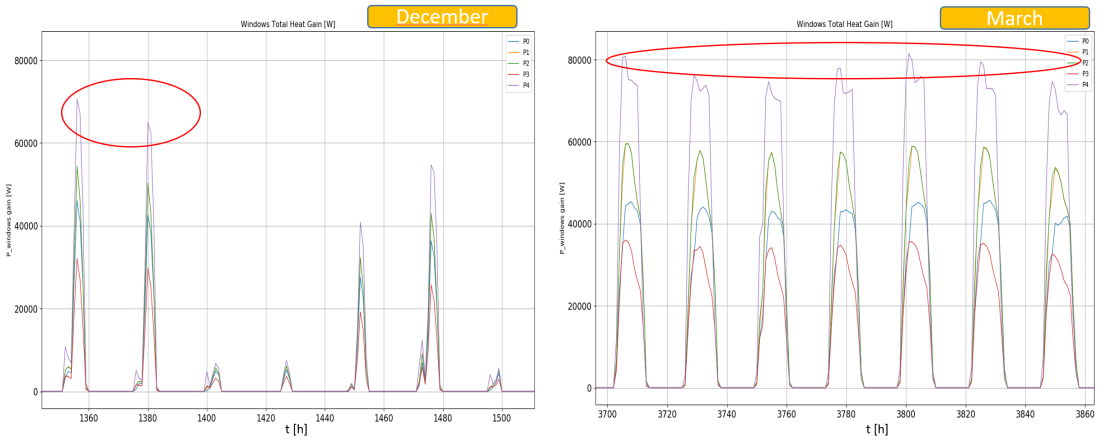

Figure 3.13. Windows Heat Gain: Typical weeks analysis

During the week of March considered, windows heat gains reached values of over 80kW, reasonably explaining the deviation between the measured and simulated temperature: this highlights the need to insert shields in the simulation.

Opaque Heat Gain

The opaque heat gains are related to the amount of solar radiation that comes from outside, released by the opaque envelope inside the building.

This aspect is related to the building thermal inertia, because during the day the envelope is hit by solar radiation that, contained inside the same, is then released at night: compared to the case of the windows, therefore, the internal contribution is released at a different time. After all, the heat gain passes directly through the windows and is released during the daytime.

The trend did not appear identical for all floors as in the case of windows (see Figure 3.14), just as there were no particular similarities between the trend of opaque heat gain and that of direct solar radiation.

The highest values were recorded for the second and fourth floor, as shown by the detail over the entire heating season.

The detail of the typical weeks chosen is also shown in Figure 3.15.

The heat gains associated with the opaque envelope reached peaks around 10 kW on the first floor. The trend, in this case, seems to be very similar for all, but not for the fourth floor.

During the week of March considered instead the heat gains linked to the opaque envelope could reach values of over 22kW, notably on the second floor.

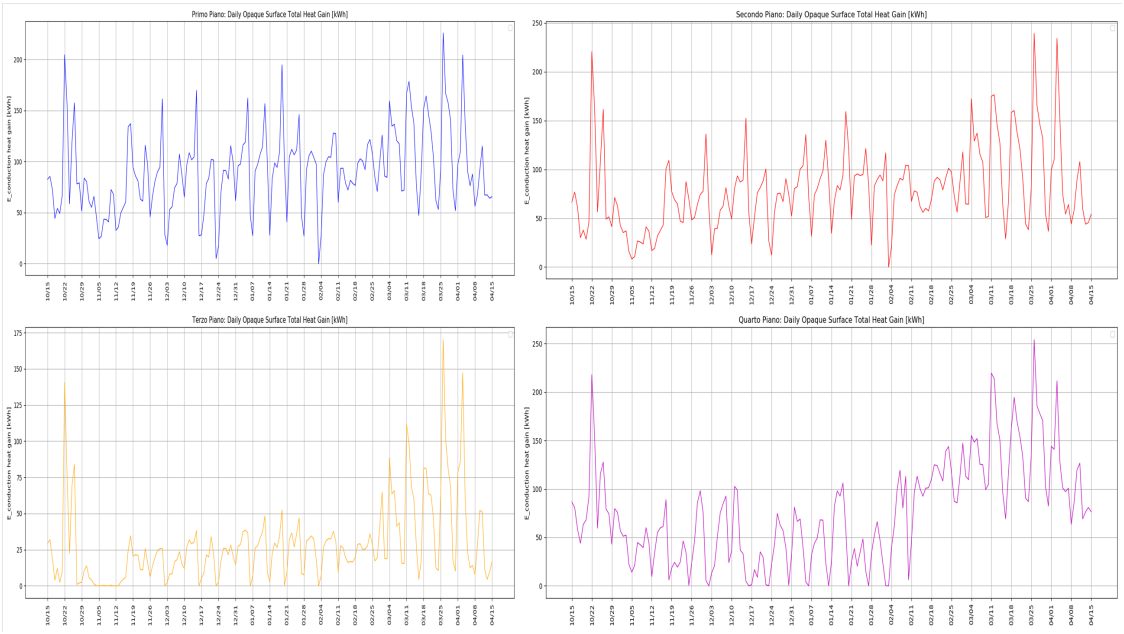

Figure 3.14. Opaque Heat Gain: Comparison between four level (from first to fourth)

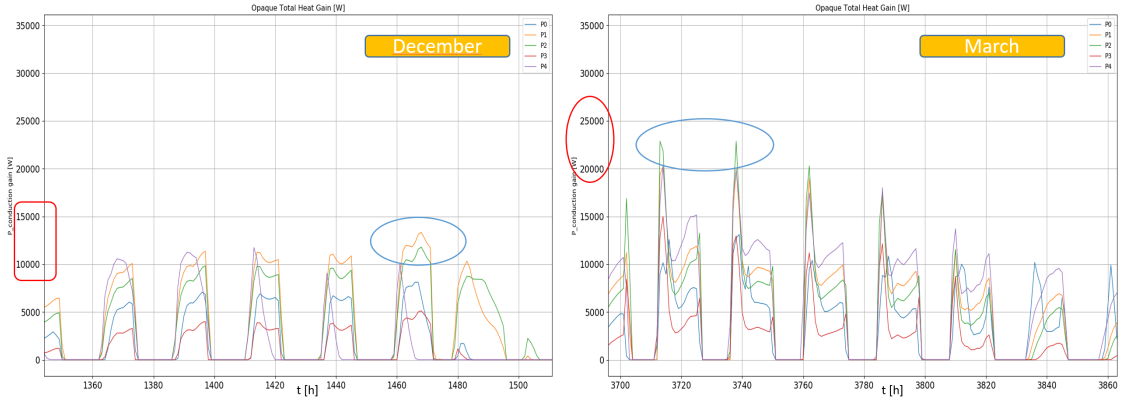

Figure 3.15. Opaque Heat Gain: Typical weeks analysis

Ventilation & Infiltration Loads

To conclude the overview of loads, ventilation and infiltration ones were considered.

Remember that in this case study, since HVAC is reduced to a system with radiators, the ventilation will be natural. Considerations about load trends were based on a daily average.

In this case, the above conditions were, respectively:

- For Ventilation: $\dot{\mathbf{V}}_{\mathbf{NAT}} = 0.3\frac{m^3}{s}$ & $T_{min,open} = 22°C$;
- For Infiltration: $\dot{\mathbf{V}}_{\mathbf{inf}} = 0.05ACH$ for all building, including basement and roof.

Figure 3.16 shows the sensible heat gain for ventilation on the ground floor, which has a trend identical to the similar case of infiltrations (where the values obtained are of the order of 10^{-3}).

Except for two days of the heating season, their value is always zero. It is also essential to specify that in the remaining part of the building, there is no sensible heat gain for ventilation and infiltration.

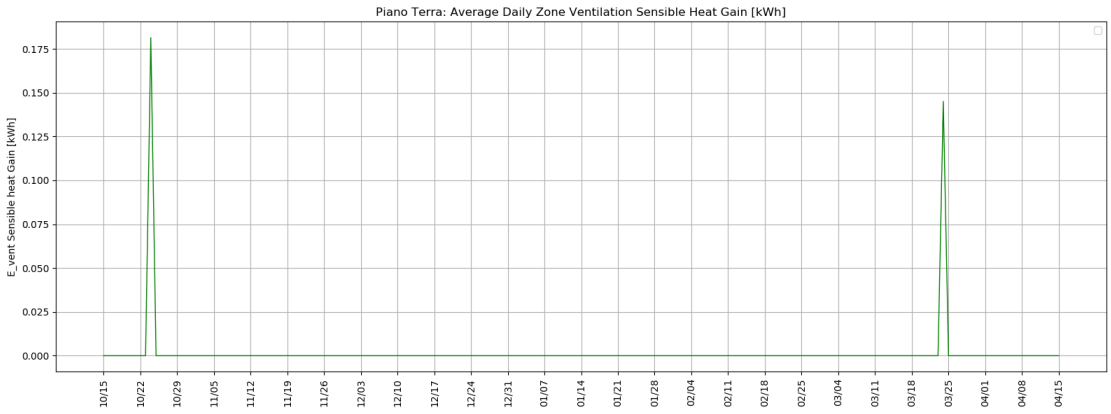

Figure 3.16. Ventilation Sensible Heat Gain for ground floor

The latent heat gain, on the other hand, shows different trends between the two cases, ventilation and infiltration. However, within each category, they are quite similar between all floors, except for infiltration, which shows differences for the ground floor and the other floors of the building (see Figure 3.17).

As far as latent infiltration heat gain is concerned, the highest values were recorded on the first and fourth floor in the final phase of the heating season.

In the case of latent ventilation heat gain, the highest values were recorded in the first three floors and during the last week of October 2018.

It is also singular to remark that the above load was eliminated in the period between November and mid-February, the central part of the heating season (Figure 3.18).

As far as infiltration heat loss is concerned, there is a similar trend between the sensible and latent cases and also almost equal between the various floors of the building. As expected, heat losses increase during the coldest season and decrease at the extremes during the mid-seasons.

Figure 3.19 exhibits the trend only for the ground floor, highlighting that the peak values reached are higher in the sensible rather than latent cases.

This analysis is closed with ventilation heat loss, which, as shown in Figure 3.20, have opposite trend compared to the infiltration case, as there is an increase during the mid-seasons and a definite decrease during the intermediate phase of

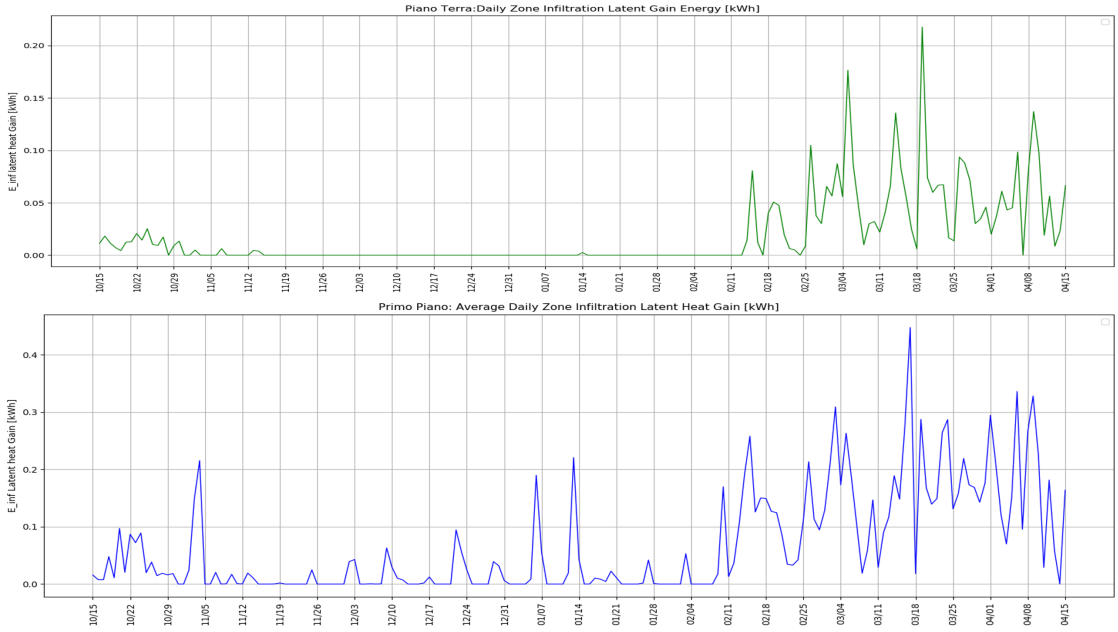

Figure 3.17. Infiltration Latent Heat Gain: comparison between ground & first floor

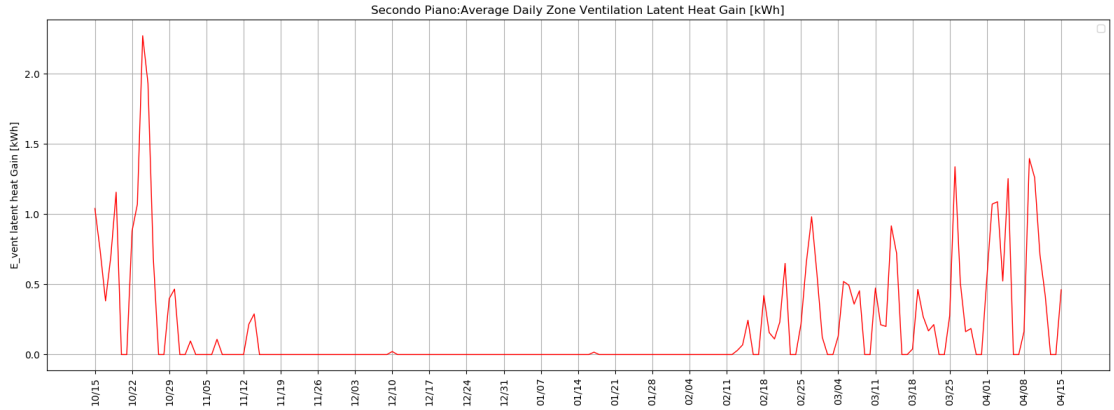

Figure 3.18. Ventilation Latent Heat Gain for second floor

the heating season.

The sensible heat loss has a very similar trend to the extremes of the period considered. At the same time, in the middle part, the peaks are more or less accentuated according to the floor considered.

The same concepts apply to latent heat losses as the previous ones, but they are about ten times smaller than the sensible ones. The highest value in absolute is recorded from the ground floor.

However, the second floor is reported in in Figure 3.21 for the particular trend in the central part because the ground floor finds in this context zero loads.

The detail shown on ventilation and infiltration loads does not present particular

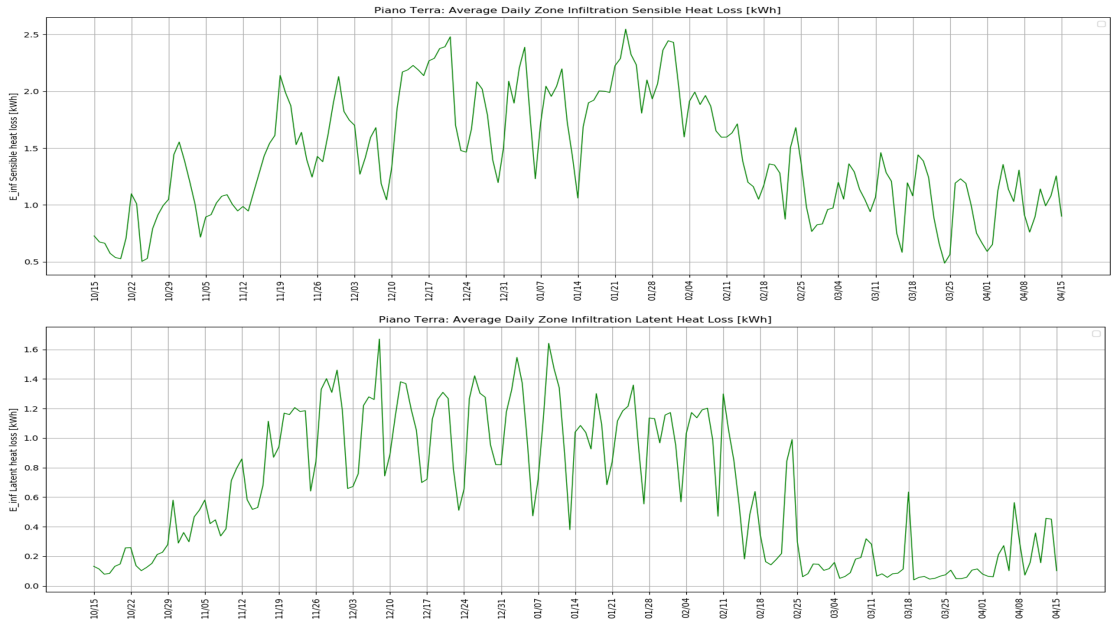

Figure 3.19. Infiltration Heat Loss: comparison between sensible & latent cases on ground floor

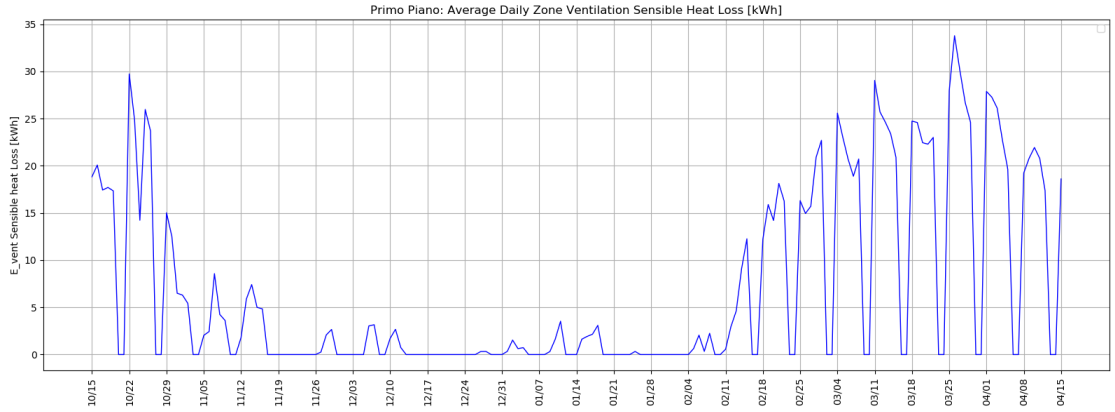

Figure 3.20. Ventilation Sensible Heat Loss on first floor

anomalies, so it was decided to conclude the analysis with a daily detail of the heat gains over two days considered significant within the heating season. It was chosen to analyze them on 24 October and 24 March.

On 24 October (see Figure 3.22) it is possible to notice that sensible heat gain energy was null, except between 3 PM and 6 PM, wherein any case peak values of little more than $0.01kWh$ were reached and in any case only for the ground floor. The other floors have zero values, except for a slight amount on the first floor.

For the latent case, it is possible to observe a different trend compared to the sensitive case: all floors had different energies from zero at night and between 2 PM and 8 PM. The trend is more similar at night compared to the afternoon one,

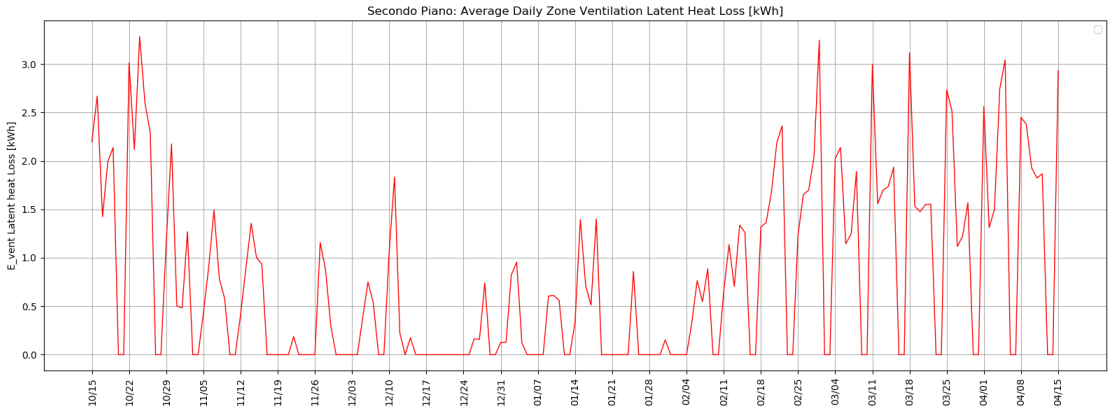

Figure 3.21. Ventilation Latent Heat Loss on second floor

where it appears slightly different, with higher values reached on the second floor.

The behaviour of sensible ventilation energy is the same as in the infiltration one. The only difference is the value reached, which is much higher than in the infiltration case.

In this case, the highest values for heat gains are reached. They are around $25kWh$ and are obtained following an upward and downward trend between 2 PM and 7 PM. The trend is almost equal between floors.

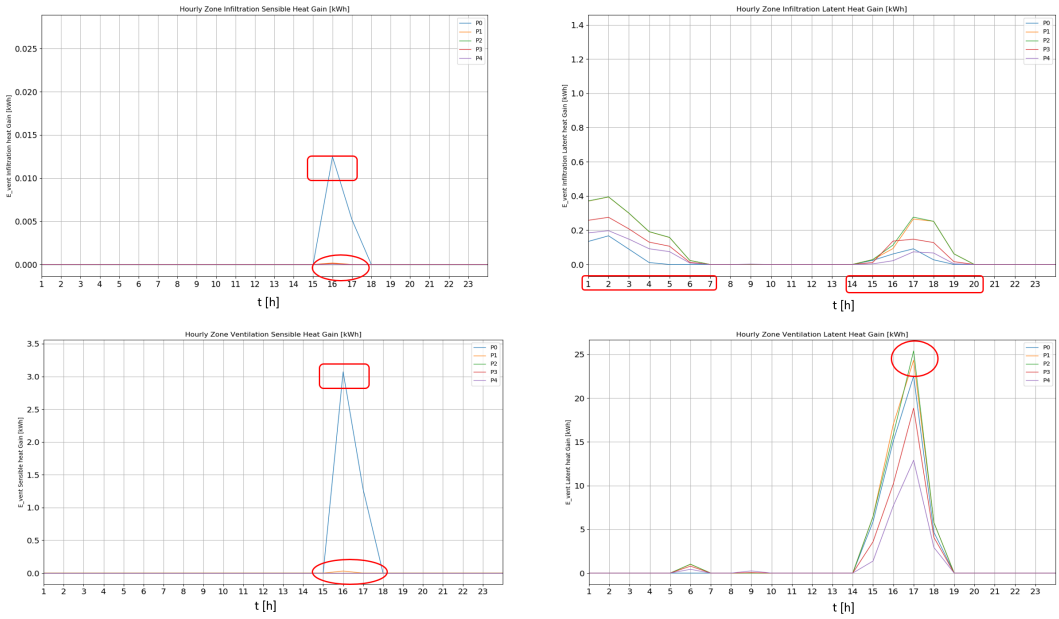

Figure 3.22. Infiltration & Ventilation Loads in 24 October

As already seen on 24 October, also on this day of March there was no Sensible Infiltration Heat Gain during the day, with a peak between 1 PM and 4 PM slightly higher than the autumn day, and only for the ground floor.

Latent infiltration leads to zero heat gain between 10 AM and 5 PM, with an identical trend for the first and second floor: they also record the highest values, this in the evening hours. The ground floor trend deviates from the others after 5 PM.

Note in Figure 3.23 that Sensible Ventilation Heat Gain graph is identical to the Infiltration case for the same day. The only difference lies in the peak value reached at 2 PM, around $3.5kWh$.

Ventilation Latent Heat Gain is zero during the day for the whole building except for the ground floor, where there is a zigzag pattern between zero values and peaks, whose maximum recorded value is slightly higher than $10kWh$ and around 7 PM.

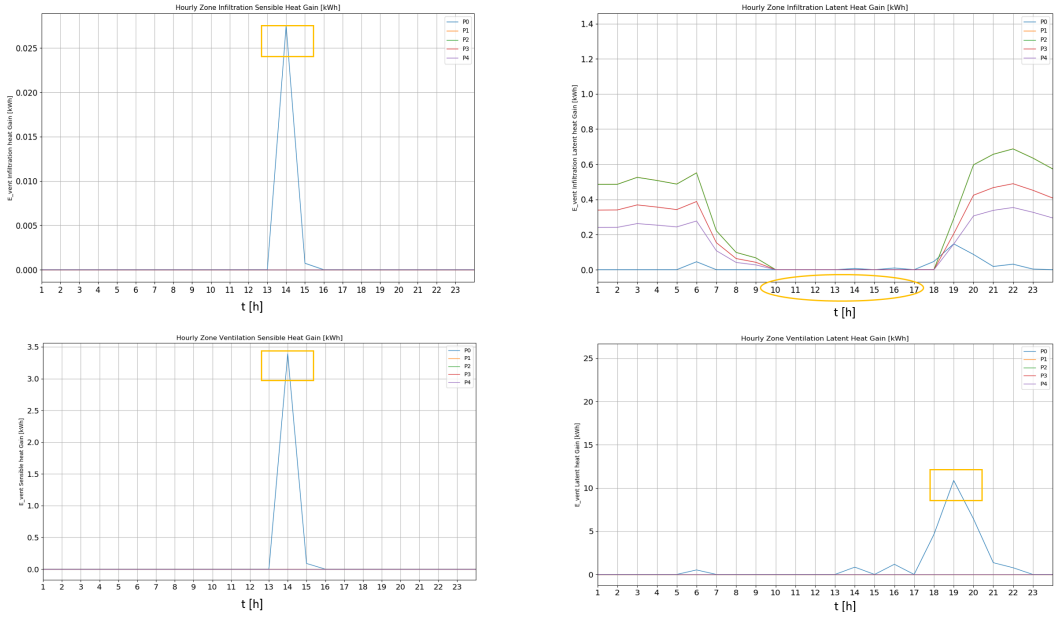

Figure 3.23. Infiltration & Ventilation Loads in 24 March

3.3.2 Second Iteration: Trial-and-error approach

From the previous section concerning the first iteration, it emerges the need to implement some model changes, because the calibration metrics could be improved (mainly for offices). Moreover, the internal temperature trends of the two main areas of our model were still quite different from the one measured in reality.

In this regard, a ***trial-and-error method*** was used which, as explained in section 2.1.1 concerning calibration theory, consists in a proceeding by trial-and-error attempts to modify the value of some variables considered as having the most considerable influence and evaluating the result obtained.

The variables must be modified according to combinations so that they are as close as possible to the real case; otherwise, the result would not be relevant.

Firstly it was modified the layout of the HVAC system, which initially consisted of a single loop with a single boiler serving the entire building. In reality, there are two loops, each with its boiler and serving the "Vigili" area and the Office area respectively: it was, therefore, necessary to insert a separate cycle for the police area, removing it from the only loop initially present.

Holidays according to the calendar were also included in heating season 2018-2019, to make the simulation even more accurate.

As evidenced by the high values of windows heat gain (especially on the fourth floor and during the spring season), it was also necessary to insert **interior shading** on the windows.

The choice is not causal, as it corresponds to the reality, as the glass surfaces of the building find these shadings rather than the exterior blinds.

For the same windows shading, a control of the type *"OnIfHighSolarOnWindow"* was chosen, active for all-day hours. The shading activation depends on the incident solar radiation value.

The trial-and-error method was used to act on the following quantities:

- Solar Radiation Power Windows shading activation;
- Infiltration Airflow rate;
- The number of occupants per each floor;
- Temperature set-point;
- Lights heat gain;
- Equipment heat gain;
- Lights radiant fraction.

The following method has led to changes in internal temperature profiles, total consumption values but especially in the values of the calibration metrics.

The Table 3.2 shows, for each quantity indicated above, the initial value, the lower and upper limit but above all the final value considered as the best.

By entering in input to the model for all quantities the respective final value, it is possible obtaining the temperature profiles for the "Vigili" and Office areas respectively in the Figure 3.24 and 3.25, compared both with the real case but also with the previous iteration.

Table 3.2. Parameters modification during Calibration optimization

Parameter	Initial Value	Lower Bound	Upper Bound	Final Value
Solar Radiation Windows shading activation $[\frac{W}{m^2}]$	300	100	350	200
Infiltration Airflow rate $[ACH]$	0.05	0.05	1.1	0.8
Occupants per each floor $[-]$	90	50	100	60
Temperature set-point $[°C]$	21	20	23	22
Lights heat gain $[\frac{W}{m^2}]$	10	5	10	7
Equipment heat gain $[\frac{W}{m^2}]$	10	5	10	6
Lights radiant fraction $[-]$	0.42	0.20	0.50	0.35

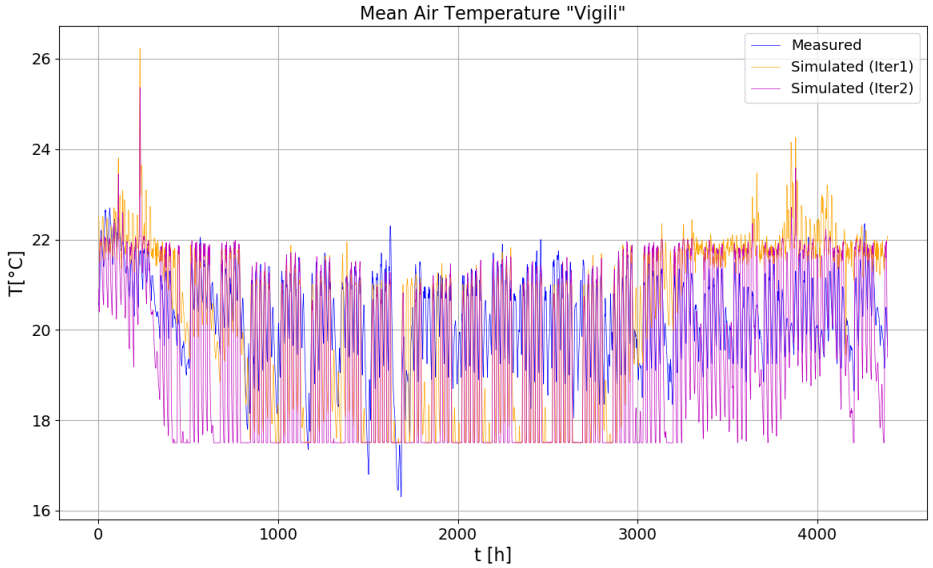

Figure 3.24. "Vigili" zone indoor Temperature trend after second iteration

It is possible to notice that the temperature profile is improved, compared to the previous step and in both areas: the result obtained appears quite satisfactory as far as the temperature is concerned.

The overall value of natural gas consumption is then shown, both in real and simulated cases at the end of the second iteration:

$$E_{gas,real} = 239361.6kWh \tag{3.5}$$

$$E_{gas,simulated} = 236824.6kWh \tag{3.6}$$

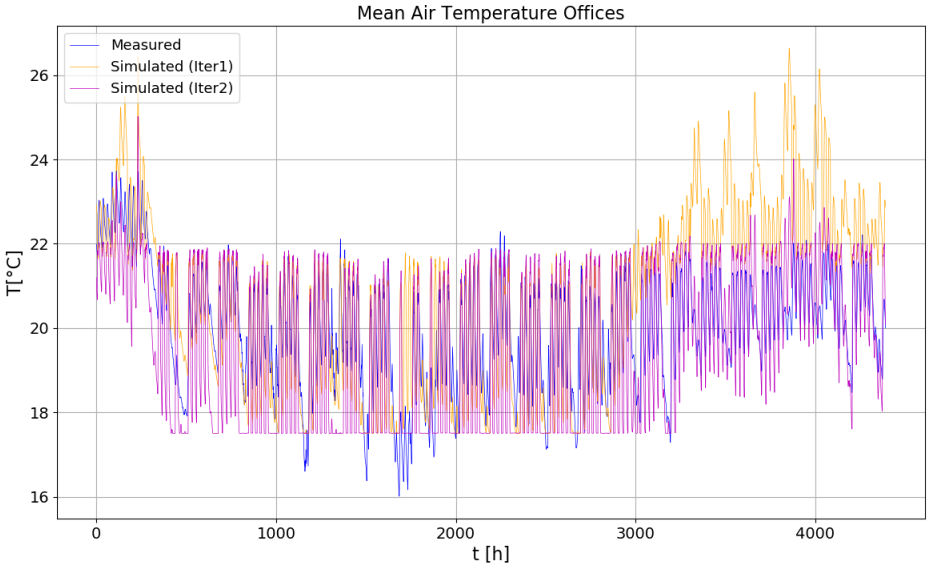

Figure 3.25. Office zone indoor Temperature trend after second iteration

Consumption has values that disagree little between the real and simulated case, so the result obtained for consumption is also satisfactory.

To conclude, the Table 3.3 shows the temperature and consumption calibration metrics values. For the latter, they are reported on an hourly, daily (for which, however, there is no indication of limits by the *[ASHRAE Guideline 14, 2002]* and monthly basis.

Table 3.3. Calibration metrics results after second iteration

Variable	$MBE[\%]$	$Cv(RMSE)[\%]$
Temperature ("Vigili")	0.15	7.09
Temperature (Office)	3.63	8.57
Consumption (Hourly)	-1.06	150.94
Consumption (Daily)	-1.06	36.02
Consumption (Monthly)	-1.06	5.62

The results obtained are within limits imposed by the *ASHRAE standard* (see Table 2.1), such that the building model can be defined well-calibrated.

In absolute value, an advantageous improvement of the values is obtained, except for the "Vigili" zone Temperature, where a slight increase for MBE and $Cv(RMSE)$ is noted.

As far as consumption is concerned, the MBE value has improved a lot considering that on the hourly scale there was a value of -85.40% for iteration 1 and -1.06% for the second one. In contrast, for hourly $Cv(RMSE)$ the value remains very high: this is however due to the different behaviour of the boiler between the real and the simulated case, even if, as it was already seen, the thermal power demand is almost equal overall. The results are within monthly limits provided by [ASHRAE Guideline 14, 2002].

From this it is possible saying that the model can be considered **well-calibrated** based on the results obtained following **iteration 2**: this will be the model used for the control implemented with algorithms based on Reinforcement Learning, as will be discussed in the next chapter of this work.

3.4 Design of DRL Control Logic

The last three chapter paragraphs describe all the SAC control agent features the setting of the two phases of training and deployment, according to the workflow provided by Figure 3.26.

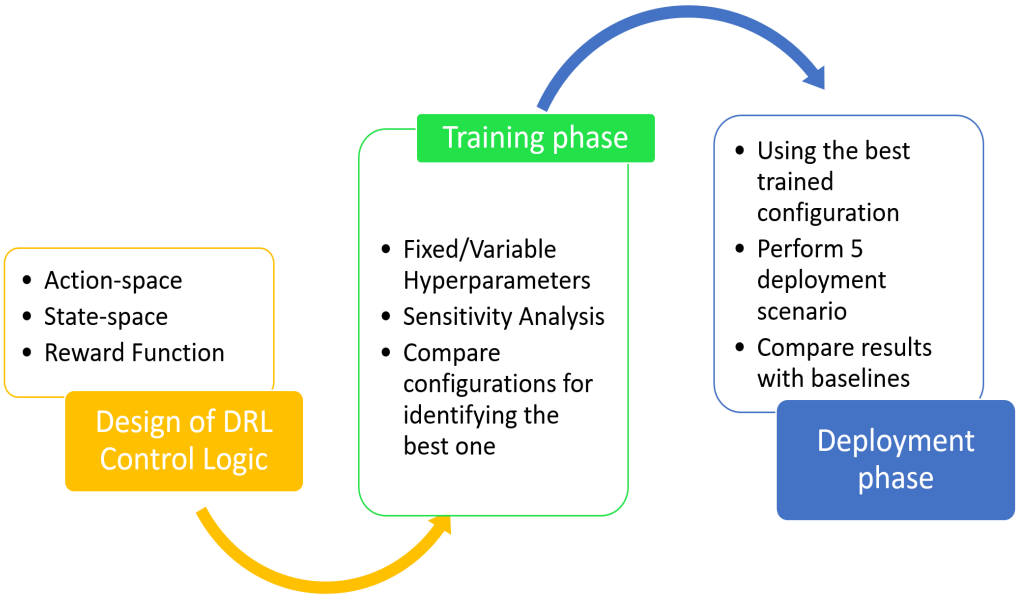

Figure 3.26. From the definition to the application of DRL control agent

Starting with the Design of the DRL-SAC control agent, it is essential defining its main features:

- action-space;

- state-space;
- reward function.

In the next subsection, these three elements are discussed.

3.4.1 Description of action-space

The action chosen by the controller belongs will be the supply water temperature to the radiators.

Compared to [Brandi et al., 2020], however, there is no discrete but continuous space, as required by the SAC algorithm: therefore, the supply water temperature will be chosen between a lower limit of $20°C$ and an upper limit of $70°C$.

So the action chosen will be:

$$A_t : 20 \leq SWT_t \leq 70 \tag{3.7}$$

The same action range was selected to match that of the baseline.

It is essential to specify that simulation environment was set in order to shut down the circulation pump if the chosen action (corresponding to supply water temperature) is equal or lower than $25°C$.

Furthermore, the system will be switched off when the occupants leave the building. This switching on and off process takes place employing the variable $BCVTB_{BOILER}$, information that is exchanged between Python and EnergyPlus through the tool described in Section 3.2.

3.4.2 Description of state-space

The state space is composed of a series of observations displayed by the agent. It is of fundamental importance because based on the values assumed, a certain action is chosen.

In this thesis work, the state-space was made of 9 features, indicated in the Table 3.4 with their lower and upper extremes.

The variables chosen are all made available in output from EnergyPlus to Python so that it is possible to provide the agent with the information necessary to evaluate the reward.

Outdoor Air Temperature and **Direct Solar Radiation** were included because are exogenous factor with the greatest influence on energy consumption but also on the indoor air temperature.

The **Indoor Air Temperature** information during the control step was passed as a difference between the desired setpoint and the same temperature because directly linked to the reward formulation 3.4.3.

Table 3.4. Variables included in the State-Space

Variable	Min Value	Max Value	Unit
Outdoor Air Temperature	-8	32	$°C$
Direct Solar Radiation	0	720	$\frac{W}{m^2}$
ΔT Indoor Setpoint - Mean indoor temperature	-3	8	$°C$
Time to Occupancy start	0	36	h
Time to Occupancy end	0	12	h
Supplied Heating Energy control step (15 min)	0	$4.5 * 10^8$	J
Supply Water Temperature	10	80	$°C$
Return Water Temperature	10	80	$°C$
Status	0	1	–

The occupants' presence was passed in the temporal form, considering the *Time to Occupancy Start/End*, because a simple binary variable of the type [0, 1] would dissipate the temporal information on the occupants' arrival or departure, useful to evaluate the pre-heating or energy savings in the final phase of occupation.

When the building is not occupied, **Time to Occupancy Start** represents the number of hours required for the arrival of the occupants, so during occupancy periods, it is set to zero. Conversely, when the building is occupied, **Time to Occupancy End** represents the number of hours that must elapse before occupants leave the building, so during periods of occupancy it is set to zero.

The **supplied heating energy** was expressed in such a form that it can be used in the reward formulation, although in this case, it will be evaluated in Joule. It is key information for the agent and is calculated as the sum of the energy spent between one control step and the next (three consecutive simulation steps).

The last three variables are linked to the system. Specifically, the first is the **system status**, ON from 1 AM to 7 PM, and OFF the complementary one.

The last two are the **Supply Water Temperature and Return one**. They are included in the observations as they are directly proportional to the value of energy spent by the boiler.

The relative humidity was not taken into account in the observations because the radiator heating system cannot control the latent part of the heating load.

Observations must be scaled within a range of [0, 1] in order to feed the neural network: to do this, a scaling process with min-max normalization is used.

3.4.3 Description of reward function

In order for the agent to learn what control policy is required, the reward function must be set up in such a way that it is representative of the problem under attention.

In this case study, the reward is expressed as a linear combination of two different parts, an energy-term and the temperature one. They are combined employing two weights (δ and β respectively) that have been made to vary in order to change their importance.

The reward is calculated for each control time step: the controller is enabled to choose an action since occupant were present to turn ON/OFF the heating system, every 15 minutes.

The energy-term refers to the daily energy consumption in the heating phase, calculated in kWh, and evaluated from the first moment the boiler could switch on. The remaining part referring to temperature corresponds to the temperature difference between the upper/lower threshold temperature and the average temperature of the offices during the building occupancy, evaluated in $°C$.

The general expression of the reward could be summarised as follows:

$$R = R_T + R_E \tag{3.8}$$

After the workers leave the building, **the reward will be completely set to zero**.

The objective of our agent will be therefore to maximize the reward, with a maximum (ideal) value of zero.

The energy-term reward is always present and expressed in the following way:

$$R_E = -\delta * E_{HEAT} \tag{3.9}$$

Temperature-term, on the other hand, has different expressions depending on the situation.

If no workers are present, the temperature-term is:

$$R_T = 0 \tag{3.10}$$

If people are present, the temperature-term could have three different expressions:

- if $T_{mean,off} < T_{LOW}$:

$$R_{T,OCC=1} = -\beta * (SP_{int} - T_{mean,off})^2 \tag{3.11}$$

- if $T_{mean,off} > T_{UPP}$:

$$R_{T,OCC=1} = -\beta * (T_{mean,off} - SP_{int})^3 \tag{3.12}$$

- if $T_{LOW} <= T_{mean,off} <= T_{UPP}$:

$$R_{T,OCC=1} = 0 \tag{3.13}$$

It is possible to observe the reward function composition graphically, as shown in Figure 3.27.

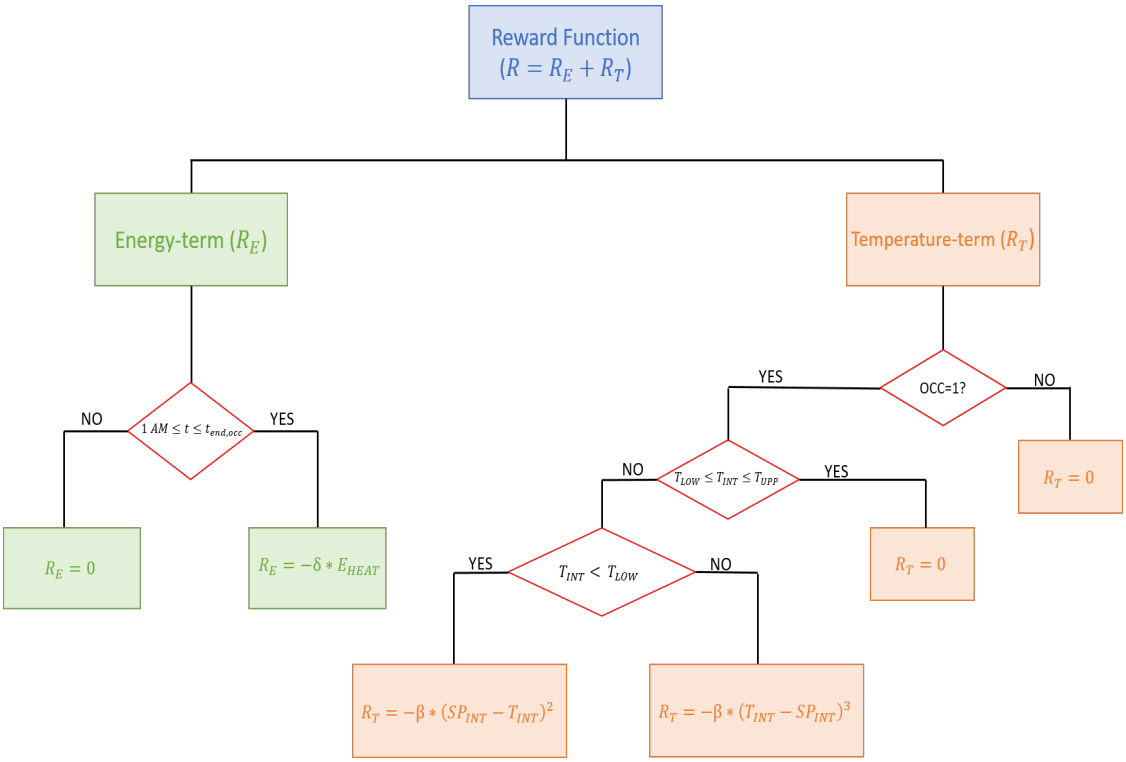

Figure 3.27. Composition of reward function and evaluation of its terms.

A similar formulation of the temperature-term was chosen to try to speed up the learning process and avoid the exploration of unacceptable states from the beginning.

The environment allows the system to reach values greater than the setpoint set by $SP_{int} = 21°C$: this is due to the presence of another variable exchanged between Python and Energyplus, called $BCVTB_{SP}$.

It assumes the following value:

$$BCVTB_{SP} = 23°C \tag{3.14}$$

3.5 Training Phase

As discussed in the initial chapter, the use of a DRL algorithm implies that several hyperparameters influence the behaviour and performance of the agent.

In order to assess their influence, a sensitivity analysis was necessary, so it was decided to try varying the set of hyperparameters and compare the results obtained. This process was carried out on the agent having the characteristics described in paragraph 3.4.

Within the set of hyperparameters, it was decided to keep some of them fixed, indicated in the Table 3.5.

Table 3.5. Fixed Hyperparameters for DRL training phase

Variable	Value
DNN Architecture	4 layers
Batch size	64 Control Steps
Episode Length	2976 Control Steps (31 days)
Training Episodes	30
Replay buffer size	50000
τ Soft update Boltzmann Temperature coefficient	0.005
Energy-term weight factor (δ)	0.01

An episode training includes the whole month of December (from 1st to 31st), for a total of 2976 control steps (every 15 minutes) and 8928 simulation steps (every 5 minutes).

Each episode was repeated 30 times for each hyperparameter configuration chosen in order to allow the agent to evaluate different control strategies, for a total of about 40 minutes (about 1 minute and a half per episode).

The weather file used in this thesis work was personally created from the necessary 2018 and 2019 meteorological data, made available by ARPA Piemonte and processed in *EPW* format using the *EnergyPlus Weather tool*. This choice took a few days for data processing and formatting. However, it allowed obtaining results closer to the present one, since the reference weather file available on the EnergyPlus website for Torino Caselle, refers to a weather situation of about 15 years ago.

The parameters involved in sensitivity analysis are different, and include:

- **neurons per hidden layer**, so that the agent can explore more or less;

- **discount factor** (γ), to determine the relative importance of future reward versus immediate reward;
- **learning rate** (α), to determine how the agent tries to overwrite old information with new information;
- **weight factor reward temperature-term** (β).

The weight factors help to define the relative importance of the two reward terms and therefore determine the agent's choices in investing greater attention to comfort or energy saving.

The different configurations of hyperparameters tested are shown in the Table 4.1 present in the next section 4.1.

Other necessary training phase specifications are the value of the internal set-point of 21°C, so the acceptability comfort conditions range will go from 20°C to 22°C.

The building was occupied during the same period as the baseline, i.e. between 7 AM and 7 PM from Monday to Saturday.

3.6 Deployment Phase

The configuration considered as the best between those explored during the training phase was chosen, then it is used during the deployment phase.

This phase is particularly important because it allows understanding the adaptability of our agent towards the change of conditions around the training phase. The control policy learnt in the previous phase will then be reused to test the agent on five different scenarios, throughout January and February and for a single episode per scenario.

The trained agent can be deployed statically or dynamically:

- in **static deployment** the agent is used as a static function. This process requires less computational time than dynamic one but the energy model should be continuously calibrated, and the control agent, in this case, should be towed back to the new model with the modified features;
- in **dynamic deployment**, the RL agent is characterized by continuous learning. In fact, for each moment of control, the agent receives the observations from our system and proposes a control action, observes the reward and the next state and proceeds to the updates in the training phase. This phase requires a high computational time, and the agent is more flexible to changes. The problem is that the control policy may have instabilities.

In this thesis work, a static deployment was performed.
The scenarios assessed were:

- **Scenario 1**: in this case, it was assessed how the agent adapts to different weather conditions (i.e. outdoor temperature and direct solar radiation). Therefore no parameters related to building physics or schedules were changed. The control environment remains the same as the training phase.

- **Scenario 2**: in this scenario, the internal setpoint value was increased to $22°C$. The tolerance band for maintaining comfort conditions always remains between [-1.1], so the lower and upper limits will also increase accordingly.

- **Scenario 3**: in this scenario, it was tested the agent's adaptability if the energy performance of the building opaque envelope is improved. No changes were made to the transparent envelope.

  Therefore, the external stratigraphy modification has been foreseen, introducing an external coat that allows for more excellent insulation and a reduction in heat loss, also improving the comfort conditions inside.

  It is a recommended solution in the case of intermittently heated rooms, as in the case of offices heated during periods of daily occupancy in which plant is turned off at night.

  The basic thermal transmittance of our building is $U = 1.08\frac{W}{m^{2\circ}C}$, well above the current standard. In fact, the U-value for vertical walls must be lower than $U = 0.26\frac{W}{m^{2\circ}C}$ for public buildings located in the Turin (climate zone E). Thanks the introduction of an external coat, it is reached a value of $U = 0.21\frac{W}{m^{2\circ}C}$.

- **Scenario 4**: this scenario is the complementary of the previous one, as the transparent envelope was modified, improving its energy performance. No changes were made to the opaque envelope.

  It was introduced a double glazing with 16mm double glazing filled by 85 % Argon. In this case, the thermal transmittance value is reduced to $U_g = 1.1\frac{W}{m^{2\circ}C}$ with solar factor $g = 0.33$, instead of $U_g = 2.7\frac{W}{m^{2\circ}C}$ and solar factor $g = 0.75$ in the base case.

- **Scenario 5**: in this case, it was decided to modify the building's occupancy schedules already present in the primary case. The agent's behaviour was assessed concerning the situation in which building is occupied every day (Monday to Sunday) from 8 AM to 6 PM.

It is essential to specify that in all the scenarios, except for the second one, the desired setpoint and the relative comfort band have remained unchanged, according to the training phase.

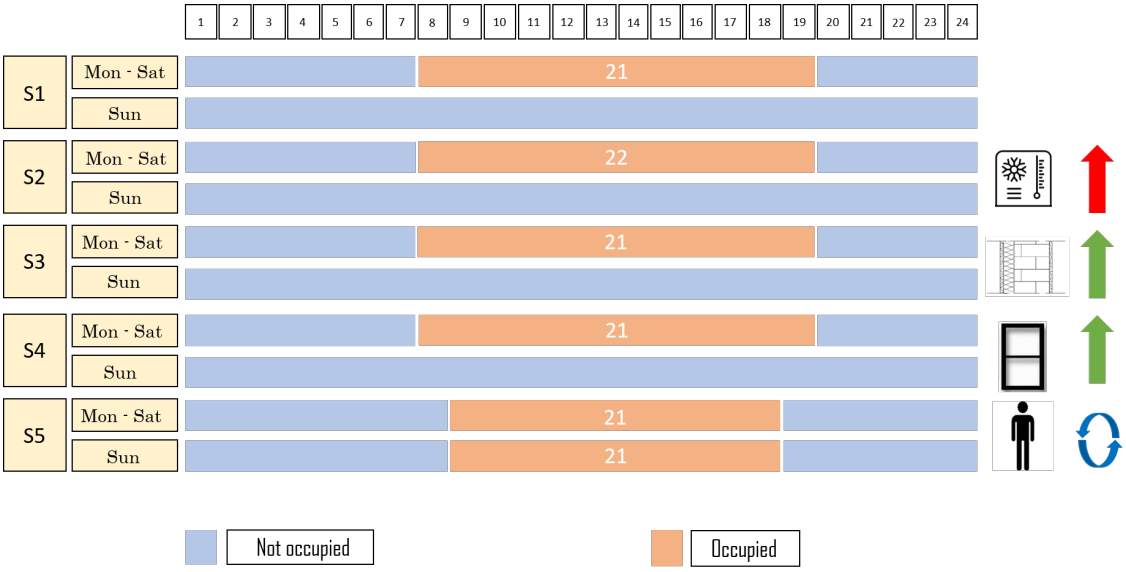

Figure 3.28. Indoor setpoint, occupancy schedules and thermal trasmittance in different deployment scenarios.

Chapter 4

Results

The DRL-SAC framework presented in the previous chapter was implemented in the simulation environment described in section 3.2.

The results obtained are exhibited to compare, both for the training phase and the subsequent deployment phase, the performance of the DRL control agent with the baseline control of the flow temperature to the radiant terminals of the heating system.

4.1 Training phase results

As shown in section 3.5, in the initial part of the training phase, it was necessary to define the hyperparameters that were kept fixed. Then, in a second step, some have been chosen as variable to evaluate their influence on the agent performance.

Specifically, in this work 24 configurations are evaluated, shown in Table 4.1.

In order to evaluate the goodness of each configuration, it was necessary to set up a comparison metric between them that was consistent with our agent's objective, i.e. to maintain comfort conditions during the occupancy period (indoor office temperature within the range [-1, 1] of the setpoint established).

At the same time, it was trying to reduce the energy consumption of the boiler by controlling the flow temperature to the radiators.

It was chosen to evaluate, from the energy point of view, the amount of energy needed for heating the water supply (both in the pre-heating phase and during the working day).

The comfort performance was evaluated choosing the calculation of the cumulative sum of temperature violations during the occupancy hours, measured in $°C$.

A temperature violation occurs when, during the presence of occupants, the temperature is not within the acceptability range [-1, 1] of the $21°C$ setpoint. It is calculated, therefore, according to the expressions:

Table 4.1. Training hyperparameters configurations for DRL agent

Configuration	Neurons per hidden layer	γ	α	β
1	256	0.9	0.001	1
2	256	0.9	0.001	5
3	256	0.9	0.001	10
4	256	0.95	0.001	1
5	256	0.95	0.001	5
6	256	0.95	0.001	10
7	256	0.99	0.001	1
8	256	0.99	0.001	5
9	256	0.99	0.001	10
10	128	0.9	0.001	5
11	128	0.95	0.001	5
12	128	0.99	0.001	5
13	64	0.9	0.0005	1
14	64	0.95	0.0005	1
15	64	0.99	0.0005	1
16	256	0.9	0.0005	1
17	256	0.9	0.0005	5
18	256	0.9	0.0005	10
19	256	0.95	0.0005	1
20	256	0.95	0.0005	5
21	256	0.95	0.0005	10
22	256	0.99	0.0005	1
23	256	0.99	0.0005	5
24	256	0.99	0.0005	10

- if $T_{INT} < T_{LOW}$:

$$T_{violation,i} = T_{LOW} - T_{INT} \tag{4.1}$$

- if $T_{INT} > T_{UPP}$:

$$T_{violation,i} = T_{INT} - T_{UPP} \tag{4.2}$$

Both terms were evaluated for each **simulation step** and then cumulated with an algebraic sum at the end of each episode.

The period between office closing time and the time when the system could be switched on the following day (from 7 PM to 1 AM) was overlooked.

These parameters were calculated also for the control baseline in order to compare the results obtained for the various trained agents.

The graphical results representation had fundamental importance in excluding some configurations and selecting others for a more detailed sensitivity analysis.

The Figure 4.1 shows, for the various configurations, the cumulative sum of temperature violation values for the last episode training (30^{th}) as a function of the percentage energy saving compared to the baseline.

In order to make the results more comprehensible, the y-axis was set on a logarithmic scale.

The graphic legend corresponds to the order of the configurations shown in table 4.1.

The graph is divided into four quadrants, whose axes are the performance values obtained from the baseline for overall energy consumption and the cumulative sum of temperature violations.

Each quadrant expresses a different condition, and for this work proposes, it is essential understanding in which quadrant the acceptable results fall.

Specifically, the common objective of maintaining comfort and energy-saving concerning the baseline is achieved by all the agents with configuration falling in the bottom-left quadrant.

The worst case is represented in the upper-right quadrant, where the two objectives would be not respected: in this work, however, no configuration fall in this situation.

The upper-left and bottom-right quadrants, respect only one of the two conditions, respectively only energy-saving or thermal comfort.

For the choice of the best configuration, any configuration with higher temperature violations than the baseline was also excluded, so the sensitivity analysis focused on the agents present in the third and fourth quadrant.

The Figure 4.1 highlights that the agents in the third quadrant do not have significant differences depending on the number of neurons per hidden layer, therefore, to avoid that the computational time is excessively high without having enormous benefits in respect to the other configurations, it has been chosen to consider the three agents with a number of neurons per hidden layer equal to 128, corresponding to the three configurations reported in the Table 4.2.

Configuration 10 and 11 manage to meet both requirements, while configuration 12 achieves excellent results in terms of comfort.

These three configurations are all characterized by:

- learning rate $\alpha = 0.001$;

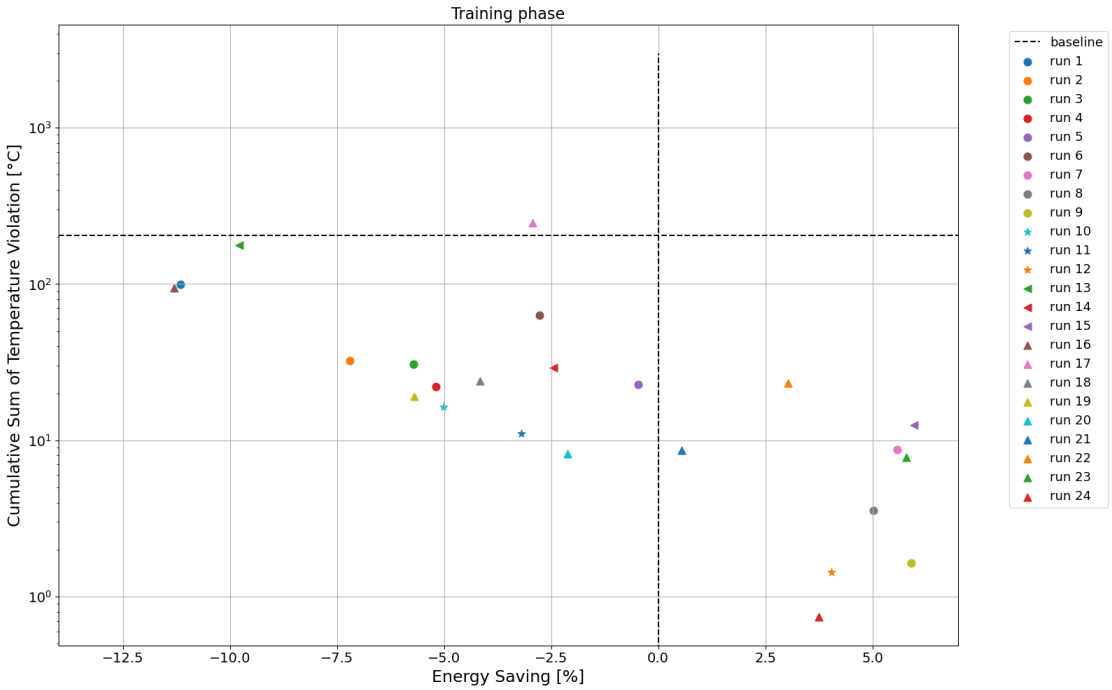

Figure 4.1. SAC control performance in 30th training phase episode of the training phase

- 128 neurons per hidden layer;
- temperature-term reward weight $\beta = 5$.

First of all, it was evaluated the learning process goodness of these three DRL agents. Then, it was studied the cumulative reward evolution, evaluating the comfort-term and energy one separately, as proposed in Figures 4.2.

The reward does not have a direct physical meaning: it gives indications regarding the control policy convergence of the proposed agents. A non-convergent trend could cause instability of the optimal control policy.

Figure 4.2 has two mainframes, the one above shows the trend of the comfort-term for cumulative reward, while the one below the energy-term. For each principal block, the three configurations to be analyzed, with their respective discount factors, are shown. For better legibility, the solid blue line represents the evolution of the term comfort, while the red one represents the evolution of the term energy. In both analyzed configurations and for both terms, the agent starts the exploration with relatively high values of the two terms of the reward.

During the training phase, the agent with discount factor $\gamma = 0.99$ can maximize the reward comfort-term more than all the other configurations, getting very close to its cancellation during the last episode. All this also confirms what observed in Figure 4.1, i.e. the almost absence of violations for configuration 12 during the 30^{th}

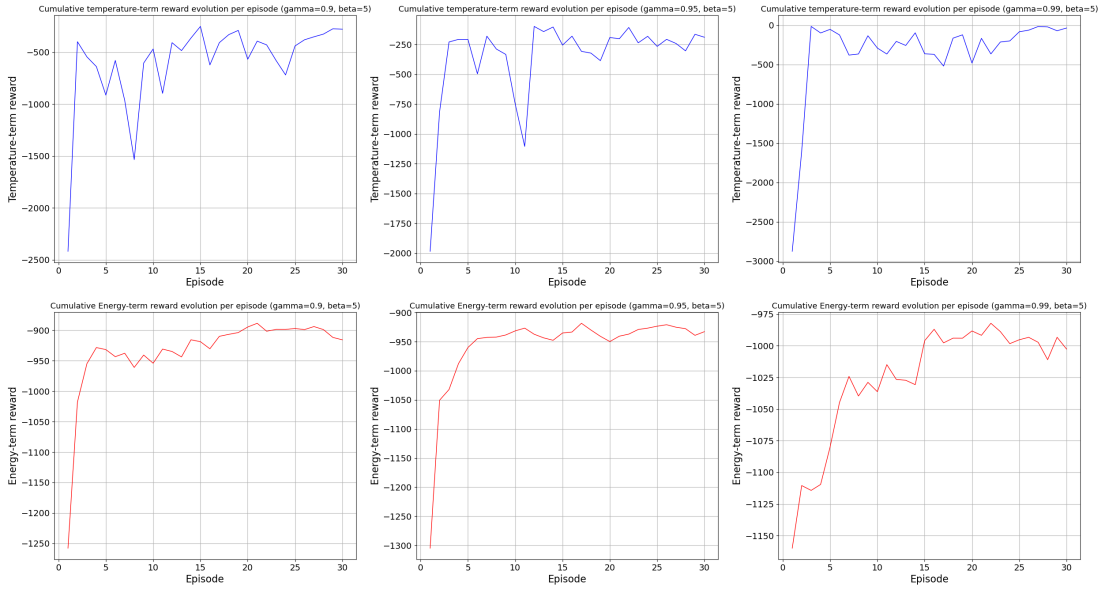

Figure 4.2. Comparison of cumulative reward energy-term and temperature-term between three agents with different discount factor during the training phase.

episode. On the whole, however, the result achieved by this last configuration was not the best, as observing the energy-term trend shows an optimal non-convergence even in the last part of its training phase.

For the other two agents, on the other hand, after about 20 episodes, a control policy converging towards a stable value seems to be established. The same trend can be observed only during the very last episodes for the cumulative reward temperature-term, more in the configuration with $\gamma = 0.9$ than in the $\gamma = 0.95$ case. On the whole, therefore, it could be said that the agent with the most excellent stability is the one proposed in configuration 10.

It would be wrong, however, to stop here the analysis and already choose this configuration as the optimal one, because the reward value alone is not an evaluation metric for assessing the goodness of the overall DRL agent performance.

Therefore, it was decided to graphically represent the results obtained at the 30^{th} episode for each of them in terms of energy consumption, indoor temperature profile and flow temperature profile. The figure 4.3 compares, daily, the three agents that differ only in the value of discount factor γ: the day chosen corresponds to one between the coldest of the training period, 14 December.

Overall, all three agents meet during the entire occupation period the necessary comfort requirements, with the utilising agent $\gamma = 0.9$ keeping the temperature close to the lower acceptability limit, without ever falling below between 7 AM and 7 PM. Then this agent could be able to use the fewest energy quantity between the three agents considered, $3.7MWh$, about $100kWh$ and $500kWh$ lower than the

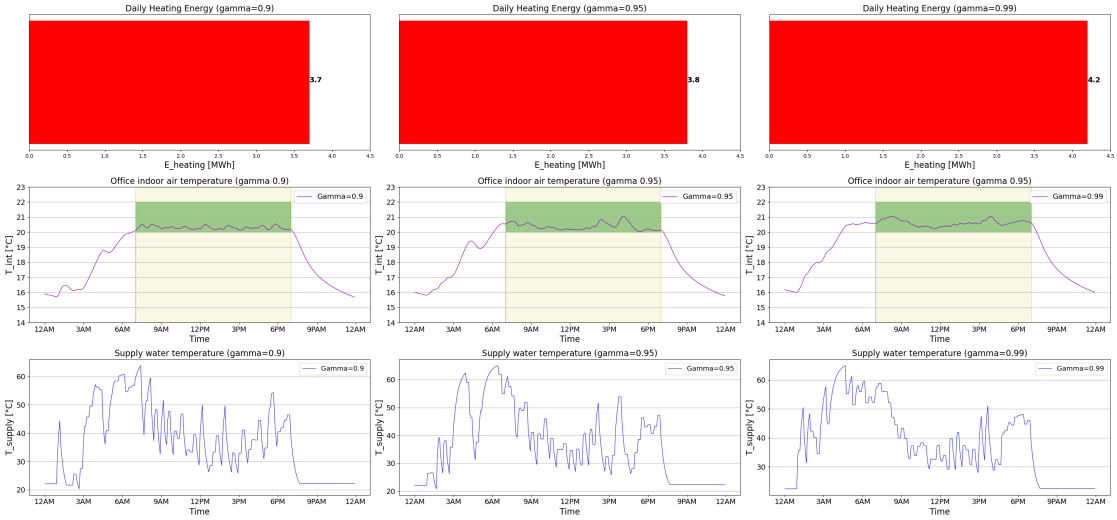

Figure 4.3. Comparison between three agents with different discount factor during the training phase.

other two.

Especially the agent with $\gamma = 0.99$ is found to consume the most significant amount of energy in the world, as also demonstrated by the internal temperature profile, which is placed around the central range of acceptability comfort more than the other two agents.

The supply water temperature trend is very similar during the worker occupancy phase for agents with $\gamma = 0.9$ and $\gamma = 0.95$. However, all three agents have their peak water supply temperature before the arrival of the occupants, so that the temperature at their arrival is greater than the lower limit.

This peak is earlier for the agent with $\gamma = 0.99$, and this explains why already about two hours before the arrival of the occupants the internal temperature falls within the range of acceptability without it being necessary, thus consuming a greater amount of energy.

In order to choose the best configuration, the comparison of the three agents based on the entire training month for the last episode (30^{th}) is shown in table 4.2.

The results obtained suggest the exclusion of the agent with configuration 12 because, although it has very low-temperature violations during the training phase, it consumes more energy than the baseline, about 4% more.

For two remaining agents, 10 and 11, it is observed that both have temperature violations of the same order of magnitude, with the agent with $\gamma = 0.95$ in slight advantage ($11.07°C$ against $16.28°C$ of the other).

This difference is, however, less marked compared to the one from the energy point of view, wherewith configuration 10, it is possible obtaining an energy saving of $1.8MWh$ more.

It results that, although they have an advantage on each side, the one trained with the **tenth configuration** is chosen as the best agent.

4.2 Deployment phase results

In this section, the deployment phase results are analysed, where the adaptability of the control agent chosen in the training phase (configuration 10) is tested in the five scenarios proposed in section 3.6.

Table 4.3. SAC control agent features for the deployment phase

Features	Value
DNN Architecture	4 layers
Neurons per hidden Layer	128
Discount factor γ	0.9
Learning rate α	0.001
Number of episode	1
Episode Length	5664 Control steps (59 days)
Energy-term weight factor δ	0.01
Temperature-term weight factor β	5

In this thesis work, a **static deployment** is used, and each simulation will consist of only one episode for each scenario. The period of the simulation will extend to two months, from 1^{st} January to 28^{th} February. Consequently, the meteorological data will change, which will always be taken from the weather file created for Turin for this work.

The figure 4.4 and the figure 4.5 provides an overview of the results obtained for all scenarios for energy consumption and cumulative sum of temperature violations for the whole episode, respectively, coloured blue. In both figures, there are also baseline values for each scenario, shown in red.

The DRL agent in the S2 scenario allows achieving significant energy savings compared to the baseline (in the range of 20%). However, above all, it highlights its behaviour from the comfort point of view: it presents the highest value of comfort band temperature violations among deployment scenarios, albeit smaller than its baseline (around $40°C$).

On the contrary, the baseline finds a considerable increase in energy consumption but also in temperature violations, especially in the initial occupation phase. Then, it is clear that our SAC agent has greater adaptability towards an increase

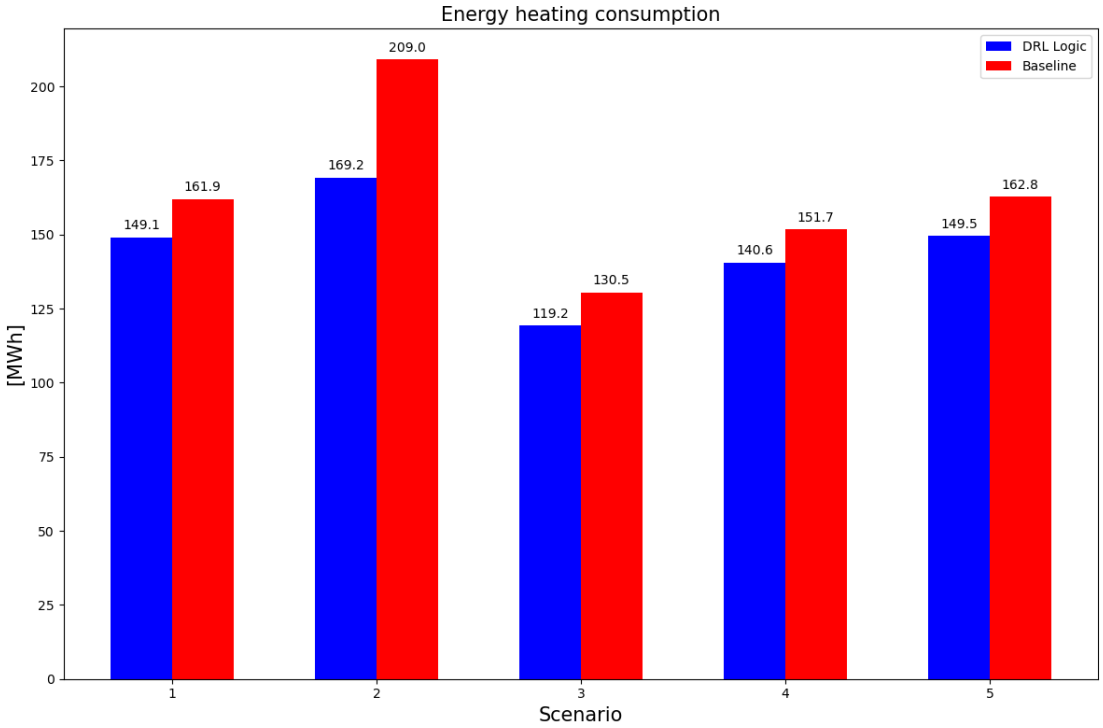

Figure 4.4. Energy consumption per each scenario in deployment phase, comparing DRL and Baseline control logic.

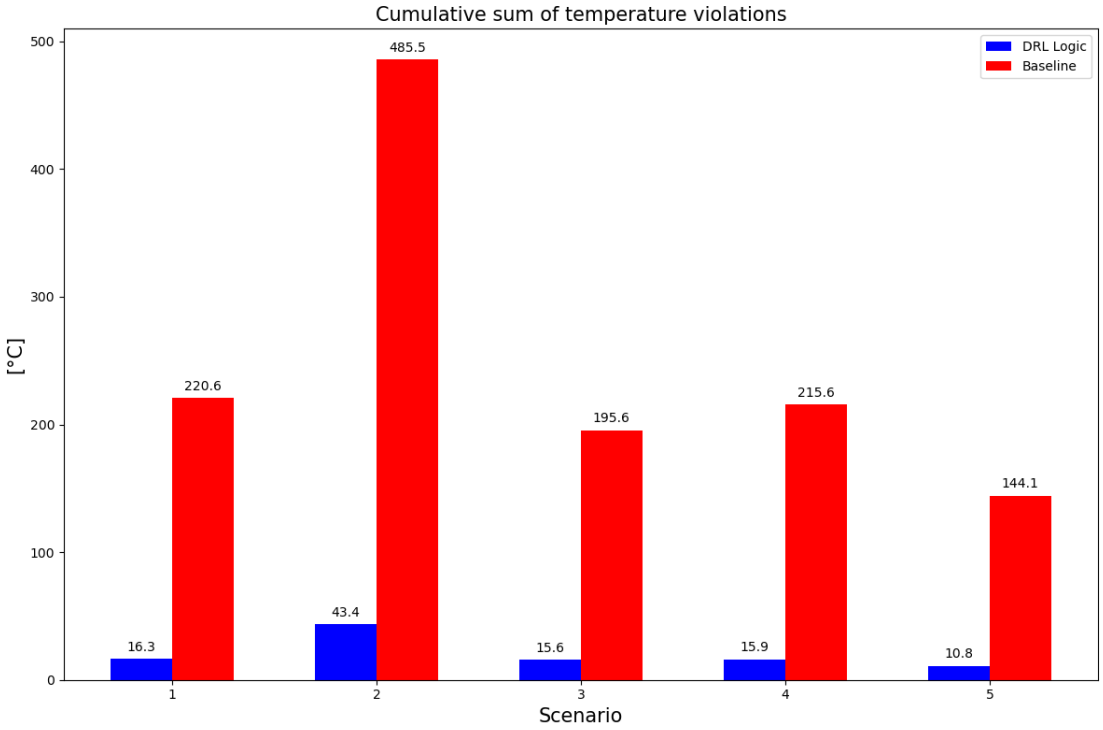

Figure 4.5. Cumulative sum of temperature violations per each scenario in deployment phase, comparing DRL and Baseline control logic.

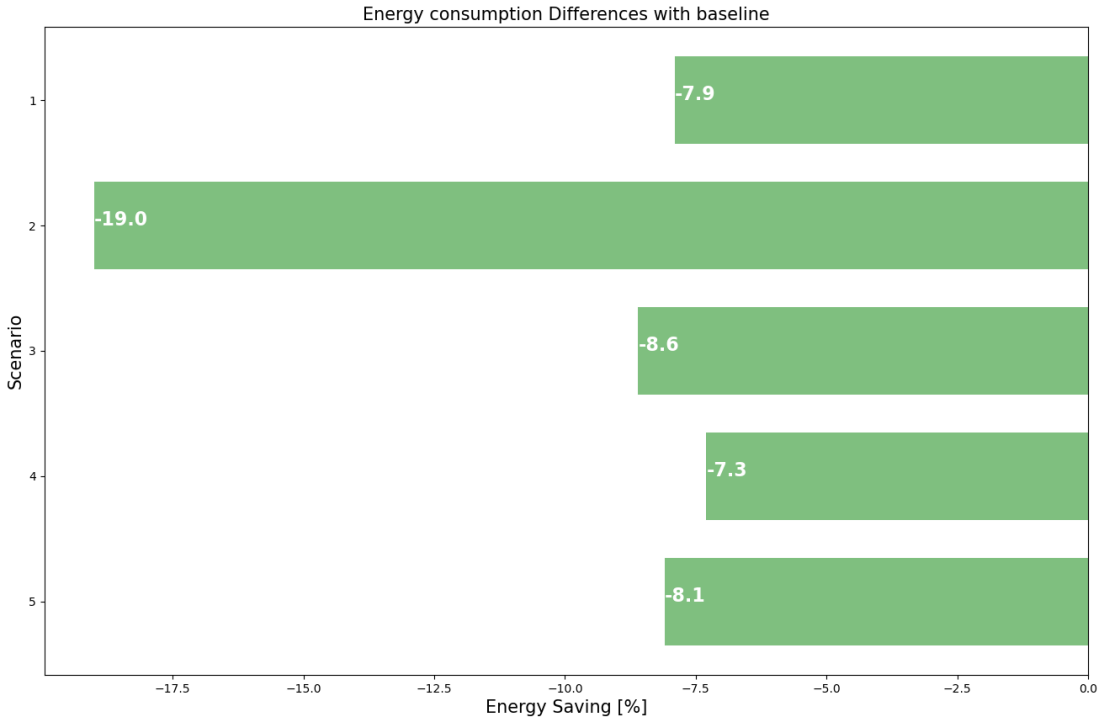

Figure 4.6. Energy savings per each deployment scenario.

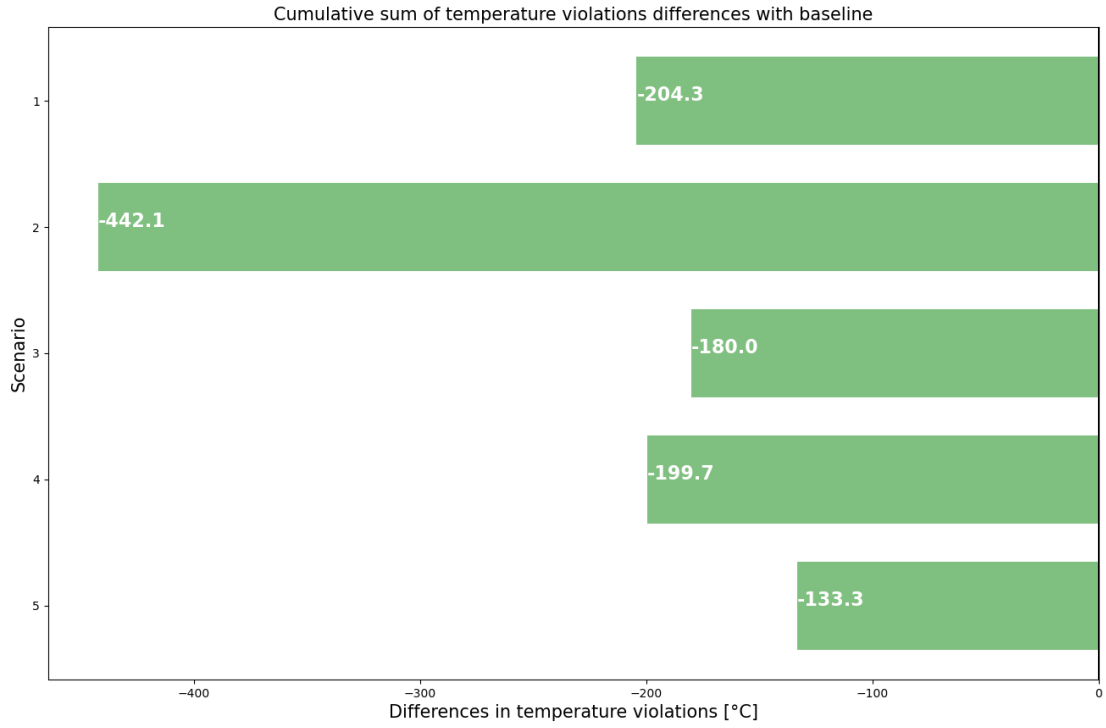

Figure 4.7. Differences in cumulative sum of temperature violations per each deployment scenario.

in the setpoint value than the baseline, and therefore an excellent control policy able to adapt to this change.

The remaining four scenarios all have energy savings of between 7 and 9 % compared to their baseline counterpart, with overall values varying depending on the scenario.

Scenario S1, i.e. the scenario where only the period is changed, finds an overall energy consumption that is lower than scenario S3 and scenario S4 consumption, as insulation is increased in these scenarios, and therefore a similar result is achieved for S3 (-8.6%) and S4 (-7.3%). The S5 scenario has almost the same energy savings as the S1 scenario (8.1%): despite the inclusion of Sunday as the day of occupation, remember that it is reduced by two hours compared to the S1 scenario, for each day, the occupancy band, so this makes the consumption is almost similar.

In the comfort field, on the other hand, there are values of temperature violations in line with what was already seen in the agent training phase. Values are between 10 and 16 $°C$.

The other two figures, 4.6 and 4.7, show an overview of savings and differences for each scenario between the deployment and baseline. Energy savings are shown as percentage savings in the deployment phase compared to the baseline counterpart, while temperature violations are assessed as a simple difference between the two cases.

In this situation, the colour of the bar chart is always green to indicate that, compared to the baseline, the DRL control logic is in an advantageous condition and therefore brings the expected benefits.

The following overview of images is used to give more details on the results for each deployment scenario.

Figure 4.8 shows the comparison between the deployed control agent and the control baseline in the S1 scenario, during a typical week of the period analysed. The internal temperature trend and the supply water temperature profile are compared in light of the external temperature profile.

The DRL control logic allows to reduce temperature violations during the first day of the week through a better managing of the pre-heating phase and, at the same time, ensures that the temperature trend remains almost stable in a narrower temperature band near the lower limit of the acceptability range.

During the weekly band that goes between the third and sixth day, when the outdoor temperature has lower values during the week, the energy savings compared to baseline are considerable. It is due to the attitude of the latter control logic to keep the system ON until the upper limit is reached, and then turn it off and find a new restart when it drops below the lower temperature threshold. It is, therefore, demonstrated how the SAC control logic allows adaptation to exogenous factors.

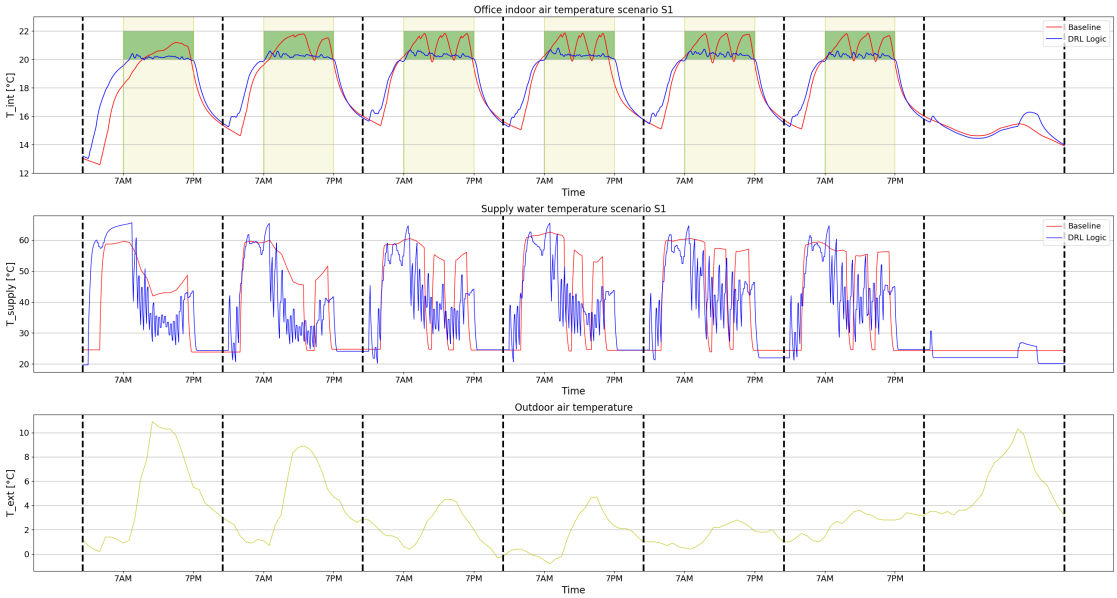

Figure 4.8. Comparison between SAC control deployed agent and baseline controller during a deployment period week in Scenario S1.

Figure 4.9 shift focus on the internal temperature profile for offices for scenario S2, during the same typical week considered in the scenario S1. Compared to the latter, the difference consists of the desired setpoint value within the office zone, which is increased by $1°C$ ($22°C$). For this changing, the comfort conditions acceptability range shift their limits to the other by the same amount.

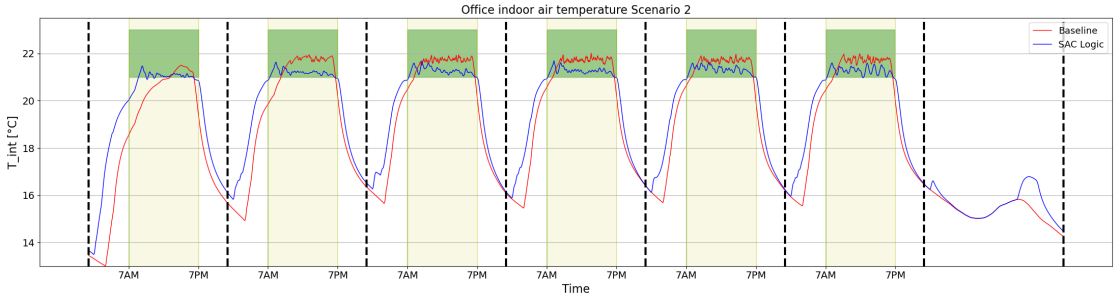

Figure 4.9. Comparison between SAC control deployed agent and baseline controller during a deployment period week in Scenario S2.

The first substantial difference from the previous scenario is that the baseline never manages to reach the upper limit of the comfort band, maintaining a temperature profile in the middle of the day more similar to that of the SAC control agent.

This detailed study allows having the motivation for which in Figure 4.5 it is perceived a significant increase in temperature violations for the baseline. Never

during the week, the baseline control manages to ensure that the internal temperature at the arrival of the occupants is at least equal to the lower limit of the comfort band, even if it manages to reduce this gap in the last days of the week. This mismanagement of the pre-heating phase also determines that the energy consumption is high because, in order to reach the desired temperature by 7 AM, a late ignition causes a massive amount of energy to be required over a limited time.

On the contrary, the tested DRL logic, by managing the pre-heating phase at best, allows to reduces the temperature violations, then maintaining a constant temperature profile in order to optimise the energy consumption.

There is also an abnormal increase in temperature on Sunday, no-occupancy day. In this work, it was decided to set the control environment so that the agent could learn by himself not having to turn on on Sundays. At certain times of this day may be sudden ignitions of the plant due to the behaviour of the agent, rightly penalised through the term of consumption in the reward.

Figure 4.10 compares the internal temperature and water temperature supply profiles over the same typical week considered for S1, but in this case for the couple S3 and S4 scenarios.

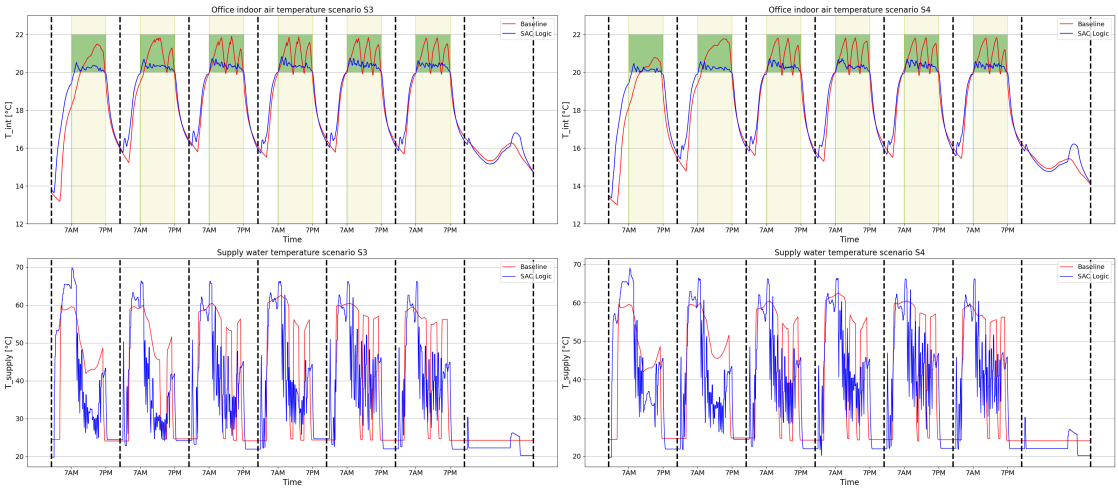

Figure 4.10. Comparison between SAC control deployed agent and baseline controller during a deployment period week in Scenario S3 and S4.

Both scenarios show a very similar indoor temperature profile, especially during the occupancy phase, where comfort range violations are almost nonexistent. It could also explain the comparable value of energy savings compared to baselines and the values of temperature violations for each scenario that differ by just over $0.3°C$.

The supply water temperature profile is similar during this typical week, with the peaks that are reached around 7 AM, with slightly higher values for scenario depending on the specific day considered.

The results just discussed highlight the need to compare more the S1 scenario with the two S3 and S4 but separately, so that it is better to assess the differences of the chosen DRL agent concerning the improvements of the opaque and transparent envelope. The comparisons are shown respectively in Figure 4.11 and Figure 4.12.

The attitude of the DRL agent in the different S1, S3 and S4 scenarios is compared based on the indoor temperature profile.

Comparing in Figure 4.11 S1 and S3, it can be stated that in the occupancy phase the agent's behaviour is similar, even if in the last days of the week considered there are more significant differences, such as a slightly earlier system shutdown in scenario S3.

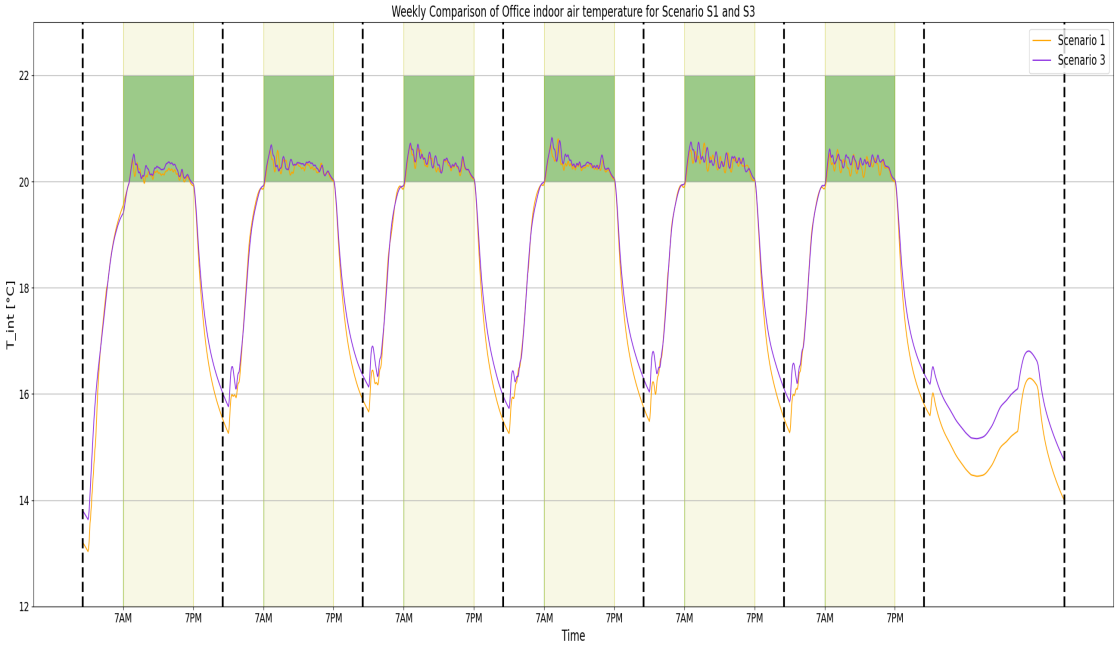

Figure 4.11. Comparison between SAC control deployed agents in scenario S1 and S3 during a typical week in deployment phase.

In the pre-heating and post-shutdown phase of the system, the indoor temperature profile of the S1 scenario appears as shifted upwards to give rise to the S3 case.

After discharge phase of heat accumulated by vertical walls to the outside, when the system was switched on again for the pre-heating phase, it finds a building about half a degree Celsius warmer in scenario S3 than in scenario S1. It is demonstrated by the phenomenon of exploiting the thermal inertia of the building, as in the case where there is an external coat (S3) the building discharges more slowly than in the case where the building does not have one (S1).

Therefore, with the same switch-on time in the two scenarios, the system will require a lower energy amount in the S3 scenario compared to S1 near about 20 %,

as shown in Figure 4.4 considering the whole deployment period. This same phenomenon makes it possible to reduce, but very little, the violations of the comfort band by switching from the real opaque envelope to a more performing one.

Similar considerations can be made in the comparison between scenarios S1 and S4, even if in this situation the differences related to the greater exploitation of the thermal inertia of the building envelope are little, as demonstrated by observing in Figures 4.12 the phase of system ignition after the building envelope discharge.

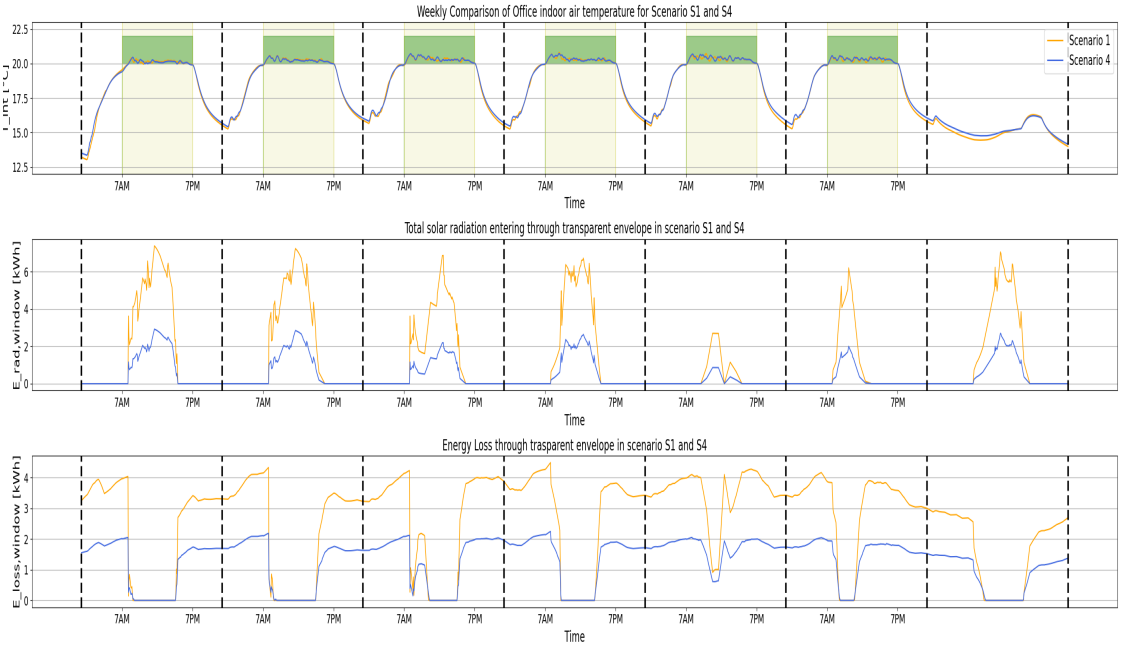

Figure 4.12. Comparison between SAC control deployed agents in scenario S1 and S4 during a typical week in deployment phase.

It is because in scenario S4 the transparent envelope was simply modified compared to scenario S1 using glazing with a much lower U-value but at the same time seeing a reduction in the solar thermal transmittance of the glass to $g = 0.33$ against the value of $g = 0.75$ in case S1. For this reason, it was also decided to plot the solar radiation entering the office zone through the glazing to show how it is affected by the reduction of the solar factor.

As expected, a g-factor reduction leads to a reduction in entering solar radiation, which in the winter case represents an heat gain to reduce the thermal load required by the system. As shown in Figure 4.12, passing from scenario S1 to S4, it is remarked a reduction in energy linked to incoming solar radiation by a factor approximately equal to that of the g-factor reduction, so that in S4 scenario the above profile appears as in case S1 but scaled down to lower values.

At the same time, however, the reduction of the thermal transmittance value U_g in the S4 scenario leads to a reduction in heat loss, as shown in the bottom part of

the Figure 4.12. In particular, this represents a much greater share compared to the value of the magnitude discussed above. During the typical week considered, the windows energy loss in the S4 scenario, $E_{loss,wind} = 2.62MWh$ is reduced to about half of the value recorded in the S1 case $E_{loss,wind} = 5.21MWh$. It represents a more significant energy saving than the reduction of internal solar radiation contributions through the glazing, linked to the reduction of the g-factor (from $2.56MWh$ in the S1 scenario to $0.91MWh$ in S4.

Therefore causes energy consumption to be reduced by 5.7% compared to the S1 scenario considering the entire deployment period.

The differences between the two scenarios are almost nonexistent if the violations of the temperature range set between $[20, 22]°C$ are considered, as already shown in Figure 4.5 where the difference is only $0.4°C$ considering the two months of the deployment phase.

To conclude the detailed analysis of deployment phase results discussed in this section, the indoor office temperature profile for the S5 scenario is analysed over four Sundays of four consecutive deployment period weeks. In this case, the behaviour of the two control logics DRL and baseline will be compared again, also considering the external temperature profile, as reported in Figures 4.13.

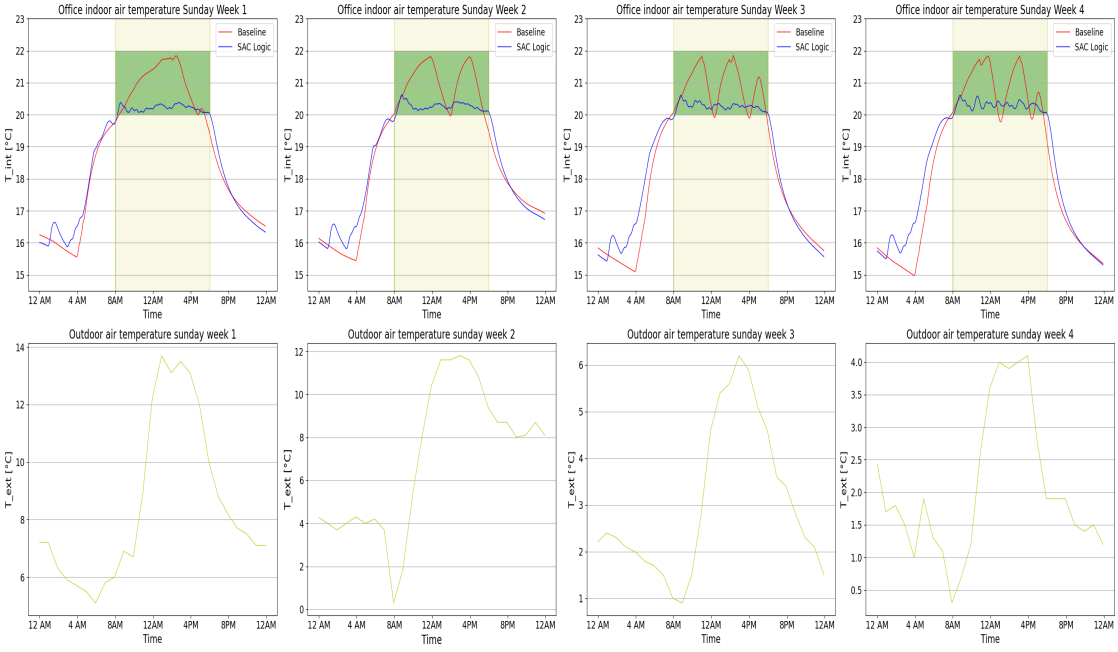

Figure 4.13. Comparison between SAC control deployed agent and baseline controller during 4 different sundays in Scenario S5.

The plot shows that in the first two Sundays, the DRL control is in comfort penalty due to a slight time delay in reaching the lower limit of the temperature range. For the other two Sundays, occurring a reduction of two or three times in

the external temperature value respectively, the controller manages to cancel the violations in the initial phase and to maintain a fairly stable internal temperature profile, contrary to what happens in the control baseline. In this latter case, the alternating ON/OFF phases lead to double wrong behaviour, both for the more significant amount of energy spent but also for the temperature violations given the exceeding of the lower limit of the comfort band during the occupancy phases.

Table 4.2. Performance comparison at the end of training for three agent employing different discount factor γ.

Configuration	SAC Control Logic		Climate-based Baseline Logic		Energy Saving [%]	Differences in Temperature violation [$°C$]
	Consumption [MWh]	Temperature Cumulative [$°C$]	Consumption [MWh]	Temperature Cumulative [$°C$]		
10 ($\gamma = 0.9$)	91.5	16.28			-5.0	-189.19
11 ($\gamma = 0.95$)	93.3	11.07	96.4	205.47	-3.2	-194.40
12 ($\gamma = 0.99$)	100.3	1.43			+4.0	-204.04

Chapter 5

Conclusion and future work

This work is focused on the development of an adaptive model-free control strategy for a real radiant system, whose current baseline is of the climatic-based type.

The developed controller was trained and deployed in a simulation environment composed by EnergyPlus and Python.

This case study has particular importance because most of the buildings in Italy have heating systems of this type and could therefore be advantageous in terms of increasing their efficiency, without considering the possibility of modifying the system with new technologies, such as the implementation of RES, which involves considerable costs in some situations.

The goal of this work was to deploy a control agent that could maintain comfort during the occupied period and reduce the boiler energy consumption. Then, the controller try to identify the trade-off between these two contrasting function, in order to improve both of them.

Before the implementation of the adaptive controller, however, it was necessary to calibrate the energy model of the building through a trial-and-error approach. The calibration was therefore carried out on an hourly basis for the internal temperature and monthly for the energy consumption of natural gas and at the end of this process, it is possible to consider the model as well-calibrated.

The energy model coming out from the calibration stage was then used for the implementation of the DRL control logic. It is essential to specify that although the calibration is performed for the entire energy model, the new control logic was only tested in the office area to simplify the problem.

It was decided to use an algorithm based on a continuous action-space, so it is employed the Soft Actor-Critic, which respects the prerogative required.

The DRL framework requires, however, that first of all, the agent features were defined, such as the action space, the observations space and the reward function. This last item is a fundamental component for obtaining a correct control policy of our DRL agent and therefore, representative of our objective, which is to maintain

adequate conditions of comfort and at the same time try to save energy.

The action-space consists of all the supply water temperature values and the state-space consists of 9 features, that are chosen in order to provide accurate information to the agent for the calculation of reward function.

The latter consists of two terms, one related to comfort and one related to the energy aspect. The sum of these two terms is weighed using two terms representing the weights of temperature and energy, appropriately chosen during the sensitivity analysis.

The values assumed by the hyperparameters greatly influence the performance of the DRL control algorithm. However, the SAC can reduce it in part by improving the exploration phase of new optimal policies thanks to the presence of the entropy term and the temperature coefficient.

However, during the training phase a sensitivity analysis on the hyperparameters was necessary due to their strong dependence on the performance of the DRL controller. For this reason, it is advisable rely on simulated environments for RL application, at least in the first stage of training.

In this work, 24 different configurations were analysed over a training period of 1 month (December), varying the number of neurons per hidden layer, the γ discount factor, the α learning rate and the temperature-term weight of the reward (β).

The best configuration was chosen after careful analysis, carried out in the section 4.1, and leads to energy savings of 5 % compared to baseline but above all to a significant reduction of violations of comfort conditions.

The trained agent chosen was then used for a static deployment phase, in which the simulation period was extendend to two months, including January and February.

Five deployment scenarios were chosen, so that the adaptability of the DRL control logic can be assessed in respect to changes in boundary conditions, such as the variation of external weather conditions, but also changes in occupancy schedules, structural conditions and indoor comfort conditions.

The results showed excellent adaptability of the DRL control agent and considerable advantages compared to baseline counterpart, as comfort conditions are considerably improved and energy savings are significant. In scenario S2, where the desired temperature setpoint was increased by $1°C$ the DRL control logic was able to achieve savings of 19 % but it has the highest value of comfort band temperature violations among deployment scenarios, albeit smaller than its baseline.

In the remaining four scenarios, the comfort conditions were improved if compared with the baselines, managing to achieve at the same time energy savings between 7 and 9 %. Then, the trade-off concerning the improvements of these two

features is well-respected.

In this thesis work, the results obtained through a careful choice of the present observations in the state-space belonging to an adaptive set, have evidenced the necessity to avoid the exploration of a dynamic deployment phase, that could create eventual problems of instability, due to its ability to update online the control strategy in front of the changing of certain boundary conditions. This analysis suggests that a DRL controller designed with a carefully set of state variables is sufficient to provide good flexibility and adaptability considering the changes on outdoor conditions but also indoor comfort requirements, structural components and occupancy schedules in a static deployment without sacrificing adaptability.

It is clear that, using a non-adaptive set of variables could lead to poorer performance for the static deployment if compared with a dynamic one.

The work carried out could, however, be expanded or improved so that **future works** could focus on the following aspects:

- The developed controller could be implemented in the real world, although switching from simulations to implementation is very complicated and still represents one of the main challenges, especially concerning infrastructure to make the controller available. This problem could be explored in the future so that in-field results can be measured.

- Apply the SAC control logic to more modern HVAC systems, characterised by a much higher level of complexity than the radiator system considered in this work. It is possible thinking to implement it in systems characterised from the presence of renewable energy sources to demonstrate the excellent results achieved in that field by this type of control logic.

- It is possible to introduce into this study the analysis of the Predicted Mean Vote (PMV) and Predicted Percentage of Dissatisfied (PPD) and then take them into account in the objective function. It will be necessary to monitor new parameters useful for their calculation such as the air velocity and the mean radiant temperature, which makes it difficult to evaluate in the real world given the need for adequate probes with which to improve the Building Management System (BMS).

- The DRL control logic could be tested in a non-adaptive set of variables, in order to evaluate its performance in the static deployment case.

- The possible dynamic deployment phase could be explored in order to make a comparison with the current static phase assessed with similar scenarios and assess any instability that results in the learned control policy. Dynamic

deployment may be useful if the thermal inertia of the controlled environment changes, as in the S3 and S4 scenarios.

- The aspects of reproducibility and standardization should be analyzed, as the control agent implemented in this case study does not exhibit the same behaviour under different conditions, such as another building or plant, but also different climatic conditions.

Appendix A

Acronyms

Abbreviazione	Significato
A3C	Asynchronous Advantage Actor-Critic
ANOVA	Variance Analysis
BCVTB	Building Control Virtual Test Bed
BMS	Building Management System
CS	Calibrated Simulation
$C_v(RMSE)$	Coefficient of Variation of the Root Mean Square Error
DNN	Deep Neural Network
DQN	Deep Q-Network
DP	Dynamic Programming
DRL	Deep Reinforcement Learning
DSA	Differential Sensitivity Analysis
EMS	Energy Management System
FAST	Fourier Amplitude Sensitivity Test
HVAC	Heating Ventilation and Air Conditioning
LHV	Lower Heating Value
LSTM	Long Short Term Memory
MADRL	Multi-Agent Deep Reinforcement Learning
MAAC	Multi-actor attention-critic
MBE	Mean Bias Error
MDP	Markov Decision Process
OAT	One At a Time
OCC	Occupant-Centric Controller
PMV	Predicted Mean Value
PR	Polynomial Regression
RL	Reinforcement Learning
SA	Sensitivity Analysis
SAC	Soft Actor-Critic
SI	Sensitivity Index
TD	Temporal Difference
UA	Uncertainty Analysis

Bibliography

[Fabrizio et al., 2015] Enrico Fabrizio & Valentina Monetti (2015), *Methodologies and Advancements in the Calibration of Building Energy Models*, Energies, 8, 2548-2574.

[Coakley et al., 2014] Coakley D., Raftery P., Keane M. (2014), *A review of methods to match building energy simulation models to measured data*, Renew. Sustain. Energy Rev., 37, 123–141.

[Davin et al., 2015] Enrico Fabrizio, Elisabeth Davin, Valentina Monetti, Philippe Andrè & Marco Filippi (2015), *Calibration of building energy simulation models based on optimization: a case study*, Energy Procedia, 78, 2971-2976.

[Raftery et al., 2011] Raftery P., Keane M. & Costa A. (2011), *Calibrating whole building energy models: Detailed case study using hourly measured data*, Energy and Buildings, 43, 3666-3679.

[Reddy et al., 2007] Agami Reddy T., Maor I. & Panjapornpon C. (2007), *Calibrating Detailed Building Energy Simulation Programs with Measured Data—Part II: Application to Three Case Study Office Buildings (RP-1051)*, HVAC&R Research, 13.2.

[Naidu & Rieger, 2011] Naidu D. Subbaram & Rieger Craig G. (2011), *Advanced control strategies for heating, ventilation, air-conditioning, and refrigeration systems—An overview: Part I: Hard control*, HVAC&R Research, 17.1, 2-21.

[Heo, 2011] Heo Y. (2011), *Bayesian calibration of building energy models for energy retrofit decision-making under uncertainty*, Ph.D. Thesis, Georgia Institute of Technology, Atlanta, USA.

[Iren, 2016] Iren (2016), *Report di Diagnosi Energetica, Via Bazzi 4*, Torino, Piemonte, Italia.

[ASHRAE Guideline 14, 2002] ASHRAE Guideline 14, 2002, *Measurement of Energy and Demand Savings, American Society of Heating, Refrigerating and Air-Conditioning Engineers, Inc.,Atlanta*

[Yaser, 2012] Abu-Mostafa Y. (2012), *Lesson 1: The Learning Problem*, *http* : *//work.caltech.edu/slides/slides*01.*pdf* , Caltech, Pasadena, CA, USA.

[Silver, 2015] Silver D. (2015), *Lecture 1: Introduction to Reinforcement Learning*, *https* : *//www.davidsilver.uk/wp* − *content/uploads*/2020/03/$intro_RL$, UCL, London, UK.

[Brandi et al., 2020] Brandi S., Savino Piscitelli M., Martellacci M. & Capozzoli A. (2020), *Deep Reinforcement Learning to optimise indoor temperature control and heating energy consumption in buildings*, Energy and Buildings.

[Pinto et al., 2020] Pinto G., Brandi S., Capozzoli A., Vázquez-Canteli J.R. & Nagy Z. (2020), *Towards Coordinated Energy Management in Buildings via Deep Reinforcement Learning*, 15th SDEWES Conference, Cologne, Germany.

[Dalamagkidis et al., 2007] Dalamagkidis K., Kolokotsa D., Kalaitzakis K. & Stavrakakis G.S. (2007), *Reinforcement Learning for energy conservation and comfort in buildings*, Building and Environment, 42, 2686-2698.

[Chen et al., 2018] Chen Y., Norford L.K., Samuelson H.W. & Malkawi A. (2018), *Optimal control of HVAC and window systems for natural ventilation through reinforcement learning*, Energy and Buildings, 169, 195-205.

[Qiu et al., 2020] Qiu S., Li Zhenhai, Li Zhengwei, Li Jiajie, Long S. & Li Xiaoping (2020), *Model-free control method based on reinforcement learning for building cooling water systems: Validation by measured data-based simulation*, Energy and Buildings, 218.

[Uhn Ahn & Soo Park, 2019] Ki Uhn Ahn & Cheol Soo Park (2019), *Application of deep Q-networks for model-free optimal control balancing between different HVAC systems*, Taylor & Francis, Science and Technology for the Built Environment.

[Wang et al., 2017] Wang Y., Velswamy K. & Biao Huang B. (2017), *A Long-Short Term Memory Recurrent Neural Network Based Reinforcement Learning Controller for Office Heating Ventilation and Air Conditioning Systems*, Processes, 5, 46.

[Wang & Hong, 2020] Zhe Wang & Tianzhen Hong (2020), *Reinforcement learning for building controls: The opportunities and challenges*, Applied Energy, 269.

[Zou et al., 2020] Zou Z., Yu X. & Ergan S. (2020), *Towards optimal control of air handling units using deep reinforcement learning and recurrent neural network*, Building and Environment, 168.

[Hasselt et al., 2016] Hasselt H.V., Guez A. & Silver D. (2016), *Deep reinforcement learning with double Q-Learning*, Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, Association for the Advancement of Artificial Intelligence (AAAI) Press, Phoenix, Arizona, USA, 2094–2100.

[Datacamp, 2020] Datacamp (2020), *Neural Network Models*, *https* : *//www.datacamp.com/community/tutorials/neural − network − models − r* , last visit dated 18/06/2020.

[Sutton & Barto, 1998] Sutton R. S. & Barto A.G. (1998), *Reinforcement learning an introduction*, a Bradford book. England: MIT Press.

[Haarnoja et al., 2018] T. Haarnoja, A. Zhou, K. Hartikainen, G. Tucker, S. Ha, J. Tan, V. Kumar, H. Zhu, A.Gupta and P. Abbeel & S. Levine (2018), *Soft Actor-Critic Algorithms and Applications*, arXiv 1812.05905.

[Nair et al., 2015] Nair, A., P. Srinivasan, S. Blackwell, C. Alcicek, R. Fearon, A. De Maria, V. Panneershelvam, M. Suleyman, C. Beattie, S. Petersen, et al. (2015), *Massively parallel methods for deep reinforcement learning*, arXiv preprint 1507.04296.

[Yang et al., 2015] Yang L., Nagy Z., Goffin P. & Schlueter A. (2015), *Reinforcement learning for optimal control of low exergy buildings*, Applied Energy, 156, 577-586.

[Zhang et al., 2019] Zhang Z., Chong A., Pan Y., Zhang C. & Lam K.P. (2019), *Reinforcement learning for optimal control of low exergy buildings*, Energy and Buildings, 199, 472-490.

[Claub et al., 2017] Claub J., Finck C., Vogler-finck P. & Beagon P. (2017), *Control strategies for building energy systems to unlock demand side flexibility - A review Norwegian University of Science and Technology*, Trondheim, Norway Eindhoven University of Technology, Eindhoven, Netherlands Neogrid Technologies ApS/Aalborg, 15th Int Conf Int Build Perform 2017, 611–20.

[Finck et al., 2017] Finck C., Beagon P., Clauss J., Thibault P., Vogler-Finck PJC, Zhang K., et al. (2017), *Review of applied and tested control possibilities for energy flexibility in buildings*, Technical report from IEA EBC Annex 67 "Energy Flexible Buildings" 2017, 1–59. *https* : *//doi.org*/10.13140/*RG*.2.2.28740.73609.

[Afram & Janabi-Sharifi, 2014] Afram A. & Janabi-Sharifi F. (2014), *Theory and applications of HVAC control systems – A review of model predictive control (MPC)*, Building and Environment, 72, 343-355.

[Yun et al., 2012] Yun K, Luck R., Mago P.J. & Cho H. (2012), *Building hourly thermal load prediction using an indexed ARX model*, Energy and Buildings, 54, 225-233.

[Yoon & Moon, 2019] Yoon Y.R & Moon H.J. (2019), *Performance based thermal comfort control (PTCC) using deep reinforcement learning for space cooling*, Energy and Buildings, 203.

[Nagarathinam et al., 2020] Nagarathinam S., Menon V., Vasan A. & Sivasubramaniam A. (2020), *MARCO - Multi-Agent Reinforcement learning based COntrol of building HVAC systems*, In The Eleventh ACMInternational Conference on Future Energy Systems (e-Energy'20), June 22–26, 2020, Virtual Event. ACM, New York, NY, USA, 11 pages, *https* : *//doi.org*/10.1145/3396851.3397694.

[Park & Nagy, 2020] June Young Park & Zoltan Nagy (2020), *HVACLearn: A reinforcement learning based occupant-centric control for thermostat set-points*, In The Eleventh ACM International Conference on Future Energy Systems (e-Energy '20), June 22–26, 2020, Virtual Event. ACM, New York, NY, USA, 434-437, *https* : *//doi.org*/10.1145/3396851.3402364.

[Ding et al., 2020] Ding X., Du W. & Cerpa A. (2020), *OCTOPUS: Deep Reinforcement Learning for Holistic Smart Building Control In The Eleventh ACMInternational*, In The 6th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation (BuildSys '19), November 13–14, 2019. ACM, New York, NY, USA, 326-335, *https* : *//doi.org*/10.1145/3360322.3360857.

[Yu et al., 2020] Yu L., Sun Y., Shen C., Yue D., Jiang T. & Guan X. (2020), *Multi-Agent Deep Reinforcement Learning for HVAC Control in Commercial Buildings*, IEEE Transactions on smart grid, Vol. 20, No. 20, *https* : *//arxiv.org/pdf*/2006.14156.*pdf*.

Acknowledgements

My sincere gratitude goes to BAEDA team, particularly to Professor Alfonso Capozzoli, who believed in me from the first meeting, always trying to bring out the best in me.

Thanks Silvio and Giuseppe, co-advisors but also friends, for always being available and encouraging me in every moment in front of the troubles, making me give my very best on this route.

The BAEDA team support was essential in making the choose about my near future and I am very excited to join the team.

Thanks, Enerbrain Srl, for allowing me to join as a trainee in their team, to learn important aspects for my future profession.

Five years ago, I remember as if it was yesterday, I entered for the first time through the main entrance of this University.

If that was possible, I must, first of all, declare my heartfelt thanks to my parents, Enzo and Giusy, who made it plausible for me to achieve what I have always dreamed of becoming: an engineer. Thanks also to my brother Paolo, who encouraged me at all times to give my best, and my sister Noemi, with whom I was able to share family moments during her stay in Turin.

Thanks to my sweetheart, Emilia, who has always believed in me in each particular moment, accepting my determination to leave home to grow professionally.

Thanks to Nicolò and Simone, roommates since the beginning but mainly brothers, for practically destroying the homelessness.

This stay in Turin was represented by moments of study but above all of joy and fun. I indeed could not have wished for better classmates and life.

Thanks to Giuseppe, Davide, Roberto, Gianvito, Andrea, Enzo L., first classmates but real friends and brothers for me. I have grown up academically and not, and shared many joyful moments with them: I hope to spend more in the future.

Thanks, Miriana, Alessia and Mariacarla thanks for having considered me a friend you can rely on.

My gratitudes goes to my dear friends Francesco, Giuseppe D., Roberto G., Mario, Cristiano, Matteo, for having shown that they are friends even before being simple classmates. I will always bring you all with me.

Thanks to those who arrived at the advent of this experience and have never forgotten the friendship that joins us: Enzo C., Giorgio, Giovanni, Riccardo, Simona, Alessia, Federica, Lucia.

To conclude, thanks to the ladies and gentlemen from "Il Posto", my dear sicilian friends from Alcamo, the city that I left to arrive here in Turin, for making a party every time I came back home.