POLITECNICO DI TORINO

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

TESI DI LAUREA MAGISTRALE

Implementation of an adaptive control strategy to regulate heating systems in residential building

Relatore:

Prof. Alfonso Capozzoli

Correlatori:

Ing. Silvio Brandi

Ing. Giuseppe Pinto

Candidato

Davide Borello

Anno Accademico 2020-21

To my grandparents,
for all the advices and
for being examples to follow.

Abstract

In recent years, control systems able to predict the continuous adjustments of dynamic factors, which allow the adaptability in the building energy management, have become necessary due to the increasing complexity of HVAC systems, and the impact occupants' behaviour.

Classic control systems, including On/Off or PID, cannot perform these tasks because they do not provide any prediction capabilities. Moreover, model-based predictive control strategies, such as Model Predictive Control (MPC), are complex to apply because they both need a model for the optimisation, which is difficult to achieve and have a high computational cost.

For these reasons, recent researches are focusing on model-free control strategies, and in particular on the application of Reinforcement Learning (RL).

Since RL does not require a prior known model, the agent learns the best action through trial-and-error interactions within the environment, following an action-reward process.

In this dissertation, a control algorithm based on Soft Actor-Critic (SAC) is implemented to control a radiant floor heating system in an existing residential building. Since the occupants' behaviour have a significant impact on the heating energy consumption, in particular the windows behaviour, four different models simulating the windows opening and closing are tested, and the model which estimate the windows' state with best performance has been implemented in the building model.

The initial phase consists of the construction of geometrical and energy models. It is necessary to implement the control agent, which is then tested in the training stage to estimate potential energy savings and temperature violations' reduction.

As a consequence, through a sensitivity analysis, conducted on the hyperparameters to determine the best configuration, an energy saving of 5% and a significant decreasing in the sum of temperature violations are obtained.

After the training phase, the agent is tested in the deployment phase by analysing four different scenarios to examine its adaptivity in different conditions.

In conclusion, the agent obtains a significant reduction in temperature violations, these reductions range between 750 °C and 950 °C, in all scenarios, and, at the same time, the energy-saving obtained ranges between 2% and 6%.

Summary

List of figures

List of tables

Chapter 1:
Introduction

1. Introduction

As the importance of climate change themes has grown, programmes focused on reducing primary energy consumption, and CO_2 emissions have been encouraged in recent years. In this context, it is also incentivised the use of Renewable Energy Source (RES).

Since people spend most of their time in buildings, they consume a considerable amount of energy, and this is due to different factors like occupants' behaviour, building characteristics and the context in which the building is located.

In particular, as shown in *Figure 1*, the residential buildings in Italy are responsible for around 30% of the total energy consumption.

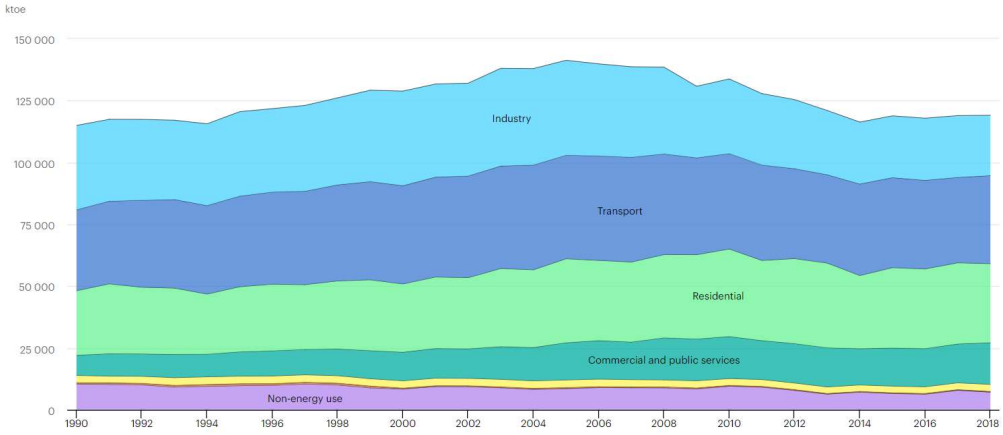


Figure 1 - Italian energy consumption by sector [1]

This energy consumption is mainly due to the HVAC systems that have to satisfy the occupants' comfort. However, very often these systems are either inefficient or not optimally controlled.

In recent years, HVAC systems became increasingly complex, and, consequently, the design of control systems which has to take into account several factors related to grid requirements (Demand Response), occupant preferences and external forcing variables.

These factors, being stochastics, lead to the non-linearity of the system, complicating, even more, the control actions. Therefore, recent researches have

focused on adaptive control systems and new control methods for HVAC systems to maintain the comfort conditions for the occupants and to reduce the energy consumption.

Figure 2 shows the general scheme for a single-level control and the principal factors that influenced the controller.

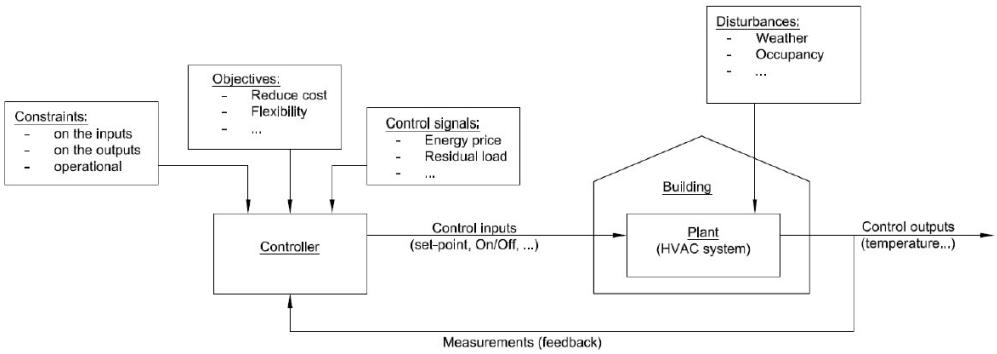


Figure 2 - General scheme of a single-level control [2]

In this context, it is essential the introduction of energy flexibility, even if its definition is complex. For Finck et al. [2], the energy flexibility can be seen as the ability to manage a building's demand and generation according to local climate conditions, user needs and grid requirements.

These tasks lead to overcoming the classic control methods, which will be described in paragraph 1.1, and to increase the research in new strategies

Furthermore, predictive control allows buildings to use better available energy flexibility from the building passive thermal mass. However, due to the complex nature of the building, developing computationally efficient control-oriented models, which are capable to handle the nonlinear thermal-dynamics of buildings, is showing to be a significant barrier. Data-driven predictive control, linked to the "Internet of Things", is the promise for an initial and transferrable approach, with data-driven models replacing traditional physics-based models [3].

Model Predictive Control (MPC) is a model-based predictive control technique, which applies a building model to predict the future states and to optimise the cost

function over a prediction time horizon. The main drawback of this method is the high computational cost for the construction of the model.

Therefore, researchers start to study the application of Reinforcement Learning (RL) to control the HVAC systems, which is a control technique belonging to the Machine Learning family. This technique does not require prior knowledge neither of the system nor of the buildings to be controlled. In fact, an agent learns an optimal policy directly by the interaction with the environment, receiving a reward influenced by the action taken from a certain state.

In this thesis, a residential building, located in Turin, was controlled, through a control algorithm based on SAC, (Soft Actor-Critic), which is a branch of RL that allows the use of continuous action and state space. In addition, it introduces an entropy term on the reward definition. The objective of the control agent is the maintenance of the comfort conditions by control the supply power to be provided in each floor.

The geometrical model of the building was first built on SketchUp, while the energy model was made on EnergyPlus v9.2.0.

Then, the reinforcement learning control logic is implemented by matching EnergyPlus and Python. This two software are connected through Building Control Virtual Test Bed (BCVTB) and the ExternalInterface of EnergyPlus.

The objective is to maintain the indoor temperature inside a comfort range when there is the presence of occupants, trying to obtain also a reduction of energy consumption. To do this, the designed control agent chooses the supply power to be given to each floor.

Moreover, since the large importance of the occupants in energy consumption, some models that simulate the windows opening and closing behaviour were tested and implemented to try to represent the reality as close as possible.

Sections 1.1 and 1.2 present a description, respectively, of the main control techniques for HVAC systems, and an overview of previous works on the Reinforcement Learning. Section 2 provides a description of the different windows'

models analysed and the qualitative choice of the best. In contrast, section 3 illustrates the background for this work introducing the RL and the SAC. In section 4, the framework of this analysis is described, in particular, the case study and the construction of the simulation environment.

In section 5, the development of the SAC control agent is illustrated, together with the introduction of the training and deployment phases.

The result of the simulation environment and the results of both phases are described in section 6. Finally, section 7 and 8 provide a discussion of the results and the conclusion.

1.1 Control of HVAC systems

A Heating, Ventilation and Air Conditioning (HVAC) system is designed to maintain a certain level of comfort of the users of the building. There are different type of HVAC systems and each of them operates on various parameters of the building; in particular, the most controlled parameter is the indoor air temperature. Fink et al. [4] classified the HVAC control methods in four categories:

- Classical control;
- Hard control;
- Soft control;
- Hybrid controls

Figure 3 highlights the classification of the control techniques for HVAC systems.

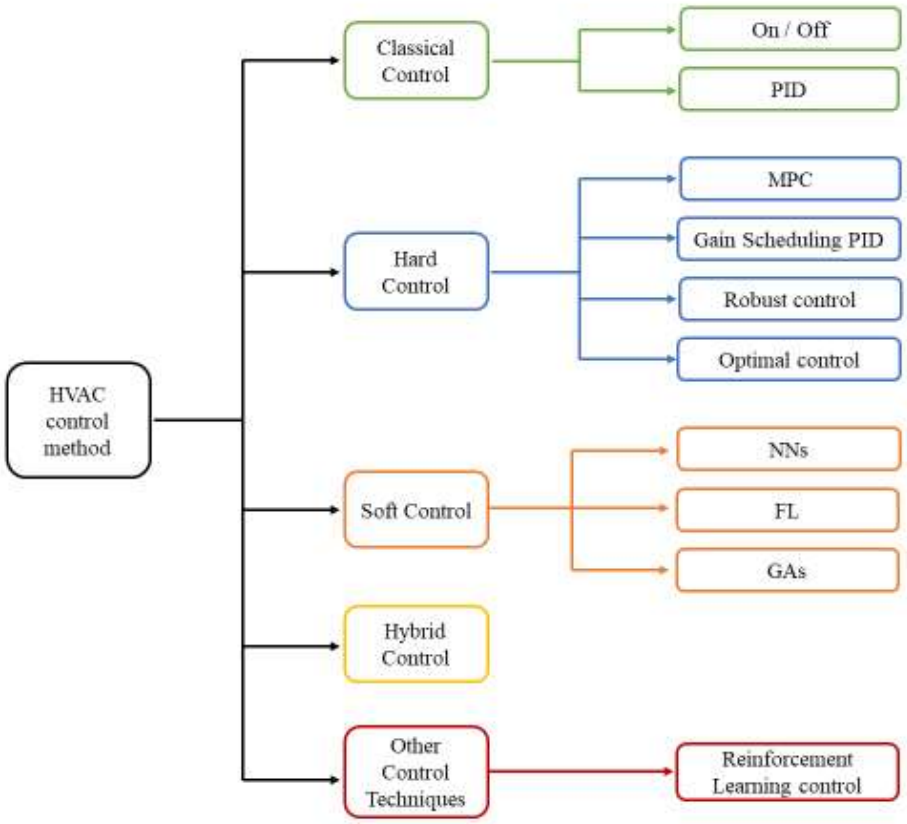


Figure 3 - Classification of control methods for HVAC systems

The first category includes On / Off control and PID control. On / off controllers regulate the process within a defined lower and upper value, to maintain the process between this range.

Figure 4 show the On / Off control logic and the behaviour of the controlled variable. The system stays turn on until the controlled variable reaches the upper limit of the threshold, then the system is turned off and remain in this position until the variable reaches the lower limit.

On the other hand, the PID control a variable by using error dynamics, in particular applying three different action: proportional, integral and derivative. The signal (u(t)) provide by the controller can be defined with the following equation:

$$u(t) = K_P e(t) + K_I \int_{to}^{t} e(t)dr + K_D \frac{de(t)}{dt}$$

in which e(t) indicates the control error, and it is given by the difference between the controlled variable and the setpoint value, K_P is the proportional gain, K_I is the integral gain, and K_D is the derivative gain.

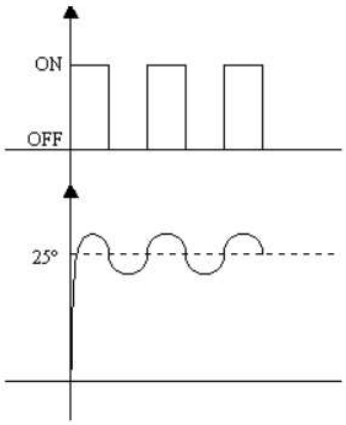


Figure 4 - On/Off control logic [5]

The proportional action produces a difference between the actual value of the variables and the desired one. This difference can be reduct increasing the proportional gain. *Figure 5* illustrates the response of the variable subject to the proportional action.

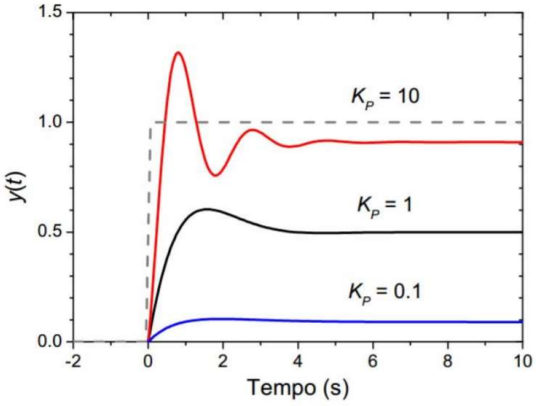


Figure 5 - Proportional action [6]

The integral action tends to reduce the offset between the controlled variables and its setpoint. *Figure 6* shows the response of the system to both the integral and the proportional action. The following equation defines the parameter T_I:

$$T_I = \frac{K_P}{K_I}$$

With a higher value of K_I, and consequently lower value of T_I, the offset can be reduced.

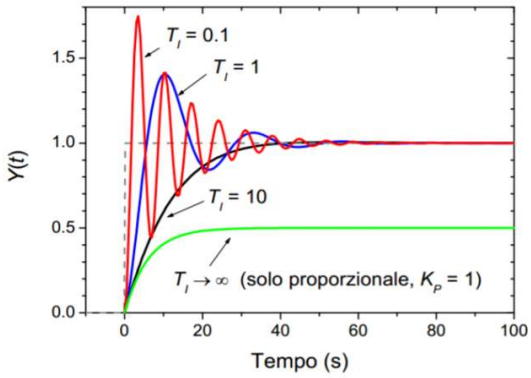


Figure 6 - Integral action [6]

Finally, the derivative action increases the stability of the response decreasing its oscillations. *Figure 7* shows the response to a PID controller with all of the three actions. The parameter T_D is defined by the following equation:

$$T_D = \frac{K_D}{K_P}$$

The stability of the response increases with the increasing of T_D.

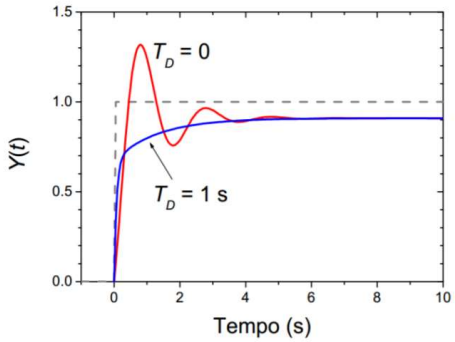


Figure 7 - Derivative action [6]

The main drawback of this control logic is the tuning of the three parameters K_P, K_I, and K_D to minimize the offset, respond fastly to disturbances and increase the stability. The two most used tuning method are the two proposed by Ziegler-Nichols.

The Hard control category includes control techniques such as Optimal Control, Robust Control and Gain Scheduling PID, but the most important control method in this family is Model Predictive Control (MPC). It applies a building model to predict the future states and to optimise the cost function over a prediction time horizon; it also takes into account disturbances and constraints [7].

The goal of the MPC is to minimize the cost function, which is influenced by different factors, such as building dynamics, type of the HVAC system and user preference. For example, Picard and Helsen [8] in their work modelled only the building envelope and their cost function aims to minimize the heat inputs from the two different heating and cooling system, which each of them has an associated cost. In contrast, Jorissen [9] also modelled the HVAC system, and he controls the setpoints of different components to minimize energy consumption.

The objective of the cost function can be different in each case, for example, Cigle et al. [10] and Yang et al. [11] introduce the PMV value to maximize the occupancy thermal comfort. Jorissen et al. [12] developed a model based on statistical data to estimate the future air quality and an occupancy model.

Another objective of the cost function can be the minimization of the cost of the energy. This task takes more importance in particular for the systems that are electricity-based, such as chiller and heat pump, because the electricity cost is variable [13].

Avci et al. [14] and Bianchini et al. [15] studied the response of MPC in demand-response problem applying the real-time pricing. Instead, Oldewurtel et al. [16] focused their works on the reduction of the peak electricity demand optimizing the economic cost. Moreover, Qureshi and Jones [17] and Patteeuw et al. [18] studied the effects on stability and flexibility of the MPC controller with the introduction of RES.

In recent year, since the introduction of programs that aim to the reduction of the greenhouse gas emissions, Knudsen and Petersen [19] and Vogler-Finck et al. [20] introduced as the objective function the minimization of greenhouse gas emissions.

Finally, Vandermeulen et al. [21] and Vogler-Finck et al. [22] designed a cost function that aims to maximize the use of RES, or to minimize the use of fossil fuels.

Figure 8 shows the schematic representation of the standard close loop with MPC which can describe most of the applications in building control. The building is affected by disturbances, such as weather conditions, and it is subjected to some constraints, such as the acceptability range for indoor temperature.

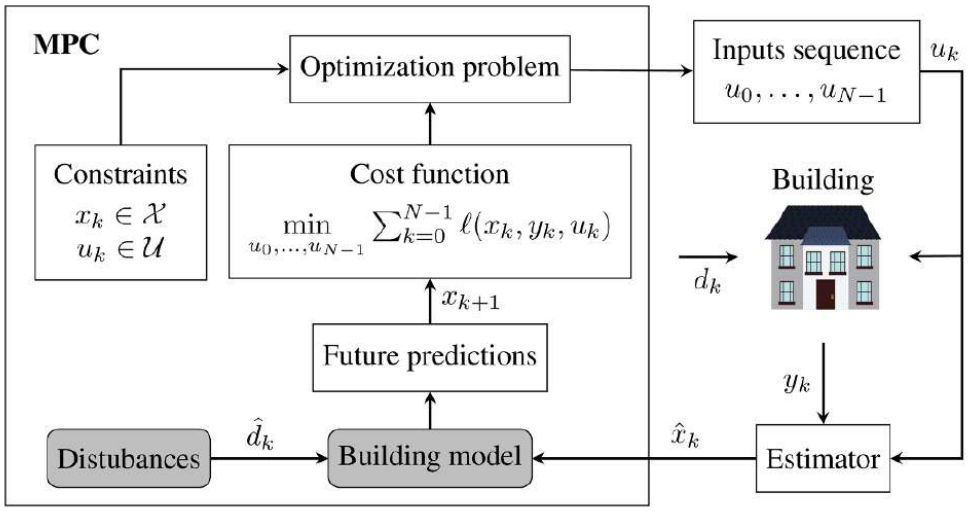


Figure 8 - Schematic representation of the standard closed-loop system with MPC [13]

The most important feature of this technique is the building model, this can be obtained by using three different modelling paradigms:

- White box models describe the building in details with physical knowledge; therefore, they are based on the conservation of mass and energy and principles of heat transfer. They require information about building geometry, material properties, and equipment. The obtained models often include thousands of parameters, so there al lot of potential sources of inaccuracy. These models are difficult to implement, and they have a high computational cost.
- Grey box models simplify the physical representation of the building using the RC (resistance and capacitance) analogy. *Figure 9* illustrates an example

of this electric analogy. The thermal mass of the construction is represented by a capacitor, while the resistor represents the building elements, such as wall or floor, and the nodes of the networks indicate the temperature of the construction. The order of the dynamic system is defined by the number of capacitors. These models require less computational time than the white box models, but they are less accurate.

- Black box models use mathematical and empirical equations to describe the buildings. They require a large dataset to calibrate and train the model.

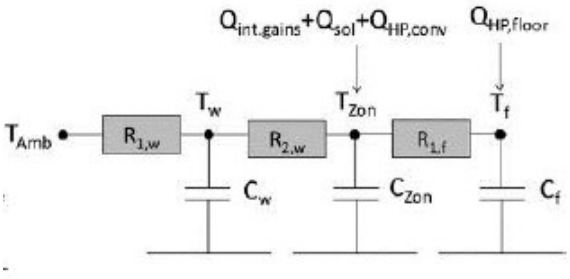


Figure 9 - Example of RC analogy for a radiant floor system [4]

One of the main drawbacks is the computational time for both the buildings' model construction and the solving of the optimization problem. Furthermore, a sudden change in the variables can lead to instability of the solution. Other issues are the availability of data and the possible building-model mismatch or the inaccurate measurements. Since the fundamental element of the MPC techniques is the building model, a non-accurate model can lead to wrong future predictions.

As regards Soft control, it is an emerging control method based on the application of Neural Networks (NNs), Fuzzy Logic (FL) and Genetic Algorithms (GAs) [23]. NNs are a mathematical representation of biological neurons which associate the input and output actions. They are used to control systems, in which the models are not fully known. Curtiss et al. [24] made a comparison between the control performance of an NNS and a PID controller on the decentralized and centralized control of an HVAC system. So et al. [25] design a NNs-based controller for an air

handling unit (AHU) to minimize the offset between the temperature and its setpoint and the energy consumption.

FL is a control method which use a series of if-then logic to imitate the human actions to control the output variable. *Table 1* illustrates an example of Fuzzy Logic.

Rules	IF	AND	THEN
Rule 1	T is low	T is decreasing	Increase heating energy
Rule 2	T is high	T is increasing	Decrease heating energy
…	…	…	…

Table 1 - Example of Fuzzy Logic

Huang and Nelson [26] and Arima et a. [27] designed a FL controller to maintain the temperature near a the setpoint for a HVAC system.

GAs are derivate-free optimization method of multiobjective functions. Wright et al. [28] design a GA controller for a single-zone AHU to define the supply air temperature and the flow rate to maximize the thermal comfort and minimize the operating cost.

The last category refers to the fusion of Hard and Soft controls.

Finally, in recent years, studies on a RL-based technology for the control of HVAC systems have increased. RL is a model-free control technique which can also be implemented without a priori knowledge of the controlled environment or process. In this control approach, a designed agent learns a control policy from its interactions with the environment through a reward.

This typology will be described in the following sections. In particular, an analysis of the existing works will be analysed in section 1.2, while chapter 2 will go into the details of the methodology.

1.2 Previous works on RL

Reinforcement learning, being a control algorithm that has very contrasting features compared to traditional control systems, such as ones applied in classical building

control, has raised interest in recent years, even if its applications persist limited. Consequently, the reinforcement learning method is becoming more distinguishing and applicable in control networks for buildings. Furthermore, this aspect is even more expanded in the Deep Reinforcement Learning because it is not only a more futuristic but also a more dynamic section of these algorithms.

Table 2 lists some works which use RL and their respective objectives.

One characteristic that distinguishes different RL algorithm is the action selection method, in particular, the two most applied approach are the ϵ-greedy and the Boltzmann method. Among the analysed works, the ϵ-greedy method results the most used, while in some case the implemented method is not specified. The number of publications of each technique is illustrated in *Figure 10*.

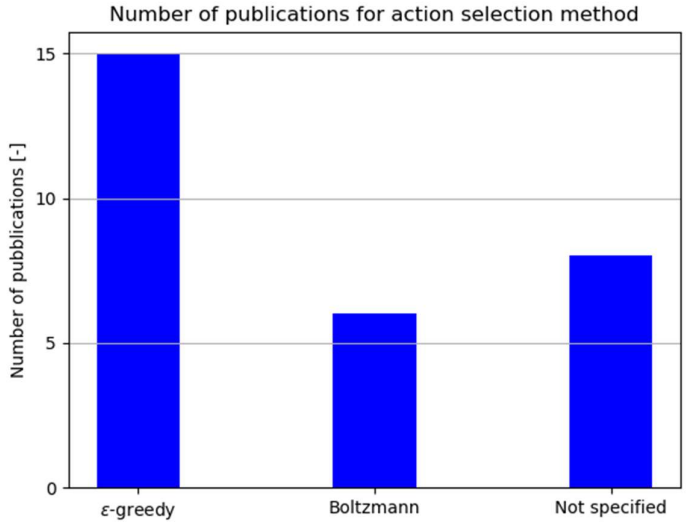


Figure 10 - Number of publications for action-selection method

Another parameter that influences the performance of the RL control algorithm is the control timesteps. In these studies are used five different timestep from 5 minutes to 60 minutes. *Figure 11* show the distribution of the publications among the various control timesteps. The most used is a timestep equal to 15 minutes, followed by the one equal to 5minutes.

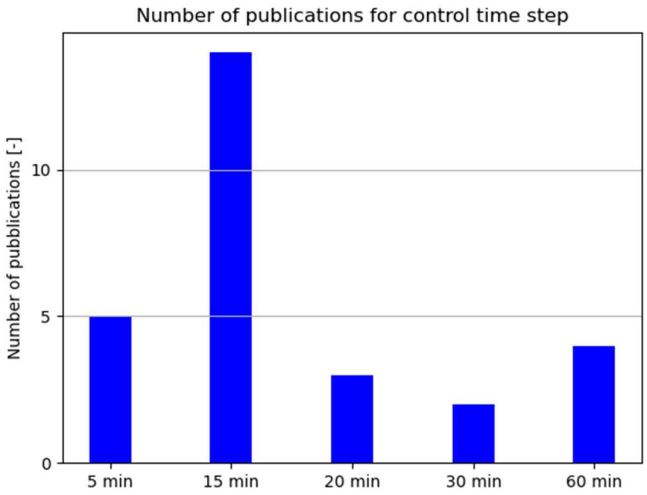


Figure 11 - Number of publications for control timestep

The formulation of the reward equation varies in different RL works. Between the studies analysed in this dissertation, five different reward term are individuated:

- Energy consumption
- Comfort term, which contains different formulation related to the maintenance of the required zone temperature, or purely comfort values such as PMV and PPD;
- Cost function, this term is strictly related to the first, but focuses more on the energy's price;
- CO_2, this term aims to control the CO_2 concentration in the controlled environment;
- RES, which takes into account the energy produced by renewable energy sources.

In some cases, the reward equation is formed by two or more competing terms. In fact, between the 27 analysed studies, 20 use multiple terms into for the reward.

As observable in *Figure 12,* the most present terms are the energy-related one and the comfort-related term.

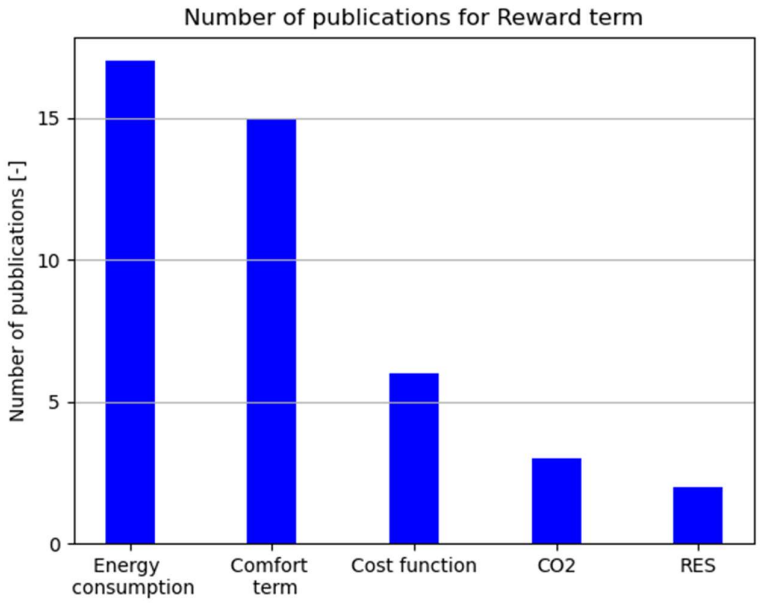


Figure 12 - Number of publications for reward term

Among the study cited above, some deserve more attention for the results obtained. For example, Vazquez-Canteli et al. [29] used a batch reinforcement learning (BRL) algorithm with fitted Q-iteration to control the heat pump and two water tanks (one for heating and once for cooling). At the end, they obtained a reduction in energy consumption while maintaining adequate thermal comfort.

A similar goal has been reached by Ki Uhn Agn and Cheol Soo Park [30], who minimize the building's energy usage by 15.7% in comparison with the baseline operation while maintaining the indoor CO2 concentration below 1,000 ppm. These targets are reached through a Q-network (DQN) for model-free optimal control balancing between different HVAC systems, and it was designed with two hidden layers.

An example of Multi agent RL problem was proposed by Nagarathinam et al. [31]. In their work they introduced MACRO (Multi-Agent Reinforcement learning COntrol), which is based on Double Deep Q-Network algorithm and it used two separated control agents to control both the AHUs and chillers setting the building and chillers setpoints to optimize the HVAC operating phase. The control objective was the minimization the HVAC energy consumption respecting the comfort

constraint. The designed agents was trained and deployment in real configuration, and as result MACRO learn the optimal policy, improving the comfort and obtaining an energy saving of about 17%.

Park and Nagy [32] execute another study on the application of a Q-Learning (Tabular Q-Learning) algorithm on an HVAC system. The agent learns the occupant behaviour and indoor environments by monitoring indoor air temperature, occupancy, and thermal vote, and estimates adaptive thermostat set-points to match between occupant comfort and energy efficiency.

Zhang et al. [33] proposed a control algorithm based on A3C (Asynchronous Advantage Actor-critic) and it was implemented to a radiant heating system. In particular, this control agent regulates the supply water temperature set-point of the Mullion system in order to reduce the heating demand consumption and to maintain the indoor thermal comfort.

An example of Deep Reinforcement Learning (DRL) on the control of the radiant heating system with a boiler and radiators is issued by Silvio Brandi et al. [34].

The controller is implemented to manage the supply water temperature setpoint to terminal units. Moreover, two sets of input variables are analysed for estimating their impact on the adaptability capabilities of the DRL controller; so, a static and dynamic deployment of the DRL controller is performed. The trained control agent is tested for four different scenarios to fix its adaptability to the variation of forcing variables. Consequently, when the set of variables are appropriately selected, the energy saved ranges between 5 and 12 %.

Last but not least, a more complex usage of a Q-Learning (Double Deep Q-Network) has been done by Ding et al. [35], who developed a system called OCTOPUS, which uses a data-driven method to find the optimal control series of all building's subsystems, including HVAC, lighting, blinds and window systems. Overall, they demonstrated that OCTOPUS could achieve 14.26% and 8.1% energy savings related with the state-of-the-art rule-based system and the latest DRL-based

method available in the literature respectively while maintaining human comfort within the aspired range.

Title	Learning algorithm	Control objective
Balancing comfort and energy consumption of a heat pump using batch reinforcement learning with fitted Q-iteration [29]	batch reinforcement learning (BRL)	Minimize the energy consumption maintaining the thermal comfort
Whole building energy model for HVAC optimal control: A practical framework based on deep reinforcement learning [33]	A3C (Actor-Critic)	Reduce the heating demand consumption and maintain the indoor thermal comfort level.
Model-free control method based on reinforcement learning for building cooling water systems: Validation by measured data-based simulation [36]	Q-learning	Reduce the cooling demand consumption and improve the system efficiency
Reinforcement learning for optimal control of low exergy buildings [37]	Q-Learning (both Tabular and Batch)	Maximize the net thermal output
Reinforcement Learning Applied to an Electric Water Heater: From Theory to Practice [38]	Q-Learning (fitted Q-Iteration)	Minimizes the cost of energy consumption of EWH, given an external price profile to the agent at the start of each day.
Model-free control of thermostatically controlled loads connected to a district heating network [39]	Q-Learning (fitted Q-Iteration)	Peak shaving and energy arbitrage responding to an external price
Beyond Theory: Experimental Results of a Self-Learning Air Conditioning Unit [40]	Q-Learning (fitted Q-Iteration)	Minimize the quadratic difference between the locally produced photovoltaic power and the power consumption of the ACU.
Application of deep Q-networks for model-free optimal control balancing between different HVAC systems [30]	Deep Q-Network (DQN)	The optimization goal was to minimize the building's energy use maintaining the CO2 concentration below 1,000 ppm.
Data-driven simulation of a thermal comfort-based temperature set-point control with ASHRAE RP884 [41]	Q-Learning (Tabular Q-Learning)	Improve the thermal comfort maintaining temperature set-point.
Learning Based Bidding Strategy for HVAC Systems in Double Auction Retail Energy Markets [42]	Q-Learning (Tabular Q-Learning)	Reduce the energy cost
Optimal control of HVAC and window systems for natural ventilation through reinforcement learning [43]	Q-Learning (Tabular Q-Learning)	Reduce the energy consumption maintaining constant thermal comfort
Experimental analysis of data-driven control for a building heating system [44]	Q-Learning (Fitted Q-Iteration)	dynamic pricing
Gnu-RL: A Precocial Reinforcement Learning Solution for Building HVAC Control Using a Differentiable MPC Policy. [45]	Differentiable MPC & REINFORCE	Reduce the energy consumption maintaining constant thermal comfort
Residential Demand Response of Thermostatically Controlled Loads Using Batch Reinforcement Learning [46]	Q-Learning (Fitted Q-Iteration)	Minimize any deviation between the day-ahead consumption plan and the actual consumption, minimizing the cost
Performance based thermal comfort control (PTCC) using deep reinforcement learning for space cooling [47]	Q-Learning (Fitted Q-Iteration)	Minimize the total energy consumption while maintaining the thermal comfort performance within a desired range
On-Line Building Energy Optimization Using Deep Reinforcement Learning [48]	Deep Q-learning, Deep Policy Gradient	Reduce the Peak Power and minimize costs.
Online tuning of a supervisory fuzzy controller for low-energy building system using reinforcement learning [49]	Q-Learning (Fuzzy Q-Learning)	Reduce the energy consumption maintaining thermal comfort.
A Long-Short Term Memory Recurrent Neural Network Based Reinforcement Learning Controller for Office Heating Ventilation and Air Conditioning Systems [50]	Model-free actor-critic RL using a variant of Artificial Recurrent Neural Network	achieve thermal comfort while maintaining a certain level of energy efficiency.

Thermal and Energy Management Based on Bimodal Airflow-Temperature Sensing and Reinforcement Learning [51]	A3C	Minimize the energy consumption
Advanced Building Control via Deep Reinforcement Learning [52]	Not described in detail	Reducing Energy consumption
Energy optimization associated with thermal comfort and indoor air control via a deep reinforcement learning algorithm [53]	DQN	Optimization of energy consumption of air-conditioning systems in association with thermal comfort and indoor air quality.
Towards optimal control of air handling units using deep reinforcement learning and recurrent neural network [54]	Q-Learning (Deep Q-Network with LSTM)	Minimizing energy consumption while maintaining thermal comfort for occupants.
HVACLearn: A reinforcement learning based occupant-centric control for thermostat set-points [32]	Q-Learning (Tabular Q-Learning)	Calculating thermostat set-points to balance between occupant comfort and energy efficiency
MARCO - Multi-Agent Reinforcement learning based Control of building HVAC systems [31]	Q-Learning (Double Deep Q-Network)	Maintaining thermal comfort with the lowest energy consumption.
OCTOPUS: Deep Reinforcement Learning for Holistic Smart Building Control [35]	Q-Learning (Double Deep Q-Network)	Minimize the energy consumed by all subsystems in the building and maintain the human comfort metrics within a particular range.
Deep Reinforcement Learning to optimise indoor temperature control and heating energy consumption in buildings [34]	Q-Learning (Double Deep Q-Network)	Reduce the amount of thermal energy while maintaining indoor air temperature within an acceptability range during occupied periods
Multi-Agent Deep Reinforcement Learning for HVAC Control in Commercial Buildings [55]	Multi Agent Deep Reinforcement Learning (MA-DRL)	Minimizing 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

Table 2 - RL previous works

Chapter 2:
Modelling occupants' behaviour

2. Modelling occupants' behaviour

As many researchers describe, occupant behaviour has a more considerable influence on residential consumption. In particular, van den Brom et al. [56] found that at least 54% of the variance in energy consumption in similar buildings can be explained by "building characteristics, 17% by the occupants' lifestyle, 15% by the change of occupants and 13% by house-related quality differences. Therefore, occupants contribute approximately 50% of the variance.

While in older buildings, the physical characteristics have a more decisive impact on the variance, in more recent ones households cause a larger percentage of the variance.

In the occupants' lifestyle, one of the most impacting actions on energy consumption is the windows opening behaviour.

In this study, to better represent the occupant behaviour and in particular, the windows behaviour, are analysed different window models found in the literature.

The first one is the model proposed by Rouleau and Gosselin [57], and it is based on the study of eight apartments in Quebec City (Canada). This article points to create a probabilistic window opening model based on a logistic regression to predict the state (open/close) of windows according to different parameters related to indoor and outdoor environments and time-related terms. After a sensibility analysis, the two most relevant parameters are the indoor and outdoor temperatures, and consequently, two equations are created, one for the opening of the windows and one for the closing:

$$logit(p_{op}) = ln\left(\frac{p_{op}}{1-p_{op}}\right) = -6.216 + 0.059 * T_{in} + 0.33 * T_{out}$$

$$logit(p_{clo}) = ln\left(\frac{p_{clo}}{1-p_{clo}}\right) = -0.871 - 0.091 * T_{in} - 0.028 * T_{out}$$

Instead, Andersen et al. [58] made their study in 15 dwellings in Denmark and proposed a model based on the logistic regression. However, concerning the previous model, the coefficient of the equation varies with both the day of the week

and the time of the day. The parameters involved in the equations and their coefficient are shown in *Figure 13.*

		Open		Close	
Variable		Coefficient	magnitude	Coefficient	Magnitude
Intercept during weekend	Night	-8.55	-	-4.08	-
	Morning	-5.08		5.57	
	Day	-6.67		7.35	
	Evening	-6.61		6.36	
Intercept during workday	Night	-8.32	-	-3.88	-
	Morning	-4.85		5.77	
	Day	-6.44		7.55	
	Evening	-6.38		6.56	
Indoor temperature	Night	0.002585	1.10	-0.8107	-12.2
	Morning	0.009908		-0.3025	
	Day	0.07336		-0.1871	
	Evening	0.011616		-0.2357	
Indoor Relative humidity	-	-	-	0.03942	1.6
CO2 concentration	Night	0.001018	2.38	-0.0037	-7.8
	Morning	0.000566		-0.00059	
	Day	0.000158		-0.00179	
	Evening	0.001134		-0.00039	
Outdoor temperature	Night	0.060408	2.30	-0.5343	-20.3
	Morning	0.043587		-0.267	
	Day	0.012418		-0.2153	
	Evening	0.026525		-0.2019	
Wind speed during weekend	Night	0.002489	0.03	0.36406	4.7
	Morning	0.002489		0.05866	
	Day	0.002489		0.01184	
	Evening	0.002489		0.02274	
Wind speed during workday	Night	-0.04236	-0.55	0.3241	4.2
	Morning	-0.04236		0.0187	
	Day	-0.04236		0.0518	
	Evening	-0.04236		0.0627	
Outdoor Relative humidity	-	-	-	-0.02261	-1.6
Solar radiation during Weekend	Night	0.001089	1.09	-0.00045	-1.7
	Morning	0.001089		-0.00167	
	Day	0.001089		-0.00086	
	Evening	0.001089		-0.00098	
Solar radiation during workday	Night	0.000482	0.48	-0.00045	-1.7
	Morning	0.000482		-0.00167	
	Day	0.000482		-0.00086	
	Evening	0.000482		-0.00098	

Figure 13 - Equations' coefficient [58]

Calì et al. [59] made their study in 90 dwellings in Germany, applying the same method of the other two models. They found that the most common driver which influence the opening was the time of the day and the carbon dioxide concentration, while the most common driver which leads to closure was the outdoor temperature, and the time of the day. The days are divided into:

- Night, low probability of action: 7 hours, between 11:00 p. m. and 5:59 a.m.
- Morning, high probability of action: 3 hours, between 7:00 a. m. and 9:59 a. m
- Rest of the day, medium probability of action: 14 hours, between 6:00 a. m. and 6:59 a. m. and between 10:59 a. m. and 22:59 p. m.

The following two equations show, respectively, the model for the opening and the closure.

$$logit(p_{op}) = \alpha - 551.15 * \frac{1}{CO_2} + 0.134 * T_{in}$$

$$logit(p_{clo}) = \alpha - 785.7 * \frac{1}{CO_2} - 0.268 * T_{in} - 0.058 * RH_{in} - 0.105 * T_{out} + 0.022 * RH_{out}$$

In which α vary with the time of the day, and its value is reported in *Table 3*

	Open	*Close*
α_{night}	-10.089	2.539
$\alpha_{morning}$	-8.214	3.317
$\alpha_{rest\ of\ the\ day}$	-7.795	3.955

Table 3 - Value of the coefficient α [59]

Lastly, Jones et al. [60] work on 10 dwellings located in Torquay (south-west of the UK). They applied the logistic regression model, and they studied the influence of the indoor and outdoor temperature and relative humidity, wind speed, solar radiation and rainfall on the two probability. In both cases, the equation coefficients vary with both the time of the day and the season.

Figure 14 shows the coefficients of the windows opening model, while the coefficients for the closing model are illustrated in *Figure 15*.

Variable		All year		Spring		Summer		Autumn		Winter	
		Coef.	Mag.	Coef.	Mag.	Coef.	Mag.	Coef.	Mag.	Coef.	Mag.
Intercept (α)	Morning			−10.126		−7.529		−18.147		−6.845	
	Afternoon			−3.837		−3.358		−17.252		−14.406	
	Evening			−6.392		−8.980		−6.914		−3.653	
	Night			−2.747		−8.580		−34.202		−18.420	
	All year	−9.275									
Indoor air temperature (°C)	Morning			0.413	5.00	0.174	2.02	0.498	6.32	–	–
	Afternoon			–	–	−0.155	1.71	0.602	7.04	0.363	4.54
	Evening			0.062	0.69	0.110	1.17	–	–	−0.117	1.60
	Night			−0.043	0.48	−0.148	1.67	0.617	7.03	0.709	9.78
	All year	0.233	3.80								
Indoor RH (%)	Morning			0.053	2.78	–	–	0.064	2.99	−0.008	0.39
	Afternoon			−0.002	0.09	–	–	0.087	3.94	–	–
	Evening			–	–	–	–	0.024	1.07	–	–
	Night			–	–	–	–	–	–	0.107	4.89
	All year	0.038	2.11								
Outdoor air temperature (°C)	Morning			−0.137	3.38	−0.151	3.40	−0.215	5.31	–	–
	Afternoon			–	–	–	–	−0.250	5.87	–	–
	Evening			–	–	–	–	−0.097	2.20	−0.285	3.02
	Night			–	–	0.269	3.90	–	–	–	–
	All year	−0.105	3.62								
Outdoor RH (%)	Morning			−0.077	3.93	–	–	–	–	–	–
	Afternoon			−0.037	0.85	–	–	−0.051	2.42	–	–
	Evening			−0.007	0.37	–	–	–	–	–	–
	Night			−0.053	1.62	–	–	0.160	4.03	−0.107	2.73
	All year	−0.042	2.45								
Wind speed (m/s)	Morning			–	–	–	–	–	–	–	–
	Afternoon			–	–	–	–	0.127	1.59	–	–
	Evening			–	–	0.164	1.67	–	–	0.191	3.74
	Night			–	–	0.301	3.34	0.149	2.22	0.054	0.88
	All year	0.057	1.27								
Global solar radiation (W/m^2)	Morning			–	–	–	–	0.002	1.72	0.003	2.08
	Afternoon			−0.001	1.14	–	–	–	–	0.004	2.77
	Evening			−0.009	1.91	–	–	–	–	–	–
	Night			–	–	0.017	3.26	–	–	–	–
	All year	–	–								
Rainfall (mm)	Morning			0.039	0.63	0.013	0.26	–	–	–	–
	Afternoon			–	–	0.058	1.21	–	–	–	–
	Evening			–	–	–	–	−0.093	1.99	0.051	1.44
	Night			–	–	–	–	–	–	–	–
	All year	0.034	0.96								

Figure 14 - Windows opening model coefficient [60]

Variable		All year		Spring		Summer		Autumn		Winter	
		Coef.	Mag.	Coef.	Mag.	Coef.	Mag.	Coef.	Mag.	Coef.	Mag.
Intercept (α)	Morning			−3.727		−8.215		−4.017		5.563	
	Afternoon			1.226		−5.032		−11.306		−14.617	
	Evening			−4.300		−9.306		−3.049		−5.852	
	Night			−1.995		−14.165		3.132		43.857	
	All year	−2.984									
Indoor air temperature (°C)	Morning			−0.102	1.23	–	–	−0.161	2.04	−0.454	5.81
	Afternoon			−0.128	1.47	−0.263	2.89	0.268	3.14	0.337	4.21
	Evening			−0.276	3.09	–	–	−0.060	0.72	–	–
	Night			−0.244	2.71	–	–	−0.667	7.60	−0.299	4.13
	All year	−0.178	2.90								
Indoor RH (%)	Morning			–	–	0.053	2.44	–	–	−0.089	4.33
	Afternoon			–	–	–	–	0.036	1.63	0.052	2.47
	Evening			–	–	0.039	1.62	−0.054	2.40	−0.038	1.84
	Night			–	–	0.055	2.10	–	–	–	–
	All year	−0.017	0.94								
Outdoor air temperature (°C)	Morning			–	–	−0.038	0.85	0.099	2.44	0.145	1.74
	Afternoon			−0.115	2.63	–	–	−0.201	4.72	−0.189	1.78
	Evening			0.239	4.90	–	–	0.124	2.81	–	–
	Night			–	–	0.216	3.13	0.204	3.81	−0.841	9.84
	All year	0.062	2.14								
Outdoor RH (%)	Morning			–	–	–	–	–	–	–	–
	Afternoon			−0.060	3.26	0.050	2.61	–	–	–	–
	Evening			–	–	–	–	–	–	–	–
	Night			–	–	–	–	–	–	−0.508	12.95
	All year	–	–								
Wind speed (m/s)	Morning			–	–	–	–	–	–	–	–
	Afternoon			0.184	2.67	–	–	0.146	1.82	0.095	2.00
	Evening			–	–	0.247	2.52	–	–	0.199	3.90
	Night			–	–	–	–	–	–	−0.449	7.32
	All year	0.063	1.40								
Global solar radiation (W/m^2)	Morning			–	–	−0.001	1.12	−0.003	2.58	–	–
	Afternoon			–	–	0.002	2.25	–	–	–	–
	Evening			0.003	0.81	–	–	−0.392	19.60	–	–
	Night			–	–	0.019	3.65	–	–	–	–
	All year	–	–								
Rainfall (mm)	Morning			–	–	–	–	–	–	0.067	1.74
	Afternoon			–	–	0.035	0.73	–	–	0.042	1.15
	Evening			–	–	–	–	–	–	–	–
	Night			0.182	1.93	–	–	–	–	–	–
	All year	0.032	0.90								

Figure 15 - Windows closing model coefficient [60]

The main problem in the application of these models is the climatic difference between locations and these differences can lead to a broad diversity between the simulated conditions and the real ones.

To show the weather difference *Table 4* presents the Heating Degree Days (HDD) in the various localities.

Locality	*HDD [°C]*
Turin - Italy	2747
Quebec City - Canada	5608
Copenhagen - Denmark	3984
Stuttgart - Germany	3573
Torquay - UK	3186

Table 4 - HDD of the different localities

To choose the model that best fits the case study, each of them was implemented in a different simulation, and through a qualitative analysis of the evolution of the window state, the one with the most realistic behaviour was chosen.

Figure 16 shows the comparison of the windows states applying the four different models. In these pictures, the open state is represented by 1, while the close state is represented by 0.

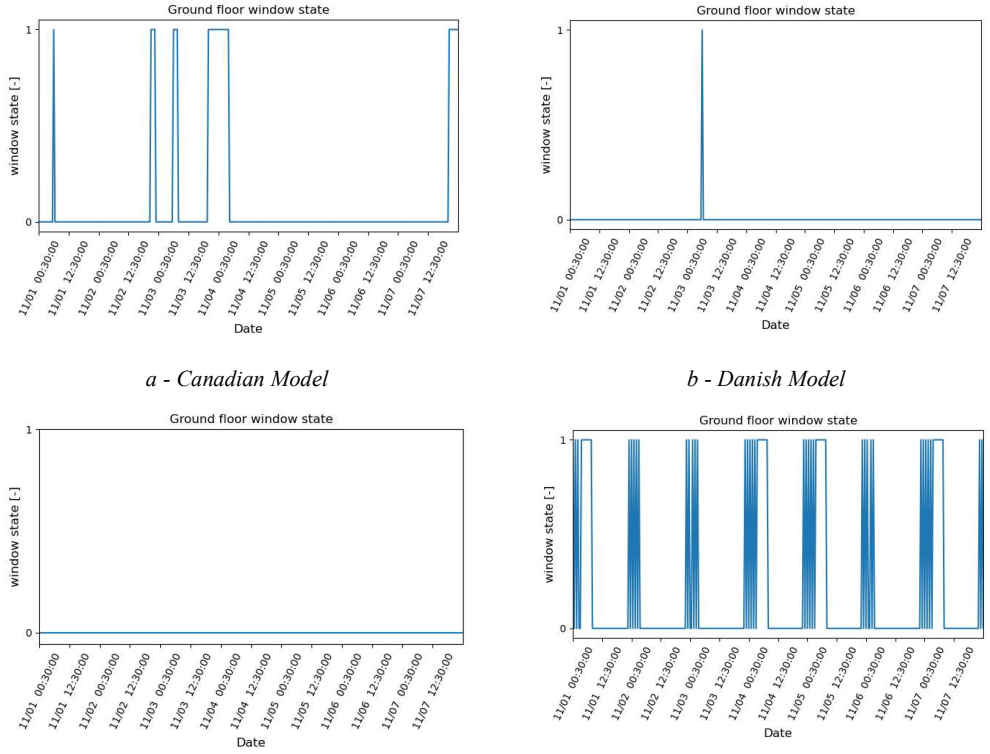
The implementation of the model proposed by Rouleau and Gosselin [57], illustrated in *figure 16a*, leads the simulation to have a prolonged open state of the windows, in contradiction with the night hour and the autumn season.

Meanwhile, the use of the model proposed by Calì et al [59] does not give good results because the state of the window never changes, in fact, it always stays closed, as can be seen in *Figure 16c*.

On the other hand, applying the English model proposed by Jones et al. [60], it results that the window state changes too frequent and the windows stay open for a long time, in contrast with the season.

The best model seems to be the Danish model proposed by Andersen et al [58] and illustrated in *Figure 16b*, which gives results consistent with reality, with less frequent openings and for a single time step.

Therefore, this last model was implemented in the simulation environment for both the training and deployment phase of the SAC control agent.



a - Canadian Model

b - Danish Model

c - German Model

d - English Model

Figure 16 - Window's model comparison

Chapter 3:
Reinforcement Learning

3. Reinforcement Learning

Reinforcement learning is a branch of Machine learning, as illustrated in *Figure 17*, with supervised learning and unsupervised learning. Concerning the other two types of machine learning, it requires inputs, and it provides outputs and a score for them [61].

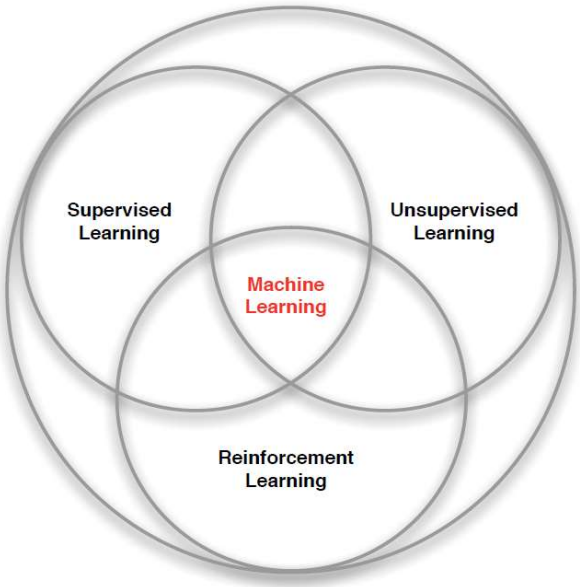


Figure 17 - Machine Learning branches [62]

Necessary for this technique is a control policy, which is learnt through interaction with the environment. Consequently, an agent is created to choose the best actions to achieve a specific objective, *Figure 18* shows a typical RL loop structure. Since RL does not require a priori known model, it is defined model-free.

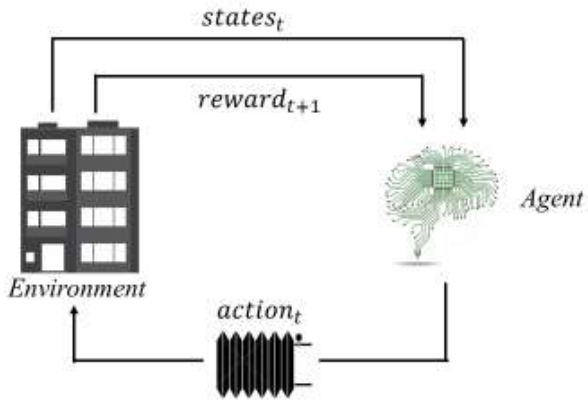


Figure 18 - Typical control loop based on RL

RL is based on the Markov Decision Property (MDP), in which the future step is independent on what has already happened; so, it is conditioned only on the current level. In particular, both the reward and the transition probability between two states depend on the actual state and the chosen action.

In the MDP, the mathematical formalization of the interaction between environment and agent is described by the following components [50]:

- Action space ($a \in A$), which is the set of all possible actions. The agent selects one of them at each time step;
- State space ($s \in S$), which is the set of all possible environment's state. It can be divided into time-dependent, controllable and exogenous (uncontrollable) state information;
- Reward (r), which is a scalar value released by the environment after the evaluation of both the action chosen and the new state;
- Policy (π), that is a mapping between states and the probability of selecting each action. As a result, the agent aims to learn the optimal policy.
- Transition probability distribution, which represents the likelihood of the transition to the next state when the agent is in the initial state.

Furthermore, the object of the control agent is learning the optimal policy that maximizes the total reward/return.

In addition, the two value-functions called the state-value and the action-value, respectively, are very valuable to determine the optimal policy [50]:

- The first one represents the expected return of the agent, starting from a state *s* and following the policy π: [34]

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

  where $\boldsymbol{\gamma}$, which is ranged between [0,1], is the discount factor for future rewards:

  a) if $\boldsymbol{\gamma}$=0, the agent gives greater importance to immediate reward, neglecting the future one.

  b) if $\boldsymbol{\gamma}$=1, the agent provides more weight to the future reward.

- The second one represents the expected return of the agent when it takes the action *a*, being in a certain state *s* and following the policy π: [34]

$$q_{\boldsymbol{\pi}}(s, a) = E[rt+1+\gamma q_{\boldsymbol{\pi}}(s', a')|St=s, At=a]$$

These two functions are obtained through the experience of the agent, and they are updated online during the training phase.

The RL agent is trained through a trial-and-error approach by a technique called on-policy learning, which means that after having tried and evaluated the performance of various policies, it improves them as much as possible. On the other hand, in the analogue method named the off-policy learning, the agent learns from other policies already created for other cases but, the main issue is the lack of skill to explore the action space.

Furthermore, all RL problems can be divided into two main categories [63]:

- Episodic problems have one or more terminal states. An episode is repeated many times through the agent's training phase in order to explore all possible states' combination and rewards. So, when an agent reaches a certain state, the episode ends, the environment will be reset to the initial state, and a new episode starts;
- Continual problems do not end, and they continue indefinitely.

One of the peculiarities that characterize reinforcement learning is the compromise between exploration and exploitation for the action-selection, which has to be optimised by a right control agent.

During the exploration phase, the agent selects new random actions by neglecting the maximization of the reward, while, during the exploitation, the agent selects actions already undertaken in order to maximize the rewards.

The exploration phase is more focused on the initial phase when the agent explores all action-space. Consequently, after a certain period, the exploitation becomes more significant in order to reach the objective.

Moreover, to balance these two main actions, two methods can be witnessed:

- In the ε-greedy, the agent would choose the currently known action with the highest estimated value with a probability of 1-ε and selects a random action with the probability of ε [64]. ε represent the exploration rate, and it can decrease over time in order to support the exploitation phase. This method can be described by the following two equation:

$$P\left(a_i = argmax\big(Q(a_i)\big)\right) = 1 - \epsilon$$

$$P(a_i = random) = \epsilon$$

- The Soft-max method selects the action based on the action's performance and τ, which is the Boltzmann temperature constant. The agent tends to exploit more when most of the action space has been explored already [64]. The next equation represents this method:

$$P(a_i) = \frac{\exp\left(\frac{Q(a_i)}{\tau}\right)}{\sum_{i=1}^{n} exp\left(\frac{Q(a_i)}{\tau}\right)}$$

3.1 Q-Learning

Q-learning is one of the most popular methods of model-free RL. It belongs to the Temporal Difference (TD) technique, and it is used when the model issues incomplete data.

The TD methods, in comparison to the other two RL techniques (Monte-Carlo (MC) and Dynamic Programming (DP)) converge towards an optimal policy faster [63].
In Q-learning all transition are represented by a table, called Q-Table, in which each entry represents a state-action tuple, and then state-action value or Q-Value is stored.
Overall, Q-learning tries to evaluate the Q-values from experience, and they are updated according to Bellman's equation:

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

Where α, which is the learning rate, behaves between [0,1]. It determines with which capacity new information overrides old knowledge, for example, α = 1 means that the new data overrides completely the old one, while α = 0 means that no learning occurs [34].

3.2 Deep Q-Learning

Since Q-learning involves tables to store and retrieve state-action values, in which each entry denotes a state-action tuple (s, a), the representation may be unfeasible in a real problem where action and state spaces are wide [34].
For this reason, to improve this technique, the Deep Q-Learning can be used, in which a function approximator allows state-action values to be represented by using a fixed amount of memory. In particular, by using Deep Neural Networks (DNN) as a function approximator, it changes Q-Learning into Deep Q-Learning, or Deep Q-Network.
Moreover, 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, and between them, there are hidden bands that receive information from the previous one. *Figure 19* shows an example of DNN of N hidden layer.

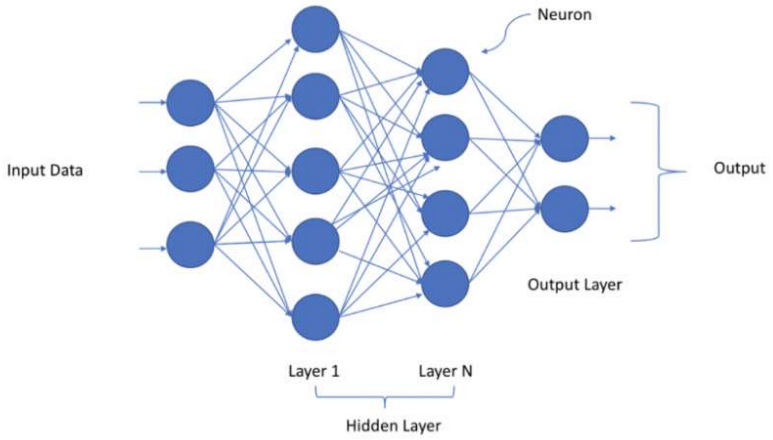


Figure 19 - Example of Deep Neural Network [65]

Moreover, the Q-values are indicated with the following formula, taken from [66]:

$$Q(s,a) = Q(s,a,\theta)$$

The equation represents the Q-network, in which the term θ, which is the weights of the network, parameterizes the Q-value function. The neurons' number in the input layer is equal to the variables' number that composes the state space, while the number of neurons in the output layer corresponds to the size of the action space. [34].

This structure, which is represented in *Figure 20*, is helpful because the network allows learning the relation between states and the Q-value for each action, which is unknown a priori, and it is learnt over successive interaction with the environment. Overall, the Q-value is updated following the Bellman's equation (introduced in section 2.3).

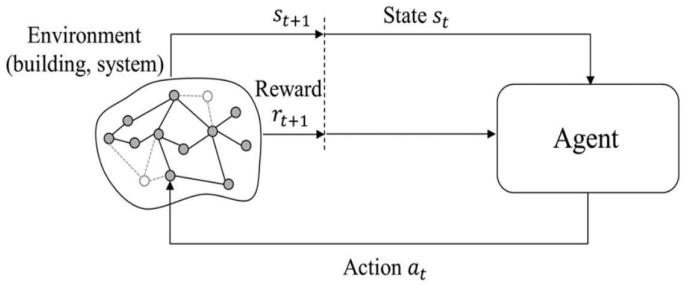


Figure 20 - Reinforcement Learning Deep Q-Network [30]

3.3 Soft Actor-Critic

Model-free deep reinforcement learning (RL) algorithms suffer from two significant hurdles, very high sample complexity and brittle convergence properties, which necessitate meticulous hyperparameter tuning. Both of these challenges severely limit the applicability of such methods to complex, real-world domains.

In particular, algorithm like TRPO (Trust Region Policy Optimization) and PPO (Proximal policy optimization) have stochastic policies, and they use on-policy process to improve them. Besides, they suffer from low sample efficiency because they require new samples to be collected after each policy update.

On the other hand, algorithm like DDPG (deep deterministic policy gradient) and TD3 (Twin Delayed DDPG) use deterministic policies, and they adopt off-policy approach for the optimization. Compared to previous algorithms, they present a better sample efficiency, thanks to the replay buffer, but they are extreme brittleness and suffer from hyperparameter sensitivity [67].

To try to overcome these obstacles, a new algorithm, called Soft Actor-Critic (SAC), is proposed like a combination of the properties of the two previous group. SAC is an off-policy algorithm which combines stochastic policy and replay buffer, and it introduces the entropy regularization. This algorithm allows the use of continuous action space instead of the discrete one used in traditional RL algorithm. SAC aims to maximize a new target function composed of two term, the expected reward and the entropy term. This last term expresses the attitude of choosing random actions.

High entropy is necessary to encourage exploration, to promote the policy to assign same probabilities to actions with same Q-values and to guarantee that it does not always select a particular action that could lead to inconsistency in the approximated Q function. Consequently, SAC supports the policy network to

explore and not assign a very high probability to any one part of the range of actions [67].

As proposed by Haarnoja et al. [68], the soft actor-critic algorithm includes three principal elements: an actor-critic architecture with separate policy and value function networks, an off-policy formulation that allows reuse of the previous sample, and entropy maximization to encourage stability and exploration. The structure of actor and critic neural networks are shown in *Figure 21*.

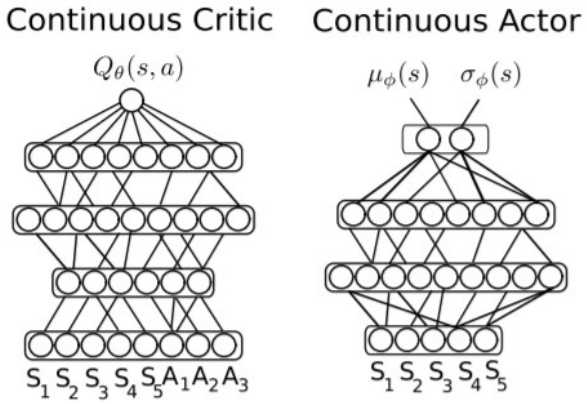


Figure 21 - Actor and critic neural network structure [69]

The maximum entropy objective requires an optimal policy π^* like this:

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

In which α, the temperature parameter, defines the relative importance of the entropy term against the reward, and therefore controls the stochasticity of the optimal policy.

In order to find the optimal policy, SAC uses three function approximator and the parameters of these function are ψ, θ, and φ. The parameterized function are respectively the state value function $V_\psi(s_t)$, a soft Q-function $Q_\theta(s_t, a_t)$, and a tractable policy $\pi_\varphi(a_t \mid s_t)$. The algorithm requires the train of three functions:

1. . The soft value function is trained to minimize the squared residual error:

$$J_V(\psi) = E_{s_t \sim \mathcal{D}} \left[\frac{1}{2} \left(V_\psi(s_t) - E_{a_t \sim \pi_\phi} \left[Q_\theta(s_t,a_t) - \log \pi_\phi(a_t|s_t) \right] \right)^2 \right]$$

where $\mathcal{D}$ is the distribution of previously sampled states and actions, or a replay buffer.

2. The soft Q-function parameters can be trained to minimize the soft Bellman residual:

$$J_Q(\theta) = E_{(s_t,a_t)\sim\mathcal{D}}\left[\frac{1}{2}\left(Q_\theta(s_t,a_t) - \hat{Q}(s_t,a_t)\right)^2\right]$$

3. The policy parameters can be learned by minimizing the expected KL-divergence:

$$J_\pi(\phi) = E_{s_t\sim\mathcal{D}}\left[D_{KL}\left(\pi_\phi(\cdot\,|s_t)||\frac{exp\left(Q_\theta(s_t,\cdot)\right)}{Z_\theta(s_t)}\right)\right]$$

Where D_{KL} is the Kullback - Leibler Divergence, and Z_θ is the normalization function.

Chapter 4:
Framework of the analysis and Case Study

4. Framework of the analysis and Case Study

In this section the methodological framework is illustrated with the goal of introducing the stages of the SAC control agent development. The present framework is based on three different levels as shown in *Figure 22*.

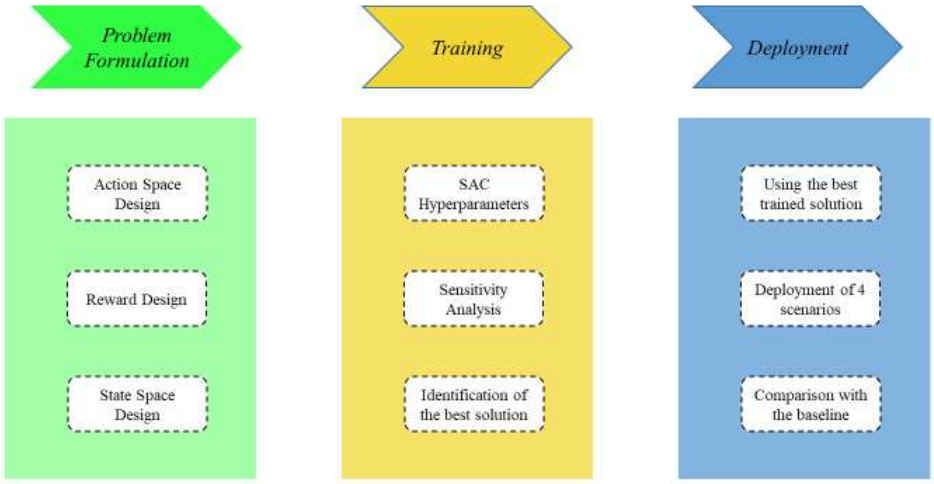


Figure 22 - Framework of the application of SAC control.

The first phase of the framework is the problem formulation, which has the aim to define the principal components of the learning problem. The action-space includes all the possible control actions that the agent can take. The reward is the function that describes the performance of the control agent concerning the objectives. Lastly, the state-space is a set of variables that describe the environment. These variables are sent to the control agent.

The second stage of the procedure is the training phase in which the control agent was trained. In this phase, a sensitivity analysis was performed on the most relevant hyperparameters by training the control agent with different configurations.

The training process repeats multiple time a training episode in order to improve the agent's control policy. At the end of this phase, after the comparison between different solutions and baseline, the best configuration was selected.

In the last phase the trained agent was tested through a static deployment in one episode of a different period from the training episode. The deployment was carried

out in four different scenarios. Finally, a comparison between the different scenarios and the baseline was performed.

4.1 Case Study

The following subsections provide the description of the simulation environment and the building under study. In particular, section 4.1.1 defines the interaction and the exchange of data between the simulation environment and the control agent, section 4.1.2 illustrates the geometric model of the building, while in section 4.1.3 describes the construction of the building model. Finally, the building's heating system and its control logic is described in section 4.1.4.

4.1.1 Description of the simulation environment

The interaction between the control agent and the building is simulated within a surrogate environment which connects EnergyPlus and Python.

The EnergyPlus model of the building is enveloped in the Python interface, based on OpenAI Gym. Through this program, a SAC control agent, always developed in Python, can virtually interact with a simulated building to learn the optimal control policy. The whole environment is based on the Building Control Virtual Test Bed (BCVTB) and the External Interface function of EnergyPlus.

Lastly, the interaction between the two software is dynamic, and during the simulation, a continuous exchange of data takes place.

The temporal features, which are characteristic of the data transaction, are:

- Control time step that represents the time step in which the agent takes action, in this case, the control time step is set equal to 30 minutes;
- Simulation time step, which is defined in the Energy Plus environment. It is defined equal to 30 minutes;

- Episode, which represents simulation's duration performed by EnergyPlus. During the training phase, one episode is repeated multiple time in order to explore different paths. In this case, a training episode lasts two months.

For the exchange flow of data between the control agent and Energy plus simulation, the model proposed by [34] is used, and it is illustrated in *Figure 23*, in which the green lines illustrate the exchange of data between Python and EnergyPlus that is managed using *BCVTB*.

In particular, the loop is characterized by four functions in Python:

- Init(), a function used for the initialization of the environment. Every simulation starts with this function;
- Step(), which receives the action selected by the agent and translates the encoded value into a physical control action. This function also returns four objects: next state, reward, done (True/False) and info;
- Reset(), that is called at the beginning of each episode to re-initializes the Energy plus simulation process and returning the first state of the environment;
- Render(), a function used to render one frame of the environment.

The simulation start with the initialization of the OpenAI gym environment using the init() function, therefore a socket server is created for the communication between EnergyPlus and Python.

After, at the beginning of each episode, the reset() function is called to re-initialize the Simulation process and to return the initial state of the environment. The physical state ,which is provided by Energy plus, is processed before the communication with the SAC control agent that receive the processed state and the reward (it is used as a feedback signal) and it chose one of the possible action. The step() function translate the chosen action into a physical control action This action is passed to EnergyPlus as a schedule value through the ExternalInterface function to simulate the next control step.

If the episode is the last one, the process ends here, otherwise a new episode start from the reset() function.

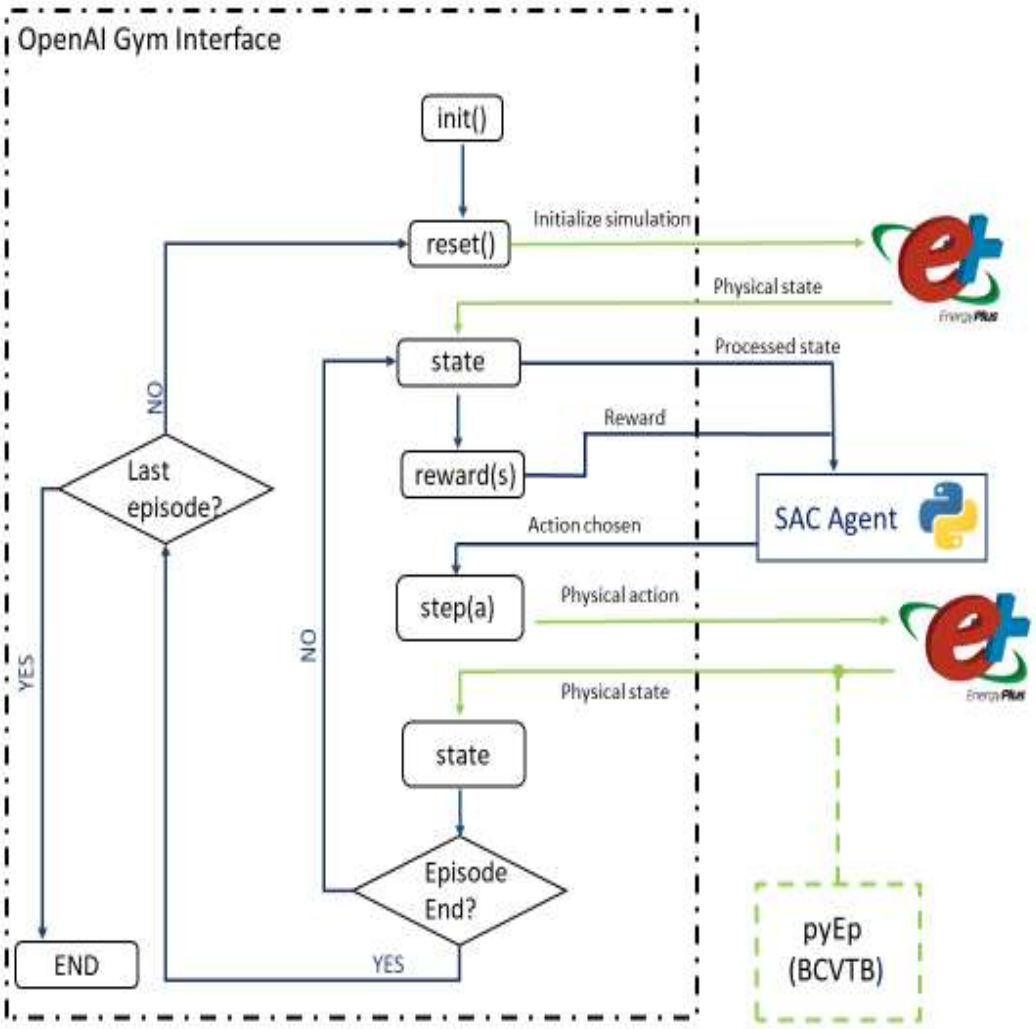


Figure 23 - Simulation environment for SAC controller [34]

4.1.2 Geometric model

The geometric model of the building has been created with SketchUp 2016, with the help of the OpenStudio's tool. They allow the development of the structural model of the building, which simplifies the development of the energy model implemented.

The building under study is located in Turin, Italy, and it is representative of a big portion of the Italian building stock in terms of both heating system configuration and building construction characteristics. It is a five-level building with a net heated surface of 527 m^2, and each floor represent a thermal zone, *Table 5* shows both the volume and the floors' surface for each level.

	Volume [m^3]	Surface [m^2]
Ground Floor	562.02	140.50
First Floor	371.97	96.61
Second Floor	294.67	96.61
Third Floor	280.18	96.61
Fourth Floor	193.23	96.61

Table 5 - Building parameters

As a building of old construction, it has transparent and opaque envelope components with poor energy performance, in particular, the transmittance values for the principal components are shown in *Table 6*, and they are compared with the limits value imposed by standards [70].

Component	U-Value [W/m^2K]	Limit U-Value [W/m^2K]
External Walls	0.985	0.30
Partition Walls	2.174	0.80
Horizontal partition	1.376	0.80
Roof	1.215	0.26
Windows	2.681	1.90

Table 6 - Building's components U-value

For a better understanding, *Figure 24* witnesses the developed model, in which the various floors are easily recognizable, and it is already possible to distinguish opaque walls, windows and ceiling.

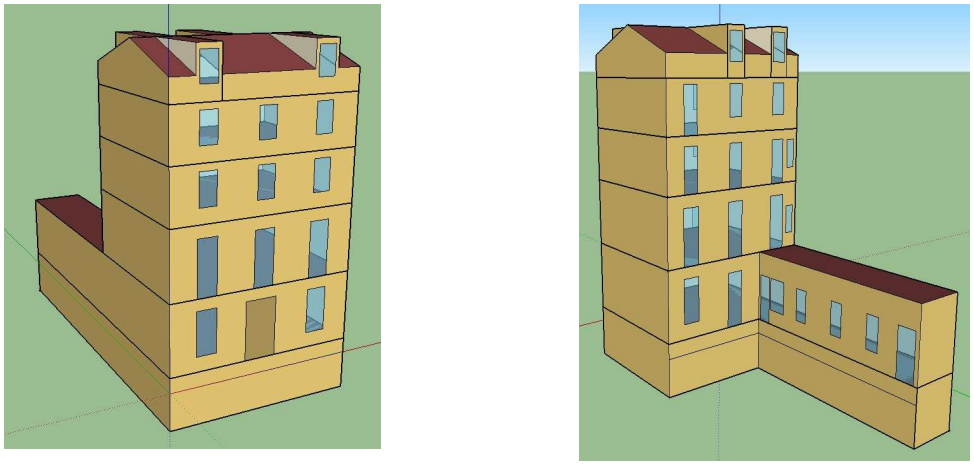


Figure 24 - Geometrical model

Figure 25 represents the boundary condition for each surface required for the energy balance calculation. Blue surfaces indicate the external ones, while the browns are in contact with the ground. Furthermore, boundary conditions for external walls, that are in contact with nearby buildings, have been considered adiabatic supposing that the other structures are also subjected to air conditioning, and they are represent by pink surfaces.

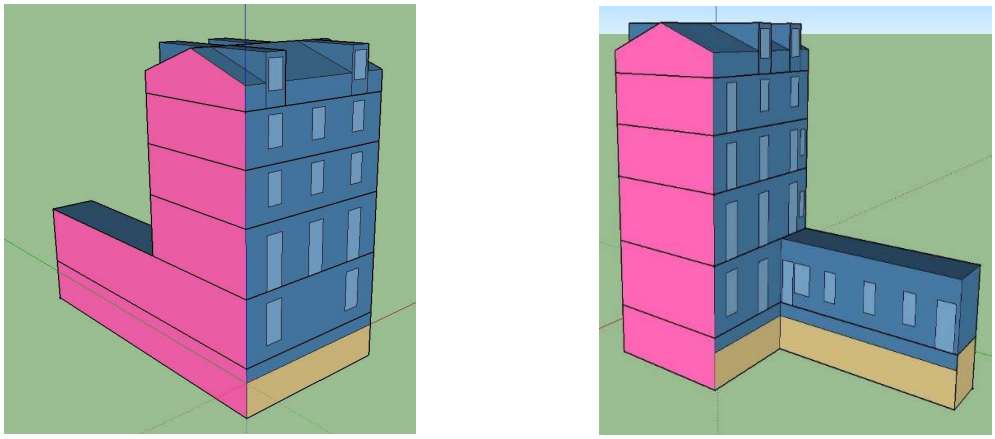


Figure 25 - Building's boundary conditions

4.1.3 Energy Model

The designed geometrical model was used as a base for the construction of the energy model in EnergyPlus. The simulation was carried out during the heating

season, which in Turin goes from 15th October to 15th April, and the simulation time step is defined equal to 30 minutes.

To complete the definition of the building, there were inserted every material presents in the construction. For these materials are defined some properties, which are fundamental to determine the dynamics of the building, such as density, conductivity and the specific heat. After, the constructions are created using the materials according to the following stratigraphies.

The external walls are composed of two layers of lime plaster and one layer of brick. there is not the presence of some insulant materials, in fact, the transmittance value for this type of construction (0.985 W/m^2K) results higher than its limit value (0.300 W/m^2K). *Table 7* summarises the thermal properties of the external walls.

Layer	Thickness [mm]	Density [kg/m^3]	Conductivity [W/mK]	Thermal resistance [m^2k/W]	Specific heat [J/kgK]
Internal Surface R_t	-	-	-	0.13	-
Lime plaster	15	1800	0.90	0.017	840
Brick	520	1800	0.72	0.722	1000
Lime plaster	15	1800	0.90	0.017	840
External Surface R_t	-	-	-	0.04	-

Table 7 - External walls Stratigraphy

Table 8 highlights the roof stratigraphy for this building. It is a wooden roof with two layers of air gap, and the outside layer is made of tile. Also this construction presents a U-Value (1.215 W/m^2K) higher than the limit one (0.26 W/m^2K).

Layer	Thickness [mm]	Density [kg/m^3]	Conductivity [W/mK]	Thermal resistance [m^{2}k/W]	Specific heat [J/kgK]
Internal Surface R_t	-	-	-	0.10	-
Lime plaster	15	1800	0.90	0.017	840
Spruce	30	450	0.12	0.250	1380
Air gap	280	1	1.88	0.149	1000
Spruce	20	450	0.12	0.167	1380
Air gap	40	1	0.50	0.080	1000
Tile	15	1800	0.72	0.021	1000
External Surface R_t	-	-	-	0.04	-

Table 8 - Roof Stratigraphy

The internal vertical partition have a construction similar to the exterior walls one, the only difference is the thickness of the brick layer, that in this case, it is 120 mm. Since the absence of insulation material and the small thickness of the wall, this construction presents a very high transmittance value (2.174 W/m^{2}K) compared to the limit one (0.80 W/m^{2}K). *Table 9* shows the properties of this type of construction.

Layer	Thickness [mm]	Density [kg/m^3]	Conductivity [W/mK]	Thermal resistance [m^{2}k/W]	Specific heat [J/kgK]
Internal Surface R_t	-	-	-	0.13	-
Lime plaster	15	1800	0.90	0.017	840
Brick	120	1800	0.72	0.722	1000
Lime plaster	15	1800	0.90	0.017	840
External Surface R_t	-	-	-	0.13	-

Table 9 - Partition walls stratigraphy

In the horizontal partitions host the radiant floor system, which provide the power to each floor. The stratigraphy of this construction is described in *Table 10.* The pipes of the heating system are located in the screed layer. The construction has a

transmittance value of 1.376 W/m^{2}K, and also in this case, it is higher than the limit value (0.80 W/m^{2}K).

Layer	Thickness [mm]	Density [kg/m^3]	Conductivity [W/mK]	Thermal resistance [m^{2}k/W]	Specific heat [J/kgK]
Internal Surface R_t	-	-	-	0.13	-
Lime plaster	15	1800	0.90	0.017	840
Block brick	200	1100	0.598	0.334	1000
Concrete	80	2200	1.65	0.048	1000
Screed	55	1700	1.06	0.052	1000
Tile	15	2300	1.00	0.015	840
External Surface R_t	-	-	-	0.13	-

Table 10 - Horizontal partition stratigraphy

Finally, the windows are composed by a two-layer glazing of 4 mm and an internal air gap of 12 mm. The resulting transmittance value of the glass is 2.849 W/m^{2}K. The windows have a wooden frame with a U-Value of 1.767 W/m^{2}K. In the calculation of the total window's transmittance, it is also taken into account the linear thermal bridge caused by the connection between the frame and the glazing ψ_g of 0.06 W/mK. Therefore, the total U-Value of the windows can be obtained by applying the following equation:

$$U_w = \frac{U_g * A_g + U_f * A_f + \psi_g * l_{tb}}{A_g + A_f}$$

The resulting transmittance is 2.681 W/m^{2}K that do not respect the limit value imposed by the standards.

Table 11 shows the thermal properties of the window's glazing stratigraphy.

Layer	Thickness [mm]	Density [kg/m^3]	Conductivity [W/mK]	Thermal resistance [m^{2}k/W]	Specific heat [J/kgK]
Internal Surface R_t	-	-	-	0.13	-
Glass	4	2500	1.00	0.004	840
Air gap	12	1	0.025	0.173	1010
Glass	4	2500	1.00	0.004	840
External Surface R_t	-	-	-	0.13	-

Table 11 - Window stratigraphy

Figure 26 and *Figure 27* show an example of the material and construction definition in EnergyPlus.

```
Material,
    brick,                  !- Name
    MediumRough,            !- Roughness
    0.52,                   !- Thickness {m}
    0.72,                   !- Conductivity {W/m-K}
    1800,                   !- Density {kg/m3}
    100;                    !- Specific Heat {J/kg-K}
```

Figure 26 - Example of the Material definition

```
Construction,
    roof,                   !- Name
    roof tile,              !- Outside Layer
    air 40mm,               !- Layer 2
    spruce 20mm,            !- Layer 3
    air 300mm,              !- Layer 4
    spruce 30mm,            !- Layer 5
    lime plaster;           !- Layer 6
```

Figure 27 - Example of the Construction definition

In order to simplify the design of the energy model, the windows are defined using the object *WindowMaterial:SimpleGlazingSystem*, in which the global U-value of the windows is added. *Figure 28* shows an example of the definition of the window property.

```
WindowMaterial:SimpleGlazingSystem,
    glazing,                !- Name
    2.681,                  !- U-Factor {W/m2-K}
    0.75;                   !- Solar Heat Gain Coefficient
```

Figure 28 - Example of window definition

Furthermore, the building was split into five thermal zones, each of them corresponding to a different floor. Each thermal zone is characterised by its own schedules and temperature setpoint.

Since the presence of occupants in the building is stochastic, different schedules for the occupancy is implemented on each floor. The schedules differ in both the arrival and leaving hours and the number of people in the zone; this is done to represent different occupants' behaviour. It has been hypothesised that in each flat there is a family of four people. *Table 12* illustrates the different occupancy schedule of each floor.

Floor	Range	Number of people
Ground floor	7.00 – 17.00	0
	17.00 – 19.00	1
	19.00 – 7.00	4
First floor	8.00 – 19.00	0
	19.00 – 22.00	3
	22.00 – 8.00	4
Second floor	6.00 – 7.00	3
	7.00 – 16.00	0
	16.00 – 18.00	2
	18.00 – 6.00	4
Third floor	6.00 – 8.00	2
	8.00 – 18.00	0
	18.00 – 20.00	3
	20.00 – 6.00	4
Fourth floor	7.00 – 16.00	0
	16.00 – 18.00	2
	18.00 – 7000	4

Table 12 - Occupancy schedules

The windows' state, open or closed, is passed to EnergyPlus with a schedule that is updated each simulation timestep according to the model proposed by Andersen et al. [58], which can be represented by the following equations:

$$\log\left(\frac{p_{op}}{1-p_{op}}\right) = \alpha_0 + \alpha_1 T_{in} + \alpha_2 CO_{2,in} + \alpha_3 T_{out} + \alpha_4 v_{wind} + \alpha_5 I_{dir}$$

$$\log\left(\frac{p_{cl}}{1-p_{cl}}\right) = \alpha_0 + \alpha_1 T_{in} + \alpha_2 RH_{in} + \alpha_3 CO_{2,in} + \alpha_4 T_{out} + \alpha_5 v_{wind} + \alpha_6 RH_{out} + \alpha_7 I_{dir}$$

Where the coefficients α_i can be find in *Figure 13.* These values change with both the hour of the day and the day of the week.

Figure 29 shows the Python function used for the implementation of the two windows' equations.

```
def popen(time, tin, co2, tout, wind, solar):
    if time <= 11 and time >= 4:  # morning
        popening = 1/(1+math.exp(4.85-0.009908*tin-0.000566*co2-0.043587*tout+0.04236*wind-0.000482*solar))
    elif time > 11 and time <= 18:  # day
        popening = 1/(1+math.exp(6.44-0.07336*tin-0.000158*co2-0.012418*tout+0.04236*wind-0.000482*solar))
    elif time > 18 and time <= 22:  # evening
        popening = 1/(1+math.exp(6.38-0.0011616*tin-0.001134*co2-0.026525*tout+0.04236*wind-0.000482*solar))
    else:  # night
        popening = 1/(1+math.exp(8.32-0.002585*tin-0.001018*co2-0.060408*tout+0.04236*wind-0.000482*solar))
    return popening


def pclose(time, tin, rhin, co2, tout, wind, rhout, solar):
    if time <= 11 and time >= 4:  # morning
        pclosing = 1/(1+math.exp(-5.77+0.3025*tin-0.3942*rhin+0.00059*co2+0.267*tout-0.0187*wind+0.02261*rhout+0.00167*solar))
    elif time > 11 and time <= 18:  # day
        pclosing = 1/(1+math.exp(-7.35+0.1871*tin-0.3942*rhin+0.00179*co2+0.2153*tout-0.0518*wind+0.02261*rhout+0.00086*solar))
    elif time > 18 and time <= 22:  # evening
        pclosing = 1/(1+math.exp(-6.56+0.2357*tin-0.3942*rhin+0.00039*co2+0.2019*tout-0.0627*wind+0.02261*rhout+0.00098*solar))
    else:  # night
        pclosing = 1/(1+math.exp(3.88+0.8107*tin-0.3942*rhin+0.0037*co2+0.5343*tout-0.3241*wind+0.02261*rhout+0.00045*solar))
    return pclosing
```

Figure 29 - Windows' model equations

Through a probabilistic function, the model will receive the value 1 if windows are open, and 0 if windows are closed. The state value of the windows is determined by the Python function, which is illustrated in *Figure 30*, and it is passed through BCVTB to the EnergyPlus's schedule that regulates the *ZoneVentilation:DesignFlowRate*.

In this function the variable *pc* verifies the presence of occupants in the thermal zone, *ss* indicates the state of the window, *ww* is the schedule value to be sent to the building model, and *pp* is the value obtained by the application of the equation. Being a probabilistic model, the value of *pp* is compared with a random number,

which is generated each timestep, and if the probability is higher than the random number, the state of the window changes.

```
def windows(pc, ss, time, tin, rhin, co2, tout, wind, rhout, solar):
    if pc >= 1:
        if ss == 0:
            pp = popen(time, tin, co2, tout, wind, solar)
            if pp >= rand(1):
                ss = 1
                ww = 1
            else:
                ss = 0
                ww = 0
        else:
            pp = pclose(time, tin, rhin, co2, tout, wind, rhout, solar)
            if pp >= rand(1):
                ss = 0
                ww = 0
            else:
                ss = 1
                ww = 1
    else:
        ww = 0
        ss = 0
    return ww, ss
```

Figure 30 - Implementation of windows' model

The application of the windows' model will mainly affect natural ventilation losses and consequently, the load required for heating. Each thermal zone has its own schedule independent from the others.

The air flow due to the opening of the window was modelled in EnergyPlus by using the object *ZoneVentilation:DesignFlowRate*, assuming a natural air flow of 2 Air Changes per hours (ACH):

$$\dot{V}_{nat} = 2\ ACH$$

These air flows are considered only when the windows' model returns the open state for the windows on each floor. The windows states are updated every 30 minutes, corresponding to the simulation timestep. Therefore, if the window is open, there will be a natural ventilation heat loss which lasts until the state change again.

The other sources of internal heat gain considered are the lights and the electric equipment. Both are controlled by two schedules, which regulates the heat gain coherently with the presence of occupants.

Concerning the lights, it has been assumed that in the entire building are installed standard light with a lighting power of 7 W/m^2.

Infiltrations, which are the unplanned air flows from the external environment directly into the thermal zone. Are generally caused by the uncorrected sealing of windows and doors or through building elements. When the outdoor temperature is lower than the internal one, the infiltrations lead to heat loss. In this energetic model they are modelled by using the object *ZoneInfiltration:DesignFlowRate,* and it has been hypothesised that the air flow is equal to 0.1 ACH in each thermal zone:

$$\dot{V}_{inf} = 0.1ACH$$

Since the objective of the SAC control agent will be the maintaining of the temperature in the acceptability range by controlling the supply power to be provided in each thermal zone, the simplest way to regulate this power is the introduction of an internal source of heat gain in each zone. The EnergyPlus objected *OtherEquipment* is used. The designed power level in Watt, defined in this object, must be equal to the maximum power that can be supplied by the existing system in that zone. The schedule linked to this object the schedule related to this object will specify the percentage of power to be supplied in that timestep. *Figure 31* forgives an example of the definition of the *OtherEquipment* object.

```
OtherEquipment,
    gf_load,                 !- Name
    ,                        !- Fuel Type
    ground floor,            !- Zone or ZoneList Name
    BCVTB_gf_equip,          !- Schedule Name
    EquipmentLevel ,         !- Design Level Calculation Method
    11000,                   !- Design Level {W}
    ,                        !- Power per Zone Floor Area {W/m2}
    ,                        !- Power per Person {W/person}
    ,                        !- Fraction Latent
    ,                        !- Fraction Radiant
    ;                        !- Fraction Lost
```

Figure 31 - Example of OtherEquipment definition

However, for the application of this strategy in real cases, the control of the supply power depends on the variable that the control system can regulate, such us flowrate or supply temperature.

4.1.4 Heating system and Baseline control logic

The building is heated through a radiant floor heating system. The hot water loop is composed of a gas-fired boiler of 70 kW, a variable speed pump and a collector to separate the flow in the five radiant floors. A simplified scheme of the building's heating system is provided by *Figure 32*

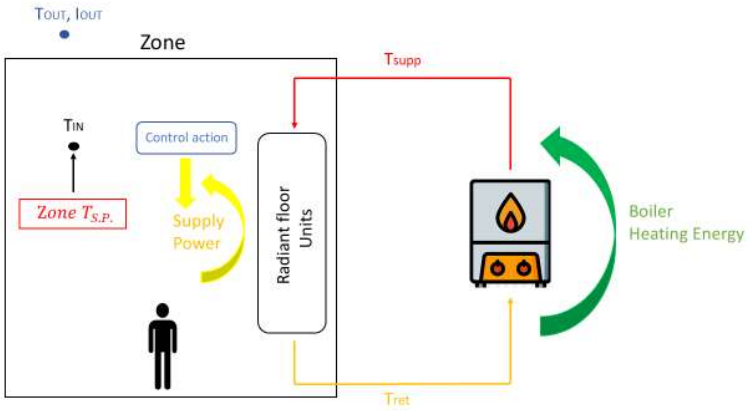


Figure 32 - Case study heating system

Since the radiant floor can only operate on the sensible load, and the comfort parameter is not monitored, the work focuses on the thermal zone internal temperature.

The baseline control logic is a combination of rule-based and climatic-based for the control of the supply power.

The climatic curve follows a step function in which the fraction of nominal power depends on the outdoor air temperature. *Figure 33* forgives a graphical representation of this function.

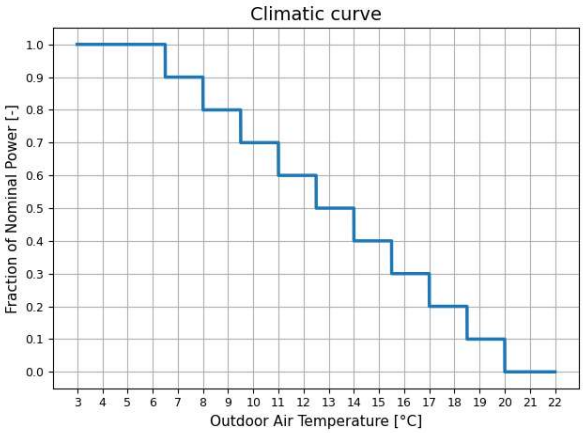


Figure 33 - Baseline Logic - Climatic curve

The time in which the system is switched on/off is based on indoor temperature when there is the presence of occupants, following this logic:

- The system is switched on two hours before the arrival of people;
- If the indoor temperature is larger than 21 °C, the system is switched off;
- If the indoor temperature is less than 19 °C, the system is switched on;
- If there are no occupants, the system is switched off.

Chapter 5:
SAC Development

5. SAC development

In this chapter will be described the design of the principal elements of the SAC control algorithm and it will introduce the methodologies used in the training and deployment phases.

5.1 Design of SAC control problem

The Soft Actor-Critic control algorithm described in chapter 2 is trained and tested in a developed simulation environment. On the other hand, the design of the action space, the reward function and the state-space in the next sub-sections are discussed.

5.1.1 Design of action-space

Since the SAC is chosen as control agent, the action-space is shipped in a continuous space. Every control time steps, the agent selects a value of the supply power for each floor.

The action-space includes the following actions related to the supply power (SP) in kW:

$$A_{gro\quad floor} = 0 \leq SP_{ground\ fl.} \leq 11$$

$$A_{first\ floor} = 0 \leq SP_{first\ fl.} \leq 6.5$$

$$A_{second\ floor} = 0 \leq SP_{second\ fl.} \leq 5.0$$

$$A_{third\ floor} = 0 \leq SP_{third\ fl.} \leq 5.0$$

$$A_{fourt\ \ floor} = 0 \leq SP_{fourth\ fl.} \leq 6.5$$

These values are selected to provide to the SAC agent the same range of supply power as the baseline controller. Furthermore, the simulation environment is set to shut down the system when the supply power reaches a value below the 30% of the nominal power.

5.1.2 Design of reward function

The reward that the agent receives after having taken actions at each control time step depends on two competing values: the energy and temperature-related terms. The energy-related one is proportional to the energy provided to each floors to reach the desired setpoint, while the temperature-related is quadratically proportional to the distance between zone air temperature setpoint and its actual value on each floor.

The coefficient δ and β are introduced to weight the importance of the two terms of the reward function. The weight factor β determines the relative importance of indoor temperature requirement concerning energy consumption. A high value of this factor guarantees lower temperature violations at the expense of lower energy-saving and vice-versa.

While the energy-related term is always present, the temperature-related one is inserted when there is the presence of occupants inside the zone and the temperature falls outside the acceptability range.

The following equation expresses the reward function:

$$R = \begin{cases} -\delta * \dfrac{\sum Supply\ Energy_i}{3600000} - \beta * \sum (T_{S.P.} - T_i)^2 & if\ occ \geq 1 \\ -\delta * \dfrac{\sum Supply\ Energy_i}{3600000} & if\ occ = 0 \end{cases}$$

Figure 34 shows the structure of the reward function graphically to facilitate its understanding.

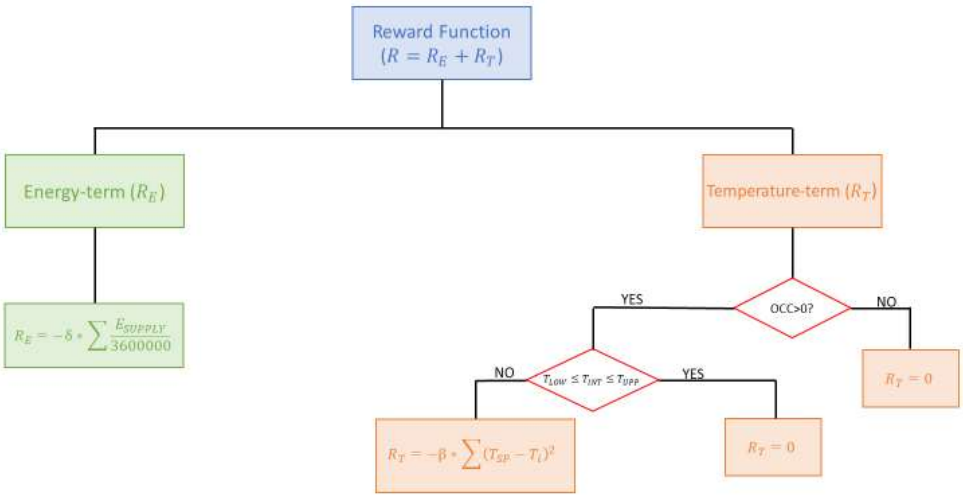


Figure 34 - Reward function structure

5.1.3 Design of state-space

The state represents the environment as the control agent observes it. The agent, at each control time step, chooses among the available actions the best one according to the values assumed by the state. The variables are selected in line with the following criteria:

- The variables must provide to the agent all the necessary information to predict immediate future rewards;
- The variables must be feasible to be collected in a real-world implementation.

The set of variables are shown in *Table 13*.

External Air Temperature and Direct Solar Radiation are inserted because they are exogenous factors with a significant impact on energy consumption and consequently on the indoor air temperature.

Information about Internal Air Temperature of each floor is given as the difference between the selected setpoint and the temperature himself because this term is direct associated to the temperature-related term of the reward.

Moreover, to take into account previous information about the indoor temperature, the above-cited difference with one, two and four hours lag are also included in the state-space. These informations are useful to the agent to correct his past actions.
To guarantee a satisfactory indoor air temperature during the occupancy period, It would be helpful that the agent can learn when it is convenient to pre-heat the zone or when to turn off in the final phase of occupancy.
For this scope, Brandi et al. [34] suggest the introduction of two variables: Time to Occupancy Start and Time to Occupancy End.
When the zone is not occupied, Time to Occupancy Start represents the number of hours left for the return of the occupants, therefore during occupancy periods the variable is equal to zero
Conversely, when the building is occupied, Time to Occupancy End represents the number of hours that have to pass before the occupants' leaving time, so during the absence of occupants this variable is equal to zero.
Since the heating system is based on radiant floors, and it is able of controlling only the sensible load, the Relative Humidity can not be included in the state-space.

Variable	Min Value	Max Value	Unit
Hour of the Day	1	24	h
Day of the Week	1	7	-
External Air Temperature	-8	32	°C
Direct Solar Radiation	0	1100	W/m^2
ΔT Indoor Setpoint - Mean indoor temperature	-5	10	°C
Time to Occupancy Start	0	10	h
Time to end Occupancy End	0	15	h
ΔT Indoor Setpoint - indoor temperature, one hour lag	-5	10	°C
ΔT Indoor Setpoint - indoor temperature, two hours lag	-5	10	°C
ΔT Indoor Setpoint - indoor temperature, four hours lag	-5	10	°C

Table 13 - State-Space

Lastly, in order to feed the variables to the neural network, they are scaled in the (0, 1) range according to a min-max normalization.

5.2 Training phase

The Reinforcement Learning framework is defined by several hyperparameters that strongly influence the performance of the control agent. Consequently, to examine their influence on the performance of the control agent, different configurations of the most interesting hyperparameters are questioned and linked in this work.
Table 14 shows the hyperparameters kept unchanged during the training.

Variable	Value
DNN architecture	3 layers
Episode length	2928 Control time steps (61 days)
Buffer Size	11520

Table 14 - Fixed Hyperparameters

Furthermore, *Table 15* lists in detail each hyperparameter used in each run for the sensitivity analysis. In this analysis, different hyperparameters are involved:

- The discount factor γ;
- The learning rate;
- The weight factors β and δ;
- The Batch size;
- The number of Neurons for Hidden Layer;
- The number of episodes for each run.

Successively, the hyperparameters of the run leading to the best performance in terms of both energy savings and temperature control are selected and used in the deployment phase. As stated in section 3.3.3, a training episode includes two months, from 1st of November to 31st of December. The required indoor setpoint

was set equal to 20 °C and the temperature acceptability range between 19 °C and 21 °C.

	hyperparameters						
run	γ	Learning rate	β	δ	Batch size	Neurons for Hidden layer	episodes
1	0.9	0.001	1	0.1	256	256	10
2	0.95	0.001	1	0.1	256	256	10
3	0.99	0.001	1	0.1	256	256	10
4	0.9	0.001	1	0.5	256	256	10
5	0.9	0.001	1	0.1	512	256	10
6	0.9	0.001	1	0.1	128	256	10
7	0.9	0.0001	1	0.1	128	256	10
8	0.9	0.0001	1	0.1	256	256	25
9	0.9	0.001	1	0.1	256	256	25
10	0.9	0.0001	1	0.01	256	256	25
11	0.9	0.0001	5	0.1	256	256	25
12	0.9	0.0005	1	0.1	256	256	25
13	0.9	0.0001	10	0.1	256	256	25
14	0.9	0.0001	1	0.1	256	128	25
15	0.9	0.0001	1	0.1	256	512	25

Table 15 - Different hyperparameter configurations

5.3 Deployment phase

As said before, the best configuration between those analysed in the training phase was chosen and it is applied in the deployment phase.

In this phase the trained agent was deployed in four different scenarios to test the adaptability of the learned control policy to changes in the controlled environment. The deployment period lasts one episode that includes two months, from the 1st January to the 28th February.

The four scenarios are:

- Scenario 1: in this scenario, the control environment does not change, the aim is to evaluate the adaptability of the control agent to different weather

conditions such as outdoor temperature and direct solar radiation. The parameters related to the occupancy schedule and building don't change.

- Scenario 2: in this case, the zone setpoint temperature was set equal to 21 °C. Consequently, the new acceptability range was between 20 and 22 °C. The goal of this test is to evaluate the adaptability of the SAC controller in satisfying different temperature requirements.
- Scenario 3: in this scenario, the agent's adaptability was tested improving the energy performance of the transparent building envelope. Since the existing windows have a thermal transmittance U_w of 2.681 $\frac{W}{m^{2\circ}C}$, as reported in *Table 6,* and U-value for the transparent envelope of buildings located in Turin (climate zone E) must be lower than 1.9 $\frac{W}{m^{2\circ}C}$, there were introduced double glazing windows with U_w of 1.1 $\frac{W}{m^{2\circ}C}$ and a solar factor g of 0.33.
- Scenario 4: in this case, it was assessed the agent's adaptability to different building characteristics while all other parameters were unchanged. The internal mass was increased to rise the thermal inertia of the building.

There are two types of deployment:

- Static deployment: the updating of the control policy does not take place in the deployment phase and it requires less computational time;
- Dynamic deployment: the agent is characterized by continuous learning. In fact, for each control step, the agent receives the observations from the environment, it selects an action, observes the reward and the next state and it proceeds to the update of the control policy on-line. In this type of deployment, the agent has more adaptability, but it requires a high computational time and the main problem is that it can causes instabilities in the learned policy.

In this work the static deployment was selected.

Chapter 6:
Results

6 Results

This section illustrates the results of the different phase of the framework. In particular section 6.1 highlights the results of the energy model implementing the baseline control logic, which is a combination of the rule-based control and the climatic curve, as described in section 4.2.

The designed SAC algorithm was implemented in the simulation environment described in section 4.3.1.

Sections 6.2 and 6.3 show the results of the training and deployment phases and the comparison between the SAC control agent and the baseline control logic.

6.1 Result of the baseline

In this section there are illustrated the principal results of the simulation of the baseline model, in particular the analysis focuses on the heating energy consumption and the major heat loads involved in the building energy balance.

Figure 35 shows the total heating energy consumption through the entire heating season.

As expected, the ground floor presents the highest energy consumption because it has the biggest volume, as reported in *Table 5.* Furthermore, it also has the highest number of dispersing surfaces with the outside, and it communicates with the basement, which is not heated.

The last floor has the second higest energy consumption, although it has the smallest heated volume. This can be explained by its boundary conditions, as the top floor is bounded by the roof, which has poor energy performance, increasing dispersion to the outside.

Since the second and third floor have similar both the heated volume and the boundary conditions, they present the heating energy consumption relatively equal.

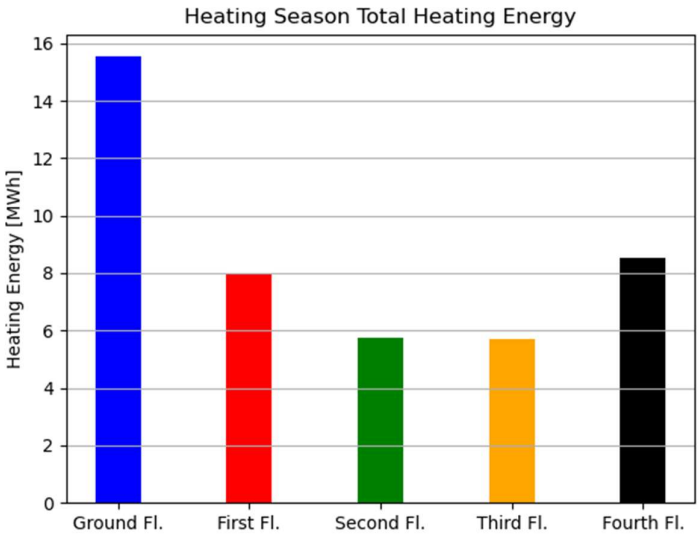


Figure 35 - Heating energy consumption of the heating season

Analysing the heating consumption in more detail, *Figure 36* shows the mean daily energy consumption of each floor.

In all floors, energy consumption has a similar shape. The system is switched on at the end of October. As mentioned before, the ground floor presents the highest consumption, while the other floors have a similar consumption in pairs, respectively the first with the fourth and the third with the second.

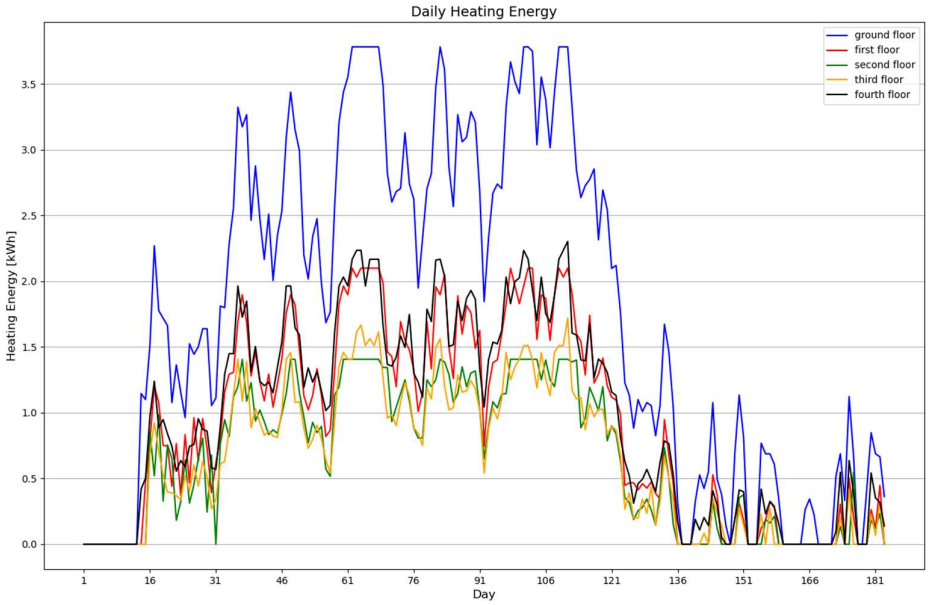


Figure 36 - Daily heating consumption comparison

Figure 37 highlights the comparison between the daily heating energy consumption pattern and the outdoor temperature one. These two variables are strictly correlated, in fact, when the temperature goes down, the energy consumption goes up and vice-versa. If there is a minimum of the mean daily outdoor temperature, there is a maximum in the mean daily energy consumption.

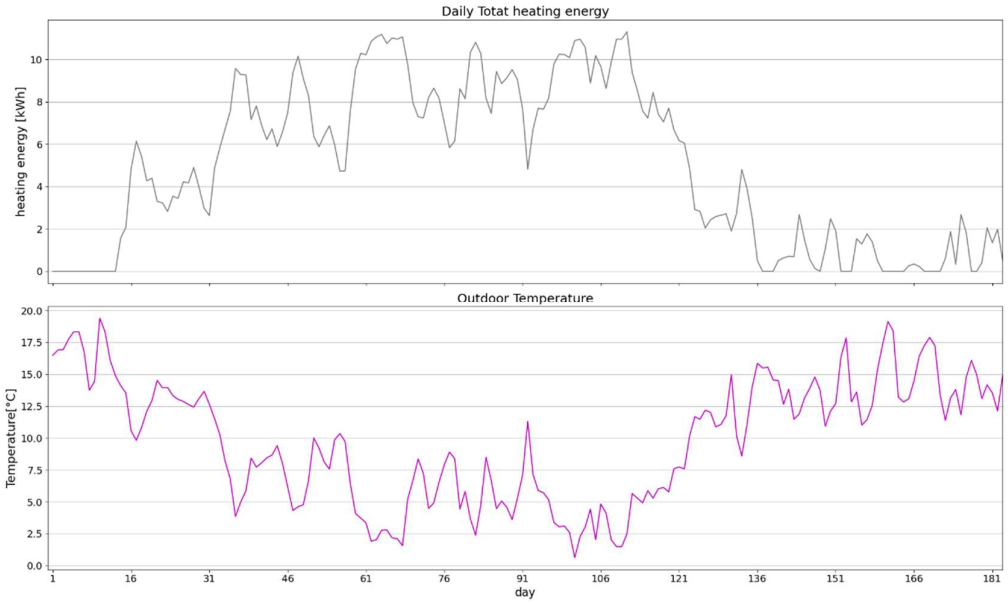


Figure 37 - Comparison between the energy consumption and the outdoor temperature

Now, the more relevant building loads are analysed, including the natural ventilation and infiltration losses and the windows heat gain.

Since the building under study has a large number of windows on each floor, the heat gain from the transparent surfaces takes importance in the building energy balance. This term is strictly correlated with the amount of solar radiation, as can be observable in *Figure 38,* in which the daily solar radiation was compared with the ground floor daily window heat gain. In fact, the two shapes are quite similar during the heating season, in particular, an increase in the mean daily solar radiation leads to a high mean daily heat gain. The highest values are reached at the end of the heating season, when the temperature and the solar radiation are higher.

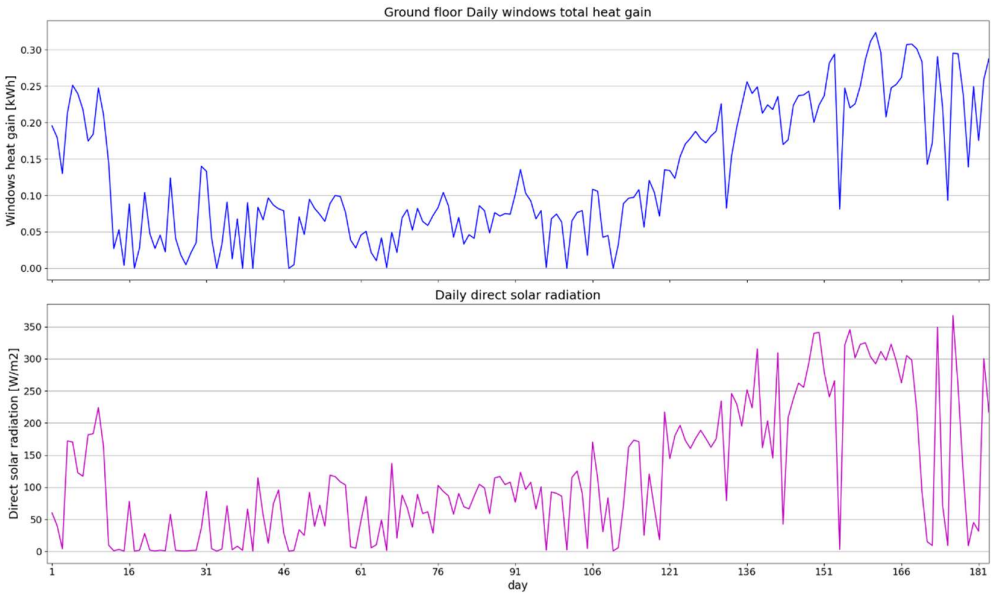


Figure 38 - Windows Heat Gain: Comparison between direct solar radiation and ground floor heat gain

Figure 39 and *Figure 40* show the comparison between the windows heat gain through different floors. This analysis was performed in two different periods:

- From the 1st December to the 6th December;
- From the 1st March to the 6th March.

In both analysed periods, each floor's gain follows the solar radiation pattern. The highest values for the heat gain are reached at midday conditions. The amount of the heat gain varies in each zone, according to their windows' number. The ground and the first floor have a similar transparent surface, and they obtain the highest amount of heat gain from the windows. Since the top floor only has four small windows, as a consequence, it has the smallest heat gain.

In the first analysed period, the peak value for the heat gain is less than 1.5 kW in the first floor, while in the second period, this value goes over 2 kW

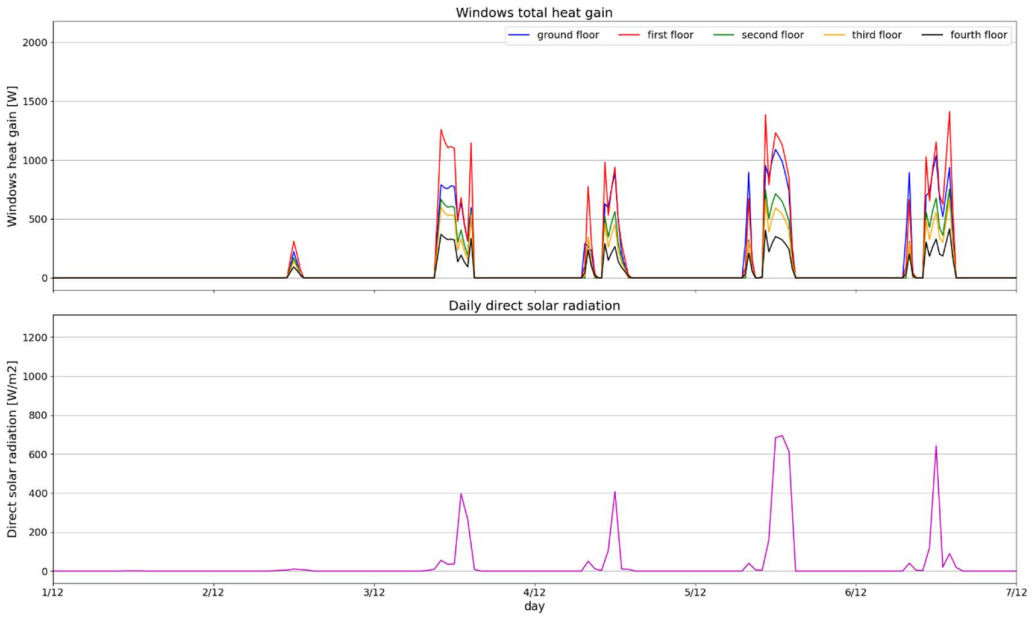


Figure 39 - Window heat gain comparison between floors in December

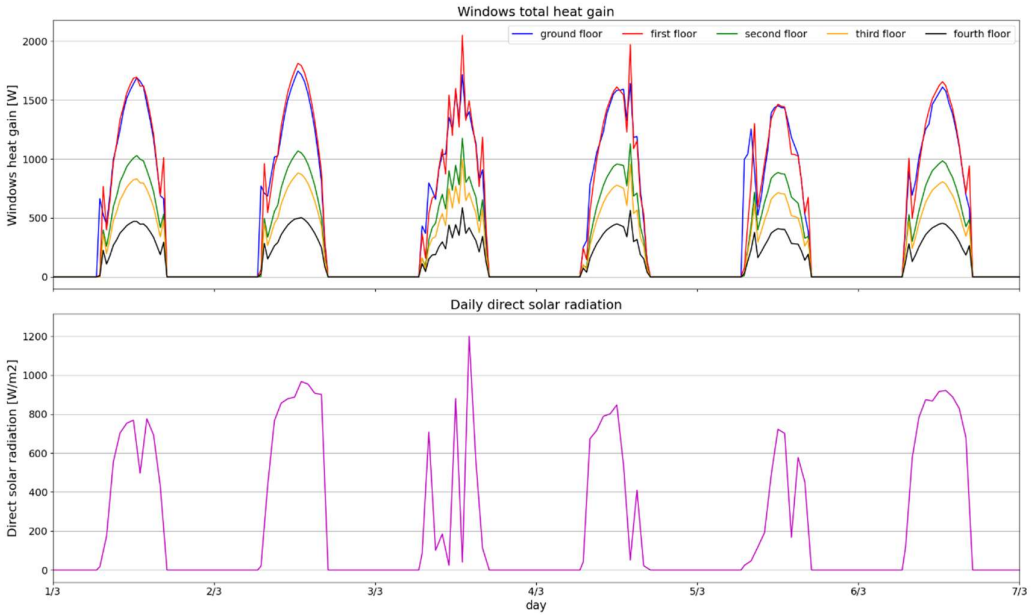


Figure 40 - - Window heat gain comparison between floors in March

Infiltrations are the unplanned air flows from the external environment directly into the thermal zone. These flows are generally caused by the uncorrected sealing of windows and doors or through building elements.

Being the outdoor air temperature lower than the internal one during the heating season, infiltration leads to heat losses that affects the building energy balance.

Figure 41 shows the relation between the outdoor temperature and the daily infiltration loss.

The two variables have opposite patterns, in fact, when the external temperature reached its lowest value, the mean daily infiltration heat loss significantly increases. These losses take importance particularly during the winter period.

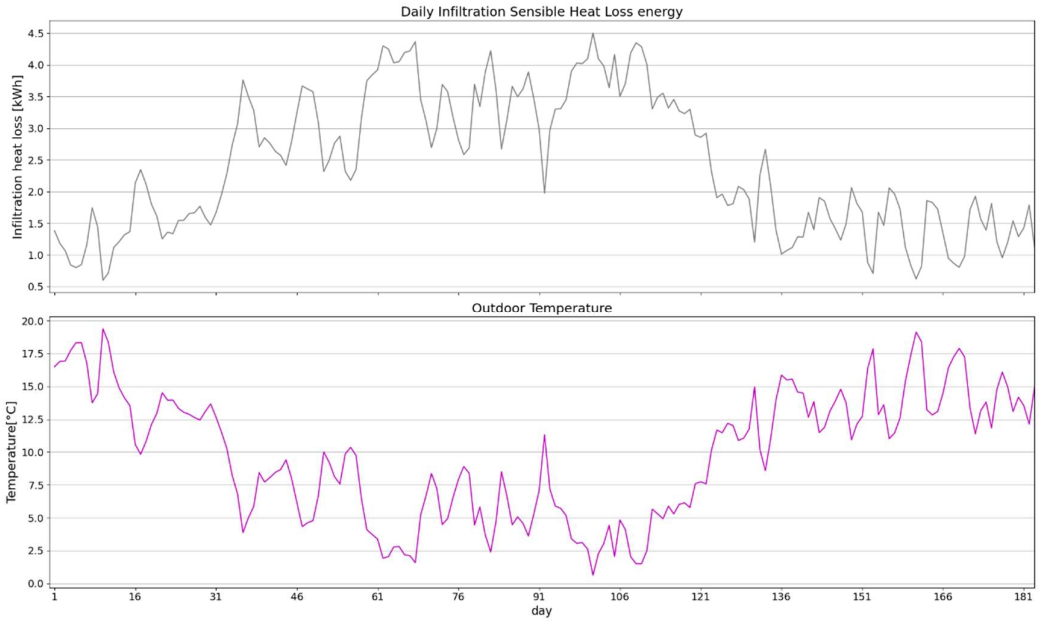


Figure 41 -Infiltration Heat Loss: Comparison between outdoor temperature and total infiltration heat loss

Figure 42 and *Figure 43* show the comparison between the infiltration heat loss through different floors. This analysis was performed in the same two periods of the window heat gain analysis.

The other parameter which influences the infiltration losses is the volume of the thermal zone, higher is the volume, higher will be the loss.

The infiltration heat loss has similar behaviour on all floors. In each floor, the heat loss reaches the highest value at midnight and the lowest value near the midday. Having the biggest volume, the ground floor presents the highest infiltration heat loss, while the last floor, having the smallest volume, has the lowest loss. In fact, at midday, the ground floor loss is three time the fourth floor one. These losses during

the period between the 1st and the 6th December reach value of about 2,5/3 kW in the first floor, that is two times the value reached in March, that is 1.5 kW.

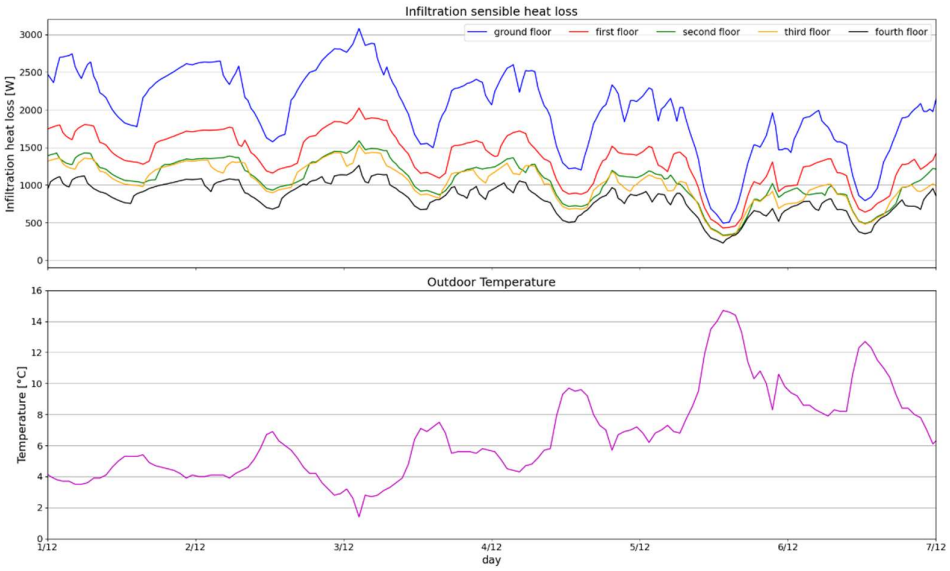


Figure 42 - Infiltration Heat Loss: Comparison between different floors in December

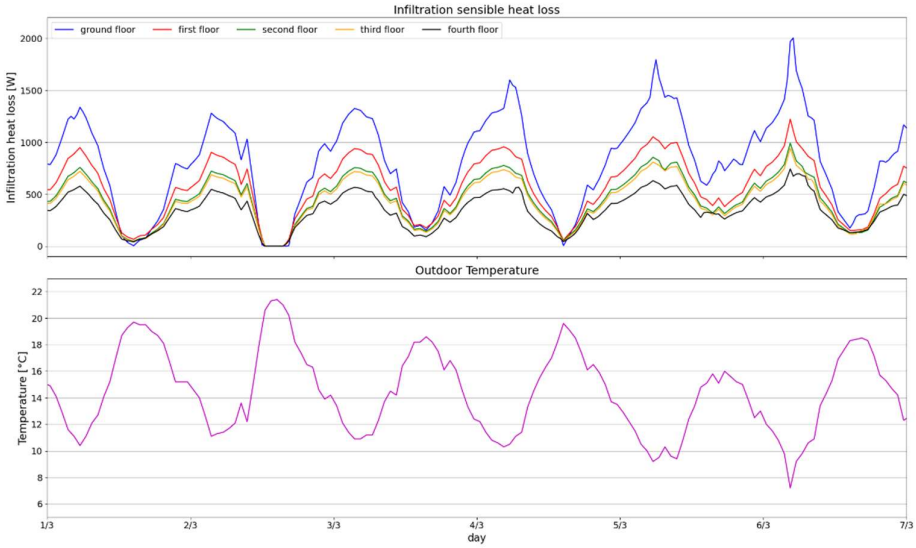


Figure 43 - Infiltration Heat Loss: Comparison between different floors in March

Finally, the analysis focuses on the ventilation heat loss.

Since the terminals of the HVAC system are the radiant floors, and they control the temperature without the use of air flow, the ventilation loss is due to the natural ventilation. These losses are influenced by two factors, the outdoor temperature and the windows state. In this work, the windows state is determined by the application

of the window opening and closing models individuated in section 2. Therefore, the ventilation heat loss is present only when the window is open, and its magnitude is influenced by both the external temperature and the opening time.

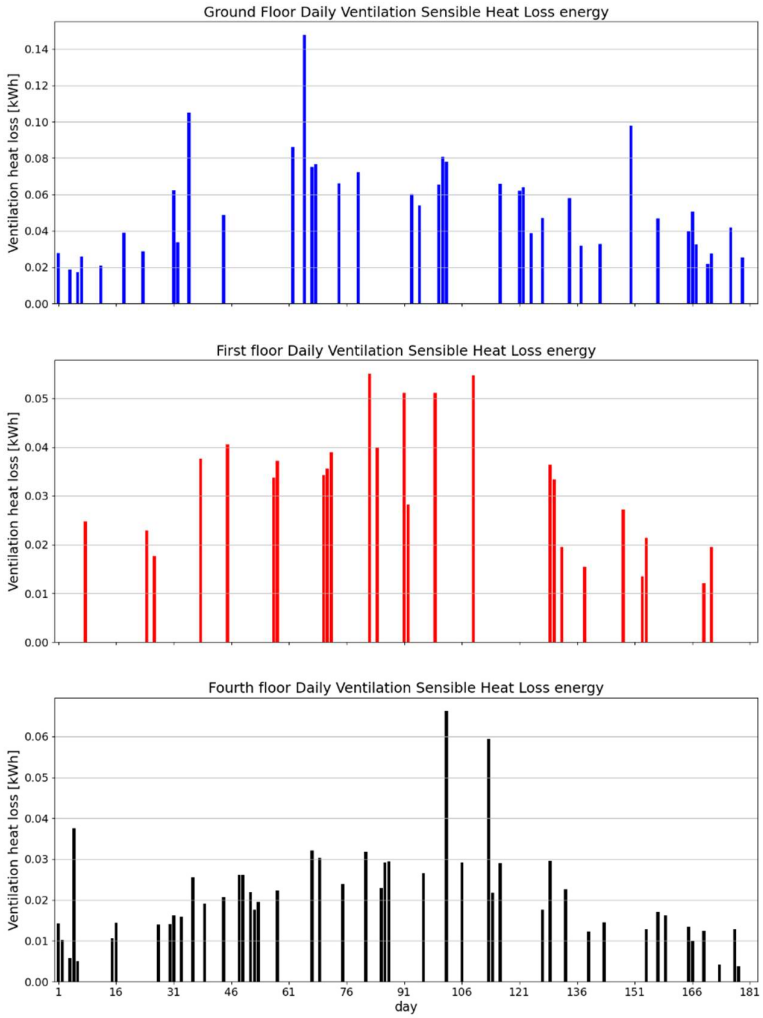


Figure 44 - Daily ventilation heat loss comparison

Figure 44 illustrates the mean daily ventilation heat loss on three different floors. The amount of the heat loss is higher during the winter period on all floors. These losses are not located in the same days in each floor because the windows opening and closing models are implemented separately on each floor. The ground floors reached the highest values.

6.2 Results of the training phase

As mentioned in section 5.4, in the training phase, a sensitivity analysis of the most relevant SAC hyperparameters was performed to study their impact on the performance of the control algorithm.

Two parameters were used to evaluate different configuration's actions: the energy-saving with respect to the baseline and the cumulative sum of temperature violations. While both terms were also calculated for the baseline control algorithm as reference parameters, the last one was introduced to evaluate comfort control performance.

A temperature violation is taken into consideration only when the indoor temperature is outside the acceptability range, and there are occupants. The magnitude of the temperature violation is then determined as the absolute difference between the indoor temperature and the designed set point value at each simulation step. The cumulative value over a whole episode returns the performance of the control algorithm expressed in °C.

Figure 45 exhibits the cumulative sum of temperature violations for the last training episode as a function of the energy saving, compared to the baseline control logic for the different configurations stated in *Table 15.*

The graph presents a y-axis on a logarithmic scale for better understanding. The performances of the baseline, which are indicated with black dashed lines, separate the plot in four-quadrants, and to each of them corresponds a different agent's behaviour.

Since the objectives of the designed control agent are the energy saving and the reduction of the temperature violations, the left-bottom quadrant illustrates the configurations that have performed better than the baseline. On the other side, the worst cases fall in the right-top quadrant, and it includes the solutions with higher energy consumption and temperature violations. Moreover, the other two quadrants witness configurations that fulfil only one requirement.

However, in this study, there are not configurations that fall in the worst region.

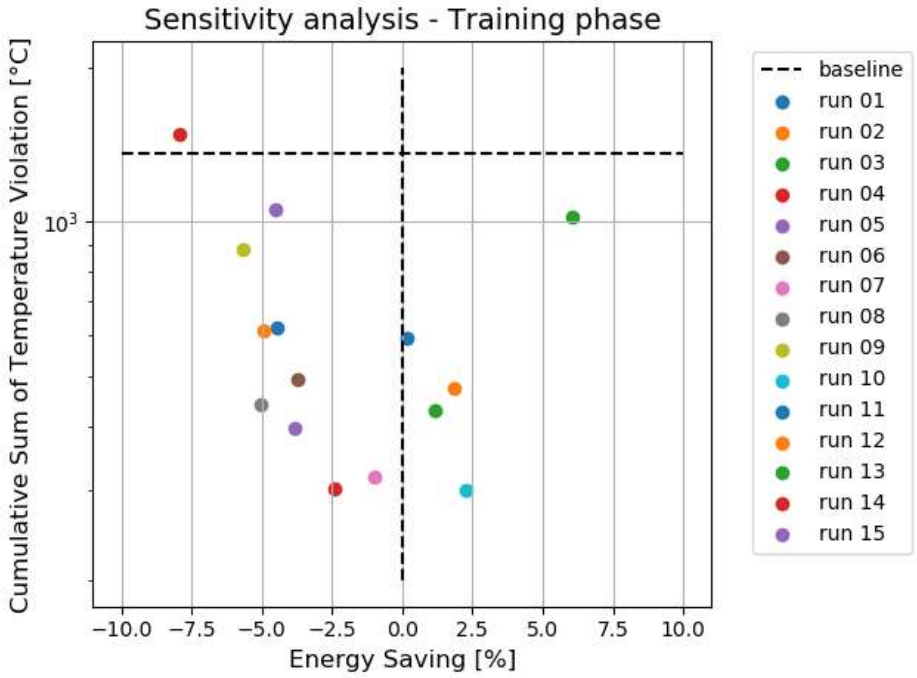


Figure 45 - SAC control performance in the last episode of the training phase

A high value of the weight of the energy related term (δ) leads to significant energy savings at the expense of temperature violations, as in the case of run 4 with a δ of 0.5.

On the other hand, a low value for δ causes a relevant reduction in the sum of temperature violation, but the agent consumes more heating energy than the baseline, as for run 10 with a δ of 0.01.

A similar discourse can be made for the weight of the temperature-related term β, in fact, with the other hyperparameters being equal, a too high value of β leads to a relevant reduction of the temperature violations, but an increase in consumption, as in the case of run 11 (β = 5) and 13(β = 10).

In configuration 2 and 3, with a discount factor γ of 0.95 and 0.99 respectively, only one requirement is satisfied because the controllers consume more energy than the baseline. Therefore, solutions with γ equal to 0.9 are preferred.

In particular, solutions with a learning rate of 0.0001, a discount factor (γ) of 0.9 and a weight of the temperature-related term of 0.1, corresponding to run 8, 14 and 15, show the best trade-off between energy saving and reduction of temperature violations. These three solutions, which are characterized by 256, 128 and 512 neurons for hidden layer respectively, are analysed in detail.

Firstly, a good indicator to evaluate the goodness of the learning process of the SAC control agents could be the evolution of the cumulative reward during each training episode. This term has not a physical meaning, but it takes into account both the energy consumption and the zone temperature, combining them in a single value. Furthermore, it gives indications about the converge of the control policy of the control agents. On the other hand, a non-convergent trend could be caused by an agent, who failed in reaching the optimal control policy. Higher is the reward, and higher is the performance of the agents.

For this task, the convergence of the three different solutions was analysed in *Figure 46*, in which the trend of both the temperature-related cumulative reward and the energy-related one, are shown for each configuration. While the first terms are represented in blue lines, the second ones are defined in red stripes.

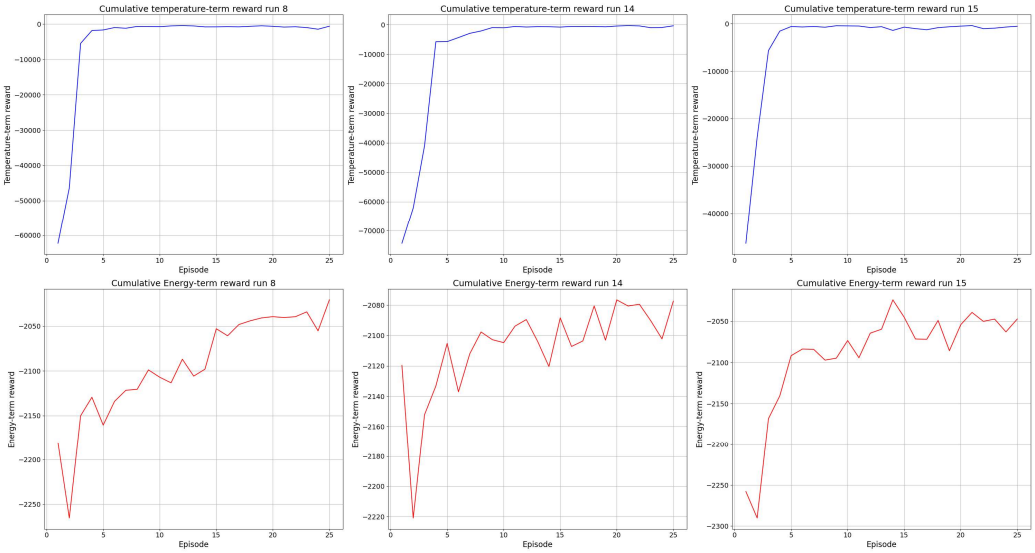


Figure 46 - Comparison of cumulative reward energy-term and temperature-term

In all configurations analysed, the agent starts the exploration with a high value for both the energy-related and the temperature-related terms.

Overall, run 8 reaches the biggest value for the energy term, while run 14 gets the peak for the temperature one.

In all solutions, the agent learns at first how to maintain the zone temperature in the comfort range during the first 10 episodes. This can be understood by looking at the temperature-related terms, which reach the convergence before the energy-related ones, due to their more oscillatory pattern. Only in the final episodes of run 8 and 15, the energy term reaches the convergence.

Taking into account these results, the control agent with the highest stability is the one proposed in solution 8.

Since the stability of the reward terms is not a complete indicator for assessing the goodness of the SAC agent performance, the analysis continues with a graphical representation of the results of the last training episode for these three configurations.

The three control agents are compared on the same day of the training episode. While the day was chosen by considering the coldest outdoor temperature in the training period, the zone was selected by looking for the level with the biggest volume, which corresponds to the ground floor.

Figure 47 shows the comparison between the three configurations, and it illustrates the daily heating energy consumption in the first line, the zone temperature in the second one and the supply power in the last one. In the graphs of the zone temperature, the period with the presence of occupants is represented by the yellow area, while the green area represents the temperature's acceptability range.

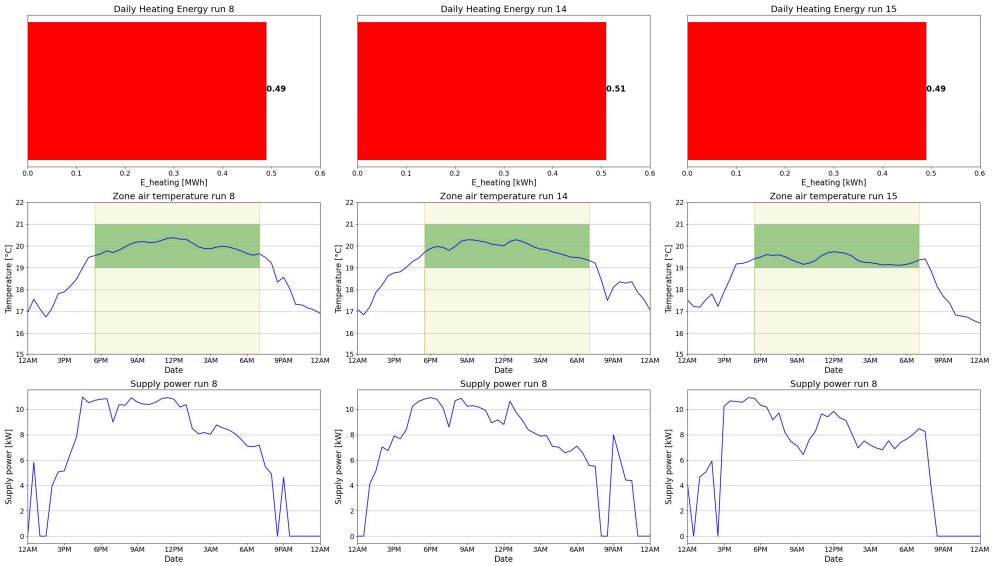


Figure 47 - Comparison between three agents during a training day

Overall, as can be noted in the central line of the figure, the agent has learnt how to maintain the zone temperature in the comfort range.

In addition, the agent of the configuration 8 keeps the zone temperature as closest as possible to the set-point of 20 °C. In contrast, the agent of configuration 15 maintains the temperature across the lower threshold. Run 14 presents the higher daily heating energy, consuming 0.51 MWh, while the other two configurations have a similar consumption. The supply power has a similar pattern in all solution. However, in configuration 15, the system switches on earlier than the other two, and during the occupants' presence, it reaches lower power values.

In conclusion, run 8 looks to adapt better the occupancy, in fact when there are no occupants the system is switched off for most time, and it switches on to pre-heat the zone, ensuring the reaching of the comfort.

To choose the best solution, *Table 16* shows the comparison of the three control agents in the entire last episode (25^{th}).

	SAC Control Agent		Baseline control logic		comparison	
configuration	Consumption [MWh]	Temperature violations [°C]	Consumption [MWh]	Temperature violations [°C]	Energy Saving [%]	Temperature violations difference
Run 8	20.21	441.73	21.83	1363.14	-5.05	-921.41
Run 14	20.77	302.25			-2.39	-1060.88
Run 15	20.47	397.15			-3.81	-965.99

Table 16 - Performance comparison at the end of the training phase

Since both the cumulative sum of the temperature violations of each configuration is significantly lower than the baseline, and the order of magnitudes are similar, the energy saving takes more importance. Therefore, solution 8 is selected as the best control agent because it presents the biggest energy saving, it has the greatest stability on the rewards, and it learns how to maintain the temperature in the comfort range. So, it will be implemented in the deployment phase.

6.3 Results of the deployment

In this section, the results of the deployment of the best configuration (configuration 8), which was identified in the previous paragraph, are analysed. *Table 17* shows the characteristics of the SAC control agent for this phase.

In this work, the control agent was deployed in one episode, which includes two months, from the 1st January to the 28th February, as mentioned in section 5.3.

The deployment of the SAC control agent was simulated in a static way in the four different scenarios.

characteristics	*Values*
DNN Architecture	3 layers
Neurons per hidden Layer	256
Discount factor	0.9
Learning rate	0.0001
Number of episodes	1

Episode Length	2832 Control steps (59 days)
Energy-term weight factor	0.1
Temperature-term weight factor	1

Table 17 - SAC control agent characteristics for the deployment phase

Figure 48 shows the results of the deployment of the SAC control agent in the four scenarios, compared with the baseline control logic. The first graph illustrates the heating energy consumption in MWh, while the second one shows the cumulative sum of the temperature violations. The blue bars represent the control agent while the baseline is illustrated with the orange bars.

In all scenarios, the control agent leads to a reduction in both heating energy consumption ad cumulative sum of temperature violation compared with the baseline control logic.

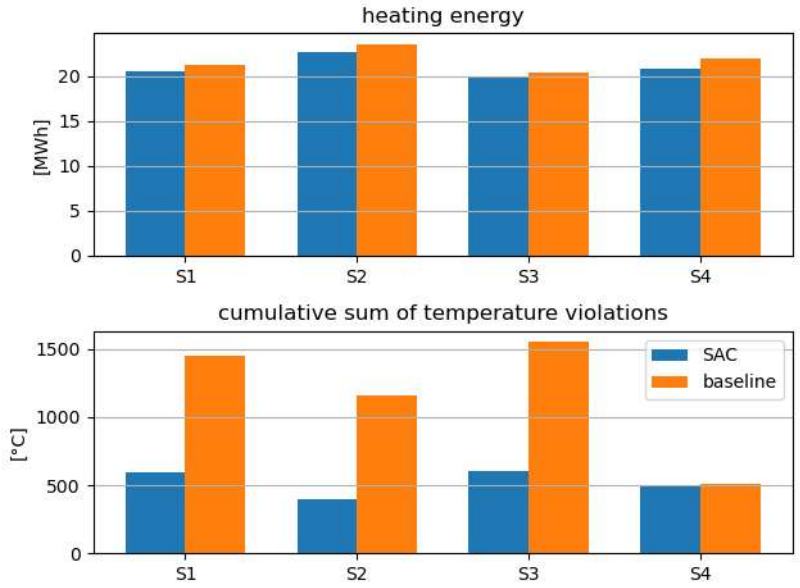


Figure 48 – Comparison of heating energy consumption and cumulative sum of temperature violations between the deployed control agent in the four scenarios and the baseline

As illustrated in *Figure 49*, in the fourth scenario, the control agent obtains the highest energy saving of about 6%, while in the third scenario, the energy saving achieved by the agent is the lowest (2%).

In the second scenario, both the control agent and the baseline have the biggest heating energy consumption, which is due to an increasing of the zone's setpoint temperature.

In all of them, the cumulative sum of temperature violations for the control agent has a similar order of magnitude, but, while in the fourth scenario this term is quite similar to the baseline one, in the other three the reduction is very significant, as can be seen in *Figure 50*. In fact, in the third one, the agents achieved a difference in temperature violation of about 950°C, while in the last scenario, this difference is only 4 °C.

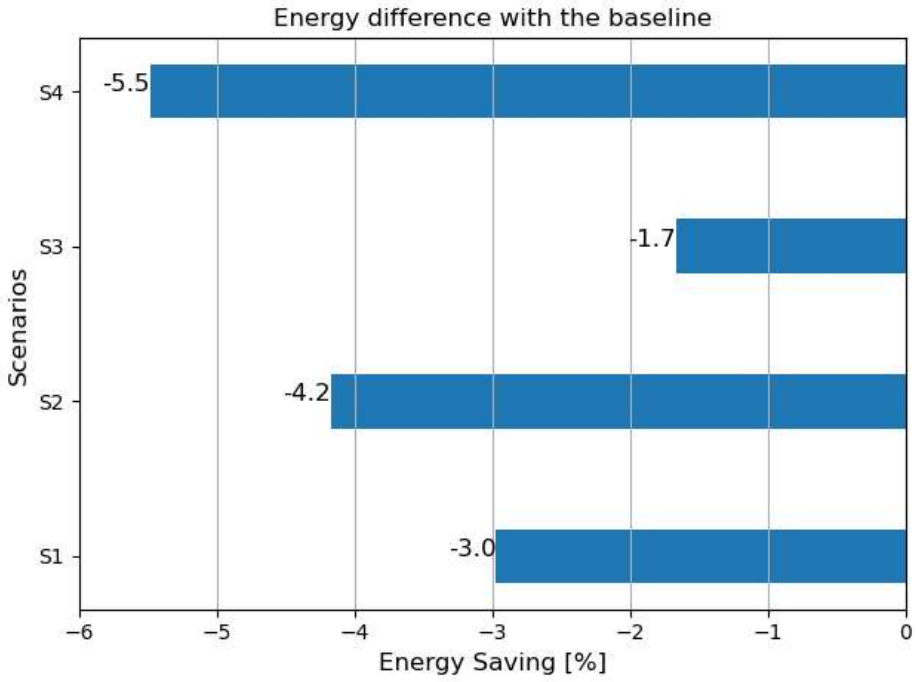


Figure 49 - Energy saving in each scenario

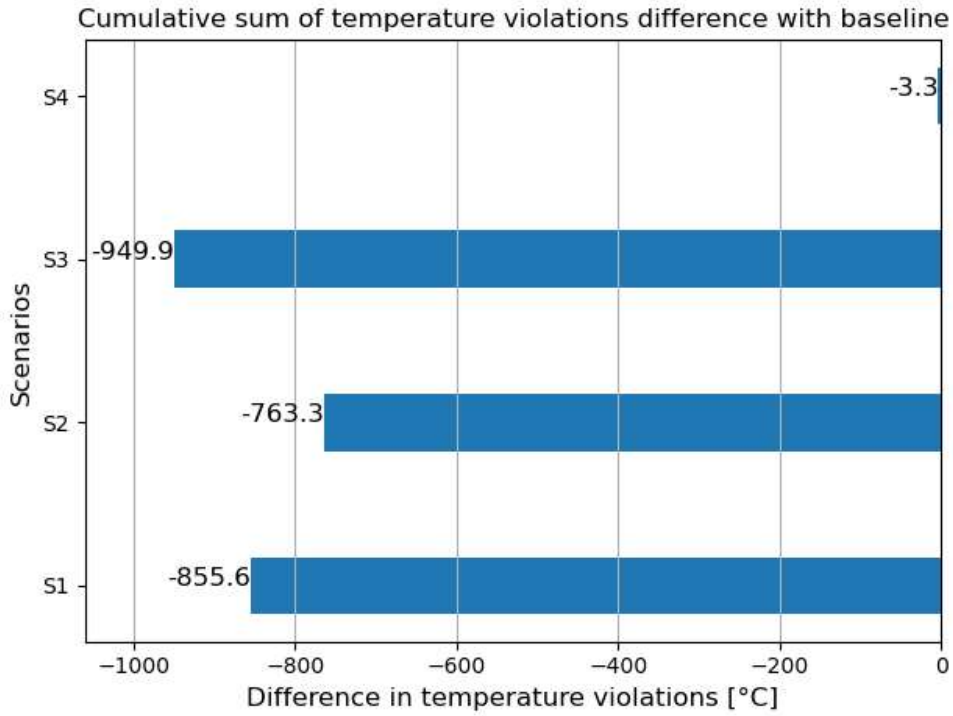


Figure 50 - Differences in cumulative sum of temperature violations in each scenario

The following pictures illustrate the comparison between the control agent and the baseline in the four deployment scenarios. In particular, in the graphs of the indoor temperature, the yellow area represents the occupancy period, while the green one indicates the acceptability range of the temperature. Moreover, while the red lines indicate the performance of the baseline, the blue ones indicate the SAC control agent.

Figure 51 shows the comparison between the baseline control and the deployed SAC agent in the first scenario, during five days on the ground floor. Since in this scenario, there was no change in the environment, the zone temperature profile and the supply power trend are compared to the outdoor temperature, represented in green line in the graph.

The SAC control agent permits the reduction on temperature violation in all days, and its temperature profile is more stable than the one of the baseline around the setpoint value (20 °C), which has a more oscillatory pattern.

Moreover, the control agent learnt how to optimize the pre-heating phase and the switch-off phase. In fact, the SAC agent tends to switch-on the system later, but with a high value of the supply power, reducing the time to reach the comfort range. On the other hand, the agent tends to switch-off the system before than the baseline, when there is still the presence of occupants, decreasing the energy consumption. When the outdoor temperature reached the lower value, the supply power provided by the control agent is lower than the one provided by the baseline; therefore, it leads to a reduction in energy consumption. In general, the SAC control agent has good adaptability to the change on exogenous factors.

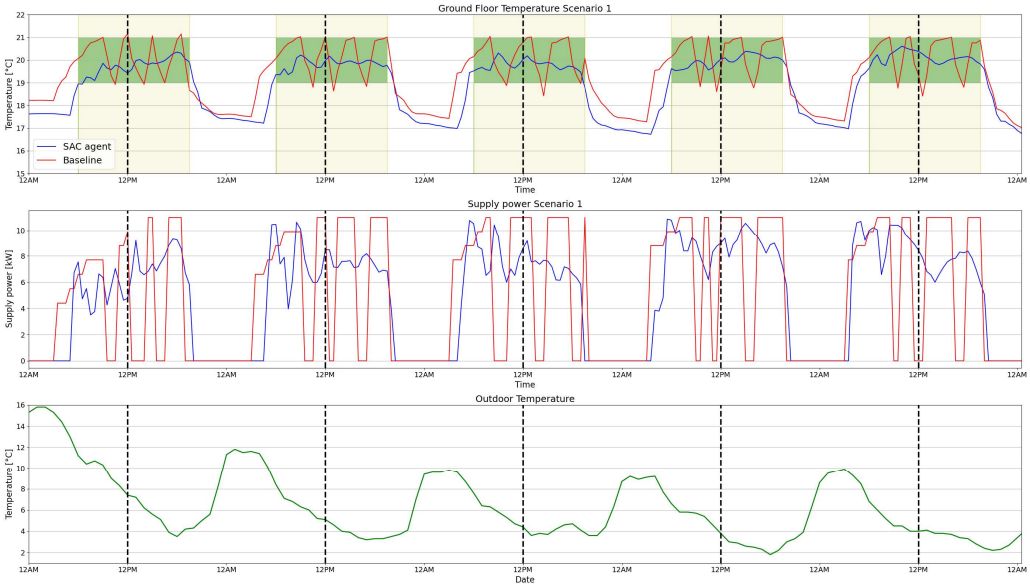


Figure 51 - Comparison between SAC control agent and baseline in Scenario 1 of the deployment phase

Figure 52 highlights the comparison between the deployed agent and the baseline in the second scenario, in which the setpoint temperature is higher (21°C) than the training one (20 °C). The analysis was performed on the same five days of the previous case.

As can be seen in *Figure 48*, in this scenario, the control agent reached the lowest value for the cumulative sum of temperature violation. In fact, the agent has learnt how to maintain the temperature in the new comfort range, and in particular, how to optimise the pre-heating phase; so, when people arrive in that zone, the

temperature is always very close to the lower limit of the threshold (20 °). A similar discussion can be made for the switch-off phase, in fact, when people leave the building, the temperature returns close to 20 °C.

In the middle of the occupancy period, the temperature is between the setpoint and the lower bound of the range, reaching high values near the midnight.

In the first two days of this period, the baseline has some issues to maintain the temperature in the acceptability range, in particular on the second day when it decreases until 19 °C.

Looking at the supply power graph, the baseline turns on the heating system before than the control agent, while the turn off period starts later.

Overall, the SAC agent maintains the indoor temperature in the acceptability range, providing a lower supply power to the zone.

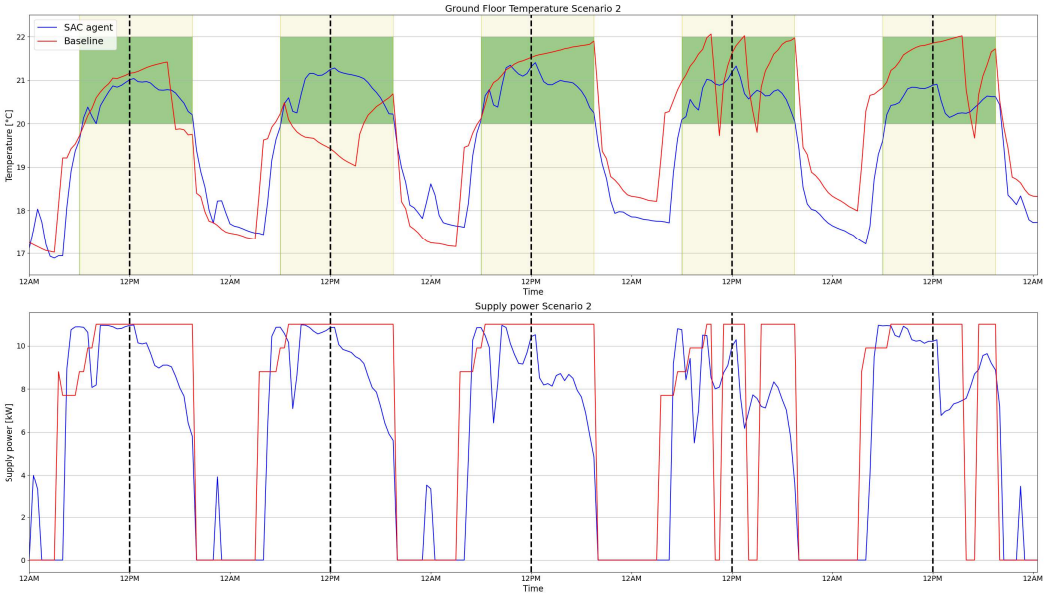


Figure 52 - Comparison between SAC control agent and baseline in Scenario 2 of the deployment phase

The comparison between the baseline and deployed agent in the third scenario, in which were installed more efficient windows, is illustrated in *Figure 53*. The analysis was performed in the same days of the first case.

As in the previous scenarios, the agent obtains a temperature profile within the comfort range during the occupancy period. Furthermore, this profile is more stable than the one provided by the baseline.

Here, both the control agent and the baseline consume less energy, compared to other scenarios, thanks to the better performance of the windows.

Moreover, the SAC agent learnt how to maintain the desired temperature by consuming lower heating energy compared to the baseline.

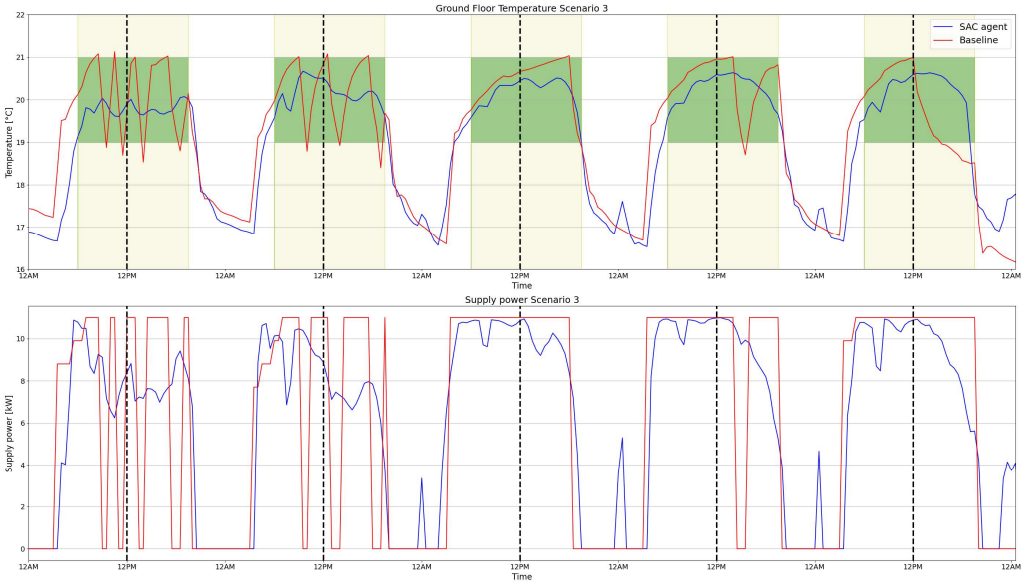


Figure 53 - Comparison between SAC control agent and baseline in Scenario 3 of the deployment phase

Figure 54 illustrates the comparison of the control agent in the first and third scenarios. In this case, the red line indicates the first one, while the third scenario is represented with the blue one.

Both the supply power pattern and the zone temperature profile are very similar. Moreover, even if the supply power is the same, the indoor temperature is slightly higher in the third scenario. However, the temperature in the third scenario decreases slightly slower.

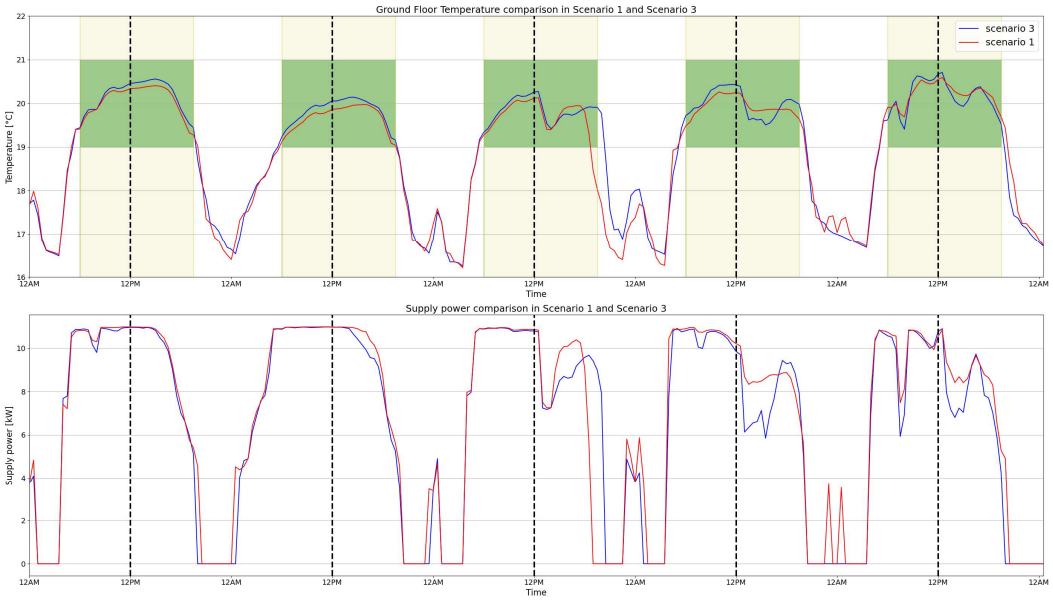


Figure 54 - Comparison between SAC control agent in Scenario 1 and Scenario 3

To analyse these two scenarios in more detail, *Figure 55* highlights the temperature comparison of each floor on the same day.

In general, both agents learnt how to maintain the desired temperature in all zones during the occupancy period, and the two temperature patterns are very similar. Except for the second floor, the deployed agent in the third scenario reaches higher zone temperatures than in the first one. Furthermore, the temperature decreases slowly in each floor.

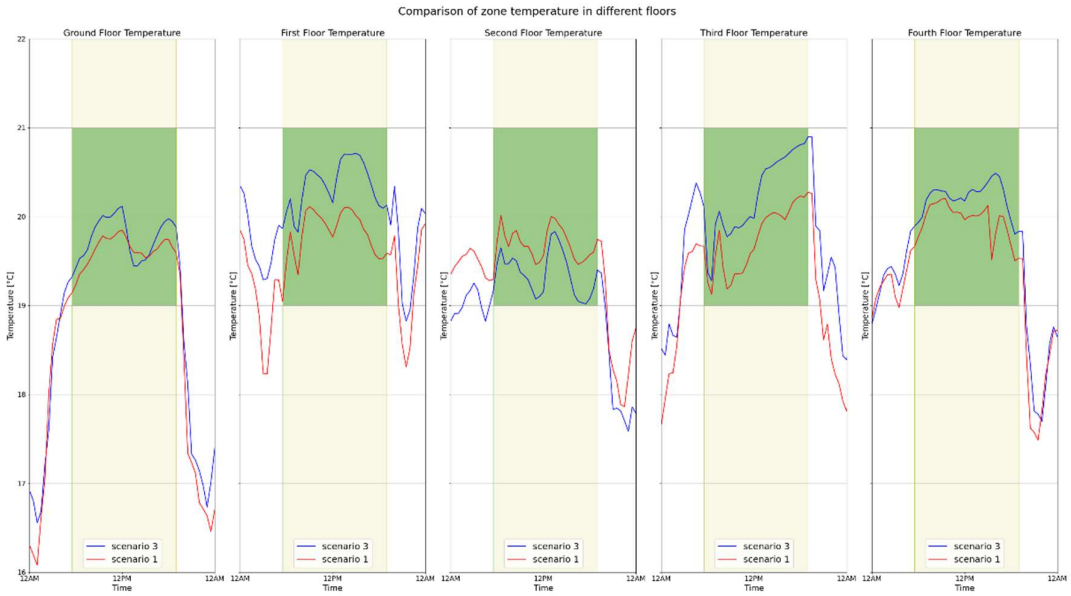


Figure 55 - Zone temperature comparison in different floors for Scenario 1 and Scenario 3

Figure 56 shows the comparison of the two controls logic in the last scenario.

While in the baseline control logic, the temperature frequently reaches the upper threshold of the acceptability range, the SAC control agent maintains the temperature close to the setpoint temperature in the occupancy time.

As said before and reported in *Figure 49*, the agent obtains the highest energy saving compared with the baseline. In fact, for most of the time, the supply power of the agent is lower than the baseline one.

Furthermore, with respect to the other three scenarios, the temperature decreases slowly when the system is switched off, and it is mainly due to the increased thermal inertia.

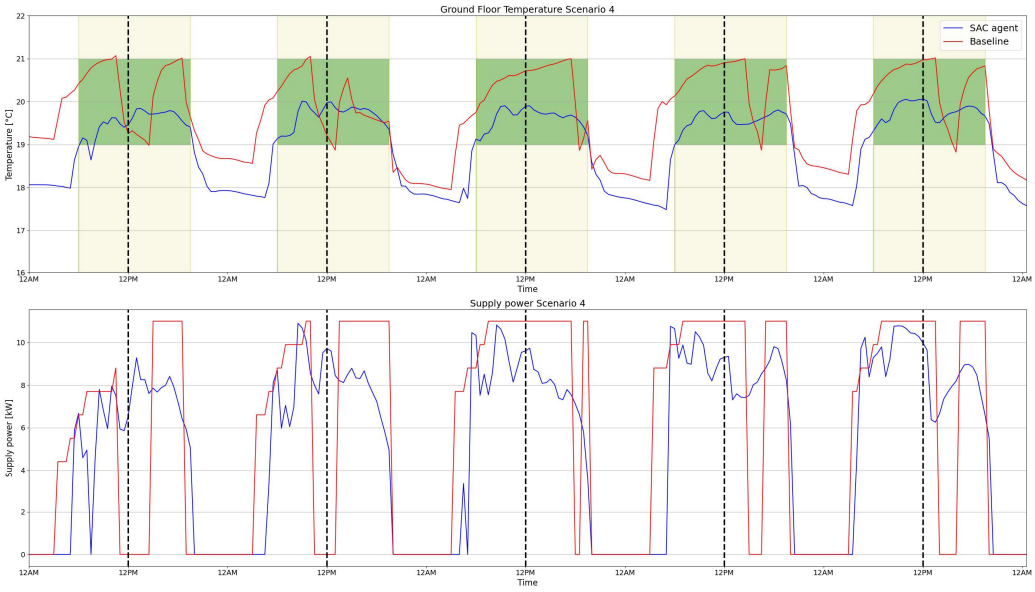


Figure 56 - Comparison between SAC control agent and baseline in Scenario 4 of the deployment phase

Finally, *Figure 57* shows the comparison of the control agent in two different scenarios: the first and the fourth. In these graphs, it is easy to observe the difference in the decrease of the temperature in the two scenarios when the system is switched off.

However, during the occupancy period, both the zone temperature and the supply power have a similar profile in both scenarios.

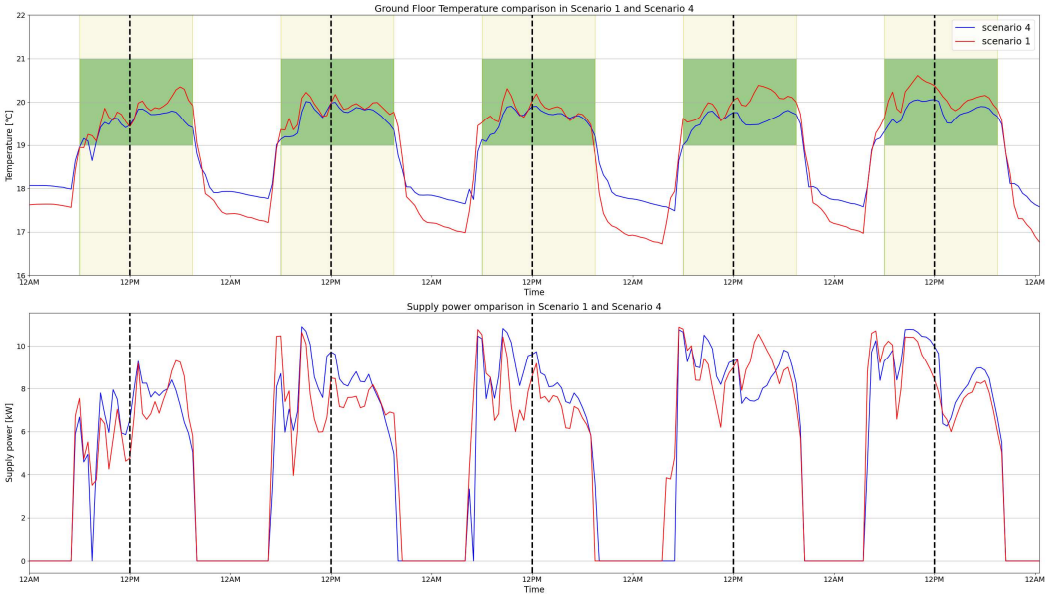


Figure 57 - Comparison between SAC control agent in Scenario 1 and Scenario 4

Chapter 7:
Discussion

7. Discussion

This work focuses on the development of a SAC control agent of a heating system to control the supply power to be provided to each zone of a residential building located in Turin, Italy.

The developed control agent was trained and deployed in a simulation environment which integrates EnergyPlus and Python.

The goal of this controller is to optimize both energy consumption and the maintenance of the desired temperature indoor air temperature withing the building. The main challenge that the controller must facies the identification of the best trade-off between these two contrasting objectives.

In RL algorithms the selection of the hyperparameters and the reward design have a fundamental role in identifying the optimal configuration of the SAC control agent. Therefore, a sensitivity analysis on the most important hyperparameters was performed to study their influence on the controller performance.

Among these hyperparameters, the two weight factors of the two terms of the reward seem to be the most influencing ones. In fact, a higher value of the weight of the energy-related term of the reward leads to a significant reduction in heating energy consumption at the expense of the zone temperature. Therefore, these configurations give too importance on the energy savings compromising the maintenance of the zone comfort.

On the other hand, high values of the weight of the temperature-related terms, giving more importance on the temperature, drive to opposite results, with a reduction in temperature violations and increased energy consumption.

Between the fifteen configurations analysed in a training period of two months, the hyperparameters of the eighth solution lead to control policy with the higher stability and the control agent presents the best trade-off between energy saving and reduction of temperature violation. Furthermore, this control agent learnt how to adapt to the occupancy patterns, in fact, the heating system is switched off when

the people leave the zone, and it is turn-on before their arrival to pre-heat the floor. During the occupancy period, the indoor temperature is inside the acceptability range. Overall, the control agent performs better than the baseline. The eighth configuration was chosen and tested in the static deployment.

The SAC control agent was tested in four different scenarios to prove its adaptability to modifications in the environment, such as different weather conditions, changes in the indoor setpoint temperature and building characteristics. Thanks to the adoption of an adaptive set of variables for the state-space, the deployment could be performed in a static configuration, that can avoid instability problem for the control policy and the high computational time of a dynamic deployment.

The SAC control agent proved to be adaptable in all the tested scenarios, and it presents better performance than the baseline controller. Both the cumulative sum of temperature violations and the heating energy consumption were lower than the baseline.

In particular, in the fourth scenario, the agent learnt how to optimise the pre-heating phase and thanks to the increased thermal inertia it obtains the highest energy saving.

On the other hand, in the second scenario, the agent has the lowest value of the cumulative sum of temperature violations. Therefore, the tested control agent has efficient adaptability to a change in the desired indoor temperature setpoint.

Overall, an accurate design of the variables of the state-space could increase the flexibility and adaptability of the SAC control agent to changes in the boundary condition even in static deployment conditions.

To implement the designed control agent in a real building, it is necessary to monitor some variables which can be collected thanks to low-cost sensor available in the market.

For the external temperature and the solar radiation could be used outdoor ambient sensor, or external weather data provider can provide those data.

Similar solutions are available for the zone temperature.

Since the stochastic nature of the occupancy in residential buildings, the biggest obstacle to overcome is the identification of the time to occupancy start and time to occupancy end.

These values can be estimated using weekly schedules based on the monitoring of people arriving and leaving hours, and updating it with the new collected data.

Chapter 8:
Conclusion

8. Conclusion

In this dissertation, the application of the SAC control agent in a radiant floor heating system was developed and analysed in a simulation environment, which combines Energy and Python.

The residential building was modelled in SketchUp to obtain the geometric structure for the energy model. With the use of External Interface and BCVTB the building model communicate with the simulation environment in Python.

Furthermore, to represent the occupants' behaviour as close as possible to the reality, the windows opening and closing were simulated with a probabilistic model.

The control agent was designed to choose the optimal action, in this case the supply power to be provided to each zone to maintain the temperature within the acceptability range.

Ten adaptive variables were included in the definition of the state-space. They include exogenous parameters, indoor air temperature-related terms and two variables which describe the occupancy status of the building.

During the training phase, a sensitivity analysis was performed on the main hyperparameters to highlight their influence on the controller's performance. The best configuration was tested in four different scenarios in the deployment phase to evaluate the flexibility and adaptability of the SAC control agent to changes of outdoor condition, indoor requirement and building characteristics.

Depending on the scenario, the deployed control agent reaches a significant reduction on the cumulative sum of temperature violations compared with the baseline, and, at same time, it also reaches an energy-saving that varies between 2% and 6%. These results are achieved by this SAC control agent in a static deployment.

The designed state-space allows the implementation of a static deployment instead of the dynamic one, which may cause instability and high computational time.

To extend and improve this work, the next studies will focus on the following aspects:

- Compare the results obtained with the static deployment with the dynamic one, using the same scenarios. This comparison can show the differences in both the stability of the control policy and the adaptability of the control agent.
- Introduce comfort parameters, such as Predicted Percentage of Dissatisfied (PPD) and the Predicted Mean Vote (PMV), in the reward function. The variables involved in these parameters may be difficult to obtain in the real-world applications, but in a simulation environment, it could be interesting.
- Compare the results obtained with the adaptive set of variables with the non-adaptive one.
- Compare the performance of SAC with model-based solution such as MPC. These two control algorithms are opposed, and a comparison in terms of performance, computational cost and modelling effort could be interesting.
- Apply the SAC controller to modern HVAC systems, which are characterized by higher level of complexity. They could introduce the application of RES generation and storage.
- Add new models to represent the occupants' behaviour in addition to the windows one. Being a residential building, the occupants have a stochastic behaviour that could be described with probabilistic models.

Acronyms

A3C	Asynchronous Advantage Actor-Critic
ACH	Air Changes per hours
AHU	Air handling Unit
BCVTB	Building Control Virtual Test Bed
BRL	Batch Reinforcement Learning
DDPG	Deep Deterministic Policy Gradient
DNN	Deep Neural Networks
DP	Dynamic Programming
DQN	Deep Q-Network
DRL	Deep Reinforcement Learning
FL	Fuzzy Logic
GA	Genetic Algorithm
HDD	Heating Degree Days
HVAC	Heating, Ventilation and Air Conditioning
MC	Monte-Carlo
MDP	Markov Decision Property
MPC	Model Predictive Control
NN	Neural Networks
PMV	Predicted Mean Vote
PPD	Predicted Percentage of Dissatisfied
PPO	Proximal Policy Optimization
RC	Resistance and Capacitors
RES	Renewable Energy Source
RL	Reinforcement Learning
SAC	Soft Actor-Critic
SP	Supply Power
TD	Temporal Difference
TD3	Twin Delayed DDPG
TRPO	Trust Region Policy Optimization

References

[1] IEA. [Online]. Available: https://www.iea.org/countries/italy. [Accessed 28 10 2020].

[2] C. Finck, P. Beagon, J. Clauß, T. Péan, P. J. Vogler-Finck, K. Zhang and H. Kazmi, "Review of applied and tested control possibilities for energy flexibility in buildings," *report from IEA EBC Annex 67 Energy Flexible Buildings,* 2018.

[3] A. Kathirgamanathan, M. D. Rosa, E. Mangina and D. P. Finn, "Data-driven predictive control for unlocking building energy flexibility: A Review," *Renewable and Sustainable Energy Reviews,* 2020.

[4] C. Finck, P. Beagon, J. Clauß, T. Péan, P. J. Vogler-Finck, K. Zhang and H. Kazmi, "Review of applied and tested control possibilities for energy flexibility in buildings," *IEA EBC Annex 67 Energy Flexible Buildings,* 2018.

[5] T. Manuel, "Controllo di temperatura on/off," [Online]. Available: http://win.maurodeberardis.it/Bridge99/control.htm. [Accessed 12 11 2020].

[6] "Controllori PID," [Online]. Available: http://www.unife.it/ing/lm.meccanica/insegnamenti/dinamica-controllo-diagnosi-di-sistemi-b/materiale-didattico/Controllori_PID.pdf. [Accessed 12 11 2020].

[7] D. S. Naidu and C. G. Rieger, "Advanced control strategies for heating, ventilation, air-conditioning, and refrigeration systems—An overview: Part I: Hard control," *HVAC&R Research,* 2011.

[8] D. Picard and L. Helsen, “MPC performance for hybrid GEOTABS buildings,” 2018.

[9] F. Jorissen, “Toolchain for Optimal Control and Design of Energy Systems in Buildings,” 2018.

[10] J. Cigler, S. Prívara, Z. Vana, E. Zacekova and L. Ferkl, “Optimization of predicted mean vote index within model predictive control framework: Computationally tractable solution.,” *Energy and Buildings ,* 2012.

[11] S. Yang, M. P. Wan, B. F. Ng, T. Zhang, S. Babu, Z. Zhang and S. Dubey, “A state-space thermal model incorporating humidity and thermal comfort for model predictive control in buildings.,” *Energy and Buildings,* 2018.

[12] F. Jorissen, W. Boydens and L. & Helsen, “Simulation-based occupancy estimation in office buildings using CO2 sensors,” 2017.

[13] J. Drgona, J. Arroyo, I. C. Figueroa, D. Blum, K. Arendt, D. Kim, E. P. Ollè, J. Oravec, M. Wetter, D. L. Vrabie and L. Helsen, “All you need to know about model predictive control for buildings,” *Annual Reviews in Control.*

[14] M. Avci, M. Erkoc, A. Rahmani and S. Asfour, “Model predictive HVAC load control in buildings using real-time electricity pricing.,” *Energy and Buildings,* 2013.

[15] G. Bianchini, M. Casini, A. Vicino and D. Zarrilli, “Demand-response in building heating systems: A model predictive control approach.,” *Applied Energy,* 2016.

[16] F. Oldewurtel, A. Ulbig, A. Parisio, G. Andersson and M. Morari, “Reducing peak electricity demand in building climate control using real-time pricing and model predictive control,” *49th IEEE conference on decision and control,* 2010.

[17] F. A. Qureshi and C. N. Jones, "Hierarchical control of building HVAC system for ancillary services provision.," *Energy and Buildings,* 2018.

[18] D. Patteeuw, G. P. Henze and L. Helsen, "Comparison of load shifting incentives," *Applied,* 2016.

[19] M. D. Knudsen and S. Petersen, "Demand response potential of model predictive control of space heating based on price and carbon dioxide intensity signals.," *Energy and Buildings,* 2016.

[20] P. Vogler-Finck, R. Wisniewski and P. Popovski, "Reducing the carbon footprint of house heating through model predictive control–a simulation study in danish conditions.," *Sustainable Cities and Society,* 2018.

[21] A. Vandermeulen, L. Vandeplas, D. Patteeuw, M. Sourbron and L. Helsen, "Flexibility offered by residential floor heating in a smart grid context: the role of heat pumps and renewable energy sources in optimization towards different objectives," 2017.

[22] P. J. C. Vogler-Finck, P. D. Pedersen, P. Popovski and R. Wisniewski, "Comparison of strategies for model predictive control for home heating in future energy systems," 2017.

[23] D. S. Naidu and C. G. Rieger, "Advanced control strategies for HVAC&R systems- An overview: Part II: Soft and fusion control," *HVAC&R Research,* 2011.

[24] P. Curtiss, J. Kreider and M. Brandemuehl., "Local and global control of commercial building HVAC systems," *Procedings of the 1994 American Control Conference,* 1994.

[25] A. So, W. Chan, T. Chow and W. Tse, "A neural-network-based identifier/controller for modern HVAC control," *ASHRAE Transactions Research,* 1994.

[26] S. Huang and R. Nelson, "Rule development and adjustment strategies of a fuzzy logic controller for an HVAC system: Part one - Analysis," *ASHRAE Transactions 100,* 1994.

[27] M. Arima, E. Hara and J. Katzberg., "A fuzzy logic and rough sets controller for HVAC systems," 1995.

[28] J. Wright, H. Loosemore and R. Farmani, "Optimization of building thermal design and control bymulti-criterion genetic algorithm.," *Energy and Buildings,* 2002.

[29] J. Vázquez-Canteli, J. Kämpf and Z. Nagy, "Balancing comfort and energy consumption of a heat pump using batch reinforcement learning with fitted Q-iteration," *Elsevier, Energy Procedia 122,* 2017.

[30] K. U. Ahn and C. S. Park, "Application of deep Q-networks for model-free optimal control balancing between different HVAC systems," *Science and Technology for the Built Environment,* 2019.

[31] S. Nagarathinam, V. Menon, A. Vasan and A. Sivasubramaniam, "MARCO - Multi-Agent Reinforcement learning based COntrol of building HVAC systems," *The Eleventh ACM International Conference on Future Energy Systems ,* 2020.

[32] J. Y. Park and Z. Nagy, "HVACLearn: A reinforcement learning based occupant-centric control for thermostat set-points," *The Eleventh ACM International Conference on Future Energy Systems,* 2020.

[33] Z. Zhang, A. Chong, Y. Pan, C. Zhang and K. P. Lam, "Whole building energy model for HVAC optimal control: A practical framework based on deep reinforcement learning," *Elsevier, Energy & Buildings 199,* 2019.

[34] S. Brandi, M. S. Piscitelli, M. Martellacci and A. Capozzoli, "Deep Reinforcement Learning to optimise indoor temperature control and heating energy consumption in buildings," *Elsevier, Energy & Buildings,* 2020.

[35] X. Ding, W. Du and A. Cerpa, "OCTOPUS: Deep Reinforcement Learning for Holistic Smart Building Control," *The 6th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation,* 2019.

[36] S. Qiu, Z. Li, Z. Li, J. Li, S. Long and X. Li, "Model-free control method based on reinforcement learning for building cooling water systems: Validation by measured data-based simuilation," *Elsevier, Energy & Buildings 218,* 2020.

[37] L. Yang, Z. Nagy, P. Goffin and A. Schlueter, "Reinforcement learning for optimal control of low exergy buildings," *Elsevier, Applied Energy 156,* 2015.

[38] F. Ruelens, B. Claessens, S. Quaiyum, B. D. Schutter, R. Babuska and R. Belmans, "Reinforcement Learning Applied to an Electric Water Heater: From Theory to Practice," *IEEE Transaction in Smart Grid ,* 2015.

[39] B. J. Claessens, D. Vanhoudt, J. Desmedt and F. Ruelens, "Model-free control of thermostatically controlled loads connected to a district heating network," *Elsevier, Energy & Buildings 159,* 2019.

[40] T. Leurs, B. J. Claessens, F. Ruelens, S. Weckx and G. Deconinck, "Beyond Theory: Experimental Results of a Self-Learning Air Conditioning Unit," *IEEE, International Energy Conference,* 2016.

[41] S. Lu, W. Wang, C. Lin and E. C. Hameen, “Data-driven simulation of a thermal comfort-based temperature set-point control with ASHRAE RP884,” *Elsevier, Building & Environment 156,* 2019.

[42] Y. Sun, A. Somani and T. E. Carroll, “Learning Based Bidding Strategy for HVAC Systems in Double Auction Retail Energy Markets,” *IEEE, American Control Conference,* 2015.

[43] Y. Chen, L. K. Norford, H. W. Samuelson and A. Malkawi, “Optimal control of HVAC and window systems for natural ventilation through reinforcement learning,” *Elsevier, Energy & Buildings 169,* 2018.

[44] G. Costanzo, S. Iacovella, F. Ruelens, T. Leurs and B. Claessens, “Experimental analysis of data-driven control for a building heating system,” *Elsevier, Sustainable Energy "Grids and Network 6" ,* 2016.

[45] B. Chen, Z. Cai and M. Bergés, “Gnu-RL: A Precocial Reinforcement Learning Solution for Building HVAC Control Using a Differentiable MPC Policy.,” *The 6th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation ,* 2019.

[46] F. Ruelens, B. J. Claessens, S. Vandael, B. D. Schutter, R. Babuška and R. Belmans, “Residential Demand Response of Thermostatically Controlled Loads Using Batch Reinforcement Learning,” *IEEE Transactions on smart grid,* vol. 8, no. 5, 2017.

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

[48] E. Mocanu, D. C. Mocanu, P. H. Nguyen, A. Liotta, M. E. Webber, M. Gibescu and J. G. Slootweg, “On-Line Building Energy Optimization Using

Deep Reinforcement Learning," *IEEE Transactions on smart grid,* vol. 10, no. 4, 2019.

[49] Z. Yu and A. Dexter, "Online tuning of a supervisory fuzzy controller for low-energy building system using reinforcement learning," *Elsevier, Control Engineering Practice 18,* 2010.

[50] Y. Wang, K. Velswamy and B. Huang, "A long-short term memory recurrent neural network based reinforcement learning controller for office Hvac," 2017.

[51] Z. Zhang, C. Ma and R. Zhu, "Thermal and Energy Management Based on Bimodal Airflow-Temperature Sensing and Reinforcement Learning," *MDPI, Energies 11, 2575,* 2018.

[52] R. Jia, M. Jin, K. Sun, T. Hong and C. Spanos, "Advanced Building Control via Deep Reinforcement Learning," *Elsevier, Energy Procedia 158,* 2019.

[53] W. Valladaresa, M. Galindo, J. Gutiérreza, W.-C. Wu, K.-K. Liao, J.-C. Liao, K.-C. Lu and C.-C. Wang, "Energy optimization associated with thermal comfort and indoor air control," *Elsevier, Building & Environment 155,* 2019.

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

[55] L. Yu, Y. Sun, C. Shen, D. Yue, T. Jiang and X. Guan, "Multi-Agent Deep Reinforcement Learning for HVAC Control in Commercial Buildings," *IEEE Transactions on smart grid ,* vol. 20, no. 20, 2020.

[56] P. v. d. Broma, A. R. Hansenb, K. Gram-Hanssenb, A. Meijera and H. Visschera, "Variances in residential heating consumption – Importance of

building characteristics and occupants analysed by movers and stayers," *Applied Energy,* 2019.

[57] J. Rouleau and L. Gosselin, "Probabilistic window opening model considering occupant behavior diversity: A data-driven case study of Canadian residential buildings," *Energy,* 2020.

[58] R. V. Andersen, B. W. Olesen and J. Toftum, "Modelling window opening behaviour in Danish dwellings," *Conference paper,* 2011.

[59] D. Calì, R. K. Andersen, D. Müller and B. W. Olesen, "Analysis of occupants' behavior related to the use of windows in German households," *Building and Environment,* 2016.

[60] R. V. Jones, A. Fuertes, E. Gregori and A. Giretti, "Stochastic behavioural models of occupants' main bedroom window operation for UK residential buildings," *Building and Environment,* 2017.

[61] A.-M. Yaser, "Lesson 1: The Learning Problem," [Online]. Available: http://work.caltech.edu/slides/slides01.pdf. [Accessed 2020].

[62] D. Silver, "UCL Course on RL, Lecture 1," [Online]. Available: https://www.davidsilver.uk/wp-content/uploads/2020/03/intro_RL.pdf. [Accessed 15 March 2020].

[63] K. Dalamagkidis, D. kolokotsa, K. Kalaitzakis and G. Starvrakakis, "Reinforcement learning for energy conservation and comfort in buildings," *Building and Environment,* 2006.

[64] Z. Wang and T. Hong, "Reinforcement learning for building controls: The opportunities and challenges," *Elsevier, Applied Energy,* 2020.

[65] J. J. Moolayil, “A Layman’s Guide to Deep Neural Networks,” [Online]. Available: https://towardsdatascience.com/a-laymans-guide-to-deep-neural-networks-ddcea24847fb. [Accessed 10 September 2020].

[66] Nair, P. Srinivasan, S. Blackwell, C. Alcicek and R. Fearon, “Massively parallel methods for deep reinforcement learning,” 2015.

[67] V. V.Kumar, “Soft Actor-Critic Demystified,” [Online]. Available: https://towardsdatascience.com/soft-actor-critic-demystified-b8427df61665. [Accessed 21 10 2020].

[68] T. Haarnoja, A. Zhou, P. Abbeel and S. Levine, “Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor,” *arXiv:1801.01290v2,* 2018.

[69] O. Sigaud, “Soft Actor Critic,” [Online]. Available: http://pages.isir.upmc.fr/~sigaud/teach/sac.pdf. [Accessed 24 10 2020].

[70] “REQUISITI SPECIFICI PER GLI EDIFICI ESISTENTI SOGGETTI A RIQUALIFICAZIONE ENERGETICA, Appendice B,allegato 1 capitolo 4,” [Online]. Available: https://www.mise.gov.it/images/stories/normativa/DM_requisiti_minimi_appendiceB.pdf. [Accessed 27 11 2020].

Acknowledgements

For the completion of this Thesis, my sincere gratitude goes to Professor Capozzoli for having believed in me and for his constant clarification throughout.

A special thanks belongs to Silvio and Giuseppe, for the enormous help and support they provide during this period; it would not have been possible to complete this journey without them.

A heartfelt thank you to my parents for the love that you have ever show and for the great sacrifices you have made, and thanks to my sister, for sharing the problems and nervousness during that period.

Thanks also to the rest of my family for their love and encouragement over the years.

Special thanks to my sweetheart Anita, for the enormous patience over the years, for believing and supporting me in achieving goals I thought I was not capable of, and especially for always making me happy.

For making these years lighter, for the enormous help you have always given me, for the good times spent outside the classrooms, I thanks the members of the "Royal Team": Giuseppe B., Roberto and Davide, a big thanks also to Matteo, despite his escape in Paris. I also thanks Mariacarla and the "Termoteam": Francesco, Giuseppe D. and Enzo, with the hope that these friendships will last forever.

Finally, thanks to all my friend in Villastellone, in particular the "96 group", which despite my limited presence in these years has always made me feel part of the group, they have always believed and supported me. Thanks for all adventures we have had over the years, and for those we will have in the future!