Universitat de les Illes Balears

MASTER'S THESIS

MULTI-AGENT REINFORCEMENT LEARNING APPLIED TO HEATING, VENTILATION, AND AIR CONDITIONING IN A BUILDING ENERGY MANAGEMENT SYSTEM

Carlos González Rotger

Master's Degree in Industrial Engineering

Centre for Postgraduate Studies

Academic Year 2020-21

MULTI-AGENT REINFORCEMENT LEARNING APPLIED TO HEATING, VENTILATION, AND AIR CONDITIONING IN A BUILDING ENERGY MANAGEMENT SYSTEM

Carlos González Rotger

Master's Thesis

Centre for Postgraduate Studies

University of the Balearic Islands

Academic Year 2020-21

Keywords:

Multi-agent, Reinforcement Learning, HVAC, BEMS

Thesis Supervisor's Name: Vicente José Canals Guinand

Acknowledgements

In the first place, I would like to thank my supervisor, Dr. Vicenç Canals, because of his trust in my work and valuable feedback. I know he has been busy throughout the full year, with more than 5 Master's Thesis on his plate, and I am glad he still found the time to supervise mine. Our tutorial meetings could last until very late in the evening and he would still be full of energy to answer any doubts.

Of course this master thesis would not have been possible without Trifork, I would like to thank Jørn Larsen and Preben Thorø especially, for providing the inspiration and support for this project, and also for their confidence in my endeavor.

I would also like to thank my colleagues at Trifork Mallorca, as they provided a shoulder to cry on when finding unexpected issues with the simulations, and they would cheer for me whenever I presented new results.

I would like to thank Prof. Nouidui for his quick response regarding a simulation issue, it helped me catch an error in the code.

Finally, I would not like to forget about my family: they have always had my best interest at heart, always curious about what I was doing. I now have understood that one needs to be able to explain a thesis to their parents.

To Beatriz, you more than anyone know how difficult it can be at times to bear with a partner that is also a full-time worker and a Master's Degree student. You have always been supportive, and I know you were looking forward to the time I would be writing the last words of my thesis. Here they are, and this is dedicated to you. I would not have been able to achieve this without you.

Carlos González Rotger
Palma de Mallorca, August 20^{th}, 2021.

Abstract

The EU aims to be climate-neutral by 2050, focusing on promoting renewable sources and energy efficiency. As of 2021 it is required that all the new buildings consume very low net energy (Nearly Zero-Energy Building, NZEB). In order to support this, improvement over HVAC systems control and predictive models for thermal conditions are pointed out as key factors. Building Energy Management Systems (BEMS) are an implementation of such systems that are gaining interest from the authorities.

This master thesis presents the case of study of a new office building in Aarhus, Denmark, where the BEMS will be tested, and focuses on the design and implementation of a proposed controller for the heating and ventilation, using two abstractions of the building—called *dev* and *test*—on a building energy simulator, Energyplus.

The contribution of this thesis is two-fold. First, it presents a state-of-the-art integration between Energyplus and a standard interface used in Reinforcement Learning problems, OpenAI Gym. Second, it develops a high-level decentralized controller using Multi-agent Reinforcement Learning (MARL) to actuate individual room setpoint temperatures and fans mass airflows. The system is trained using the previous integrated simulation tool, and can be deployed to the framed building.

Comparison to a baseline rule-based controller shows it is possible to achieve both energy savings and improved thermal comfort, with an acceptable air quality, and that there is a Pareto frontier of optimal choices in the trade-off between these conflicting goals. It is also observed that the trained controllers on the *dev* building abstraction are able to perform well on the *test* building too, meaning they can adapt to different building configurations.

Resum

La UE té per objectiu arribar a la neutralitat climàtica al 2050, promovent l'ús d'energies renovables i l'eficiència energètica. Al 2021 ja es requereix que els edificis de nova construcció consumeixin molt poca energia neta (Nearly Zero-Energy Building, NZEB). Per suportar aquest impuls, la millora dels sistemes de control dels HVAC i els models predictius de les condicions tèrmiques són assenyalats com a factors clau. Els BEMS són una implementació d'aquest tipus de sistemes que estan guanyant interès per part de les autoritats.

Aquest treball presenta el cas d'estudi d'un nou edifici d'oficines a Aarhus, Dinamarca, on es desenvoluparà el BEMS, i es centra en el disseny i la implementació d'una proposta de controlador per a la calefacció i la ventilació, emprant dues abstraccions distintes de l'edifici en qüestió (*dev* i *test*) en un simulador energètic d'edificis, Energyplus.

La contribució d'aquest treball és doble. En primer lloc, presenta una novedosa integració entre Energyplus i una interfície comuna en problemes de RL, OpenAI Gym. En segon lloc, desenvolupa un controlador d'alt nivell descentralitzat emprant MARL per actuar les temperatures de consigna de les habitacions, així com els fluxos màssics dels ventiladors. El sistema s'entrena mitjançant l'eina de simulació integrada mencionada prèviament, i es pot desplegar a l'edifici plantejat.

Una comparació respecte d'un controlador basat en regles mostra que és possible aconseguir a la vegada un estalvi energètic i una millora en les condicions de comfort, mantenint una qualitat d'aire acceptable, i que hi ha una frontera de Pareto de decisions òptimes entre aquests objectius en conflicte. També s'observa com els controladors entrenats sobre l'abstracció *dev* de l'edifici presenten un bon comportament en l'altra abstracció (*test*), indicant que es poden adaptar a diferents configuracions d'edificis.

Contents

List of Figures

List of Tables

Acronyms

ACH Air Changes per Hour.

AHU Air Handling Unit.

API Application Programming Interface.

ASHRAE American Society of Heating, Refrigerating and Air-Conditioning Engineers.

BCVTB Building Controls Virtual Test Bed.

BEMS Building Energy Management Systems.

EMS Energy Management System.

FMI Functional Mockup Interface.

FMU Functional Mockup Unit.

GAE Generalized Advantage Estimator.

GUI Graphical User Interface.

HVAC Heating, Ventilation, and Air Conditioning.

MARL Multi-agent Reinforcement Learning.

MDP Markov Decision Process.

MPC Model Predictive Control.

NZEB Nearly Zero-Energy Building.

OPC Open Platform Communication.

PPO Proximal Policy Optimization.

RL Reinforcement Learning.

Chapter 1

Introduction

1.1 Motivation

According to the study "Energy use in buildings" [1], in the EU heating consumes between 50 and 80% of the total energy in a building, especially in commercial buildings, and while this consumption is on the lower end in the Mediterranean countries, it is higher in the Northern countries. In addition, the 75% of the energy spent in heating and cooling still comes from fossil fuels [2]. Nevertheless, the EU Energy Performance of Buildings Directive requires, as of 2021, that all the new buildings are Nearly Zero-Energy Building (NZEB), i.e. very energy efficient [3].

On the other hand, previous research work shows that there is room for improvement: [4] discusses designs for glazed office buildings, and [5] estimates the potential savings in university campuses in up to 30%. The review in [6] points out an improvement of Heating, Ventilation, and Air Conditioning (HVAC) control, and predictive models for energy consumption—due to the prevalence of behavior patterns—as important factors to improve the energy consumption. Precisely, one of the targets from the EU energy policies to lower the consumption is the use of Smart Buildings [2], or Building Energy Management Systems (BEMS).

1.2 Context

The project in which this master thesis is framed is a new office building that will be developed in Aarhus, Denmark (see Fig. 1.1), with the intention to serve as a demonstration of the potential savings that come with a sustainable design and the BEMS technology.

The building will be mainly wooden, and it is designed to be a NZEB, naturally ventilated for the most part of the year, given the cold climate conditions. In the winter it will be heated by hot-water convectors and a radiant floor.

1.3 Scope

The aim of this thesis is to provide a state-of-the-art decentralized controller for the heating and ventilation in the proposed building, that saves energy without sacrificing thermal comfort or air quality.

Due to the fact that the building is still on its early development stage, the scope of this thesis will be limited to benchmarking with the use of a building simulator. Nonetheless, the proposed controller should be applicable to the real building.

Figure 1.1: 3D view of the new office building that will be developed in Aarhus, Denmark. Source: AART architects. Courtesy of Trifork.

1.4 Document structure

This thesis is organised in the following manner: it first presents a theoretical background and related work in Chapter 2. Then it further describes the case of study in context, in Chapter 3, and the design and implementation proposal, in Chapter 4. Next, it shows the analysis and results in Chapter 5, and finally it provides an overview of the conclusions and future work in Chapters 6 and 7, respectively.

Chapter 2

Background and related work

In this chapter, the theoretical background will be presented, as well as previous work that is similar to the methodology approached in this master thesis.

2.1 Building Energy Systems

Buildings are complex systems that can be decomposed into interacting subsystems [7]. In this section the energy systems in a building will be introduced, in particular, how a building exchanges energy in an environment. Thus, it is interesting to define the **thermal zones**—simply zones from now on—as areas in the building that can be assigned a single average temperature that is to be controlled because they share similar heating or cooling conditions [8]. These zones will exchange energy among each other, with their occupants and with the exterior (Fig. 2.1). Appendix A includes a short introduction to the different heat transfer methods.

2.1.1 Thermal loads

From the point of view of each zone, the energy exchanges will be seen as **thermal loads** that need to be compensated in order to maintain a set point temperature. Now the different thermal loads will be exposed.

Radiation

On the one hand the external surfaces of a zone will exchange heat in the form of long-wave radiation with the environment, which can be modelled as two different radiant surfaces with uniform temperatures: the sky, and the ground. On the other hand, there's also a short-wave radiation coming from the Sun, either directly—beam irradiance—or indirectly—diffuse irradiance—.

Radiation will heat up the building external surfaces through daylight hours, which in turn will imply a delayed conduction heat transfer through the walls into the building due to their thermal mass. However, if the surfaces receiving the radiation are windows, the effects will be immediate since it is assumed that windows don't have a significant thermal mass [9].

Conduction losses

The temperatures' difference between the zones and the exterior or the ground will drive heat exchanges in one direction or another depending on which one is hotter or colder. Again, due to the thermal mass from the walls and structure in the building, the effect will be delayed. These losses will be especially relevant during the winter periods, where the zones' temperatures will be significantly higher than the external temperatures. During most part of the summer, given the climate conditions in context, there will still be net losses, which will help towards keeping the temperatures low in those zones.

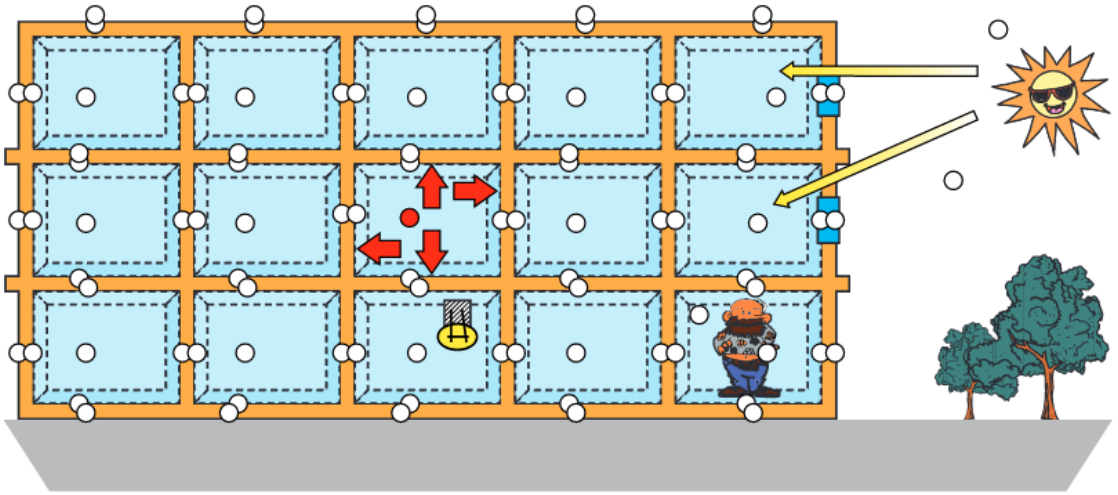


Figure 2.1: A zone exchanges heat with other zones, with its occupants and with the environment. Source: IDAE [9].

It is relevant to note that the heat exchanged among zones during the winter will not be significant, as they will be at similar temperatures. In the summer, and given the natural ventilation design, there should be both heat and air mass transfers from the perimeter zones to the atrium.

Ventilation

Renovation of air is important to ensure a good air quality and prevent related illnesses, such as the sick building syndrome [10]. The American Society of Heating, Refrigerating and Air-Conditioning Engineers (ASHRAE) provides recommendations for minimum ventilation rates per person and per zone area depending on the building type and activity [11]. From the thermal point of view, though, this represents a thermal load as air needs to be treated to preserve indoor conditions. Equations (2.1), (2.2) calculate the sensible and latent heats involved in the process.

$$Q_s = V \frac{1}{\nu_{out}} C_p (T_{out} - T) \tag{2.1}$$

$$Q_l = V \frac{1}{\nu_{out}} h_{we} (w_{out} - w) \tag{2.2}$$

where V is the dry air volume flow, ν_{out} is the specific volume of the outdoor air expressed as volume of moist air divided by mass of dry air, C_p is the heat capacity of dry air, h_{we} is the latent heat for water vaporization, T_{out} and T are the outdoor and indoor dry-bulb temperatures, respectively, and w_{out}, w are the humidity ratios likewise.

As an example, Fig. 2.2 shows a psychrometric chart with the design climate conditions for the building location in context. There the extreme climate conditions at 1 and 99% have been represented, along with plausible indoor points, and it is clear how ventilation is especially relevant in the winter, where there is an important sensible heat load to be compensated.

Incidentally, latent heat exchange can only be accomplished by humidity corrections, either humidifying or dehumidifying the air. This can be partly accomplished by recuperative heat exchangers, as they transfer both moisture and heat between the exhaust air and the fresh outdoor air [12]. These exchangers can only bring the system to an equilibrium point: they cannot remove indoor moisture if the outdoor absolute humidity is higher. However, for office buildings the latent load is in general about 20% of the total load only [13].

In addition to the forced mechanical ventilation using the fans, there is also a natural ventilation. The latter will be greater as the number of openings increases (doors, windows, and vents), and it's not possible to recover any heat or moisture from the intake or exhaust air.

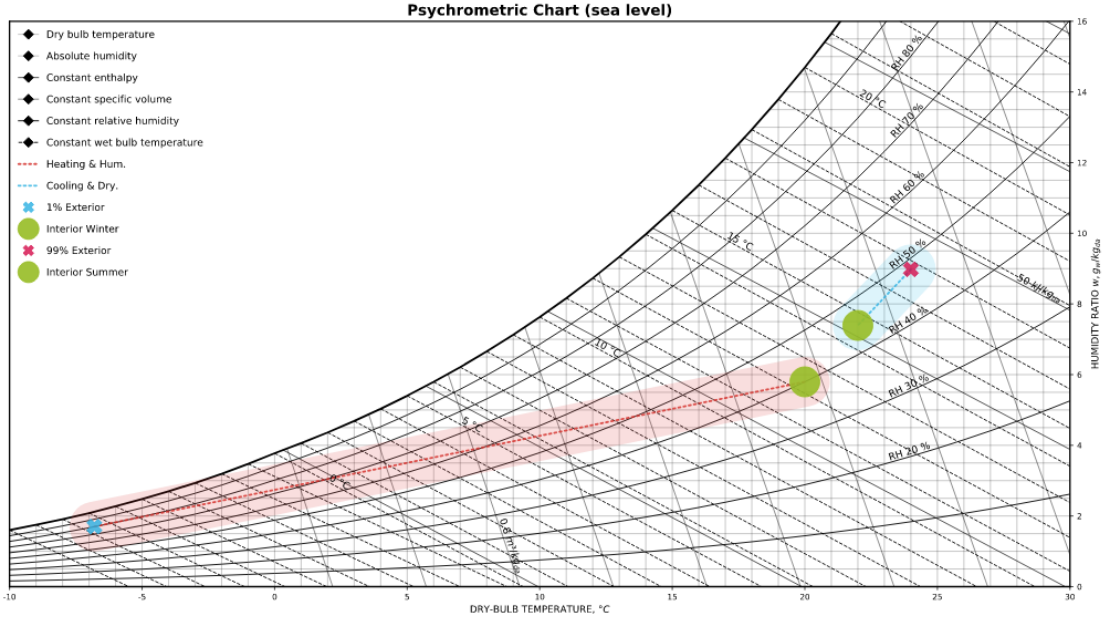


Figure 2.2: Psychrometric chart with design weather conditions. Source: own elaboration using *psychrochart* [14] and weather data from [15].

Internal gains

The last thermal load comes from the interior of each zone: the electric equipment, the lightning and the occupants generate heat that will need to be dissipated to keep indoor conditions. On the one hand the electric and lightning appliances will produce sensible heat exclusively, but the occupants exchange both sensible heat (the skin temperature is higher than the temperature from indoor air) and latent heat (due to respiration and evaporation from sweating).

2.2 Building Energy Management Systems

Building Energy Management Systems (BEMS), also known as Building Control Systems (BCS) or Building Automation Systems (BAS) have been thoroughly studied: they comprise the set of sensors, controllers, actuators, and the related infrastructure to connect them, with the goal of reducing the energy expenditure while maintaining an acceptable level of comfort [7, 16–18]. The reviews in [16, 17] do a literature classification based on a definition of different factors for comfort, and distinct controller methods. In what follows, the factors of comfort and controllers that are most relevant to this work will be outlined.

2.2.1 Factors of comfort

Thermal comfort

The operative temperature in a given room is defined as the uniform temperature of a black-body enclosure that would exchange the same amount of heat with its occupants as the radiant (from other surfaces or occupants) and convective (from the sorrounding air) fractions from the real environment combined [19]. In practice, this temperature is estimated as in Eq. (2.3), where T_{op} is the operative temperature, T_{rad} is the mean radiant temperature from the walls, and $T_{dry\ bulb}$ is the average indoor air dry-bulb temperature, using $\gamma = 0.5$ [20].

$$T_{op} = \gamma T_{rad} + (1 - \gamma) T_{dry\ bulb} \tag{2.3}$$

However, the thermal comfort is a measurement related not only to the temperature, but also to the thermal, physiological, and psychological responses of the occupants [21]. These won't

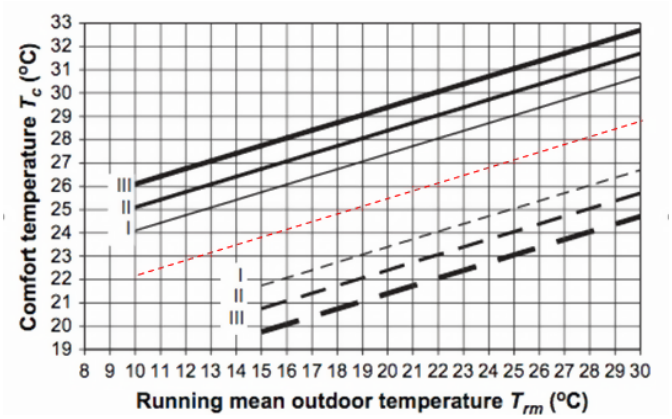


Figure 2.3: Categories of comfort according to the adaptive comfort model for naturally ventilated spaces from EN15251:2007 "Indoor environmental input parameters for design and assessment of energy performance of buildings addressing indoor air quality, thermal environment, lighting and acoustics". Category I (> 90%) is the highest level of comfort, within ±2 °C from the ideal temperature, Category II (> 80%) is the expected level of comfort for new buildings, within ±3 °C, and Category III (> 65%) defines the lowest level of comfort, only acceptable for existing buildings, within the ±4 °C range. Source: Energyplus Engineering Reference [24].

be the same for everyone, and thus there will always be a percentage of dissatisfied people. For naturally ventilated buildings adaptive comfort models have been defined [22], considering that people adapt their clothing to improve their comfort, so these models take into account running averages of outdoor temperatures from previous days to provide the ideal comfort operative temperature. Deviations from it will increase the percentage of dissatisfied people.

In the EU, the applicable standard was the EN15251:2007, being replaced in 2019 by the EN16798:2019 [23]. It sets the goal in 80% of comfort for new buildings without especial requirements (e.g. hospitals and daycare centers do need extra comfort). See Fig. 2.3 for the distinct categories defined.

Air quality

Indoor air quality is known to be affected by pollutants released by occupancy, chemical agents used for cleaning, cooking systems, and other natural sources that depend on the building location and activity [25], being CO_2 and Volatile Organic Compounds (VOCs) the most commonly studied. Other than posing a health risk, poor air quality also leads to a reduction of occupants' productivity [26].

Interestingly, ASHRAE does not provide a recommendation for the maximum CO_2 level in its recent versions [27]. However, it is known that high levels can cause drowsiness and headaches, starting from the range 2000-5000ppm, and CO_2 has been widely used [16, 17, 26], probably as a proxy to other parameters that are more difficult to measure and can be more related to the overall air quality, such as the age of air.

Relative humidity

As mentioned previously, a good ventilation prevents the occupants from being exposed to high concentrations of pollutants, whereas it can increase the relative humidity too much if humidity is not being actively regulated. This factor is relevant both for comfort and safety: having a relative humidity that is too high can lead to the growth of mould, and having it too low can cause sinus irritation and other respiratory syndromes. ASHRAE recommends that relative humidity stays within 30 and 60% to avoid these issues [28].

2.2.2 Controllers

Classic controllers

Classic control methods involve the combined action of proportional (P), integral (I) and, sometimes, also derivative (D) controllers [29]. This control is said to be purely reactive (to an error signal), and so it is limited because it will not act until the error appears. In addition, it requires manual tuning and previous experience to adequately choose the parameters [16].

Nonetheless, these controllers are still very extended in the industry for low-level control of position and velocity, and they are compatible with higher-level supervisory control methods, as loop feedback details are abstracted away. For instance, for motion control tasks, having a supervisor learning method that chooses joint angles and angular velocities over directly providing motor voltage, and having instead PD controllers for that task has shown to improve learning speed and final performance [30].

Adaptive controllers

Fuzzy logic and parameter estimation methods are an improvement over the classic control methods because there is no need for fine-tuning the PID parameters manually, but instead these are changed depending on the environment conditions [31, 32]. However, these still have problems handling non-linearities in real environments or non-modelled signals that are seen as noise.

Model Predictive Control

Another approach is to act ahead of the error by providing estimates or predictions for uncertain variables, like weather forecast or future occupancy. Model Predictive Control (MPC) uses these predictions, and an optimization model that relates input variables (current state including predictions) to output variables (total cost) to solve for the best inputs that will minimize the cost—through a convex optimization process.

It has been shown to work well in building thermal control projects [17], and some of its drawbacks are: it is difficult to build such optimization model, as it requires expertise—it is an adhoc design—and it is computationally expensive to solve at each step.

Distributed control – agent-based

Having defined that a building is divided in thermal zones, it naturally follows a control technique that has been seen frequently: to distribute the decision making into those zones, by having independent **agents** [7, 16, 17]. These agents are provided with local goals they need to meet, and can cooperate among each other. This setup is known to be flexible, extensible and robust [7]. The difficulty that arises with it is dealing with the coordination and negotiation between the agents.

Model free control – related work

Finally, there are control strategies that do not require a explicit model that relates the current state and the goal. Rather, they interact in an environment, observing inputs, and receiving delayed feedback that is used to improve continuously. It is remarkable that these strategies are compatible with a multi-agent setup (distributed control).

This is the use case for Reinforcement Learning (RL), and a whole Section (2.4) is devoted to the theory that supports it. Non-comprehensive literature reviews in [33, 34] show an extensive amount of previous works using RL, and present the state of the matter. Here some additional related work will be commented.

The works in [35, 36] control the airflow in different setups: [35] modulates the fan variable speed inverter frequency in a subway station, while [36] controls the dampers positions of a centralized Air Handling Unit and the supply airflow rates for each zone in a multi-zone commercial building. They both use custom models to define the environment with which the RL interacts.

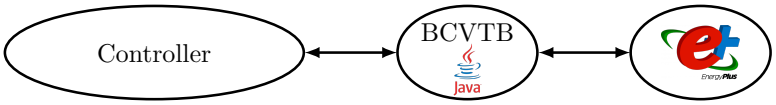


Figure 2.4: Coordination of the controller and Energyplus—slave processes—using BCVTB—the master process. Source: own elaboration using *graphviz* [39].

Regarding temperature setpoint control, [37, 38] have developed a similar framework to the one in this master thesis, and they control the hot water supply setpoint and room temperature setpoint, respectively, using their framework and RL controllers. Their frameworks integrate Energyplus simulator with a middleware layer called Building Controls Virtual Test Bed (BCVTB), which is written in Java language and becomes the master process coordinating communication between different slave programs (Fig. 2.4). One drawback of this middleware is that it slows down the calculations [38], as there is an extra tool that needs to coordinate the communication.

More information on the controllers themselves will be presented in Section 2.4.

2.3 Building Energy Simulator

In the Introduction Chapter it was stated that the aim of this work is to apply a BEMS to a real building that is under development. As a proxy to that building, the alternative is to use a model that realistically approximates the evolution of the real building in terms of energy exchanges. This is what a building energy simulator does.

2.3.1 DesignBuilder

DesignBuilder [40] is a commercial software that provides a Graphical User Interface (GUI) on top of Energyplus, the core simulator, explained next. The GUI eases development process, as the 3D model of the building can be prepared with the different thermal zones, and parameters can be specified in different tabs (Fig. 2.5).

It is important the notion of **schedules**, which allow defining rule-based values depending on the time of day, and the type of day (working day, holidays, weekend) as shown in Listing 2.1. These schedules can then be assigned in the GUI to set occupancy, HVAC heating and cooling setpoints, door and window openings, and any other schedulable control.

Finally, it is also interesting to use because the GUI also supports drag-and-drop definitions of the HVAC detailed system, in a very intuitive way (see an example in Fig. 2.6).

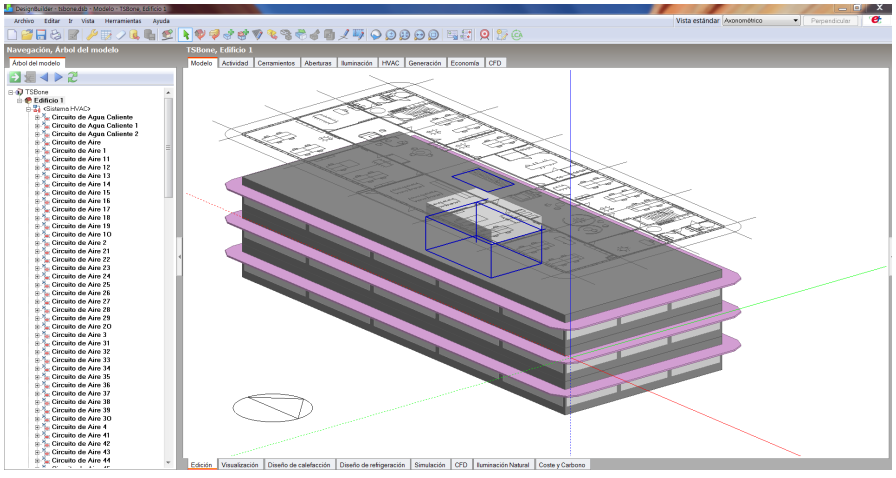


Figure 2.5: Screenshot from DesignBuilder modelling tab. Source: DesignBuilder software [40].

```
Schedule:Compact,
ASHRAE 90.1 Occupancy - Office,
Fraction,
Through: 31 Dec,
For: Weekdays,
Until: 06:00,       0,
Until: 07:00,       0.10,
Until: 08:00,       0.20,
Until: 17:00,       0.95,
Until: 18:00,       0.30,
Until: 22:00,       0.10,
Until: 24:00,       0.05,
For: Saturday,
Until: 06:00,       0,
Until: 08:00,       0.10,
Until: 12:00,       0.30,
Until: 17:00,       0.10,
Until: 19:00,       0.05,
Until: 24:00,       0,
For: Sunday,
Until: 06:00,       0,
Until: 18:00,       0.05,
Until: 24:00,       0,
For: SummerDesignDay,
Until: 08:00,       0,
Until: 23:00,       1,
Until: 24:00,       0,
For: AllOtherDays,
Until: 24:00,       0      ;
```

Listing 2.1: Schedule example. Source: Default schedule in DesignBuilder.

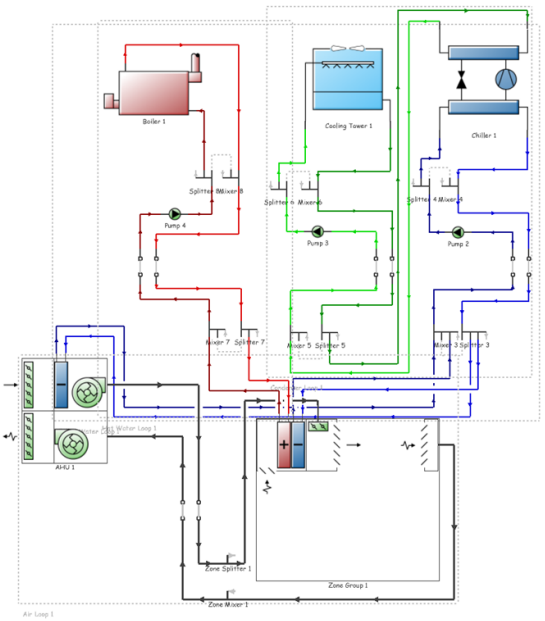


Figure 2.6: Example of detailed HVAC. Source: DesignBuilder Help [40].

The main limitation of schedules is that values are fixed during the whole simulation, so there is no chance to hook in and assign some calculated value while simulating.

2.3.2 Energyplus

Energyplus [41] is a Building Energy Simulator that is funded by the U.S. Department of Energy, it is open-source, and it has extensive documentation available [42].

In Energyplus, simulations have a constant time step size that is defined beforehand. Normal values sit between 5 and 15 minutes. It means that calculations are performed once every X minutes, being X the desired time step size.

To run the simulation, it expects the following text files as input, which are written out by DesignBuilder once the building has been modelled (see Fig. 2.7):

1. An input dictionary defining what is valid input, in ".idd" format.
2. A weather file in ".epw" format.
3. The input file describing the location and the building itself, in ".idf" format.

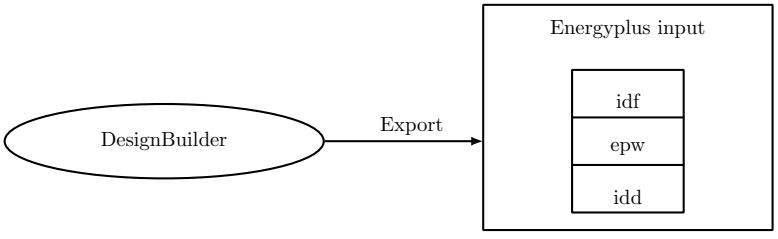


Figure 2.7: DesignBuilder writes the input files to Energyplus. Source: own elaboration using *graphviz* [39].

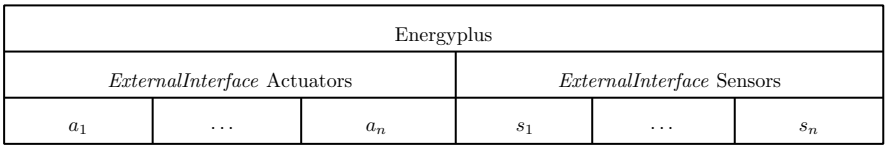


Figure 2.8: Energyplus with *ExternalInterface* actuators and sensors. Source: own elaboration using *graphviz* [39].

Energyplus shares the concept of schedules with DesignBuilder, and their limitations. To overcome that, it provides a feature called Energy Management System (EMS) that allows defining programs and hook them into certain points during the simulation execution. This lays the ground to allow dynamic actuators that take decisions based on specific conditions.

Precisely, on top of the EMS language Energyplus provides an object called *ExternalInterface* that allows an external program to read variables or set actuator values (Fig. 2.8). This is what has been used so far by previous work to connect the simulator to the BCVTB middleware, and so to the controller.

Actuators

In Energyplus actuators can overwrite schedule values during the simulation, and also other object properties—input is defined as a set of objects, some of their properties are overwritable, as documented in [42].

For each thermal zone, a dual setpoint with a deadband can be defined, i.e. a cooling setpoint over which cooling will be activated, and a heating setpoint below which heating will be activated. Both setpoints can be actuated at any timestep.

Going back to DesignBuilder, in the detailed HVAC system the airloop can be defined drawing air handling units and connections to groups of zones. It will then write to the ".idf" file all the objects and links that represent the same in the Energyplus world. Among these objects, there is the fan component, which can be either constant or variable air volume. In both cases the actuator can set the mass air flow at any timestep.

Sensors

In a similar way to the actuators, sensors can be read out at any timestep into another application by using the *ExternalInterface* object, and there are many. Some examples include all the psychrometric properties of air at any zone or outdoors, environmental conditions like wind or rain, any flow through the HVAC system, and heating and cooling rates, down to the surface level (heat exchanged through a given surface).

2.3.3 Known limitations

Being a simulation, there are some interactions that cannot be modelled correctly. First, even if CO_2 generation is supported by defining a rate that depends on the metabolic rate (in $m^3/(sW)$), and airflows are calculated between zones depending on pressure and temperature differences–through the Airflow Network algorithm [43]—Energyplus does not consider transport of pollutants across zones, CO_2 included. So, it will be necessary to turn on the mechanical ventilation to lower the concentration, even if by natural ventilation it would have dropped in a real environment. This needs to be taken into consideration when implementing the controller in the real building, as the use of ventilation might sometimes be unnecessary.

Regarding the partition into thermal zones and definition of openings, the Airflow Network model does not work well with large horizontal openings, like a rooftop vent. It is recommended that openings are vertical.

Also, the Airflow Network model will only work with constant volume fans, even though their values can still be overwritten by actuators. It is thus a minor issue.

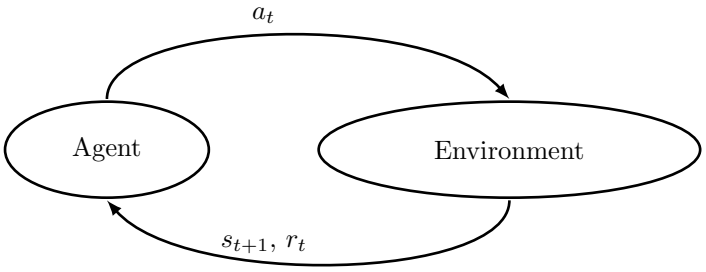


Figure 2.9: Interaction between an agent and the environment in a RL setup. Source: Own elaboration using *graphviz* [39].

2.3.4 Functional Mockup Interface

Functional Mockup Interface (FMI) is a standard [44] that defines how to group functionality in the form of compiled code and configuration files into a *zip* file, so that the resulting package—called Functional Mockup Unit (FMU)—can be exported and used across applications. The work in [45] made this standard available to Energyplus, so that it is possible to export it as a FMU. In fact, a new generation of Energyplus was proposed in [46], that will give it more flexibility in terms of modularity and integration with other tools. To achieve this, FMU is the cornerstone for the communication.

2.4 Reinforcement Learning

In Reinforcement Learning (RL) the controller is modelled as an agent interacting with an environment: at any given timestep t it observes a given *state* s_t, takes an *action* a_t, and in return it receives a *reward* r_t, along with a new observation s_{t+1}. This process is repeated at each timestep, it is represented in Fig. 2.9. The mathematical formulation for the problem RL tries to solve is the maximization of total future rewards in a possibly infinite-horizon sequence of timesteps. More formally, the sequence of steps can be formulated as a Markov Decision Process.

2.4.1 Markov Decision Process

A Markov Decision Process (MDP) [47] is defined as a tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, r, \gamma \rangle$, where $\mathcal{S}$ is the set of possible states that can be observed, $\mathcal{A}$ is the set of actions that can be taken, $\mathcal{P}$ is the set of transition probabilities $P(s_{t+1}|s_t, a_t)$, r is the reward function: $\mathcal{S} \times \mathcal{A} \to \mathbb{R}$, and $\gamma \in (0, 1)$ is defined as a discount factor for future rewards, to account for uncertainty and also as a convenient mathematical trick to bound the infinite sum of future rewards at any given time t, also called *return*, shown in Eq. (2.4).

According to the Markovian property, the any state S_t is independent of previous states because it already summarizes all the history, i.e. future states depend only on the current state and the action taken, as can be seen from the definition of the transition probabilities. In a model-free setup these are not known, and must be inferred through interactions with the environment.

$$R_t = \sum_{k=0}^{\infty} \gamma^k r(s_{t+k}, a_{t+k}) \tag{2.4}$$

At the same time, a policy π can be defined as the decisions the controller will take given some state, in case this is stochastic it follows $\pi(a_t|s_t) = P(a_t|s_t)$. Marginalization over the state and action combined allows to write the Eq. (2.5). Then, the probability of observing a given sequence of states and actions, also called *trajectory*, can be written as shown in Eq. (2.6), and the optimal policy π^* will maximize the expected return of any trajectory as shown in Eq. (2.7).

Maximizing over trajectories means that in any given situation the agent would always choose the optimal action leading to the best sum of rewards.

$$\pi(s_t, a_t) = \pi(a_t|s_t)P(s_{t+1}|s_t, a_t) \tag{2.5}$$

$$p(\tau) = P(s_1, a_1, s_2, a_2, ...) = P(s_1)\prod_{t=1}^{\infty}\pi(s_t, a_t) \tag{2.6}$$

$$\pi^* = \arg\max_{\pi} \mathbb{E}_{\tau\sim p(\tau)}\left[R_1\right] = \arg\max_{\pi}\sum_{t=1}^{\infty}\mathbb{E}_{(s_t,a_t)\sim\pi(s_t,a_t)}\left[\gamma^{t-1}r(s_t, a_t)\right] \tag{2.7}$$

2.4.2 State-action and state value functions

Other definitions that are useful for solving the presented problem are the *state-action value function*, and the *state value function*. The former is defined as the expected return given a state s_t and action a_t and following a policy π, see Eq. (2.8). The latter summarizes possible actions and it takes only the state into consideration: it is defined as the expected return given a state s_t and following a policy π, see Eq. (2.9). In the equations the sampling of state-action pairs from $\pi(s_t, a_t)$ has been shortened to simply the policy π for clarity.

From Eqs. (2.7) and (2.9) it can be seen how the optimal policy maximization objective is $\mathbb{E}_{s_1\sim P(s_1)}\left[V^{\pi}(s_1)\right]$, i.e. the expected value of the value function over any initial state s_1.

$$Q^{\pi}(s_t, a_t) = \mathbb{E}_{\tau\sim p(\tau)}\left[R_t|s_t, a_t\right] = \sum_{k=0}^{\infty}\mathbb{E}_{\pi}\left[\gamma^k r(s_{t+k}, a_{t+k})|s_t, a_t\right] \tag{2.8}$$

$$V^{\pi}(s_t) = \mathbb{E}_{\tau\sim p(\tau)}\left[R_t|s_t\right] = \sum_{k=0}^{\infty}\mathbb{E}_{\pi}\left[\gamma^k r(s_{t+k}, a_{t+k})|s_t\right] \tag{2.9}$$

Finally, the relationship between $Q^{\pi}(s, a)$ and $V^{\pi}(s)$ is given by Eq. (2.10).

$$V^{\pi}(s_t) = \mathbb{E}_{a_t\sim\pi(a_t|s_t)}\left[Q^{\pi}(s_t, a_t)\right] \tag{2.10}$$

2.4.3 Bellman optimality equation

Using the relationship from Eq. (2.10) and the state value function, a recursive decomposition of the discounted return from Eq. (2.4) can be obtained, yielding the Bellman optimality equation developed in Eq. (2.11). It defines the optimal value that can be achieved starting in state s, via the optimal policy π^*.

In the 2nd to 3rd step of the equation, the fact that the optimal policy will always choose the action that maximizes the value function (P = 1) allows to get rid of the expectation. From 3rd to 4th step, the linearity of expectation is used to move the sum inwards.

$$\begin{aligned}
V^*(s_t) &= \max_{\pi} V^{\pi}(s_t) \\
&= \max_{\pi}\mathbb{E}_{a_t\sim\pi(a_t|s_t)}\left[Q^{\pi}(s_t, a_t)\right] \\
&= \max_{a_t} Q^{\pi*}(s_t, a_t) \\
&= \max_{a_t}\mathbb{E}_{\pi*}\sum_{k=0}^{\infty}\left[\gamma^k r(s_{t+k}, a_{t+k})|s_t, a_t\right] \\
&= \max_{a_t}\mathbb{E}_{\pi*}\left[r(s_t, a_t) + \gamma V^*(s_{t+1})|s_t, a_t\right]
\end{aligned} \tag{2.11}$$

2.4.4 Algorithms

After showing the optimality equation, it is necessary a discussion on how to effectively compute the optimal policy, given that the Eq. (2.11) is recursive. In general, the different model-free algorithms in RL fall into one of the following main categories: policy gradient, value function approximation, and actor-critic. After the main categories are introduced, the Proximal Policy Optimization (PPO) algorithm is presented.

Policy gradient

To begin with, without loss of generality a policy $\pi(a_t|s_t)$ can be any function parametrized by parameters θ, giving $\pi_\theta(a_t|s_t)$. These parameters can be changed, updating how the policy will choose the actions. Policy gradient methods specify how to change these to obtain the optimal policy.

After sampling the environment by running trajectories with the current policy π_θ, the expected return can be obtained by averaging the returns for each trajectory. Then, using a Gradient Descent method [48] the policy parameters θ are updated. A step of parameters update is shown in Alg. 1.

Algorithm 1 Simple policy gradient step. This is repeated until convergence of θ.

for $k \in 1..N_{trajectories}$ **do**
  for $t \in 1..N_{steps}$ **do**
    Given current state s_t
    $a_t \sim \pi_\theta(a_t|s_t)$
    $s_{t+1} \sim P(s_{t+1}|s_t, a_t)$
    Accumulate into $R_k \leftarrow r(s_t, a_t)$
  end for
end for
$R_\tau \leftarrow \frac{1}{N\,trajectories} \sum_{\forall k} R_k$
Update $\theta \leftarrow \theta + \alpha \nabla_\theta R_\tau$

Value function approximation

Another option is to parametrize the value functions from the optimal policy, either $V_\theta^{\pi*}$ or $Q_\theta^{\pi*}$, without explicitly defining the policy π^*. At each step, the value function is updated using the Bellman optimality equation, and the optimal action can be chosen greedily, i.e. the one that maximizes the value function, or in a exploit-explore manner, where the maximizing action is chosen with great probability—exploit knowledge—, but the algorithm can also choose other actions with smaller probability—explore alternatives—.

A simple step of Q-value function approximator is shown in Alg 2. It uses an approximation for $Q_\theta^{\pi*}$ that chases a moving target—the optimal—, although it can be proved that the greedy value function updates monotonically converge to the optimal value [49].

Algorithm 2 Simple Q-value step. This is repeated until convergence of Q_θ.

for $t \in 1..N_{steps}$ **do**
  Given current Q_θ and state s_t
  $a_t \sim \arg\max_{a_t} Q_\theta(s_t, a_t)$
  $s_{t+1} \sim P(s_{t+1}|s_t, a_t)$
  if s_{t+1} is terminal state **then**
    $Q_{target} \leftarrow r(s_t, a_t)$
  else
    $Q_{target} \leftarrow r(s_t, a_t) + \gamma \max_{a_{t+1}} Q_\theta(s_{t+1}, a_{t+1})$
  end if
  Update $\theta \leftarrow \theta - \alpha \nabla_\theta (Q_\theta(s_t, a_t) - Q_{target})^2$
end for

Actor-critic

Finally, a combination of the previous two options makes it possible to define an actor-critic pair: the actor is a parametrized policy that is directly updated using policy gradient, while the critic is an estimator of the advantage function. The advantage function can take many shapes, but it is intended to be a measure of how valuable a taking a given action in a given state with respect to the value of being in that state. It has been demonstrated to decrease variance and speed up learning [50, 51].

Proximal Policy Optimization

Proximal Policy Optimization (PPO) [52] is an actor-critic algorithm. It estimates both the policy and the value function, and then uses the Generalized Advantage Estimator (GAE) algorithm [53] to calculate the advantage function.

It also implements importance sampling [54] to sample from the updated policy (after parameter update) while actually using the old policy (before the update), improving the sampling efficiency. This makes it an **on-policy** algorithm, meaning that it always takes decisions following the current policy. On the contrary, other algorithms like Q-value are said to be **off-policy**, because they might take a different decision—this is possible because they do not need the current policy to improve at each step.

At the same time, it solves an issue from simple policy gradients: when the gradient is too steep, they might take too big steps that overshoot the optimal parameters. PPO solves this issue by clipping the gradient within some boundaries, to ensure the parameter updates are "safe".

It has shown good performance on complex tasks [55, 56].

2.4.5 Parametrization with neural networks

So far, a generic parametrization with parameters θ has been presented, without specifying any form. In this subsection, neural networks are presented as a flexible way to approximate any function [57].

Mathematically, a neural network is a composition of non-linear functions, also called layers: $\text{NN}(x) = f_n \circ ... \circ f_2 \circ f_1(x)$ is the result from a neural network with n layers. Each of the non-linear functions f_i is defined as: $\mathbb{R}^{N_{i-1}} \to \mathbb{R}^{N_i}$, i.e. a function over multiple variables where N_0 denotes the number of input variables to the neural network, and each N_i the number of output variables from the i-th layer.

In addition, each layer can be further expanded to the Eq. (2.12), where $\mathbf{W_i}$ is a matrix of *weights* of shape $N_{i-1} \times N_i$, $\mathbf{x}$ is a vector of length N_{i-1}, $\mathbf{b_i}$ is a vector of *biases* of length N_i, and σ is a non-linear function that is applied element-wise to the resulting vector.

$$\mathbf{f_i}(\mathbf{x}) = \sigma(\mathbf{W_i} \cdot \mathbf{x} + \mathbf{b_i}) \tag{2.12}$$

Graphically, these concepts can be represented as a staged graph like in Fig. 2.10, where nodes represent inputs and outputs from each layer, and connections represent the weights of the layer. Each layer has an additional input fixed to 1 to account for the bias.

Backpropagation

It turns out the weights $\mathbf{W_i}$ and the biases $\mathbf{b_i}$ for all the layers are the parameters θ mentioned previously. They can be updated by using a differentiation technique called backpropagation [58], that adjusts their values proportionally to the error they introduce.

Importance of normalization

Initially, the weights and biases for all the layers are initialized randomly, and initialization plays an important role in resulting performance [59]. In addition, the input values should be

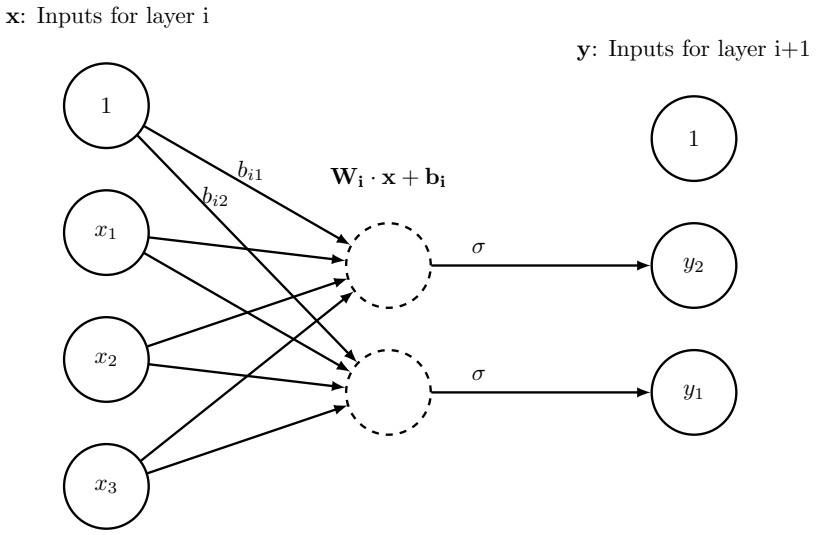


Figure 2.10: Example of i-th layer of a neural network with 3 inputs and 2 outputs. Weights $\mathbf{W_i}$ have not been drawn on each edge for clarity. The dashed circles represent the result from applying the linear mapping to the inputs, before applying the non-linearity σ. Source: self-generated using *graphviz* [39].

normalized, ideally to achieve zero mean and unit variance. This will help the learning and prevent the network getting stuck with no improvement.

Because normalization is important when using neural networks, it is critical to *shape* the actions, observations, and rewards so that the RL agent can make faster and further progress.

Batch training

Rather than doing *forward*—calculation of outputs—and *backward*—backpropagation parameter updates—passes on the neural network using a single input $\mathbf{x}$, it is better to use a batch of inputs and vectorize calculations, taking benefit from the parallelism CPUs and GPUs provide.

Not only it is a matter of speeding up the calculations, but also there is a theoretical argument behind this decision: using batches provides better estimates for the gradient in the backpropagation step [60]. However, batches that are too big may not fit in the memory available—hardware constraint—, and also lead to poorer generalization results, due to overfitting to the training data. Therefore, there is a balance between the training batch sizes that needs to be considered.

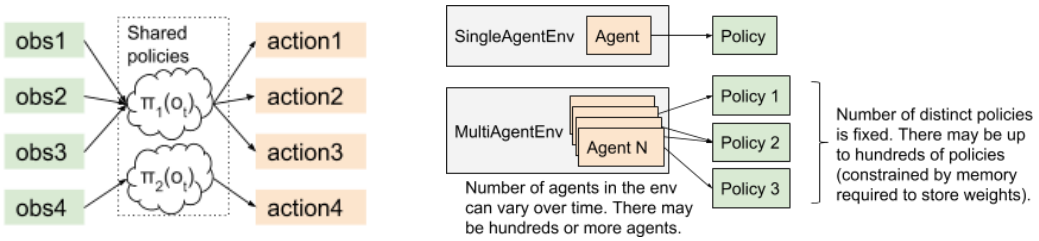
Deep Reinforcement Learning

Neural networks with many layers (more than one) are known as *Deep* Neural Networks, and using them to parametrize functions in RL, either policies or values, is coined as *Deep* Reinforcement Learning.

2.4.6 Environment

While the agent in a RL setup (see Fig. 2.9) has been described already, little information has been provided on the environment that the agent interacts with. Here the necessary interface will be described, that is, the functionality an environment should always provide.

It has been shown how the agent receives observations and rewards, and in turn it chooses actions in a timestep manner. Consequently, the environment should support them: it should be possible to request an initial observation, and it should be possible to apply an action, getting in return a new observation, the reward associated to the previous move, and an indication for the agent to know if the trajectory (recall that it is the succession of state-action pairs) should



(a) Multi-agent environment.

(b) Multi-agent vs. single-agent policy mapping.

Figure 2.11: Multi-agent environment and policy mapping. Source: RLlib [68].

be terminated because it reached a terminal state. For instance, if the agent were to control a walking robot, falling to the ground would be considered as a failure and the trajectory would be terminated, obtaining a negative reward as a result.

OpenAI Gym toolkit [61] is an example of such an interface, it is open-source, and it has been adopted as the standard in RL literature. It is written in python language, probably contributing to its adoption, given the wide range of libraries that exist in that language and make it easy to work with neural networks and reinforcement learning [62–67].

Multi-agent environments

The concept of an environment can be extended to multiple agents [68]: at each timestep, multiple observations can be returned, depending on how many active agents there are—they might not interact at every timestep, but maybe on different timescales—. Likewise, multiple actions can be returned, one from each agent, along with their rewards. This is displayed in Fig. 2.11a.

This setup brings up another choice: which policy controls which agent? It can be the case that each agent has its own policy that is learned individually, they could all share the same policy, that would learn from interactions from all the agents, or they can be grouped under a given policy depending on the type of agent (Fig. 2.11b). For instance, in a traffic control environment there could be a single traffic light agent with its own policy, and all the vehicles in the environment would be agents that share the same policy. In general, a *mapping* function should be provided, from agents to policies, in multi-agent environments.

2.4.7 Shaping of actions, observations, and rewards

In order to avoid the agent do futile actions that do not imply a progress in the task it is required to solve, it is best to reduce the number of possible actions to the ones that are related to that task. It may also be beneficial to discretize the continuous actions to a given set of options, as it will reduce the exploration space. This process is called *action space shaping*, and it has been studied under different environments [30, 69].

On the other hand, the observations will need shaping as well. It means that whenever it is possible, any variables fed into the agent as observations should be scaled to have their natural boundaries between -1 and 1, or between 0 and 1. How to do so will greatly depend on the environment and the task in hand, and sometimes it will be difficult if not impossible to achieve. Moreover, similarly to what happens with actions, the observations provided should be relevant for the task, and they should not be correlated to avoid flat gradients in the parameters' update step [59].

Regarding boolean and discrete variables, where scaling does not make sense, *one-hot encoding* can be used instead: a N-dimensional variable whose values are all 0 except for the i-th position, equal to 1, being i the value of the discrete variable with N values. See Tab. 2.1 for an example of a discrete variable with 3 possible values. Also notice that boolean variables can be thought of as discrete variables with 2 values.

0	$\langle 1, 0, 0 \rangle$
1	$\langle 0, 1, 0 \rangle$
2	$\langle 0, 0, 1 \rangle$

Table 2.1: One-hot encoding of a discrete variable with 3 values.

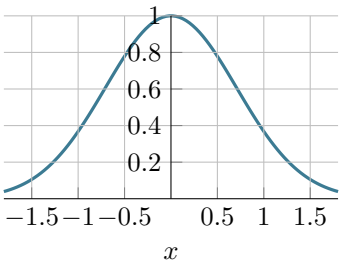


Figure 2.12: Representation of the exponential function e^{-x^2}. It can work as a reward function where x is an error measurement to be minimized.

Last but not least, the rewards are crucial to achieve success at any given task, because they define how the success is measured according to the agent. Rewards that are not very much correlated with actually solving the task will yield poor results, as the agent will learn to solve the wrong task.

Again, the relevance of normalization is highlighted when discussing how to shape the rewards. For instance, the work in [56] uses exponential functions with the negative of squared differences between values and their targets, as the resulting function is fully differentiable and is bounded between 0 and 1, with a maximum on the target (see Fig. 2.12). The authors also normalize the rewards by dividing them by the maximum attainable in each task.

Potential-based reward shaping

In RL, in addition to having the reward as a signal for the agent to learn from, giving it some information beforehand can speed up the learning process. This *prior* information can be encoded into the reward function without altering the optimal policy using a potential function, both for single-agent and multi-agent setups. This technique is called *potential-based reward shaping* [70].

The potential function, $\phi(s)$, is responsible for carrying heuristic information about the goodness of state s that will guide the agent towards better states and action choices. Using the potential values of two consecutive states, a *potential reward* is calculated as a discounted difference, see Eq. (2.13). The reward function is then augmented as shown in Eq. (2.14). Intuitively, this makes the agent care only about improvements of that signal, as comparatively, moving from one state to another one with the same potential value is not interesting.

$$F(s_t, s_{t+1}) = \gamma\phi(s_{t+1}) - \phi(s_t) \tag{2.13}$$

$$r_{aug}(s_t, a_t, s_{t+1}) = r(s_t, a_t) + F(s_t, s_{t+1}) \tag{2.14}$$

Chapter 3

Case of study

This Chapter presents in detail the framework for this thesis, first introduced in Sec. 1.2.

3.1 Building description

The building is a three-story office with a basement, an attic and a rooftop terrace with space for placing photovoltaic solar panels, as shown in Fig. 1.1. The floor plans, as well as explanatory sections, have been attached to the Appendix C.

There are a total of 50 rooms (not counting toilets), distributed as follows: 2 rooms in the basement, 14 rooms in each of the main floors, and 6 rooms in the attic. In the basement the are the mechanical and storage rooms. In the ground level there is the entrance with a reception, a kitchen, the server room, some storage spaces, and an office space. The first and second floors host office spaces and meeting rooms, as well as a canteen each. Finally, the attic has also office spaces and meeting rooms, as well as a lounge area, and connects to the rooftop terrace.

The ground, first, second floors, and the attic are all connected by an atrium—the main stairs open space, which is topped by an operable skylight.

3.1.1 Volumetry

The built-up area is $50\text{m} \times 20\text{m} = 1000\text{m}^2$, and heights are: 4m for the ground floor, and 3.5m for the rest of floors and the basement. The total exterior wall surface is 1800m^2, roof and terrace surfaces sum up to 1000m^2 and the total enclosed volume is 11200m^3. Therefore, the aspect ratios are: 2.5 length-to-width, 0.29 or 0.22 height-to-length (considering the attic or not), 0.25m^{-1} surface-to-volume.

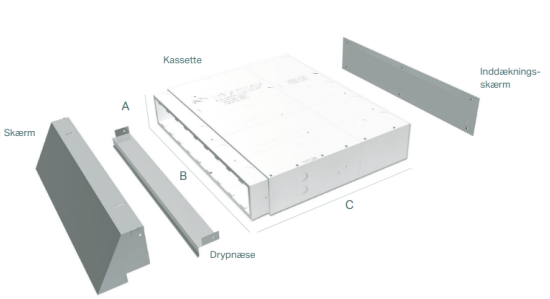
Regarding the floor areas, the total sum is 3950m^2: the basement is 135m^2, the ground floor encloses 700m^2, leaving 300m^2 for parking, the first and second floors take up the whole built-up area, and the attic occupies 400m^2, while the accessible rooftop terrace spans 200m^2 and the area reserved for solar panels is 215m^2.

3.1.2 Envelope

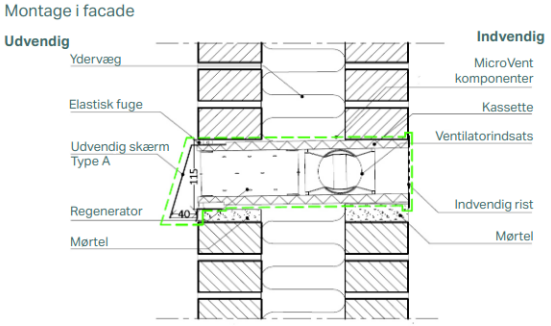
The exterior walls are light, wooden, and are filled with mineral wool insulation. All the floor decks are wooden too, except the ground floor, which is a radiant heated concrete slab.

Windows span about 450m^2, and they are operable like the skylight. Hence, the building will be naturally ventilated by the combined effect of thermal gradients and the wind pressure [71].

There is solar shading: windows have a wooden solar protection, which also serves as a support for additional solar panels. The construction detail of this element can be seen in detail in Drawing C.6.



(a) Assembly of a ventilation fan unit.



(b) Mounting detail view with the parts.

Figure 3.1: Ventilation fan units. Source: InVentilate product information brochure [72].

3.2 HVAC

The main elements of the building's HVAC system are presented in the following.

3.2.1 Heating

Other than the radiant floor in the ground floor, the rest of floors are heated using hot-water convectors distributed under the windows.

3.2.2 Ventilation

Ventilation is decentralized, per room. The construction detail view in C.6 shows how the solar shading devices are open at the bottom and at the back, so that ventilation fans can be conveniently placed above the window, hidden from plain sight. These are variable air volume and include a recuperative heat exchanger. See their assembly and a mounting detail view in Fig. 3.1.

The recuperative heat exchanger included in the fans is passive: it is a piece of hollowed plastic with many fins ("regenerator" in Fig. 3.1b), that absorbs moisture and retains heat from the flowing humid, hot air, and returns it to the cooler, dryer air. The sensible heat recovery efficiency is 85%, while the latent is about 75%.

3.2.3 Cooling

Given the cold climate conditions of Aarhus, Denmark (from Weather Data in [15]) the only cooling available will be through air changes—also known as free cooling. The air is sufficiently cold throughout the year so as to have a single hot week during the summer.

The ventilation fan units include several fans each that blow air in any desired direction—either boost or exhaust. By setting all the fans to blow in the same direction, the heat exchanger is effectively bypassed after a steady-state is reached: no heat or moisture are exchanged with the flowing air.

Chapter 4

Design and Implementation

In this Chapter the design and the implementation steps of the proposal will be explained, basing on the case of study from Chapter 3, and using the concepts presented in Chapter 2. It is presented as three stages for clarity, although they are interrelated and the development process has involved iterations between them. As mentioned in Section 1.3, the following design will be scoped to a building energy simulation, always considering the real implementation of this proposal in the building under study.

The three stages are:

1. Preparing the simulation.
2. Defining the integration with the controller.
3. Developing the controller itself.

The integration with the controller will also define the variables to be communicated between the controller and the simulator. More specifically, what variables the controller receives as observations, its rewards, and which actions it can take in return. These will be explained separately, after the stages shown above, for the sake of readability of this Chapter.

4.1 Preparing the simulation

To begin with, two levels of abstraction have been proposed, aimed at simplifying the architectural solution from Ch. 3. This is done because the methodology presented here is generic, but the building energy model is not. Therefore, two different simulations are proposed.

The first simulation, smaller, is devised to represent conditions that can be found in the building across different rooms throughout the day, while trying to keep it as simple as possible. The interactions between the thermal zones and the atrium are of especial interest, considering the natural ventilation involved. This simulation will be used for developing the controller, and will be the main source of results.

The second simulation is bigger, trying to capture interactions between different floors. However, it is still a simplification, as neither the basement nor the attic are considered, and the ground floor is a copy of the first and second floors—these are considered to be the most "representative" of the building.

The different controller versions will be tested on this second simulation to verify their wider applicability. Simulations will be called *dev* and *test*, respectively, from now on.

Among the different simulators available, Energyplus has been chosen, along with DesignBuilder as a preprocessor (see Section 2.3 for an introduction). DesignBuilder version 6 was used due to its availability, and Energyplus version 8.9 was selected because of compatibility requirements with the DesignBuilder version.

4.1.1 Building definitions

The first step in the building design is to define its geometry, its envelope properties, the schedules that govern the building behavior, the construction materials and the simulation time step size: how often simulation values are calculated.

Geometry

For the *dev* building, nine different zones have been created: one for the atrium, modelled as a central two-story room, and eight for sorrounding rooms, either adjacent—NWSE rooms—or with indirect communication to it, through another room—the corner rooms. Fig. 4.1 shows an overlay of the top view of the geometry with the floor plan of the representative floor, while Fig. 4.2 shows the top, front and lateral views.

The *test* building, on the other hand, has 17 zones per story. One of them is represented in Fig. 4.3a to show the zones' partition. The 3D view of the full building is shown in Fig. 4.3b. The atrium is modelled as a single zone spanning from the ground to the skylight, with open connections to the adjacent rooms in each floor.

Envelope and schedule

In both simulations, the windows are operable with the exception of the top window of the skylight, due to limitations in Energyplus to simulate large horizontal openings (see Sub. 2.3.3). The operation schedules have been defined to be the same as the occupancy schedules, with a lower limit on the zone operative temperatures of 22 °C. This is to avoid having them open during cold periods in which they would naturally be closed.

Construction materials

The construction materials have been selected trying to match the ones from the current architectural solution. Nonetheless, because it is in an early stage of development, they will probably change in the final execution. This is why they have been chosen among the options DesignBuilder provides.

In particular, the "best practices" within lightweight buildings has been chosen, which already defines different layers of materials for the walls, roofs and floors. Only the interior floor material has been changed to be wooden: "25mm chipboard flooring wooden-joist internal floor - industry grade" has been selected. Air tightness has been defined as "excellent", with 0.05 Air Changes per Hour (ACH) of infiltration.

Simulation time step size

Lastly, the time step size of the simulation has been chosen as 10 minutes. For the simulation of heating only it could have been increased to a longer period, as an hour, but the timescales for natural and forced ventilation are shorter, so it has been limited as a compromise.

4.1.2 Building HVAC

The next step is to define the HVAC system: as described in the case of study (Chapter 3) it should consist of decentralized ventilation outdoor Air Handling Units (AHU), and hot-water convectors in each zone. The way this has been modelled is described as follows:

- The atrium is not heated nor mechanically ventilated, so it is excluded from the HVAC modelling.
- One zone group has been created for each other zone. This is due to DesignBuilder expectations that there will be a centralized AHU serving multiple zones, but it is also possible (even if more difficult) to define a decentralized system.
- Each zone is connected to its own AHU. After the first one is created, multiple copies can be created and then they only need to be connected to their zones.

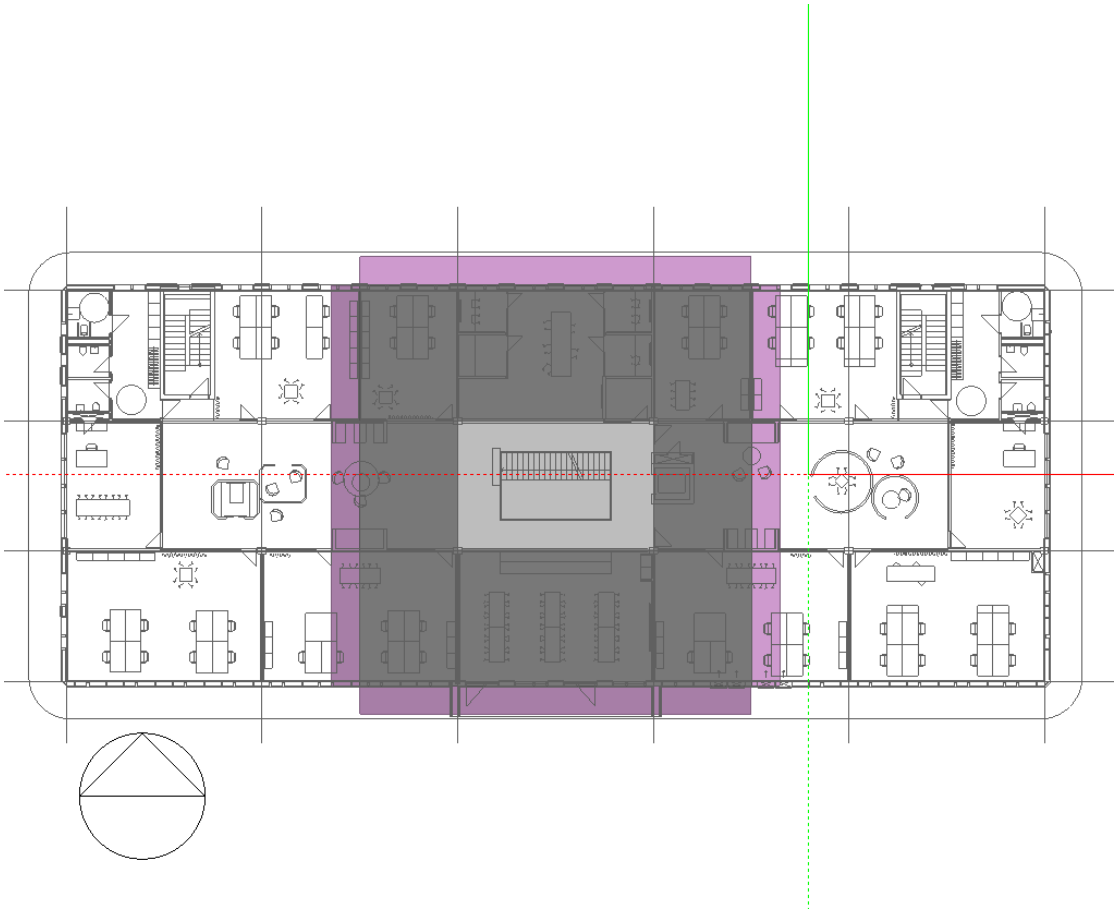
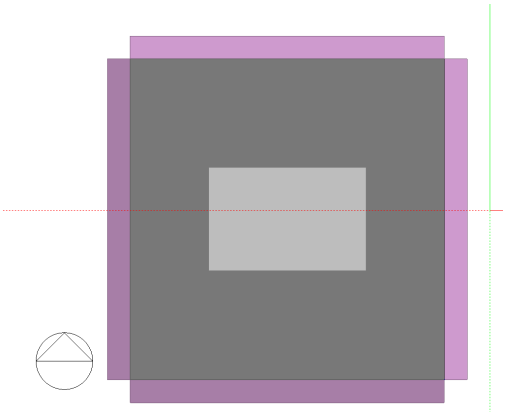
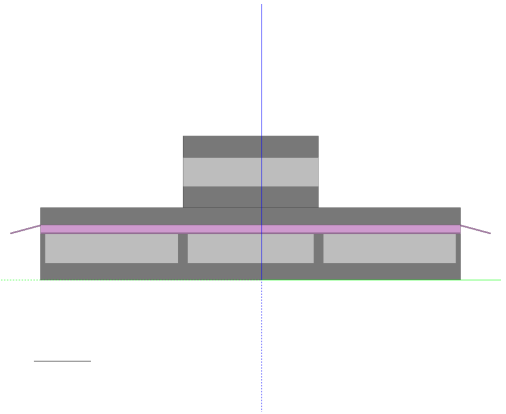


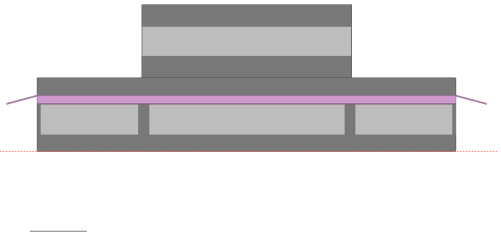
Figure 4.1: Architectural floor plan superimposed on the top view from the *dev* building. The blocks in pink are the solar shades, the block in light gray is the atrium and the eight blocks in dark gray sorrounding it are rooms. The disk in the lower left corner is pointing to the North. Source: Screenshot from DesignBuilder.



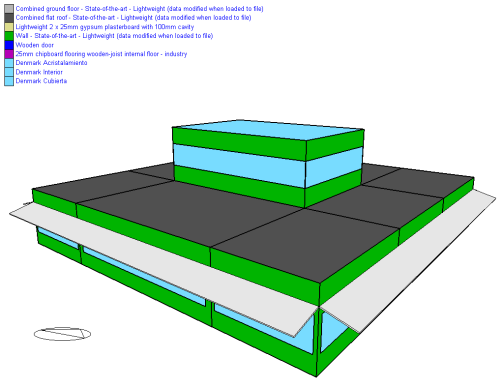
(a) Top view. The disk in the left points to North.



(b) Lateral view

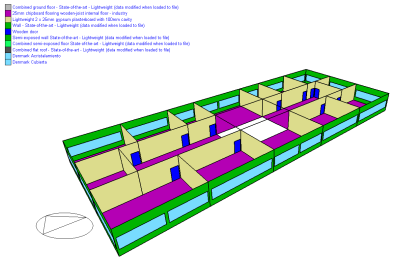


(c) Front view

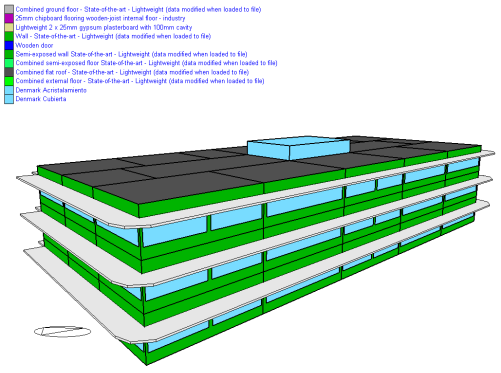


(d) 3D view

Figure 4.2: Views of the *dev* building. In the top, lateral and front views the glazings are shown in light gray, and the walls and roof in darker gray. The pink blocks are the solar shades. In the 3D view, it is easier to identify each element. Source: Screenshots from DesignBuilder.

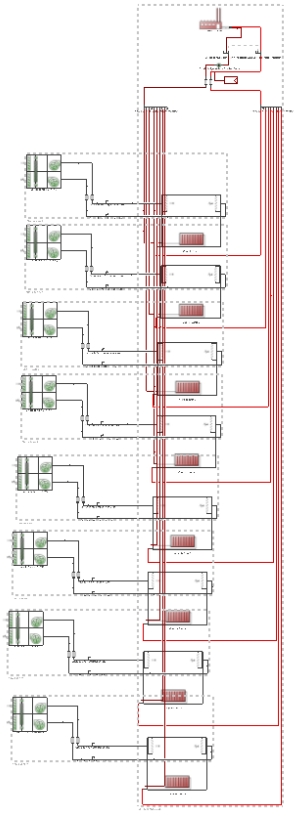


(a) Floor 3D view with partitions. The solar shadings have been removed for clarity.

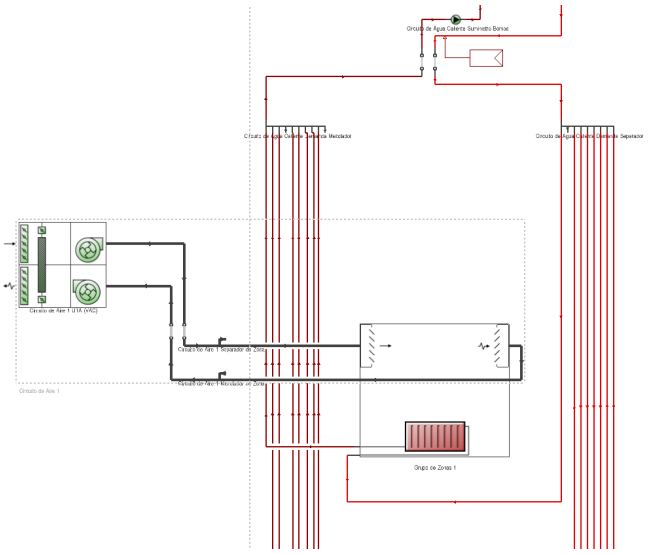


(b) 3D view

Figure 4.3: Views from the *test* building. Source: Screenshots from DesignBuilder.



(a) HVAC system in the *dev* building



(b) Detailed HVAC zone with its own AHU

Figure 4.4: HVAC system in the *dev* building, along with a closer view of a thermal zone. Source: Screenshot from DesignBuilder.

- Each zone has a hot-water convector, and all of them are connected to a single district heating. Using a district heating abstracts away how the heat is generated, and still serves for the purpose of the simulation.

The AHUs have been defined with 100% outdoor air, i.e. without recirculation. The heat exchanger has been defined as plate-based as the most suitable model available. The heat recovery availability is defined as a schedule: always available, although this value will be overwritten by the controller using an actuator. Fig. 4.4 shows a sketch of the HVAC system for the *dev* building. The *test* building has a similar setup, but with many more zones.

The sizing of the ventilation airflow and the hot-water convector units is left to Energyplus, using the "autosize" option. It will follow the ventilation requirement specified in the *Activity* tab, as well as the setpoint temperature specified in the *HVAC* tab in DesignBuilder. The ventilation requirement has been set to 2 ACH per zone, given that in Denmark there is not any regulation fixing it, and considering that the building will be, in addition, naturally ventilated for the most part of the year.

4.2 Defining the integration with the controller

After fully defining the simulation, it needs to be connected to the controller. This Section explains the whole pipeline created to obtain an interface which the controller can communicate with.

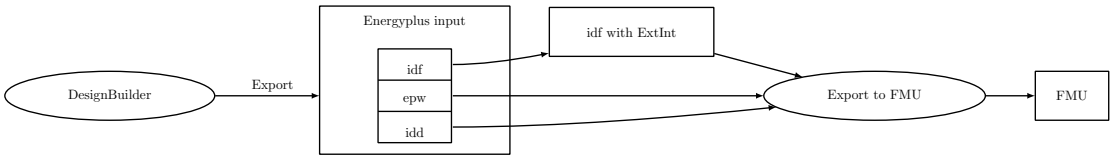


Figure 4.5: Transformation pipeline to obtain a FMU from the simulation definition. Source: own elaboration using *graphviz* [39].

4.2.1 Packaging the simulation

Here the pipeline to package the simulation into a FMU is explained. The transformation process is shown schematically in Fig. 4.5.

First, the input dictionary, the building definition and the weather file are exported from DesignBuilder (".idd", ".idf" and ".epw" files, respectively). Then, the ".idf" file is transformed using a custom Python script, in such way that the necessary *ExternalInterface* objects are added: this includes all the sensors and actuators—variables—, that have been divided into different levels:

1. Environment. Variables here are defined once, and they refer to the environment (e.g. wind speed, is it raining?).
2. Zone. Variables here are defined per zone, as each zone will have its own (e.g. operative temperature).
3. Surface. In case any variable is required at the surface level (each surface will have its own).

This setup allows to quickly change the variables, because these levels are defined as template files. It is only a matter of going to the proper template and adding or removing the desired variable, then running the pipeline again.

Finally, the original ".epw" and ".idd" files, along with the extended ".idf" file (resulting from the script) are fed into the tool that exports to FMU. This tool was originally written by [45], and it has been extended as part of this master thesis' contribution to support resetting the simulation—useful for the training of the controller as it will be shown in Sec. 4.3.

4.2.2 Defining the environment

Once the simulation has been packaged, the environment that runs it and connects it to the controller needs to be prepared. Here, the environment from the RL setup, previously presented in Fig. 2.9, is further expanded into what is shown in Fig. 4.6.

The environment is defined as a python class which follows the interface defined by OpenAI Gym: *reset()* resets the environment and returns an initial observation, *step(action)* applies a given action and returns a reward and a new observation, along with information to tell if the simulation—*episode*—is done yet.

When initializing the environment, the FMU file is provided, along with the original ".idf", ".epw" files, and an extra file containing the shading information for the building during a full year. This extra file can be generated using Energyplus, by enabling the corresponding output in the ".idf" file first. Also, the file containing the sizing of elements—those set to "autosize"—is fed into the environment initialization. This allows to preload necessary information during the simulation, as well as to look-ahead and provide predictions to the controller.

Then, on each call to *reset()*, an initial date is picked at random and a simulation episode begins for 4 weeks—to avoid seasonality effects in a given simulation, but to cover the full year with different simulations. Following that, normal interactions take place between the controller and the environment, until a terminal state is reached. The environment will, if desired, continue with further episodes by calling the *reset()* method to begin again.

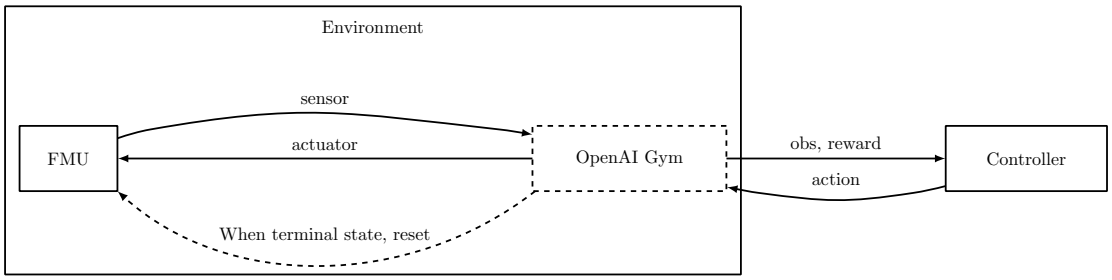


Figure 4.6: Interaction between the simulation, packaged as a FMU, and the controller, through the OpenAI Gym interface. Source: own elaboration using *graphviz* [39].

It is noteworthy to mention that the FMU can be replaced with the real building, and the environment communication with the controller will still be the same. Only the real sensor readings and actuator actions will need to be implemented. This is possible thanks to the common interface the environment implements.

4.3 Developing the controller

In this Section, the controller side of the interaction from the RL setup will be detailed, expanding the boxes seen in Figs. 2.9 and 4.6. The first thing to explain is that there will be one controller for each thermal zone. This means that the setup of the environment will be multi-agent.

The outline of this Section is as follows: first, a baseline controller is presented in Subsec. 4.3.1, and then Subsec. 4.3.2 presents the developed PPO controller.

4.3.1 Baseline controller

In this Subsection a baseline controller is defined, following the best practices given by experience, which are available in DesignBuilder and Energyplus. In this case no external actuators are needed, as everything can be scheduled or predefined beforehand. This will be a useful reference for comparison.

The heating system defines two different setpoint temperatures: a main one of 20 °C, and a setback of 13 °C, for non-occupancy periods. Given that the heating is done using hot-water convectors, the main setpoint is activated 4 hours before the arrival of the first occupants in the morning, to give enough time for the heating to reach the desired setpoint.

Regarding the ventilation, the baseline controller will keep a constant value of 2 ACH, as explained in Subsec. 4.1.2, while the heat exchanger will be bypassed to allow free-cooling according to the pattern in Fig. 4.7. It is an enthalpic free-cooling setup with dry-bulb boundaries, to avoid too cold or too hot conditions.

4.3.2 PPO controller

The developed controller implements the PPO algorithm. Since the environment is multi-agent, and the controller is crafted under the RL paradigm, the resulting framework is often called Multi-agent Reinforcement Learning (MARL).

As seen in the theory from Sec. 2.4, distributed agents generally need to coordinate and share information among them, and also a *mapping* function needs to be provided for each policy. One way to accomplish the coordination is to use a single policy for all the agents, that will share all the parameters θ. This policy will then be trained with observations from all the different agents, so for each timestep the amount of training data is increased as well. Therefore, the *mapping* function will map all the controllers to the same policy.

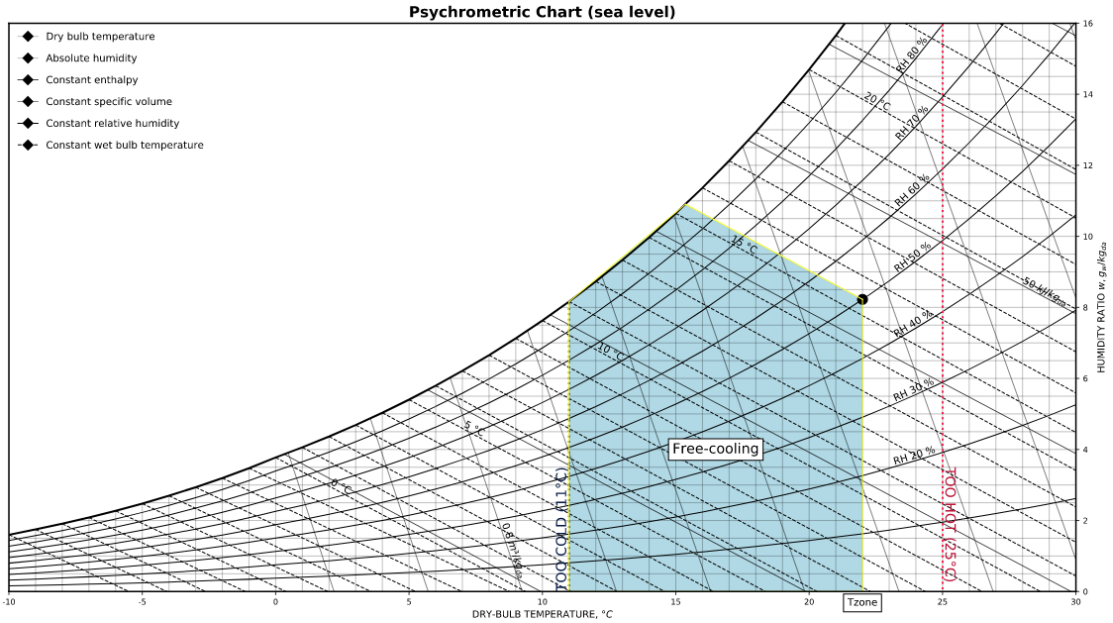


Figure 4.7: Baseline controller heat exchanger bypasss (free-cooling) region. There are dry-bulb limits of 11 and 25 °C, and a constant enthalpy limit that is set equal to the indoor enthalpy value. Source: own elaboration using *psychrochart* [14].

Figure 4.8: PPO architecture. Source: RLlib [73].

Architecture

The implementation of the controller has been done using the RLlib library in Python language [67]. Among other features, it provides implementations for multiple RL algorithms, can be connected to any environment that follows the OpenAI Gym interface—with support for multi-agent environments—, allows to run environments in parallel with different *rollout workers* that produce trajectories—collected and concatenated by the trainer process (Fig. 4.8)—, and can use hardware acceleration from a GPU. Unless otherwise stated, the default configurations have been used.

After some initial exploration, two equal neural networks have been chosen to parametrize the policy and the value function estimator, with hidden 3 layers of 512, 512 and 256 units, respectively, with ReLU non-linearity functions. The only difference is that the output layer of the policy contains the same number of units as actions available (seen in Sec. 4.5), whereas the value function estimator has a single unit as output.

The discount factor γ has been set to 0.99, and the episodes are 4 weeks long, as already mentioned. These are divided in batches of training of 3 rollout workers $\times$ 24 hours each. The goal behind this decision is to speedup the training process, but also avoid poor generalization results.

The computer setup used for developing and training the controller is an Intel Core i5-10600 CPU (6 core with Hyper-Threading @ 3.3-4.8GHz) with Nvidia 2080Ti 11GB GPU, 64GB@3000MHz RAM, 500GB TLC NVMe SSD, and 2TB SATA HDD.

4.4 Defining observations

In this Section the variables observed by the controller will be described, as well as how to *shape* and *encode* them in order to ease the learning in the controller neural network parametrization (see Subsec. 2.4.5 for an explanation).

Because each zone will have its own controller, the variables it sees should summarize their state and history, in order for the Markovian property to hold (refer to Subsec. 2.4.1). In addition, it will be helpful to provide predictions for future observations, as this will shorten the delay between observations and rewards that is inherent to a RL framework.

The structure of this Section is as follows: Subsections 4.4.1–4.4.8 describe the basic variables that conform an observation. Then, Subsection 4.4.9 builds on top to add the historic summary from past timesteps, and Subsection 4.4.10 presents the predictive module that adds information about future observations.

4.4.1 Date and time

As the controller needs to work well under different conditions every day throughout the full year, it is important to provide it with the notion of current date and time, so that it can take better—informed—decisions.

The *encoding* of these variables needs to bound them ideally between -1 and 1, while respecting their periodicity: the conditions at 23:59 from one day will be identical to the conditions one minute after, at 00:00 from the next day; likewise the conditions during the end of December will be similar to those in early January from the next year. It then follows that the sine and cosine trigonometric functions provide a natural encoding: $\langle sin(t_d), cos(t_d), sin(t_y), cos(t_y)\rangle$, where t_d is the fractional time-of-day in radians, and t_y is the fractional time-of-year, in radians too.

4.4.2 Outdoor air dry-bulb temperature and enthalpy

Knowing the outdoor air conditions is also a requirement for the controller, and the outdoor air dry-bulb temperature and enthalpy determine these conditions. These two have been chosen because they are directly related to the energy contained in the air—they are used to detect free cooling availability in the baseline controller.

Regarding the *shaping*, these two variables are continuous and follow a normal distribution each. These distributions can be extracted from the weather data in ASHRAE [15], as the dry-bulb temperatures and humidity ratios ($\mathrm{kg}_w/\mathrm{kg}_{air}$) are provided for the percentiles 99.6 and 99%, while the dry-bulb and wet-bulb temperatures are provided for the percentiles 0.4 and 1%.

Hence, the extreme values can be averaged to find the mean of the distribution, in this case because there is information from 1–99 and 0.4–99.6 percentiles, two different means can be found, and the final mean of the distribution can be approximated as the average of those two values.

Likewise, the standard deviation can also by found with Eqs. (4.1)–(4.3), where the x represents either dry-bulb temperatures or enthalpies available for the percentiles specified in the subindex. The graphical explanation can be found in Fig. 4.9).

$$\sigma_{0.98} = (x_{0.99} - x_{0.01})/(2 \cdot 2.05) \tag{4.1}$$

$$\sigma_{0.992} = (x_{0.996} - x_{0.004})/(2 \cdot 2.41) \tag{4.2}$$

$$\sigma = (\sigma_{0.98} + \sigma_{0.992})/2 \tag{4.3}$$

Once the normal distributions are found, any temperature or enthalpy can be transformed at anytime by using the corresponding normal distribution. As a final note: the outdoor air enthalpy observation has been dropped from the final controller, while it has been included as a future prediction. It will be explained in Sec. 4.4.10.

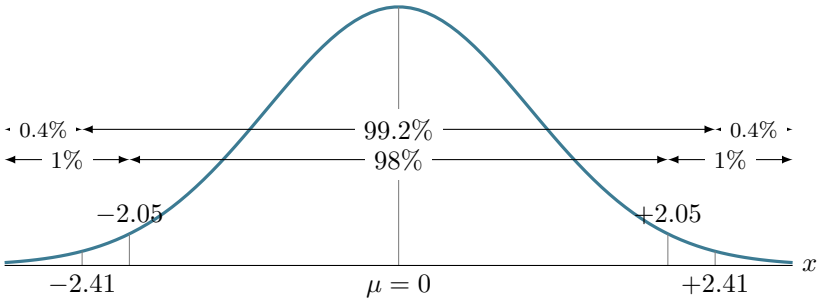


Figure 4.9: Standard distribution. The tails at 0.4–99.6 percentiles leave 99.2% centered probability, while the tails at 1–99 percentiles leave 98% centered probability. Source: own elaboration using LaTeX's *Tikz* package [74].

4.4.3 Wind speed and direction

In order to account for a possible increase in natural ventilation, each controller will also observe the wind speed strength and its direction. It has already been shown how to encode a variable with natural periodicity, and the direction is another example: it can be encoded as the sine-cosine pair of the angle between the wind direction and North.

On the other hand, the wind speed can be approximated as an Weibull distribution [75], whose tail is also given in the weather data from [15]. Concretely, the percentiles 1, 2.5 and 5% are given. Then, the Weibull parameters can be extracted by creating a system of equations based on Eq. (4.4) and solving by least squares, as it is overdetermined—3 equations, one for each percentile, and just 2 variables, a and b.

Then, Eqs. (4.5), (4.6) can be used to find c and k, and the cumulative distribution function is finally given by Eq. (4.7). The latter can be used to transform any wind speed to a number between 0 and 1, representing the probability of finding a wind speed that is less than or equal to the one observed. Values close to 1 mean the wind is very strong, while values close to 0 mean very weak.

$$\ln(-\ln(1 - P(V \leq V_x))) = a + b\ln(V_x) \tag{4.4}$$

$$c = e^{-a/b} \tag{4.5}$$

$$k = b \tag{4.6}$$

$$P(V \leq V_x) = 1 - e^{-(V_x/c)^k} \tag{4.7}$$

4.4.4 Rain and daytime

The variables here are boolean: either yes or no, and are indicators for rain and for daytime. They can be interesting for the controller because they imply a change in outdoor air conditions and radiation. As seen in Subsec. 2.4.7, these can be *one-hot* encoded.

4.4.5 Indoor air conditions

This includes the indoor operative temperature, the humidity, the indoor air enthalpy, and the CO_2 level. These variables, on the other hand, would be more useful if they were related to outdoor conditions and to ideal conditions, given that only the differences—or *deltas*—matter. That is why they are transformed according to the following, in order to get variables that are mainly distributed between -1 and 1:

- *OutdoorZoneDeltaT* and *ZoneIdealDeltaT*. The zone operative temperature (°C) is subtracted both from the outdoor temperature (°C) and the ideal temperature (°C)—according to the adaptive comfort model EN15251 presented in Subsec. 2.2.1, modified by setting a lower limit of 22 °C for days that are too cold—and divided by 10.

- $OutdoorZoneDeltaEnthalpy = (OutdoorAirEnthalpy - ZoneAirEnthalpy(kJ/kg))/20$.
- *ZoneRelHum*. The zone humidity is presented as a relative humidity, because this is already a number between 0 and 1.
- $ZoneDeltaCO_2 = (ZoneCO_2 - OutdoorCO_2(ppm))/1000$
- Boolean variables indicating whether the zone operative temperature is within 80 and 90% of comfort, according to the previous adaptive comfort model.

In addition, it is relevant to know the zone occupancy, as the controller may find beneficial to reduce the heating or ventilation when a room is not occupied. In this work, the variable *ZoneOccupied* is defined as a boolean value indicating when a room occupancy level is greater than 10%, to avoid taking into account people crossing only. Because it is boolean, it can be *one-hot* encoded as well.

4.4.6 Energy consumption

Not only sensorial information is needed in the controller, but also the following metered information: the energy consumed to heat and ventilate during the last timestep.

Here the normalization of the variables should preserve information about the size of each zone, because a bigger room will need more heating, and also how big a consumption is relative to the design value—the maximum set by the *Sizing* step in Energyplus. With that in mind, these variables are prepared:

- *AreaToTotal*. This is the fractional floor area of a zone w.r.t. the total of all the zones.
- *FanToMax*. The fraction of the maximum electric power from the zone fan.
- *HeatingToMax*. The fraction of the total thermal energy the district heating can provide that has been consumed in the zone.
- *FanToTotalConsumption*. The fraction of the zonal fan consumption over the total consumption from all the fans. This variable is a helper only, and not included in the observation presented to the controller.
- *HeatingToTotalConsumption*. Similar to the previous one, it is also a helper.
- $FanToAreaDelta = FanToTotalConsumption - AreaToTotal$ if $FanToMax > 0$, otherwise $FanToAreaDelta = 0$. This provides a guidance on how much deviation there is between the zone fan consumption w.r.t. the fraction of floor that zone represents.
- $HeatingToAreaDelta$. Calculated in a similar fashion, but with the *HeatingToTotalConsumption* instead.

4.4.7 Radiation

If each perimeter thermal zone is considered in isolation, one of the main sources of heat exchange is the solar radiation, as seen in Subsec. 2.1. This Subsection presents a simplified approach to the calculation of the observation that the controller will receive, using information available in the real building.

The information needed to begin the calculation is:

- The direct and diffuse solar irradiances (W/m^2), which can be obtained from weather forecasts.
- The solar elevation and azimuth angles (Fig. 4.10), which can be obtained from a full-year simulation—they depend only on the location of the building.
- The shading information for external surfaces throughout a year for each timestep, also attainable from a full-year simulation with the detailed *test* building.
- The external surfaces' area and orientation, which can be calculated from the geometry.

- The external surfaces' glazing information, specifically the fraction of the surface that is glazed, and the g-value of the glazing (described in App. A.4).

For the sake of the simplification of the complex heat transfer occurring on the walls due to its non-stationarity nature, it has been considered that radiation will have immediate effects on the zone if it lands on a window, and delayed effects otherwise. This is because of the thermal inertia: the incident radiation will imply an increase on the external surfaces temperatures that will later propagate the heat to the interior of the building.

So, the contributions from direct and diffuse radiation can be calculated separately for each surface of a given zone, and then added up into these two different terms: *immediate* and *inertial* zone radiations. Direct and diffuse radiation calculations are detailed in Alg. 3, where $\odot$ denotes the element-wise or Hadamard product.

Algorithm 3 Radiation calculation for a set of external surfaces $\mathcal{S}$.

Set the diffuse radiation factor $f_{dif} = 0.33$
For the timestep t,
Prepare the solar angles ϕ and β, direct (normal) irradiance i_n and diffuse irradiance i_{dif} ($\mathrm{W/m^2}$).
Prepare the following vectors $\in \mathbb{R}^{|\mathcal{S}|}$:
areas $\mathbf{a}$, window fractions $\mathbf{w_f}$, sunlit fractions $\mathbf{s_f}$, and g-values $\mathbf{g_v}$.
Prepare the matrix of surfaces' direction vectors $\mathbf{D_v} \in \mathbb{R}^{|\mathcal{S}|\times 3}$.
Solar direction vector $\mathbf{s_{dv}} = \langle \sin\phi, \cos\phi, \cos\beta \rangle / \| \langle \sin\phi, \cos\phi, \cos\beta \rangle \|$

$\mathbf{c}_\alpha = \mathbf{D_v} \cdot \mathbf{s_{dv}}$
Direct (normal) radiation for all the surfaces $\mathbf{r}_{\mathbf{n},\mathcal{S}} = i_n \odot \mathbf{c}_\alpha \odot \mathbf{s_f} \odot \mathbf{a}$
Diffuse radiation for all the surfaces $\mathbf{r}_{\mathbf{dif},\mathcal{S}} = i_{dif} \cdot f_{dif} \odot \mathbf{a}$
Total radiation for all the surfaces $\mathbf{r}_\mathcal{S} = \mathbf{r}_{\mathbf{n},\mathcal{S}} + \mathbf{r}_{\mathbf{dif},\mathcal{S}}$

The diffuse factor f_{dif} is set to 0.33 because from the total hemispherical diffuse radiation, external walls receive approximately 1/2, and from the incident radiation the shading elements may remove about 1/3. In total, $f_{dif} = 1/2 \cdot 2/3 = 1/3$.

The decomposition of the total radiation into the *immediate* and *inertial* terms is graphically represented in Fig. 4.11, where the transmittance-to-gain factor t_f signifies the fraction of solar radiation gain that is directly transmitted through the glazing, whereas 1 - t_f is absorbed and then re-emitted. The factor t_f is assumed to be 0.9—double glazing without solar control. The solar gain from the wall has been probably overestimated by assuming there will not be any reflection, although the external surfaces from the real building will be painted in dark colors, so the overshoot should be small and has not been considered to be an issue.

Normalization

Rather than presenting the radiations in W, it is better to normalize them. Here the normalization has been performed by dividing each radiation in W by the maximum irradiance possible given the location of the building—approx. $900\mathrm{W/m^2}$—, and by the average area of the external surfaces. This still carries information about surface sizes, but makes the number smaller, between 0 and 1.

4.4.8 Conduction gains

Having a controller per zone makes the setup easily extendible to any number of zones. However, it also means that each observation should include sufficient information for each zone, in particular, maybe also information that is interesting about other zones, like their operative temperatures. One problem that arises is how to deal with a variable number of related zones, e.g. a corner room has two adjacent rooms but an internal room has four.

To solve the previous problem, a distinct approach is proposed: to summarize information from the nearby zones into a single number that carries all the information. This is possible when

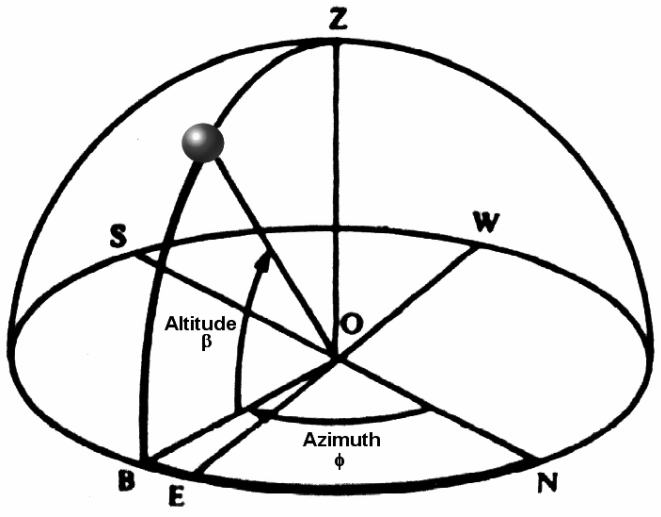


Figure 4.10: Azimuth (ϕ) and altitude (β) solar angles, with North as the reference ($\phi = 0$). Source: Energyplus Engineering Reference [76].

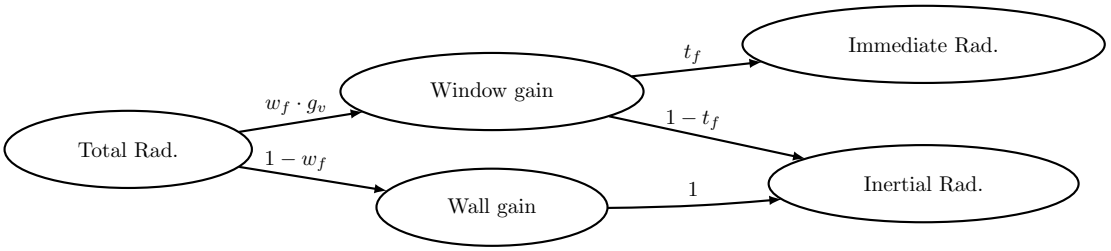


Figure 4.11: Graphical representation of solar radiation distribution into the *immediate* and *inertial* terms on a given surface. Source: own elaboration using *graphviz* [39].

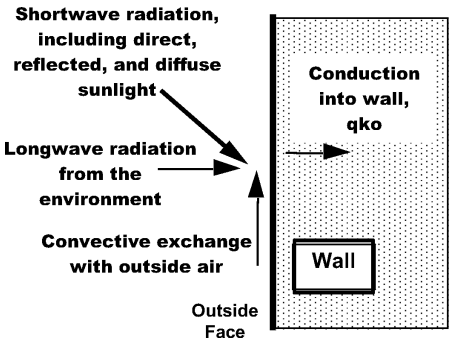


Figure 4.12: Heat balance on an external surface. The radiation, convection, and conduction transfer methods act combined. Source: Energyplus Engineering Reference [77].

dealing with the gains—or losses—by conduction, because all the controller needs is the net heat balance from its zone's perspective.

The heat balance calculation for a given zone is detailed in Alg. 4. Again, a simplification has been done: a variant of the equations from App. A.4 has been used, taking the heat balance only from the external surfaces inwards. This decouples the radiation and convection heat exchanges on the external surfaces from the wall conduction (Fig. 4.12), in spite of needing the external surface temperatures. Yet it is a viable trade-off, because this temperature can be measured per facade on the real building, and the simulation provides any desired surface temperatures as well.

Algorithm 4 Conduction gains calculation for a given zone with delimiting surfaces set $\mathcal{S}$.

Prepare the zone operative temperature T_{op}.
Initialize heat gain $h_g \leftarrow 0$
for $s \in \mathcal{S}$ **do**
  Prepare the following values for surface s:
    area a, U-factors for the wall (u_w) and the glazing (u_g), and the window fraction w_f.
  if s is adiabatic **then**
    Skip this surface, no heat exchanged.
  else if s is the ground **then**
    $T_{other} = T_{ground}$
  else if s is an external surface **then**
    $T_{other} = T_{sf}$
  else if s is adjacent to another zone z_{other} **then**
    $T_{other} = T_{op,z_{other}}$
  end if
  Global factor $U = u_g \cdot w_f + u_w \cdot (1 - w_f)$
  Update $h_g \leftarrow h_g + U \cdot a \cdot (T_{other} - T_{op})$
end for
return h_g

Normalization

In order to normalize the net heat balance (W), a corner analysis approach has been performed. A full reference year simulation running the baseline controller (Subsec. 4.3.1) provided the most extreme external surfaces' temperatures: $-12.7\,^{\circ}$C during the coldest winter day, and $54.7\,^{\circ}$C during the hottest summer day.

Then, considering corresponding zone operative temperatures of 20 and $26\,^{\circ}$C, the following ranges could be obtained: $\Delta T_- = -12.7 - 20 = -32.7$ and $\Delta T_+ = 54.7 - 26 = 28.7$.

Finally, the scaling is done by dividing the net heat balance (W) by the average surface area from the full building, and by ΔT_- or ΔT_+, depending on the sign of the heat balance.

4.4.9 History

In this Subsection the historic summary that is added to the observations is presented. Note that in order to calculate the summary, a record of the historic observations needs to be kept for each timestep. The final implementation includes a summary for the last hour, the last 30 minutes, and the last timestep—last 10 minutes—, to provide different timescales to the controller so that it can learn to map from historic states and actions to the current state.

The variables included in each summary have been described in the previous Subsections (4.4.5–4.4.6). This is the list for a summary from t_{past}:

- *ZoneIdealDeltaT*. This is the *delta* observed at timestep t_{past}.
- *ZoneRelHum*. An average of the observations between t_{past} and $t-1$. When $t_{past} = t-1$, this is simply the observation at the last timestep.
- *ZoneDeltaCO*$_2$. Idem.
- *ZoneOccupied*. Idem.
- *FanToMax*. Idem.
- *FanToAreaDelta*. Idem.
- *HeatingToMax*. Idem.
- *HeatingToAreaDelta*. Idem.

4.4.10 Predictions

Here the predictive module is presented. This is specific for the simulation only, because in a real setup it would be replaced with weather forecasts information. Like it has been done for the history, predictions are provided at different timescales: the next hour and the next 4 hours. This is the list of variables, first appearing in Subsections 4.4.2 and 4.4.7, included in a prediction for timestep t_{future}:

- Outdoor air dry-bulb temperature, *OutdoorT*.
- Outdoor air enthalpy, *OutdoorEnthalpy*.
- *Immediate* radiation.
- *Delayed* radiation.

One aspect to consider is that in a weather forecast the predicted values do not necessarily match the future observed values. This is why looking ahead in the weather file and simply providing the future observations would be to "trick", unrealistic, and misleading to the controller, that should learn instead the inherent uncertainty around them.

To overcome this issue, gaussian noise has been introduced to the predictions, controlling the accuracy of the predictions up to the desired level. It has been considered that 90% of the outdoor temperature predicted values fall within $\pm 1\,^{\circ}\text{C}$ from the real value, 90% of the relative humidities fall within $\pm 2\%$ from the real value, and 90% of the solar irradiances fall within $\pm 20\text{W/m}^2$.

4.5 Defining actions

Defining the actions involves deciding what variables will be actuated, and defining the range or set of values they can take. This is also known as choosing the **action space**. In this thesis, it has been stated that the heating and the ventilation will be controlled, so the actuators should be able to modify them. In addition, recalling the importance of *shaping* the actions (Subsec. 2.4.7), the actions should have a range wide enough to allow a correct control, without being overwhelming for the controller to explore.

Therefore, three different actuators are created for each controller: one for controlling the zone operative setpoint temperature, another for controlling the usage of the heat exchanger, and the last to control the level of airflow in the mechanical ventilation system.

4.5.1 Operative temperature setpoint

The operative temperature setpoint has been discretized into 6 different values: 7, 18, 20, 22, 24, and 26 °C. As it has inertia, the controller can learn to obtain any intermediate temperature by varying the setpoint correspondingly at each timestep. Also, the 7 °C option is provided to allow for a setback temperature like the baseline controller has. The discretization is done as a *one-hot* encoding.

4.5.2 Heat exchanger usage

Using or not the heat exchanger is a boolean value, so it is one-hot encoded as a discrete variable with 2 values. To encourage collaboration between the distributed agents, all of them will do what the majority voted, i.e. at each timestep the actions from each agent are aggregated and the final decision about using or not the heat exchanger is taken to be the most voted action.

4.5.3 Fractional mass airflow

The fractional mass airflow has been discretized into 4 different values: 0, 33, 66, and 100% of maximum airflow. It does not have inertia, but during an hour the actuator can be changed up to 6 times—the timestep size is 10 min—, so different levels of ACH can be achieved as well. Discretization has been performed with *one-hot* encoding as well.

4.6 Defining rewards

The last but equally important step for completing the RL setup is to define the rewards received by the controller at each timestep.

The first thing to notice is that there is not a unique optimal, because there are conflicting targets: on the one hand, the heating energy expenditure wants to be minimized, on the other, the thermal comfort wants to be maintained or increased. Moreover, the electric consumption is also to be minimized while maintaining a good air quality.

All of this will define optimal frontiers on which multiple optima will be found, depending on the importance weights given to each objective. This is also known as Pareto optimality. The importance weights are defined as hyperparameters, this is, they are provided to the controller on start, and they are experimented with to find out the solutions laying on the frontiers of the Pareto optimality problem. Experiment results will be discussed in Chapter 5.

In Subsections 4.6.1–4.6.4 the reward will be decomposed into the four previous targets. They are all summed to calculate the reward the controller receives at each timestep. In Subsec. 4.6.5 the extra "guidance" for the controller will be explained, using Potential-based reward shaping (theory introduced in Subsec. 2.4.7).

4.6.1 Thermal comfort

The expectation for a new building is to cover 80% of thermal comfort, but the greater the coverage, the better (Fig. 2.3). This is why the reward component from the thermal comfort is defined with two terms: one is positive, encouraging to stay close to the ideal temperature, and the other one is quadratic, negative, to penalize being far away from the comfort range.

The reward function also takes into account if the zone is occupied or not, because in the latter case it is not an issue to drift away from the comfort range, as long as the controller goes back into the control zone for occupied periods. Therefore a smaller linear penalty is applied.

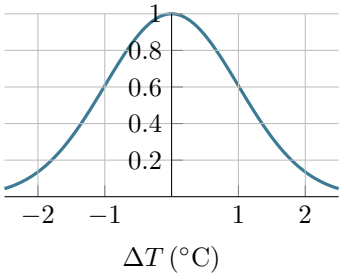


Figure 4.13: Positive reward for thermal comfort with occupancy. The value is low after ±2 °C, i.e. the 90% comfort range. Source: own elaboration using LaTeX's *Tikz* package [74].

The first term, given by Eq. (4.11), is depicted in Fig. 4.13 for occupancy periods. The second term, given by Eq. (4.12), depends on the importance weight defined for the thermal comfort, W_{comf}.

$$\Delta T = T_{op} - T_{ideal} \tag{4.8}$$

$$|\Delta T_{90\%}| = \max\{|\Delta T| - 2\,^\circ\text{C}, 0\,^\circ\text{C}\} \tag{4.9}$$

$$|\Delta T_{80\%}| = \max\{|\Delta T| - 3\,^\circ\text{C}, 0\,^\circ\text{C}\} \tag{4.10}$$

$$r_{\text{comf},+} = \begin{cases} exp\left(-\frac{1}{2}\Delta T^2\right) & \textbf{if the zone is occupied} \\ 0 & \text{otherwise} \end{cases} \tag{4.11}$$

$$r_{\text{comf},-} = \begin{cases} -W_{\text{comf}} \cdot |\Delta T_{90\%}|^2 & \textbf{if the zone is occupied} \\ -W_{\text{comf}} \cdot |\Delta T_{80\%}|/20 & \text{otherwise} \end{cases} \tag{4.12}$$

The symbol ΔT represents the temperature difference, in °C, between the operative and ideal temperatures. As for $|\Delta T_{90\%}|$ and $|\Delta T_{80\%}|$, they represent the abs. temperature difference between the the operative and comfort boundary temperatures. The 90% comfort range is ±2 °C from the ideal, while the 80% comfort range is ±3 °C from the ideal.

4.6.2 Heating

The heating term is a penalty, and the reward will be defined as the negative of that penalty. Similar to [37], here the penalty has been chosen to be smaller when the temperature difference between the operative and ideal temperatures is big, and higher when they are close. This is to embed the "common sense" that is is fine to heat more when it is colder, but not so much when temperatures are in the comfort zone, as it is an unnecessary waste. Incidentally, this is also what an integral controller would do.

The reward is defined in Eq. (4.13), and depends on the fraction of the design heating, *HeatingToMax*, presented in Subsec. 4.4.6. W_{heat} is the importance weight for the heating. The piece-wise function is designed to avoid having a singularity around $\Delta T = 0\,^\circ$C, while the values of the coefficients are chosen to achieve a continuous function up until 0.5 °C, with monotonically increasing derivatives in the full domain.

$$r_{\text{heat},-} = \begin{cases} -W_{\text{heat}} \cdot 10 \cdot HeatingToMax & \textbf{for } \Delta T > 0.5\,^\circ\text{C} \\ -W_{\text{heat}} \cdot 5 \cdot HeatingToMax & \textbf{for } -2 < \Delta T \leq 0.5\,^\circ\text{C} \\ -W_{\text{heat}} \cdot 10 \cdot HeatingToMax/|\Delta T| & \textbf{for } \Delta T \leq -2\,^\circ\text{C} \end{cases} \tag{4.13}$$

4.6.3 CO_2 level

The reward here is built in a similar way to the one in thermal comfort. This is, when the zone is occupied, a positive reward is applied for maintaining an acceptable CO_2 level, defined as

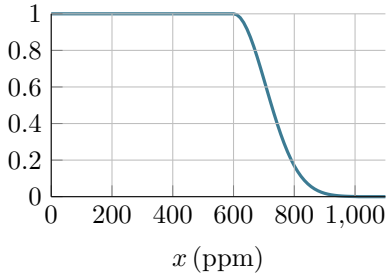


Figure 4.14: Positive reward for the CO_2 level with occupancy. The reward starts decaying after the acceptable level of 600ppm, and is very low after 900ppm. Source: own elaboration using LaTeX's *Tikz* package [74].

600ppm, and a quadratic penalty is applied when surpassing a high level of 900ppm. Otherwise, if the zone is not occupied, a less strict linear penalty is applied. This is reflected in Eqs. (4.16) and (4.17), where x is the CO_2 concentration in ppm. The positive reward for occupancy periods is shown in Fig. 4.14.

$$|\Delta x^{+}_{\text{acc}}| = \max\{(x - 600)/1000, 0\} \tag{4.14}$$

$$|\Delta x^{+}_{\text{high}}| = \max\{(x - 900)/1000, 0\} \tag{4.15}$$

$$r_{\text{CO2},+} = \begin{cases} exp\left(-16|\Delta x^{+}_{\text{acc}}|^2\right) & \textbf{if the zone is occupied} \\ 0 & \text{otherwise} \end{cases} \tag{4.16}$$

$$r_{\text{CO2},-} = \begin{cases} -W_{\text{CO2}} \cdot |\Delta x^{+}_{\text{high}}|^2 & \textbf{if the zone is occupied} \\ -W_{\text{CO2}} \cdot |\Delta x^{+}_{\text{high}}|/100 & \text{otherwise} \end{cases} \tag{4.17}$$

W_{CO2} is the importance weight for the CO_2 level.

4.6.4 Electricity consumption

The electric reward is defined as the negative of a penalty, like in the heating case. It is defined in Eq. (4.18), and depends on the fractional electric power, *FanToMax*, presented in Subsec. 4.4.6. W_{elec} is the importance weight for the electricity consumption.

$$r_{\text{elec},-} = -W_{\text{elec}} \cdot FanToMax \tag{4.18}$$

4.6.5 Potential-based reward shaping

This last Subsection does not introduce new terms into the reward function in a way that they alter the overall objective, but rather, the terms introduced here serve as an external guidance for the controller, for it to know how well it is performing.

The implementation of the potential reward follows the work in [78]. The authors propose using an episodic score, from 0 to 1, that evaluates each timestep accumulated reward since the beginning of an episode w.r.t. the max-min range, i.e. the maximum and minimum total rewards accumulated during any episode—remember that these last 4 weeks in this master thesis' setup—.

With this implementation, potential rewards close to or greater than 1 mean the controller is doing progress by improving the episodic score, while values close to or less than 0 mean that the controller is not on the right path, as it is acting worse than it did other times.

Chapter 5

Analysis and results

In this Chapter an analysis from the MARL-based BEMS is outlined and the results are discussed. It begins with the definition of measurement conditions and gradable metrics, in Secs. 5.1 and 5.2, respectively. Then the experiments carried out in the *dev* building with the implemented controller from previous Chapter are explained in Sec. 5.3. Finally the performance on the *test* building is assessed in Sec. 5.4, before diving into the discussion of results in Sec. 5.5.

5.1 Measurement conditions

The measurements are tracked during a specified episode, which needs to be the same across experiments for the comparison to be valid. Because the controller needs to run during a full year, this timespan has been taken as the evaluation episode duration, using a reference year.

Other interesting periods are a typical summer week, a typical winter week, and typical weeks from spring and autumn, which have more variance across the outdoor conditions, and should reflect whether the controller has learnt to adapt or not.

5.2 Evaluation metrics

When showing the reward function components (Sec. 4.6), an overview of the goals was casted. Here the corresponding evaluation metrics are presented, that will enable a quantitative discussion about the performance of the controller in each area. These will be calculated per thermal zone, meaning that individual zones, having *a priori* different conditions each, can be studied in isolation.

5.2.1 Violations of thermal comfort

To assess the thermal comfort, a cumulative sum of violations of comfort will be used. This means, for a given zone, every timestep when the zone is occupied and the thermal conditions are outside the 80% comfort level, a counter is increased.

5.2.2 Worst CO_2 level

The highest CO_2 level during occupied periods is recorded, and presented as a metric. It represents the worst case scenario.

5.2.3 Mechanical Air Changes per Hour

The Air Changes per Hour (ACH) is a metric that already appeared before, as it is a good indicator of overall air quality. It is related to the CO_2 level.

Here the mechanical ACH is measured: knowing the zone's volume, the fractional mass airflow at each timestep, and considering the air density this value can be calculated for the whole episode.

5.2.4 Heating and electricity consumption

Both the heating and electricity consumption can be calculated by summing over all the timesteps of an episode. They are measured in kWh.

5.3 Experiments

In Sec. 4.6, importance weights were introduced in the reward function calculations, as well as the Pareto optimality criteria. In this Section, the experiments will compare controllers running with different sets of weights, using the metrics previously introduced. Specifically, following the work in [37] they will be compared against the baseline controller and the best ones will be identified according to the optimality criteria.

The full set of weights $\mathcal{W}$ that defines a *run* is $\{W_{\text{comf}}, W_{\text{heat}}, W_{\text{CO2}}, W_{\text{elec}}\}$.

When exploring the different sets of weights—experiments from Subsecs. 5.3.1, 5.3.2—the training process is run for 150000 timesteps, while for the last experiment (Subsec. 5.3.3) the training process is run for longer, until reaching $\approx$ 5M steps. For further details please refer to the App. B, which contains results about the training proces itself and discusses these decisions.

5.3.1 Initial exploration

In the first exploration, and lacking any information that can guide a decision, the following range of values has been defined for each weight:

- $W_{\text{comf}}, W_{\text{CO2}} \in \{30, 100, 200\}$
- $W_{\text{heat}}, W_{\text{elec}} \in \{1, 3, 10, 100\}$

Because exploring each combination (144 in total) would be very expensive, a stochastic approach has been taken: for each weight, 20 samples have been drawn independently, and then joined, to obtain 20 sets of weights to test. These are represented in Table 5.1.

The comparison with the baseline is shown in Fig. 5.1, as the average across all zones—the results for each zone are presented at the end of this Chapter, in Figs. 5.5 and 5.6. It is clear how the comfort is greatly improved, at the expense of consuming more heating energy. Also, the electricity consumption is reduced, but the air quality is deemed low, because the CO_2 levels are, for most of the controllers, above 1000ppm. Further refinement is needed to adequately choose the weights $\mathcal{W}$.

In addition, a linear relationship is observed between the ACH and the electrical consumption—seen in subsequent experiments as well—, that will be commented in the discussion (Subsec. 5.5.4).

5.3.2 Refinement

After seeing that heating and CO_2 level are too high, more weight is put on them, electricity consumption is penalized less, and the comfort weight is reduced, for 10 runs:

- $W_{\text{comf}} \in \{30, 50, 70\}$
- $W_{\text{CO2}} \in \{100, 200, 300\}$
- $W_{\text{heat}} \in \{50, 100, 200\}$
- $W_{\text{elec}} = 1$

Also, an even higher penalty is considered for the CO_2 level, with the following ranges, for 10 runs more:

Run	W_{comf}	W_{CO2}	W_{heat}	W_{elec}	Run	W_{comf}	W_{CO2}	W_{heat}	W_{elec}
1	200	200	1	1	**11**	30	100	10	3
2	100	30	1	10	**12**	200	100	10	3
3	100	200	1	10	**13**	200	200	10	3
4	30	200	1	100	**14**	100	30	10	100
5	100	200	1	100	**15**	100	100	10	100
6	200	30	3	1	**16**	200	30	10	100
7	100	200	3	10	**17**	200	100	100	1
8	30	200	3	100	**18**	200	200	100	3
9	200	100	3	100	**19**	100	30	100	10
10	100	30	10	1	**20**	30	30	100	100

Table 5.1: Weights used in the initial approach, with runs labeled from 1 to 20 (although they were run in parallel).

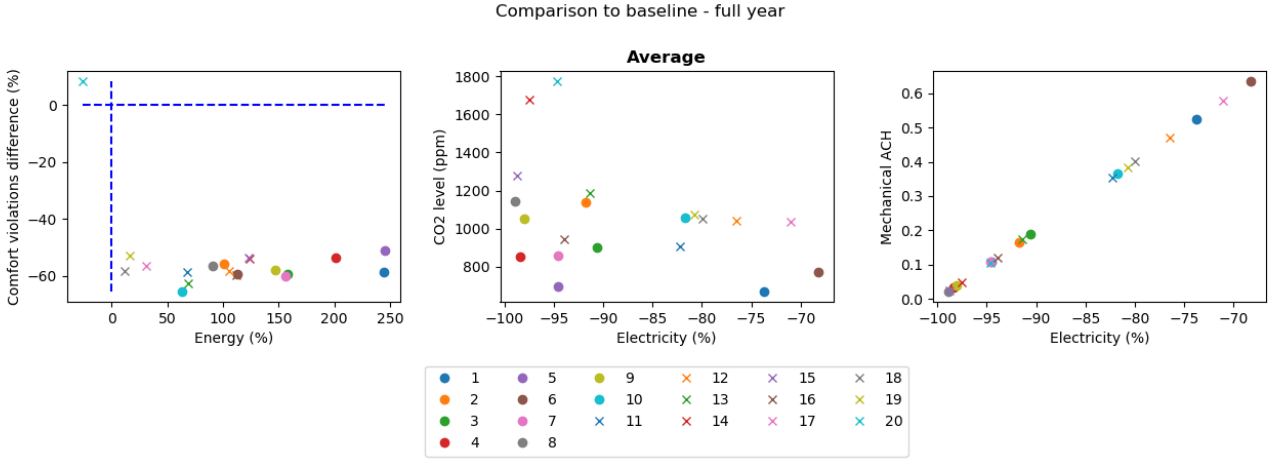


Figure 5.1: Comparison to baseline from *dev* building for runs 1–20, average of all the zones. The blue dashed line represents the baseline controller. Percentages are shown as differences with respect to the baseline. The baseline level of CO_2 has not been displayed as a dashed line because it varies across zones. Source: own elaboration using *matplotlib* [79].

Run	W_{comf}	W_{CO2}	W_{heat}	W_{elec}	Run	W_{comf}	W_{CO2}	W_{heat}	W_{elec}
21	70	300	100	1	**31**	70	500	200	1
22	70	100	100	1	**32**	70	700	200	1
23	70	200	200	1	**33**	50	700	100	1
24	50	200	50	1	**34**	70	1000	100	1
25	30	300	200	1	**35**	**70**	**700**	**200**	**1**
26	**70**	**200**	**200**	**1**	**36**	30	1000	200	1
27	70	300	100	1	**37**	**50**	**700**	**100**	**1**
28	50	200	50	1	**38**	**70**	**1000**	**100**	**1**
29	**50**	**200**	**50**	**1**	**39**	**50**	**700**	**100**	**1**
30	70	200	50	1	**40**	70	700	100	1

Table 5.2: Weights used in the refinement experiment, labelled from 21 to 40. The ones in **bold** are duplicates from others, this can happen as a result of sampling independently for each individual weight.

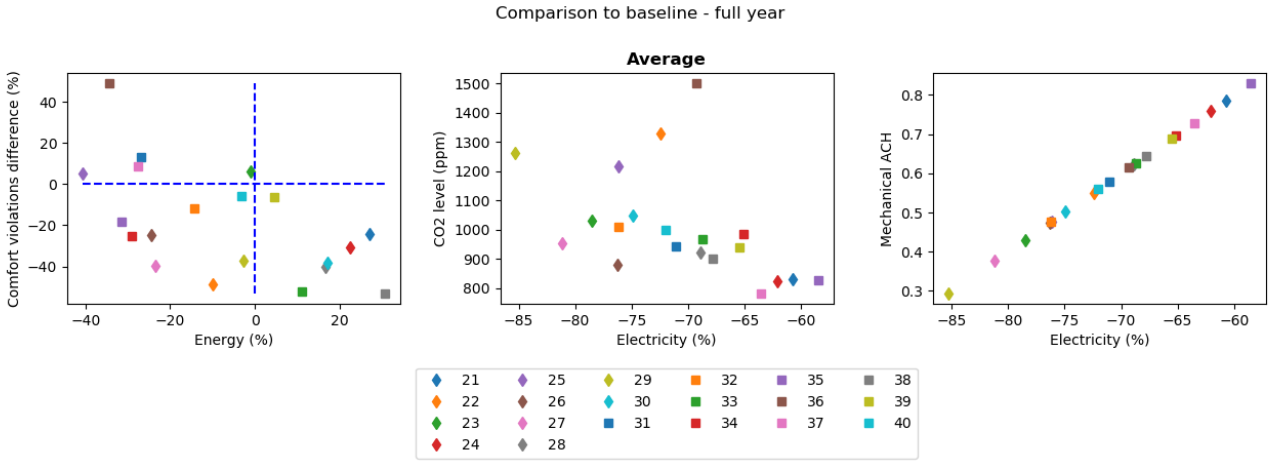


Figure 5.2: Comparison to baseline from *dev* building for runs 21–40, average of all the zones. The blue dashed line represents the baseline controller. Percentages are shown as differences with respect to the baseline. The baseline level of CO_2 has not been displayed as a dashed line because it varies across zones. Source: own elaboration using *matplotlib* [79].

- $W_{\text{comf}} \in \{30, 50, 70\}$
- $W_{\text{CO2}} \in \{500, 700, 1000\}$
- $W_{\text{heat}} \in \{100, 200\}$
- $W_{\text{elec}} = 1$

The values drawn from these runs are shown in Table 5.2, while their comparison to the baseline is shown in Fig. 5.2, as an average across all the zones. The results for each zone are represented in Figs. 5.7 and 5.8 at the end of this Chapter.

5.3.3 Final selection

From the previous experiments, the "best" weights are selected and trained for longer for their evaluation against the baseline. The criteria for selecting the "best" has been to choose the ones that consistently reduce the heating energy while improving comfort, and present a level of CO_2 below 1000ppm, across the different thermal zones.

According to the presented criteria, the runs picked are **26** (improvement in 8/8 zones), **27** and **35** (improvement in 7/8 zones), and **34** (improvement in 6/8 zones). Outside this criteria, run

Run	$W_{\mathbf{comf}}$	$W_{\mathbf{CO2}}$	$W_{\mathbf{heat}}$	$W_{\mathbf{elec}}$
41	70	300	100	1
42	70	100	100	1
43	30	300	200	1
44	70	200	200	1
45	70	700	200	1

Table 5.3: Weights used in the final experiment, labelled from 41 to 45.

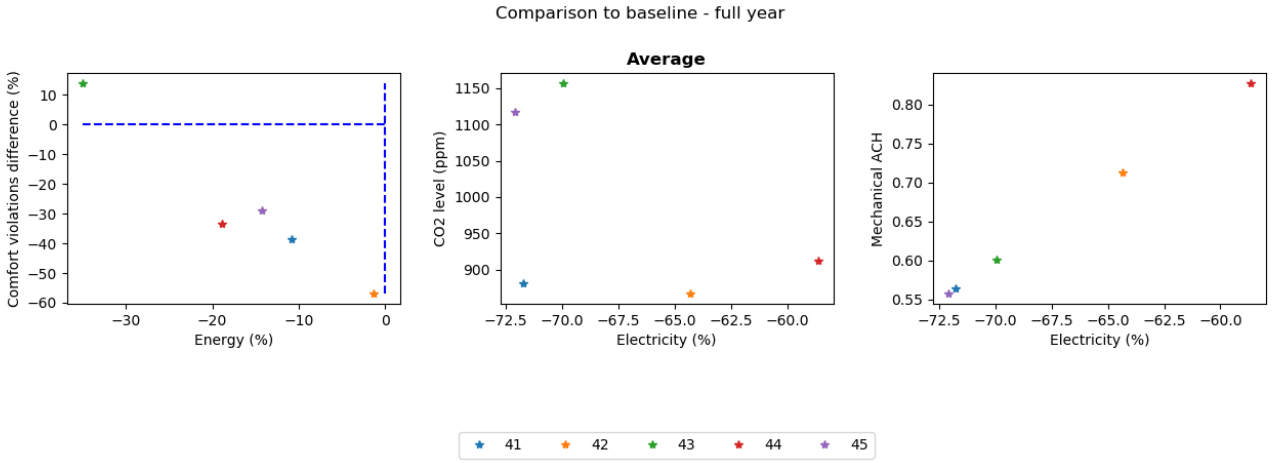


Figure 5.3: Comparison to baseline from *dev* building for runs 41–45, average of all the zones. The blue dashed line represents the baseline controller. Percentages are shown as differences with respect to the baseline. The baseline level of CO_2 has not been displayed as a dashed line because it varies across zones. Source: own elaboration using *matplotlib* [79].

25 has also been selected, to widen the range of the final selected weights. These are shown in Table 5.3, as runs 41–45.

The results from this last experiment are collected in Fig. 5.3, showing the controllers comparison to the baseline as the average across all the zones. Like in the previous experiments, the results for each zone are moved to the end of this Chapter, in Figs. 5.9 and 5.10.

5.4 Performance on the *test* building

The "best" runs have been finally tested on the *test* building, to see if the controllers learnt successfully. Here both the refinement and the final retrained models are compared. Their results for the average across all zones are presented in Fig. 5.4.

The controllers from the refinement experiment provide similar energy savings (42% on avg.) and comfort improvement (30% on avg.), while yielding different electrical consumption (55–85% savings range) and air renewal rates (0.38–0.9 ACH range).

Oppositely, the retrained final controllers provide a wider range of energy savings (8–42%), while keeping similar comfort improvements (30% on avg.). The CO_2 levels have more dispersion, and some controllers surpass 1000ppm (**43**, **45**), even though the ACH are slightly superior to the ones in the refinement experiment.

5.5 Discussion

It is first noticed that acceptable results are obtained only with the refinement and the final selection experiments, as the initial exploration, even if reducing the comfort violations, consumes

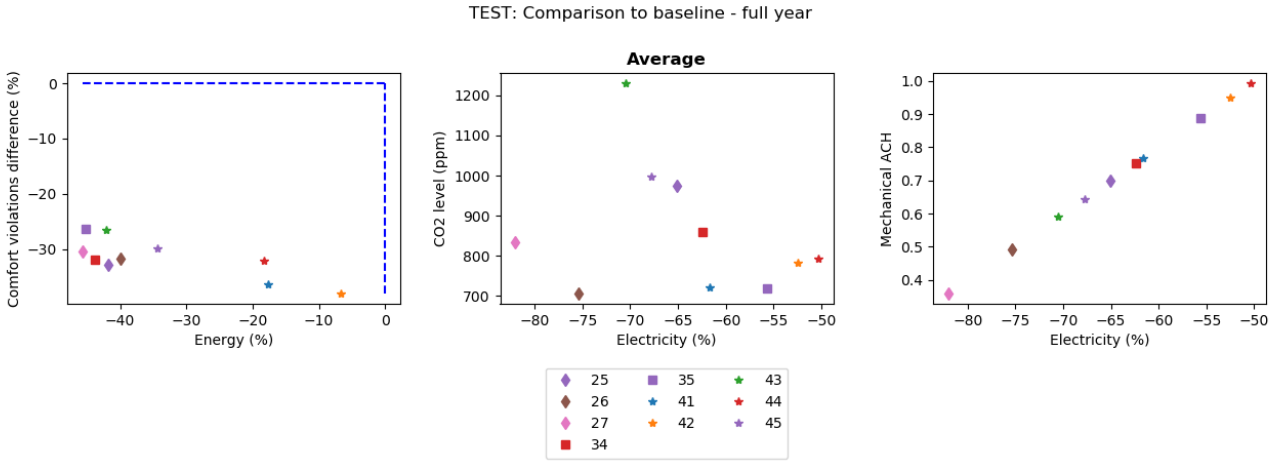


Figure 5.4: Comparison to baseline from *test* building—that is similar to the real building—for the "best" runs, average of all the zones. The blue dashed line represents the baseline controller. Percentages are shown as differences with respect to the baseline. The baseline level of CO_2 has not been displayed as a dashed line because it varies across zones. Source: own elaboration using *matplotlib* [79].

more heating energy. Hence the discussion will be centered on the former ones, both for the *dev* and the *test* buildings—the latter being similar to the the building to be constructed.

5.5.1 Pareto sets

From the refinement experiment, looking at the average of all zones (Fig. 5.2) the Pareto frontier is formed by the runs **22**, **27**, **34**, **35**, and arguably also **25**. Non-incidentally this set almost matches the "best" runs selected for the final experiment. Besides, the CO_2 levels are too high (over 1000ppm) in the extremes of the Pareto frontier only (runs **22**, **25**), staying below that mark for the other optimal runs. For these the ACH go from 0.5 to 0.9.

In the final selection experiment, also from the average of all zones (Fig. 5.3) the run **43** increases the baseline comfort violations by more than 10%, and presents a CO_2 level over 1100ppm, this is why it cannot be included in the optimal Pareto set, which is formed by runs **41**, **42**, and **44**. These show maximum CO_2 levels close to or below 900ppm, which is considered acceptable, with ACH between 0.55 and 0.9.

These Pareto sets may vary when looking at each zone individually: even if the controllers selected as the "best" often appear in the respective Pareto frontiers, other controllers might be included as well—e.g. controller **33** in the East zone from the *dev* building, Fig. 5.8. With this consideration, it is possible to improve the results not only on average, but also locally, by choosing the appropriate Pareto frontiers for each thermal zone.

5.5.2 Differing results for controllers with the same weights

Remarkably, the duplicate experiments in runs 21–40 do not result in the same energy savings, comfort violations, CO_2 levels or air renovations, despite sharing the importance weights. Appendix B provides an insight into why this can happen: there are multiple ways to achieve the same reward value, and controllers may diverge to different behaviors during the training process.

However, even if the same weights produce different trained controllers, what is relevant in the end is the evaluation of the trained controllers themselves. In this regard, the Pareto set from the refinement experiment reduces the heating energy and the comfort violations to a greater degree than the final experiment, although the latter has been trained for longer.

5.5.3 The "best" controllers on the *test* building

In the *test* building the final experiment results do not show any significant improvement over the ones from the "best" controllers in the refinement experiment. Rather, they jointly define a new Pareto frontier, as the runs **41**, **42**, **44** moderately improve the comfort (5% on avg.) at the expense of increased heating energy consumption (27% on avg.).

Overall, the "best" controllers trained on the *dev* building seem to be applicable to the *test* building, as they improve both the comfort and the energy expenditure, without incurring in too high CO_2 levels.

5.5.4 Linear relationship between ACH and electrical consumption

In the results presented there is a clear linear relationship between the air changes and the electrical consumption that deserves discussion.

Whereas there is a known cubic relationship between the volume airflow and the electrical power for continuous airflows, in the proposed implementation the actuated variable is the mass airflow, as a fraction from 0 (no flow) to 1 (maximum mass). This fraction must be interpreted as a duty cycle: % of time the fan has been on, with constant speed. This explains the linear relationship observed, as the airflow is constant while the fan is on.

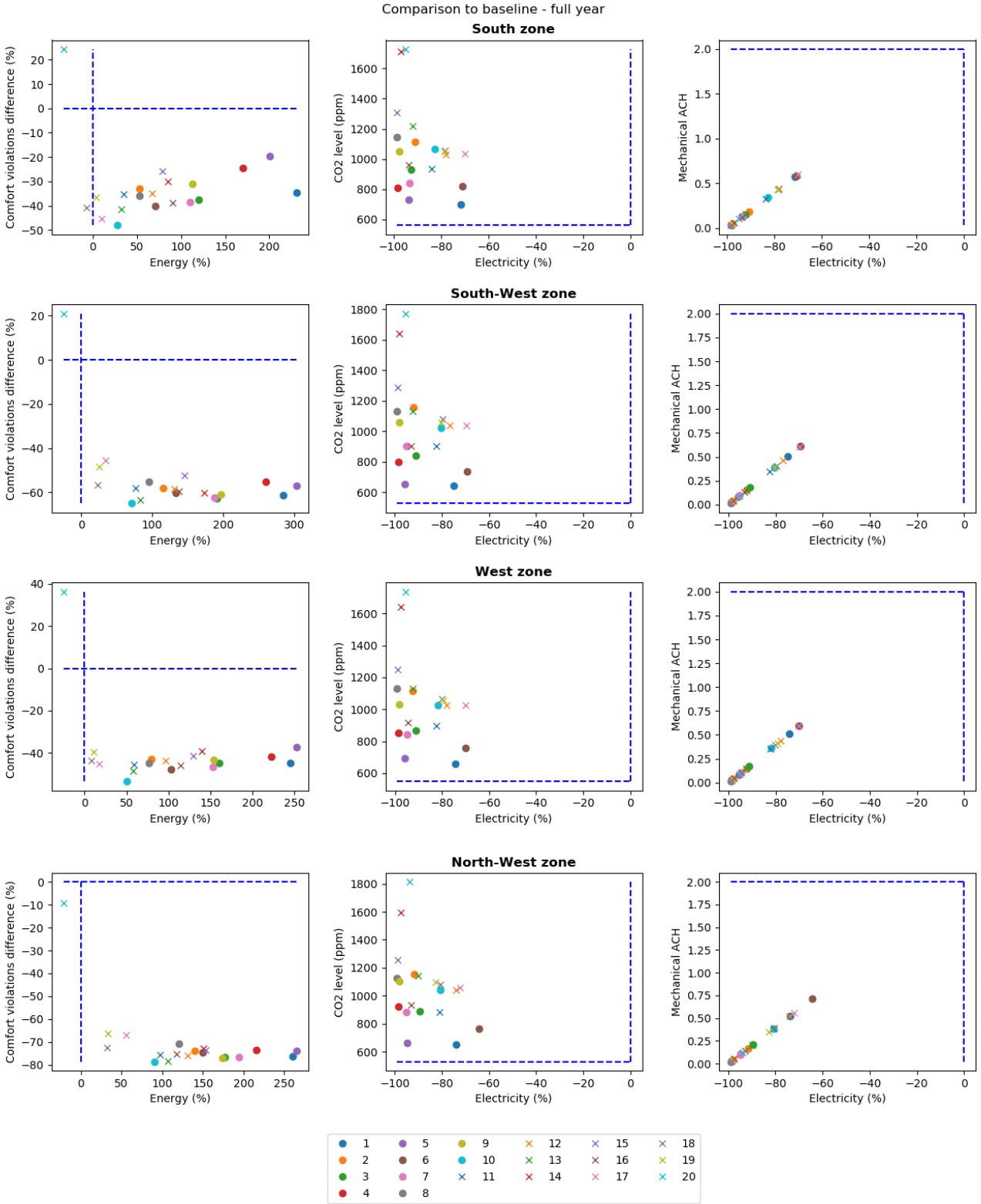


Figure 5.5: Comparison to baseline from *dev* building's zones S, SW, W, and NW, runs 1–20. The blue dashed line represents the baseline controller. Percentages are shown as differences with respect to the baseline. Source: own elaboration using *matplotlib* [79].

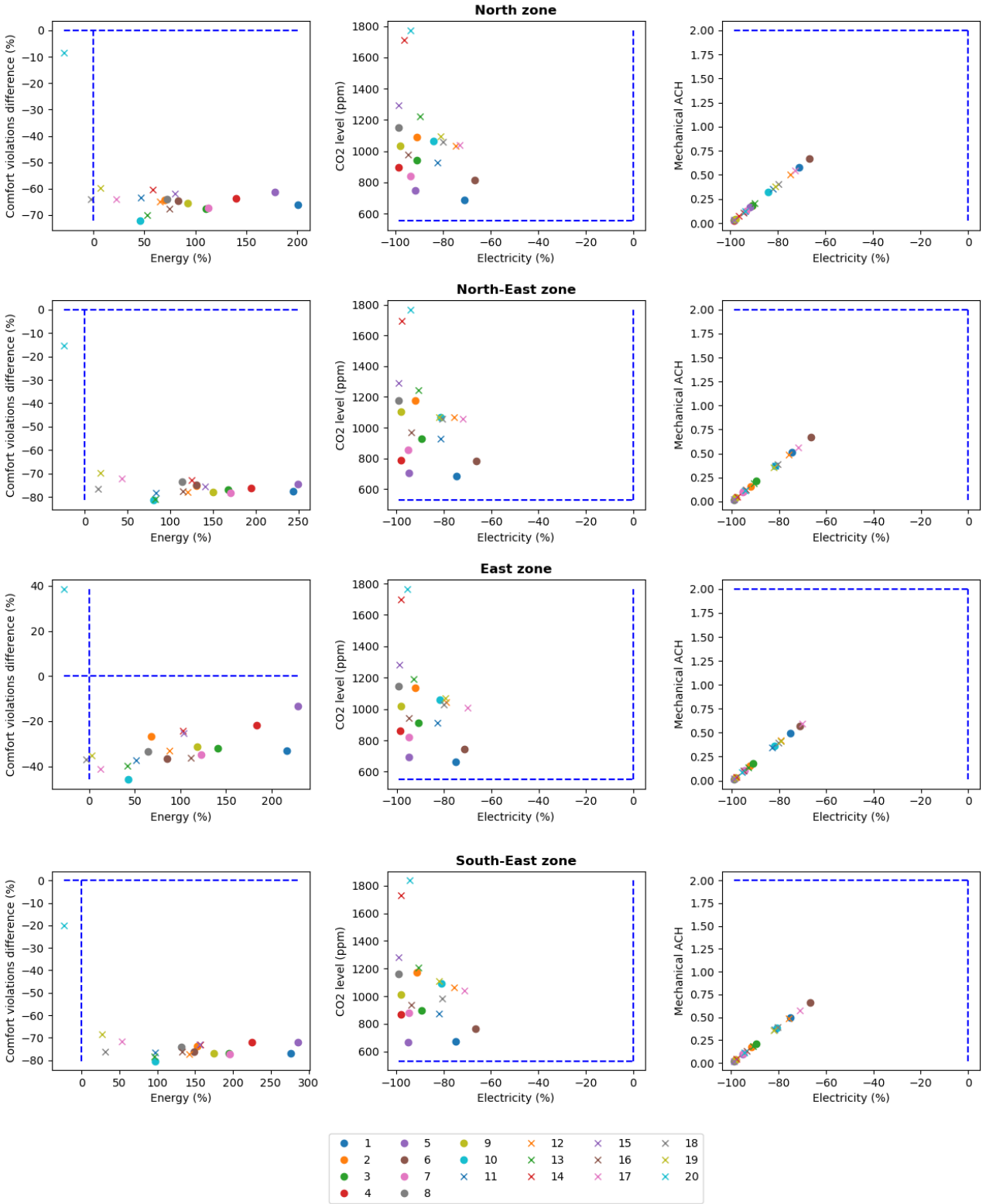


Figure 5.6: Comparison to baseline from *dev* building's zones N, NE, E, and SE, runs 1–20. The blue dashed line represents the baseline controller. Percentages are shown as differences with respect to the baseline. Source: own elaboration using *matplotlib* [79].

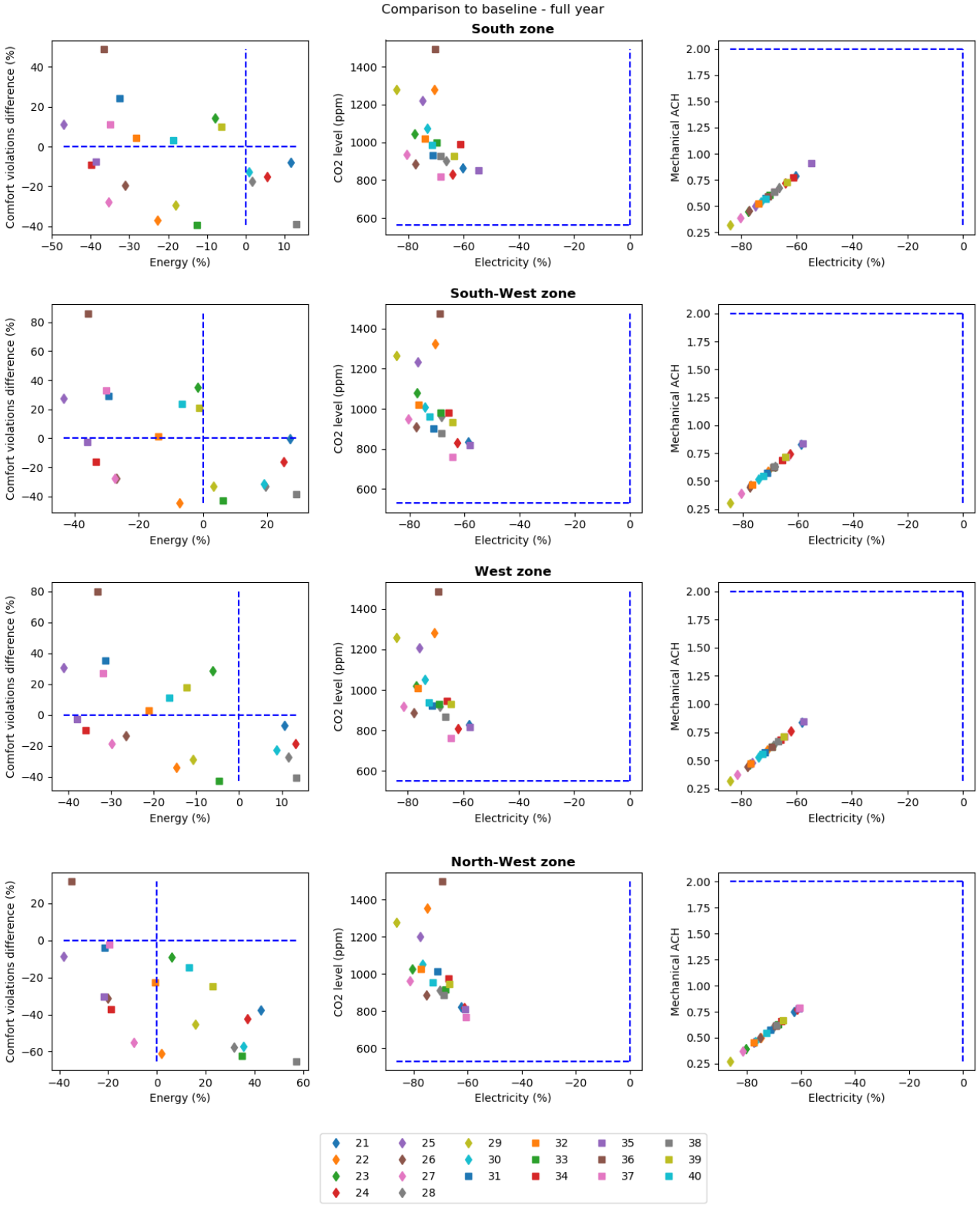


Figure 5.7: Comparison to baseline from *dev* building's zones S, SW, W, and NW, runs 21–40. The blue dashed line represents the baseline controller. Percentages are shown as differences with respect to the baseline. Source: own elaboration using *matplotlib* [79].

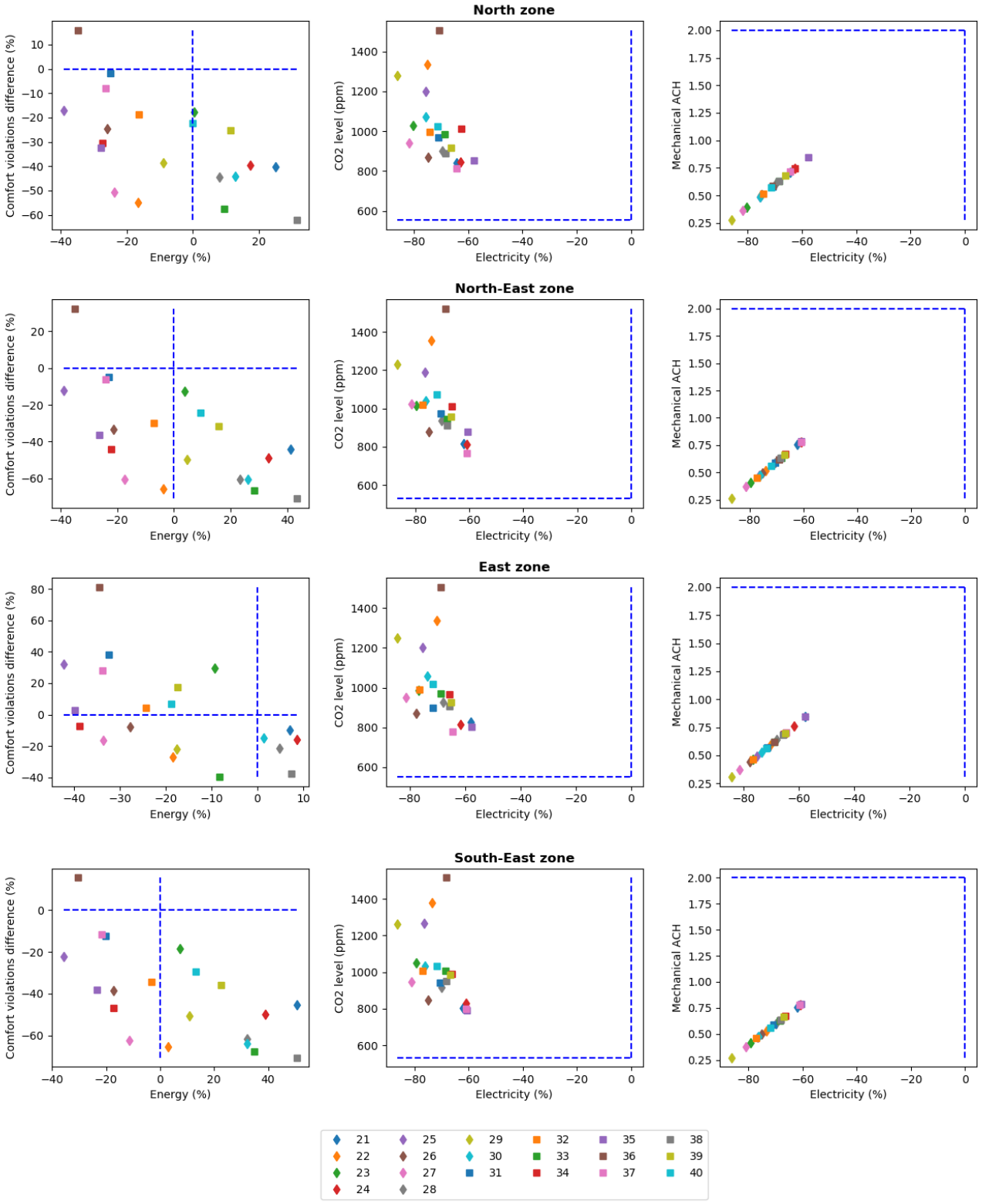


Figure 5.8: Comparison to baseline from *dev* building's zones N, NE, E, and SE, runs 21–40. The blue dashed line represents the baseline controller. Percentages are shown as differences with respect to the baseline. Source: own elaboration using *matplotlib* [79].

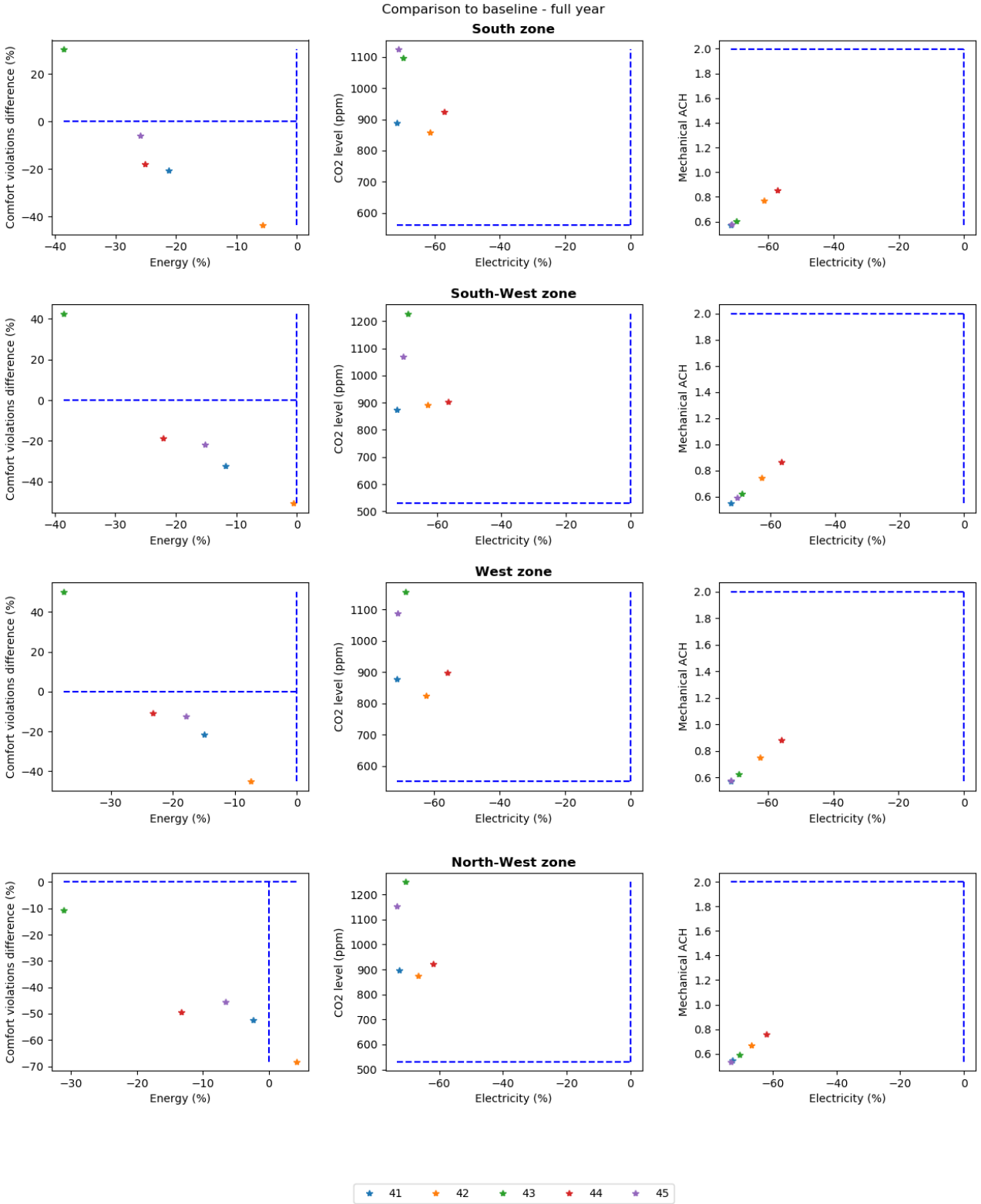


Figure 5.9: Comparison to baseline from *dev* building's zones S, SW, W, and NW, runs 41–45. The blue dashed line represents the baseline controller. Percentages are shown as differences with respect to the baseline. Source: own elaboration using *matplotlib* [79].

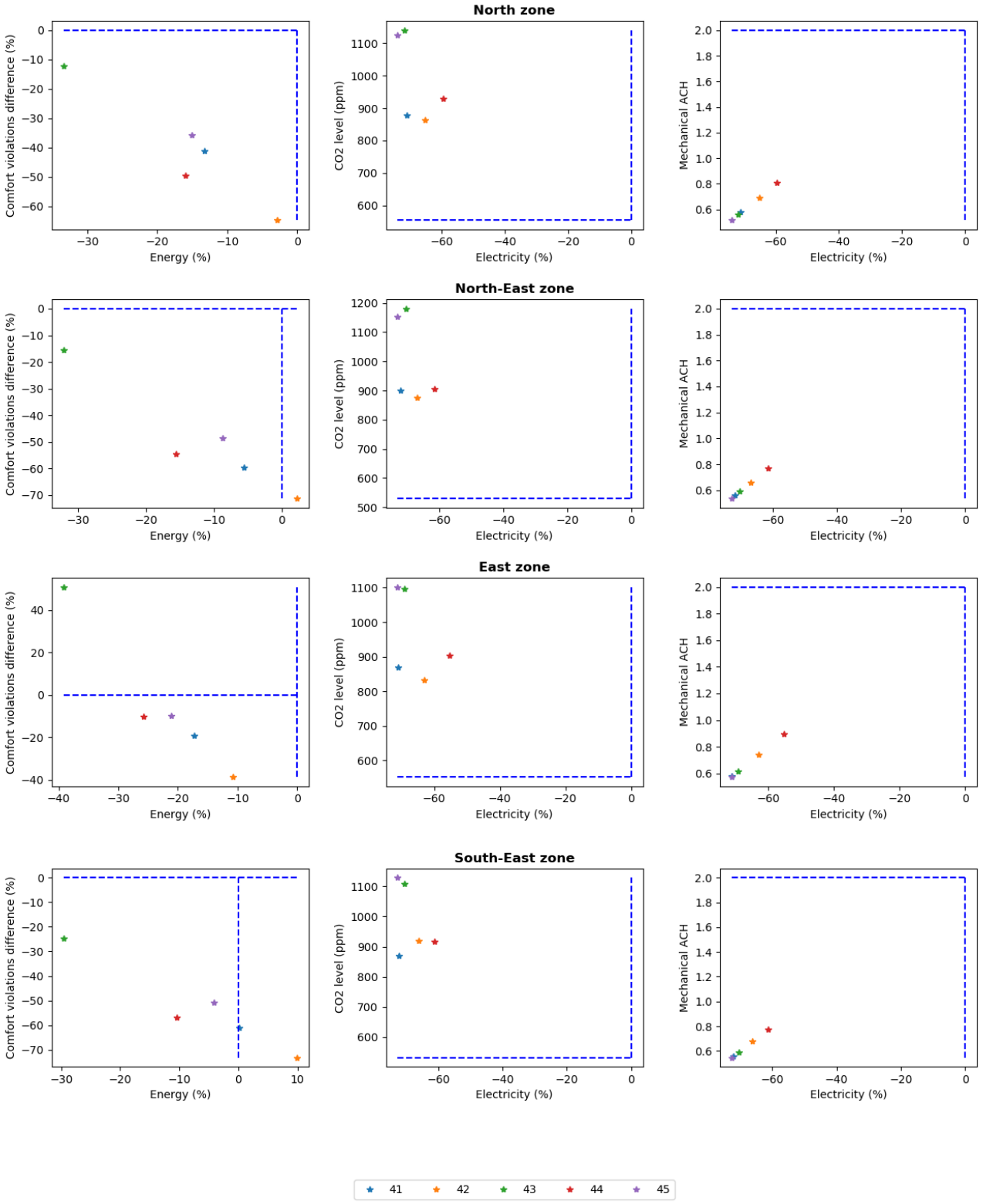


Figure 5.10: Comparison to baseline from *dev* building's zones N, NE, E, and SE, runs 41–45. The blue dashed line represents the baseline controller. Percentages are shown as differences with respect to the baseline. Source: own elaboration using *matplotlib* [79].

Chapter 6

Conclusion

After presenting a case of study with a building under development, it has been abstracted into an energy simulation tool, and a decentralized MARL-based BEMS has been proposed to control individual thermal zones' setpoint temperature, the heat exchanger bypass signal, and the fractional mass airflow for outdoor ventilation. In this Chapter, the key findings are summarized.

6.1 Abstraction of the architectural solution

The building is in a pre-design stage, so it is subject to changes. Nonetheless, the main attributes from the building, such as the atrium relationship to the sorrounding zones and the natural ventilation, have been extracted into a simplified abstraction (Figs. 4.1, 4.2, 4.4) that allowed developing and experimenting with multiple controllers.

A second abstraction, with a richer level of information (Fig. 4.3), allowed testing whether the developed controllers would be applicable to the real building, or at least a close approximation to it.

6.2 Integration between the simulation and the controller

In the first place, FMU has proven to be a useful definition to package the simulation into an executable file (Fig. 4.5). The tool from [45], extended as a contribution from this thesis, has been a cornerstone to support this.

Second, the novel integration using the FMU inside an OpenAI Gym environment has been the basis to allowing the development of any control system, as it provided the connection from the controls to the simulation. Previous integrations used BCVTB middleware (Fig. 2.4), adding a level of indirection, whereas in this master thesis the simulation process is embedded inside the same environment (Fig. 4.6). This is convenient for training, as the environment has all the information needed and can control the simulation directly, while the BCVTB requires being in control and complicates the design of the integration.

6.3 Controller proposal

A controller architecture has been proposed, and multiple versions have been developed, in order to find out an optimal set in the trade-off between comfort, air quality, and energy consumption. This described a Pareto optimality frontier.

Results show in fact the Pareto frontier exists in both building abstractions, not finding significant differences between controllers trained for 150000 steps and others trained for 5M steps. On the other hand, a given set of hyperparameters—importance weights—did not seem to guarantee a certain behavior of the controller, but rather this was found to evolve in a particular direction

during the training process. All of this indicates that, in order to maximize resources efficiency, instead of training controllers for very long times, it is better to train more controllers, and then select the subset providing the best performance against the baseline.

In addition, the Pareto optimal set has been shown to vary depending on the building, and also across zones. Although it is possible to improve the average energy consumption and comfort just by using the previous average Pareto frontiers, it is better to select the Pareto frontiers locally for each zone, thus adapting the controls to their different conditions.

It is finally highlighted that controllers trained on the *dev* building perform well on the *test* building, showing an adaption to a different setup. This increases the confidence towards a future successful deployment in the real building.

6.4 Fulfillment of the main objective

The goal presented for this master thesis has been fulfilled, as there is an optimal set of controllers that reduce energy consumption, while keeping or even improving comfort up to different degrees, and maintaining an acceptable air quality—CO_2 level below 1000ppm.

Finally, having a set of optimal controllers means that different modes can be enabled, depending on the BEMS' settings: "comfort" modes will pull the controller towards better thermal comfort conditions at the expense of some energy savings, while "ECO" modes will do the opposite. Because this can be chosen individually for each zone, it results in a completely decentralized control system, as intended.

Chapter 7

Future Work

Here the work that is left open will be discussed. It could be targeted in the future, either as the direct continuation of the work presented in this master thesis, or as new branching topics. The ideas are presented from the most specific to the most generic.

7.1 Porting the results to the real building

All of the work presented so far has considered the implementation in the real building from the case of study as an end-goal. Hence, once the building is constructed, an on-site development will be needed to use the work from this thesis. In what follows, an outline will be described.

7.1.1 Architecture

The proposed BEMS can be implemented by using a central computer that will receive sensorial information from each of the zones in the building via an Open Platform Communication (OPC) server, using any given communication protocol and hardware infrastructure (e.g. KNX standard), as well as weather forecasts from the nearest weather station via a web Application Programming Interface (API). These can be obtained from the Danish Meteorological Institute (DMI).

In turn, it will calculate actions to take every 10 min (timestep size), and send them via the same communication protocol to each zone setpoint and ventilation fan unit.

Occupants in the building could have an application on their smartphones to choose their desired level of comfort vs. energy savings, and that would drive the choice of the controller mode in the BEMS, taking into account which zone the request is made for.

7.1.2 Data collection

In order to check the controller applicability to the real building, one limitation from the algorithm used (PPO) is that it is **on-policy**, meaning it should be able to take decisions and observe the results. This might be difficult to achieve in a real world scenario, where people expect the system to always work equally well.

One alternative would be to collect data from the building while using a baseline controller, and then use the collected data to improve the abstraction of the building in the simulation, to make it more similar to the real-life results.

Another option would be to change the algorithm for an **off-policy** one, like any of the variants of the Q-learning. This would allow to tune the trained controller with the data collected from the building while running a baseline controller.

In any of the cases, collecting data is a must.

7.2 Improvements on the controller

Here a set of ideas that could improve the controller performance is presented.

7.2.1 Independent heat recovery

In the presented controller the heat recovery signal is unique, and shared across all the zones. On the one hand this encourages collaborative learning, but on the other it may be inefficient when zones have different needs.

Branching off from a partly trained controller, the environment could be changed so that each zone's decision affects its own heat recovery, rather than voting a common decision. Then the training process would complete using the environment with the new behavior.

7.2.2 Occupancy prediction

Being able to predict occupancy accurately would allow to further loosen comfort requirements and to increase the energy savings. However, this is a complex issue on its own that fell out from the scope of this thesis. If developed, it could be added as one of the predictions observed by the controller at each timestep.

7.2.3 Adding a planner

After collecting enough measurements from the building, as well as users' preference patterns, a higher-level controller could be introduced, operating on a longer timescale (e.g. 1 hour), and that would learn to switch between settings—the different controller modes—, anticipating users' choices.

7.2.4 Pareto optimality as the potential-based reward shaping

Given that Pareto optimality is the metric used to assess the goodness of a trained controller, it could be introduced to guide the controller during learning as the potential-based reward shaping, instead of the current one used.

For instance, the potential function ϕ could be the sum of comfort improvement and energy savings % with respect to the baseline during the given episode.

Bibliography

[1] Energy use in buildings, 2013. URL https://ec.europa.eu/energy/eu-buildings-factsheets-topics-tree/energy-use-buildings_en.

[2] European Union: European Commission. COMMUNICATION FROM THE COMMISSION TO THE EUROPEAN PARLIAMENT, THE COUNCIL, THE EUROPEAN ECONOMIC AND SOCIAL COMMITTEE AND THE COMMITTEE OF THE REGIONS An EU Strategy on Heating and Cooling. *COM/2016/051 final*. URL https://eur-lex.europa.eu/legal-content/EN/TXT/?qid=1575551754568&uri=CELEX:52016DC0051.

[3] European Union: European Parliament and the Council. Consolidated text: Directive 2010/31/EU of the European Parliament and of the Council of 19 May 2010 on the energy performance of buildings (recast), 2021. URL https://eur-lex.europa.eu/eli/dir/2010/31/2021-01-01.

[4] Harris Poirazis, Åke Blomsterberg, and Maria Wall. Energy simulations for glazed office buildings in Sweden. *Energy and Buildings*, 40(7):1161–1170, jan 2008. ISSN 03787788. doi: 10.1016/j.enbuild.2007.10.011.

[5] Min Hee Chung and Eon Ku Rhee. Potential opportunities for energy conservation in existing buildings on university campus: A field survey in Korea. *Energy and Buildings*, 78: 176–182, 2014. ISSN 03787788. doi: 10.1016/j.enbuild.2014.04.018.

[6] Ruben Hidalgo-Leon, Jaqueline Litardo, Javier Urquizo, Daniel Moreira, Pritpal Singh, and Guillermo Soriano. Some factors involved in the improvement of building energy consumption: A brief review. In *2019 IEEE 4th Ecuador Technical Chapters Meeting, ETCM 2019*. Institute of Electrical and Electronics Engineers Inc., nov 2019. ISBN 9781728137643. doi: 10.1109/ETCM48019.2019.9014890.

[7] L. A. Hurtado, P. H. Nguyen, W. L. Kling, and W. Zeiler. Building energy management systems - Optimization of comfort and energy use. In *Proceedings of the Universities Power Engineering Conference*, 2013. ISBN 9781479932542. doi: 10.1109/UPEC.2013.6714910.

[8] Minjae Shin and Jeff S. Haberl. Thermal zoning for building HVAC design and energy simulation: A literature review, nov 2019. ISSN 03787788.

[9] Instituto para la Diversificación y el Ahorro de Energía (IDAE). Guía técnica procedimientos y aspectos de la simulación de instalaciones térmicas en edificios, 2008.

[10] Carrie A. Redlich, Judy Sparer, and Mark R. Cullen. Sick-building syndrome, apr 1997. ISSN 01406736.

[11] American Society of Heating, Refrigerating and Air-Conditioning Engineers, Inc. American Society of Heating, Refrigerating and Air-Conditioning Engineers, Inc. (ASHRAE), 2017. ISBN 978-1-939200-58-7.

[12] Kevin Michael Smith. *Development and Operation of Decentralized Ventilation for Indoor Climate and Energy Performance*. PhD thesis, Technical University of Denmark, 2015. URL www.byg.dtu.dk.

[13] J. Pillai and R. Desai. DEHUMIDIFICATION STRATEGIES AND THEIR APPLICABILITY BASED ON CLIMATE AND BUILDING TYPOLOGY. *2018 Building Performance Analysis Conference and SimBuild co-organized by ASHRAE and IBPSA-USA*, 2018.

[14] E. Panadero. Psychrochart, 2019. URL `https://github.com/azogue/psychrochart`.

[15] American Society of Heating, Refrigerating and Air-Conditioning Engineers, Inc. Weather Data 2017 ASHRAE Handbook Fundamentals (SI Edition). In *2017 ASHRAE Handbook—Fundamentals* American Society of Heating, Refrigerating and Air-Conditioning Engineers, Inc. [11]. ISBN 978-1-939200-58-7.

[16] A. I. Dounis and C. Caraiscos. Advanced control systems engineering for energy and comfort management in a building environment-A review, aug 2009. ISSN 13640321.

[17] Pervez Hameed Shaikh, Nursyarizal Bin Mohd Nor, Perumal Nallagownden, Irraivan Elamvazuthi, and Taib Ibrahim. A review on optimized control systems for building energy and comfort management of smart sustainable buildings, jun 2014. ISSN 13640321.

[18] G. Bianco, S. Bracco, F. Delfino, L. Gambelli, M. Robba, and M. Rossi. A Building Energy Management System for demand response in smart grids. In *IEEE International Conference on Automation Science and Engineering*, volume 2020-August, pages 1485–1490. IEEE Computer Society, aug 2020. ISBN 9781728169040. doi: 10.1109/CASE48305.2020.9216880.

[19] ANSI ASHRAE. Standard 55-2010, thermal environmental conditions for human occupancy. *Atlanta USA*, page 15, 2010.

[20] Jianlei Niu and John Burnett. Integrating radiant/operative temperature controls into building energy simulations. *ASHRAE Transactions*, 104:210, 1998.

[21] ASHRAE. Physiological principles for comfort and health. In *1985 ASHRAE Handbook of Fundamentals*, chapter 8. American Society of Heating, Refrigerating and Air-Conditioning Engineers, Inc. (ASHRAE), 1985.

[22] Mohammad Taleghani, Martin Tenpierik, Stanley Kurvers, and Andy Van Den Dobbelsteen. A review into thermal comfort in buildings, oct 2013. ISSN 13640321.

[23] CEN/TC 156 - Ventilation for buildings. EN 16798-1:2019 Energy performance of buildings - Ventilation for buildings - Part 1: Indoor environmental input parameters for design and assessment of energy performance of buildings addressing indoor air quality, thermal environment, lighting and acoustics - Module M1-6. Technical report, 2019. URL `https://standards.iteh.ai/catalog/standards/cen/b4f68755-2204-4796-854a-56643dfcfe89/en-16798-1-2019`.

[24] Big Ladder Software LLC. Engineering reference on Occupant Thermal Comfort, 2018. URL `https://bigladdersoftware.com/epx/docs/8-9/engineering-reference/occupant-thermal-comfort.html#adaptive-comfort-model-based-on-european-standard-en15251-2007`.

[25] Mehzabeen Mannan and Sami G. Al-Ghamdi. Indoor air quality in buildings: A comprehensive review on the factors influencing air pollution in residential and commercial structure. *International Journal of Environmental Research and Public Health*, 18(6), 2021. ISSN 1660-4601. doi: 10.3390/ijerph18063276. URL `https://www.mdpi.com/1660-4601/18/6/3276`.

[26] N. R.M. Sakiyama, J. C. Carlo, J. Frick, and H. Garrecht. Perspectives of naturally ventilated buildings: A review, sep 2020. ISSN 18790690.

[27] Andrew Persily. Quit Blaming ASHRAE Standard 62.1 for 1000 ppm CO2. In *The 16th Conference of the International Society of Indoor Air Quality Climate (Indoor Air 2020)*, Seoul, 2020. URL `https://www.nist.gov/publications/quit-blaming-ashrae-standard-621-1000-ppm-co2`.

[28] ASHRAE ANSI. *Standard 62.1-2019 Ventilation for acceptable Indoor Air Quality.* ANSI/ASHRAE, 2019.

[29] J.M. Bartolomé Martín and M.A. Navas Martin. DTIE 11.02: Regulación y control de instalaciones de climatización. In *DOCUMENTOS TÉCNICOS DE INSTALACIONES EN LA EDIFICACIÓN - DTIE*, pages 29–48. ATECYR, 2010. ISBN 978-84-950 10-36-0.

[30] Xue Bin Peng and Michiel van de Panne. Learning Locomotion Skills Using DeepRL: Does the Choice of Action Space Matter? *Proceedings - SCA 2017: ACM SIGGRAPH / Eurographics Symposium on Computer Animation*, nov 2016. doi: 10.1145/3099564.3099567. URL http://arxiv.org/abs/1611.01055http://dx.doi.org/10.1145/3099564.3099567.

[31] PJ Lute and VAH Paassen. Predictive control of indoor temperatures in office buildings energy consumption and comfort. *Proc. CLIMA2000*, 2:290–295, 2000.

[32] C.G. Nesler. Adaptive control of thermal processes in buildings. *IEEE Control Systems Magazine*, 6(4):9–13, 1986. doi: 10.1109/MCS.1986.1105101. URL https://www.scopus.com/inward/record.uri?eid=2-s2.0-0022768312&doi=10.1109%2fMCS.1986.1105101&partnerID=40&md5=b065cf85c0c913f9447f7d3eba57525b. cited By 27.

[33] Mengjie Han, Xingxing Zhang, Liguo Xu, Ross May, Song Pan, Jinshun Wu, Hasan Fleyeh, and Xingxing Zhang a. A review of reinforcement learning methodologies on control systems for building energy. *Working papers in transport, tourism, information technology and microdata analysis*, pages 1–26, 2018. ISSN 1650-5581. URL http://www.diva-portal.org/smash/get/diva2:1221058/FULLTEXT01.pdf.

[34] Zhe Wang and Tianzhen Hong. Reinforcement Learning for Building Controls: The opportunities and challenges Energy Technologies Area. 2020. doi: 10.1016/j.apenergy.2020.115036.

[35] Sung Ku Heo, Ki Jeon Nam, Jorge Loy-Benitez, Qian Li, Seung Chul Lee, and Chang Kyoo Yoo. A deep reinforcement learning-based autonomous ventilation control system for smart indoor air quality management in a subway station. *Energy and Buildings*, 202:109440, nov 2019. ISSN 03787788. doi: 10.1016/j.enbuild.2019.109440.

[36] Liang Yu, Yi Sun, Zhanbo Xu, Chao Shen, Senior Member, Dong Yue, Tao Jiang, and Xiaohong Guan. Multi-Agent Deep Reinforcement Learning for HVAC Control in Commercial Buildings. Technical report, 2020.

[37] Silvio Brandi, Marco Savino Piscitelli, Marco Martellacci, and Alfonso Capozzoli. Deep reinforcement learning to optimise indoor temperature control and heating energy consumption in buildings. *Energy and Buildings*, 224:110225, oct 2020. ISSN 03787788. doi: 10.1016/j.enbuild.2020.110225.

[38] Donald Azuatalam, Wee-Lih Lee, Frits de Nijs, and Ariel Liebman. Reinforcement learning for whole-building HVAC control and demand response. *Energy and AI*, 2:100020, nov 2020. ISSN 26665468. doi: 10.1016/j.egyai.2020.100020.

[39] John Ellson, Emden Gansner, Lefteris Koutsofios, Stephen C North, and Gordon Woodhull. Graphviz—open source graph drawing tools. In *International Symposium on Graph Drawing*, pages 483–484. Springer, 2001.

[40] DesignBuilder Software Ltd. DesignBuilder Help. URL https://designbuilder.co.uk/helpv6.0/.

[41] Drury B Crawley, Linda K Lawrie, Frederick C Winkelmann, Walter F Buhl, Y Joe Huang, Curtis O Pedersen, Richard K Strand, Richard J Liesen, Daniel E Fisher, Michael J Witte, et al. Energyplus: creating a new-generation building energy simulation program. *Energy and buildings*, 33(4):319–331, 2001.

[42] Big Ladder Software LLC. EnergyPlus Version 8.9 Documents, 2018. URL https://bigladdersoftware.com/epx/docs/8-9/index.html.

[43] Big Ladder Software LLC. Engineering reference on the Airflow Network model, 2018. URL https://bigladdersoftware.com/epx/docs/8-9/engineering-reference/airflownetwork-model.html#airflownetwork-model.

[44] Functional Mock-up Interface. URL https://fmi-standard.org/.

[45] Thierry Nouidui, Michael Wetter, and Wangda Zuo. Functional mock-up unit for co-simulation import in EnergyPlus, 2014. ISSN 19401493.

[46] Michael Wetter, Thierry S Nouidui, David Lorenzetti, Edward A Lee, and Amir Roth. PROTOTYPING THE NEXT GENERATION ENERGYPLUS SIMULATION ENGINE. In *Proceedings of the 14th International Conference of the International Building Performance Simulation Association (BS 2015)*, Hyderabad, India, 2015.

[47] Richard Bellman. A markovian decision process. *Journal of mathematics and mechanics*, 6 (5):679–684, 1957.

[48] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004. ISBN 9780521833783. URL http://www.cambridge.org.

[49] Tanmay Gangwani, Dawei Li, and Zikun Ye. Lecture 16: Value Iteration, Policy Iteration and Policy Gradient. Technical report, 2019.

[50] Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Mach. Learn.*, 8(3–4):229–256, May 1992. ISSN 0885-6125. doi: 10.1007/BF00992696. URL https://doi.org/10.1007/BF00992696.

[51] Richard S Sutton and Andrew G Barto. *Reinforcement Learning: An Introduction*. The MIT Press, second edition, 2015.

[52] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal Policy Optimization Algorithms. jul 2017. URL http://arxiv.org/abs/1707.06347.

[53] John Schulman, Philipp Moritz, Sergey Levine, Michael I. Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. In *4th International Conference on Learning Representations, ICLR 2016 - Conference Track Proceedings*. International Conference on Learning Representations, ICLR, jun 2016. URL https://sites.google.com/site/gaepapersupp.

[54] Christopher M. Bishop. Chapter 11: Sampling methods. In *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, Berlin, Heidelberg, 2006. ISBN 0387310738.

[55] Nicolas Heess, Dhruva TB, Srinivasan Sriram, Jay Lemmon, Josh Merel, Greg Wayne, Yuval Tassa, Tom Erez, Ziyu Wang, S. M. Ali Eslami, Martin Riedmiller, and David Silver. Emergence of Locomotion Behaviours in Rich Environments. jul 2017. URL http://arxiv.org/abs/1707.02286.

[56] Xue Bin Peng, Pieter Abbeel, Sergey Levine, and Michiel van de Panne. DeepMimic: Example-Guided Deep Reinforcement Learning of Physics-Based Character Skills. *ACM Transactions on Graphics*, 37(4):18, apr 2018. doi: 10.1145/3197517.3201311. URL http://arxiv.org/abs/1804.02717http://dx.doi.org/10.1145/3197517.3201311.

[57] Kurt Hornik. Approximation capabilities of multilayer feedforward networks. *Neural Networks*, 4(2):251–257, jan 1991. ISSN 08936080. doi: 10.1016/0893-6080(91)90009-T.

[58] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. Deep Feedforward Networks. Goodfellow et al. [80], chapter 6, pages 197–217.

[59] Yann A LeCun, Léon Bottou, Genevieve B Orr, and Klaus-Robert Müller. Efficient backprop. In *Neural networks: Tricks of the trade*, pages 9–48. Springer, 2012.

[60] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. Optimization for Training Deep Models. Goodfellow et al. [80], chapter 8, pages 197–217.

[61] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. OpenAI Gym. jun 2016. URL http://arxiv.org/abs/1606.01540.

[62] Martín Abadi. Tensorflow: learning functions at scale. In *Proceedings of the 21st ACM SIGPLAN International Conference on Functional Programming*, pages 1–1, 2016.

[63] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *arXiv preprint arXiv:1912.01703*, 2019.

[64] The Theano Development Team, Rami Al-Rfou, Guillaume Alain, Amjad Almahairi, Christof Angermueller, Dzmitry Bahdanau, Nicolas Ballas, Frédéric Bastien, Justin Bayer, Anatoly Belikov, et al. Theano: A python framework for fast computation of mathematical expressions. *arXiv preprint arXiv:1605.02688*, 2016.

[65] Frank Seide and Amit Agarwal. Cntk: Microsoft's open-source deep-learning toolkit. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 2135–2135, 2016.

[66] Jojo Moolayil and Suresh John. *Learn Keras for Deep Neural Networks*. Springer, 2019.

[67] Eric Liang, Richard Liaw, Robert Nishihara, Philipp Moritz, Roy Fox, Ken Goldberg, Joseph Gonzalez, Michael Jordan, and Ion Stoica. Rllib: Abstractions for distributed reinforcement learning. In *International Conference on Machine Learning*, pages 3053–3062. PMLR, 2018.

[68] Eric Liang and Richard Liaw. Scaling multi-agent reinforcement learning. *Berkeley Artificial Intelligence Research*, 2018.

[69] Anssi Kanervisto, Christian Scheller, and Ville Hautamäki. Action Space Shaping in Deep Reinforcement Learning. Technical report, 2020. URL `https://github.com/minerllabs/baselines/tree/master/general/chainerrl`.

[70] Sam Michael Devlin and Daniel Kudenko. Dynamic potential-based reward shaping. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems*, pages 433–440. IFAAMAS, 2012.

[71] American Society of Heating, Refrigerating and Air-Conditioning Engineers, Inc. Ventilation and Infiltration. In *2017 ASHRAE Handbook—Fundamentals* American Society of Heating, Refrigerating and Air-Conditioning Engineers, Inc. [11], chapter 16. ISBN 978-1-939200-58-7.

[72] InVentilate. Produktinformation MicroVent 2-8. Brochure, 2020. URL `https://inventilate.dk/wp-content/uploads/2021/02/MV-brochure_2020.pdf`.

[73] Anyscale. RLlib Algorithms — Proximal Policy Optimization, 2021. URL `https://docs.ray.io/en/master/rllib-algorithms.html#proximal-policy-optimization-ppo`.

[74] Till Tantau. The TikZ and PGF Packages Manual for version 3.1.9a. Technical report, Institut für Theoretische Informatik, Universität zu Lübeck, 2021. URL `https://github.com/pgf-tikz/pgf`.

[75] D.G. Shepherd. *Advances in Energy Systems and Technology*, volume 1. Elsevier, 1978. doi: 10.1016/c2013-0-06147-6.

[76] Big Ladder Software LLC. Engineering reference on the Shading module, 2018. URL `https://bigladdersoftware.com/epx/docs/8-9/engineering-reference/shading-module.html#shading-module`.

[77] Big Ladder Software LLC. Engineering reference on the Outside Surface Heat Balance, 2018. URL `https://bigladdersoftware.com/epx/docs/8-9/engineering-reference/outside-surface-heat-balance.html#outside-surface-heat-balance`.

[78] Babak Badnava and Nasser Mozayani. A new Potential-Based Reward Shaping for Reinforcement Learning Agent. feb 2019. URL `http://arxiv.org/abs/1902.06239`.

[79] Matplotlib: Visualization with python, 2021. URL `https://matplotlib.org/`.

[80] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. The MIT Press, 2010.

[81] Charles H. Forsberg. Chapter 1 - introduction to heat transfer. In Charles H. Forsberg, editor, *Heat Transfer Principles and Applications*, pages 1–21. Academic Press, 2021. ISBN 978-0-12-802296-2. doi: https://doi.org/10.1016/B978-0-12-802296-2.00001-9. URL https://www.sciencedirect.com/science/article/pii/B9780128022962000019.

[82] S. Chandrasekhar. Radiative transfer. *Quarterly Journal of the Royal Meteorological Society*, oct 1950. ISSN 00359009. doi: 10.1002/qj.49707633016. URL https://rmets.onlinelibrary.wiley.com/doi/full/10.1002/qj.49707633016.

[83] Código Técnico de la Edificación. Documento Básico de Ahorro de Energía. page 38.

[84] NFRC. Energy performance label, 2020. URL https://www.nfrc.org/energy-performance-label/.

[85] G. E. Myers. Long-Time solutions to heat-conduction transients with time-dependent inputs. *Journal of Heat Transfer*, 102(1):115–120, feb 1980. ISSN 15288943. doi: 10.1115/1.3244221.

[86] J E Seem. Modeling of heat transfer in buildings. 1 1987. URL https://www.osti.gov/biblio/6425796.

[87] Kunze Ouyang and Fariborz Haghighat. A procedure for calculating thermal response factors of multi-layer walls-State space method. *Building and Environment*, 26(2):173–177, jan 1991. ISSN 03601323. doi: 10.1016/0360-1323(91)90024-6.

[88] Tensorboard: Tensorflow's visualization toolkit, 2021. URL https://www.tensorflow.org/tensorboard/.

Appendix A

Heat transfer

In this chapter, the basic principles of the heat transfer will be introduced, always considering steady state. It is not the purpose of this master thesis to dive into the math behind the equations, nor provide out-of-the-box numerical methods, considering that simulation tools already implement them.

A.1 Conduction

Energy is transferred in solids, or static liquids/gases according to the Fourier's law [81], which is shown in Eq. (A.1) for unidirectional heat transfer through a layer of depth L of a material with constant conductivity k with temperatures T_1 and T_2 on each side. Q/A is the heat rate per unit of area.

$$\frac{Q}{A} = \frac{k}{L}(T_2 - T_1) \tag{A.1}$$

A.2 Convection

Convective transfer of energy occurs between the air and a surface at different temperatures, and it is maintained through the continuous motion of air, either forced or due to natural convection.

The natural convection arises because of differences in density between the air in contact with the surface (which is heated/cooled) and the cooler/warmer air of the surroundings. Without loss of generality, assuming that the surface is hotter than the air, the difference in densities induces a local depression near the surface so that cooler air is dragged in while the hotter air moves away, creating a circular motion [81].

The Newton's law of cooling is shown in Eq. (A.2). Q/A is the heat rate per unit of area exchanged between the air and a surface at temperature T, and h is the convective heat transfer coefficient, which in practice is determined experimentally.

$$\frac{Q}{A} = h_{conv}(T_{air} - T) \tag{A.2}$$

A.3 Radiation

According to the Planck's law of black-body radiation [82], bodies exchange heat at their given temperatures by emitting radiation in all the spectrum, and the peak band depends only on the body temperature for ideal 100% absorptive bodies. However, real bodies don't have an absorptivity of 100%, but rather the equation (A.3) applies to their surface: ρ is the reflectivity,

α is the absorptivity and τ is the transmissivity. The latter can be considered 0 for opaque surfaces, also called grey bodies.

$$\rho + \alpha + \tau = 1 \tag{A.3}$$

Here it is interesting to mention that for windows, a relevant parameter is the **g-value** or **solar factor** [83], which represents how much of the normally incident solar radiation will end up being transmitted to the interior, either directly transmitted (τ) or re-emitted (a fraction of α). It is measured in summer conditions and it is used as a reference for how good a glazing is at solar control.

The Stefan-Boltzmann equation for the power absorbed by surface S out of the power emitted by surface S_{other} is shown in Eq. (A.4), where Q/A is the heat rate per unit of area absorbed, S is at temperature T (in K) and has emissivity ϵ, the surface S_{other} is at temperature T_{other}, and F_{other} is the view factor of surface S with respect to S_{other}, with σ being the Stefan-Boltzmann constant. According to Kirchhoff's law, the emissivity ϵ equals the absorptivity of the real body. The equation can be linearized using the definition in Eq. (A.5), obtaining a similar expression to the convection case, Eq. (A.6).

The view factor represents, in a scale from 0 to 1, how much of the source heat transfer the target is receiving (or viewing). It depends exclusively on the geometry and orientation of the pair of surfaces.

$$\frac{Q}{A} = \epsilon\sigma F_{other}(T_{other}^4 - T^4) \tag{A.4}$$

$$h_{rad} = \epsilon\sigma F_{other}(T_{other} + T)(T_{other}^2 + T^2) \tag{A.5}$$

$$\frac{Q}{A} = h_{rad}(T_{other} - T) \tag{A.6}$$

A.4 Combined action

Using the previous principles of transfer, it is interesting to note that an analogy with the Ohm's law can be established, where temperature differences are equivalent to a voltage, Q/A would be the intensity, and the heat transfer coefficients become the electrical conductance (inverse of the resistance).

Thus, in steady state (no heat accumulation) a wall with different layers of width L_i and conductivities k_i can be summarized as an equivalent wall of length ΣL_i and conductivity k, with $\frac{1}{k} = \Sigma\frac{1}{k}_i$.

In addition, a global heat transfer coefficient U can be found to summarize convection and radiation at both sides of a wall plus the conduction within the wall steps as follows in Eq. (A.7). The explanation is that the Ohm's law can be applied taking into account that convection and radiation happen in parallel, while they happen in series with respect to the conduction. This global heat transfer coefficient is also known as U-factor, and is used to describe heat conductivity in windows under winter conditions (it is considered that there is no accumulation of heat in the glass) [84].

$$\begin{aligned} h_1 &= h_{rad1} + h_{conv1} \\ h_2 &= h_{rad2} + h_{conv2} \\ \frac{1}{U} &= \frac{1}{h}_1 + \frac{1}{h}_2 + \frac{L}{k} \end{aligned} \tag{A.7}$$

More advanced calculations, outside the scope of this thesis, that take into account thermal mass of walls and that provide valid results for transient state are the Conduction Transfer Functions (CFTs) [85–87].

Appendix B

Training results

In this Appendix the charts representing the training process are presented, for the experiments from Sec. 5.3.

B.1 Results

In particular, the mean episodic reward shows reward's progress through the training process. It represents the average reward for each episode, and it is displayed for runs 1–20 (Fig. B.1), runs 21–40 (Fig. B.2), and runs 41–45 (Fig. B.3).

In addition, the metrics mentioned in Sec. 5.2 are also collected per zone during the training process for each episode, so that their evolution can be plotted and assessed in real time. An example, taken from the South zone, is shown for runs 1–20 (Fig. B.4), runs 21–40 (Fig. B.5), and runs 41–45 (Fig. B.6).

B.2 Discussion

To begin with, it needs to be clear that episodic rewards cannot be compared directly between different runs, other than to assess convergence. This is because the different runs differ in their set of weights $\mathcal{W}$, so their rewards will be different too. What is important is the metrics' result, which tells how well the controller performs in each of the defined areas, and allows a Pareto optimality comparison (see Sec. 4.6).

Based on the previous statement, timestep 150000 seems to be the earliest convergence point in the reward function for the controllers in runs 1–20 (Fig. B.1) and 21–40 (Fig. B.2). This is why these initial exploratory processes are stopped at that point.

A look at the training metrics, though, reaveals not all of them have converged (e.g. Figs. B.4a, B.5c). On the one hand, some variance is expected, because the training episodes are reset at their end and start randomly in any period of the year. However, an additional reason could be that the controller is trying different strategies that do not improve the reward. In any case, the comparison is fair in the sense that all of them have had the same training time.

The controllers from runs 41–45, which are based on the best weights from the previous experiments, have been trained for longer, to see if the rewards improve any further past the step 150000, and what happens to the metrics. Fig. B.3 shows a slight improvement of the reward functions, whereas Fig. B.6 confirms the tendency observed: the controllers are exploring different strategies with the goal of improving the reward.

From the training process only it is thus unclear whether a longer training time will improve the results or not. What can be extracted is that there are multiple ways to achieve the same reward, and the learning agents will try many of them in their attempt to improve the reward function.

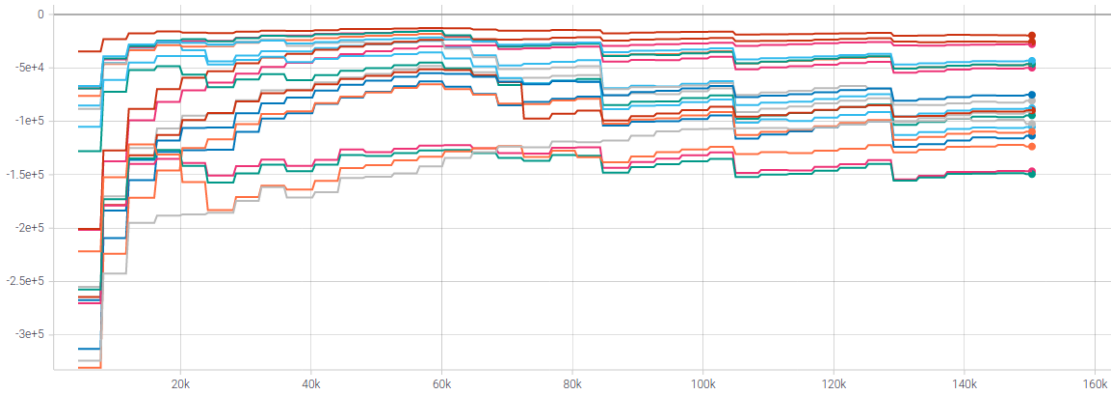


Figure B.1: Mean episodic reward during training for runs 1–20. X-axis represents the timesteps, and Y-axis the mean reward for each episode. By step 150000 it has already plateaued for all the runs. Source: training data, displayed with Tensorboard [88].

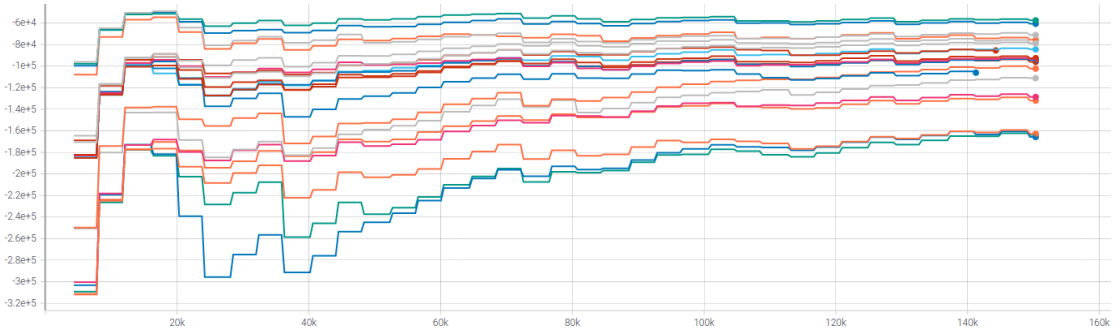


Figure B.2: Mean episodic reward during training for runs 21–40. X-axis represents the timesteps, and Y-axis the mean reward for each episode. By step 150000 it has already plateaued for all the runs. Source: training data, displayed with Tensorboard [88].

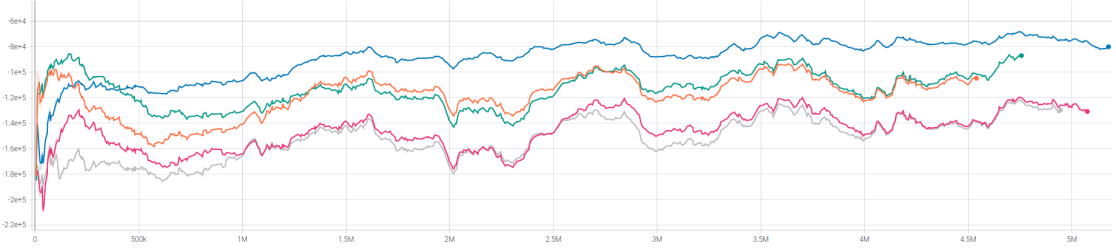
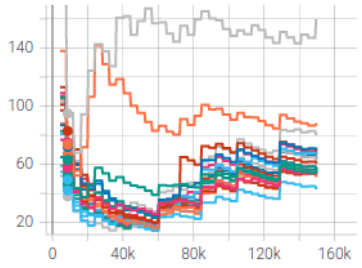
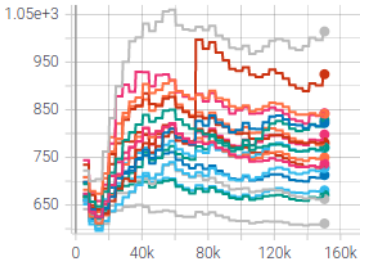


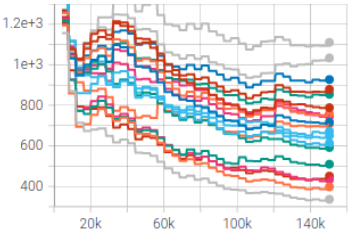
Figure B.3: Mean episodic reward during training for runs 41–45. X-axis represents the timesteps, and Y-axis the mean reward for each episode. Note that the training reaches $\approx$ 5M timesteps. Source: training data, displayed with Tensorboard [88].



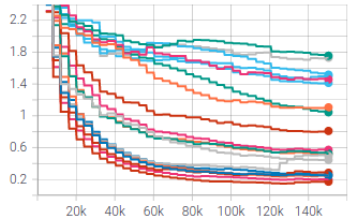
(a) Violations of 80% comfort, accumulated sum.



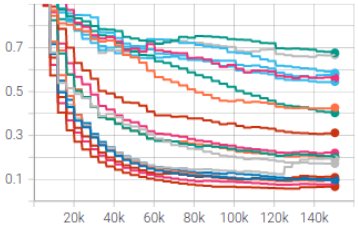
(b) Highest CO_2 level, in ppm.



(c) Heating energy, in kWh.

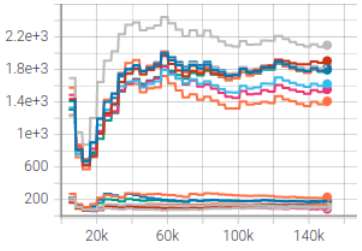


(d) Electrical energy consumed, in kWh.

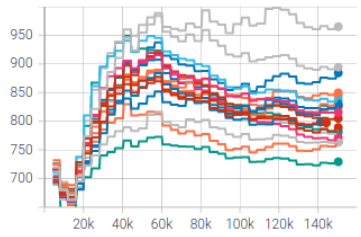


(e) Air Changes per Hour.

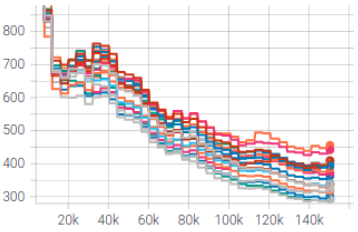
Figure B.4: Training metrics (Y-axes) as functions of the training step (X-axes) over each episode, for runs 1–20, South zone. Source: training data, displayed with Tensorboard [88].



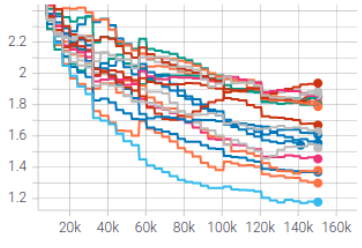
(a) Violations of 80% comfort, accumulated sum. Runs 21–30 show the greatest numbers.



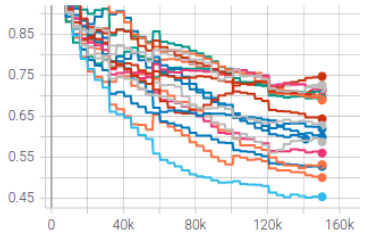
(b) Highest CO_2 level, in ppm.



(c) Heating energy, in kWh.

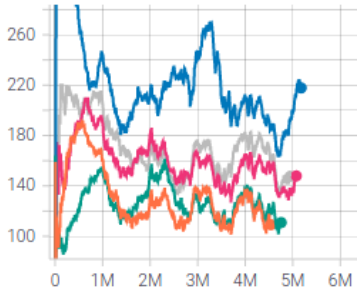


(d) Electrical energy consumed, in kWh.

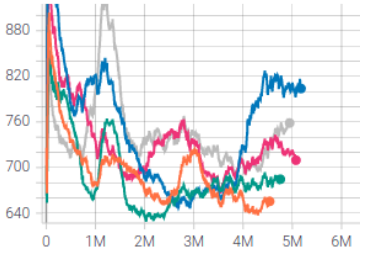


(e) Air Changes per Hour.

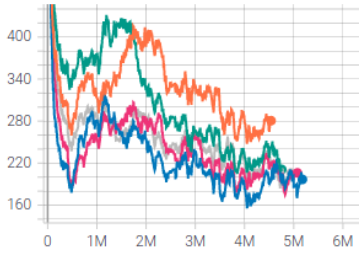
Figure B.5: Training metrics (Y-axes) as functions of the training step (X-axes) over each episode, for runs 21–40, South zone. Source: training data, displayed with Tensorboard [88].



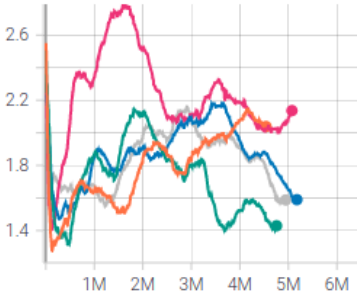
(a) Violations of 80% comfort, accumulated sum.



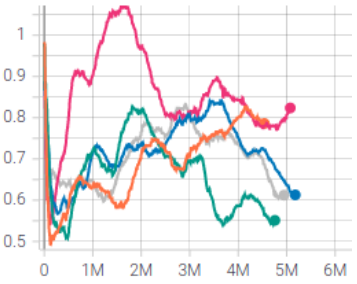
(b) Highest CO_2 level, in ppm.



(c) Heating energy, in kWh.



(d) Electrical energy consumed, in kWh.



(e) Air Changes per Hour.

Figure B.6: Training metrics (Y-axes) as functions of the training step (X-axes) over each episode, for runs 41–45, South zone. Source: training data, displayed with Tensorboard [88].

Appendix C

Drawings

In this Appendix the floor plans of the building are presented (Drawings C.1–C.4), as well as two explanatory sections (Drawing C.5) and a detail view of the solar shading elements (Drawing C.6). This is the list:

C1. Basement floor plan

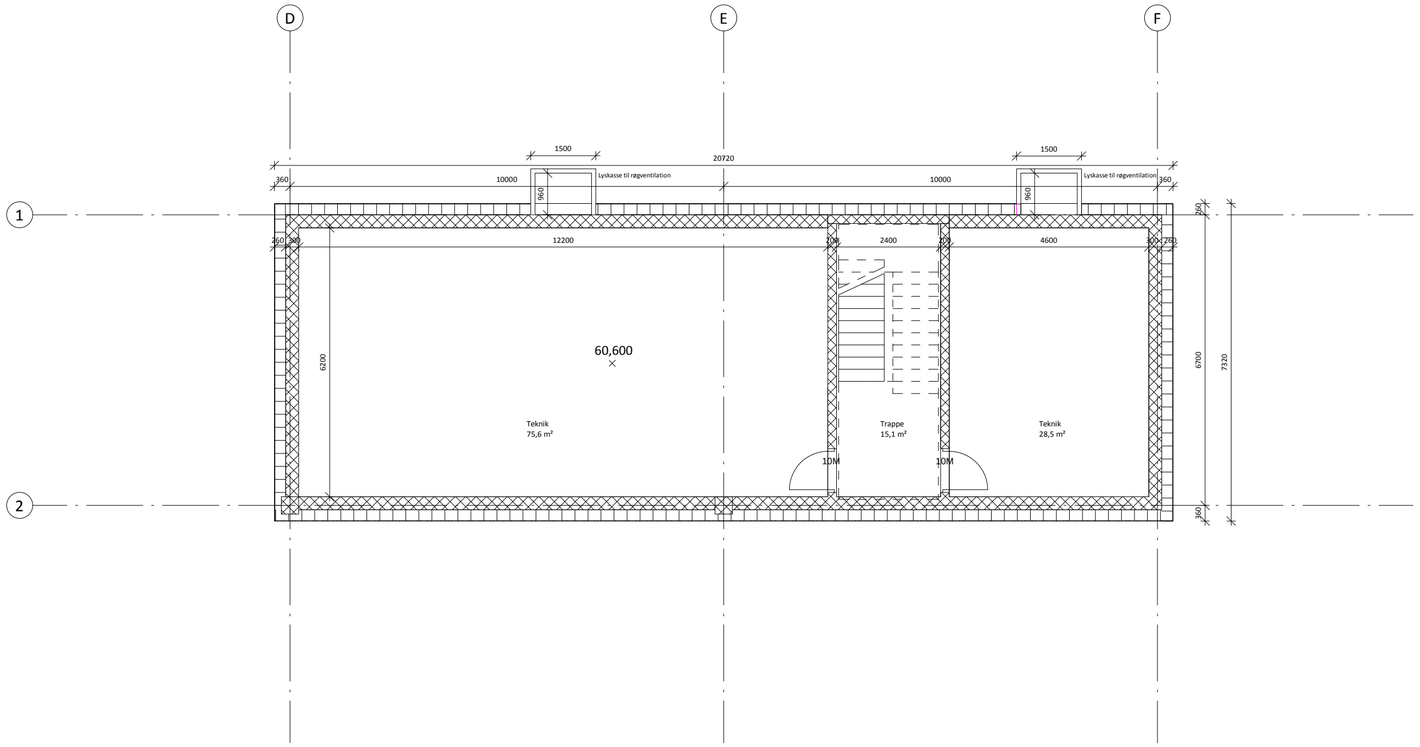
C2. Ground floor plan

C3. First and second floors plan

C4. Attic floor plan

C5. Explanatory section

C6. Construction detail view of the solar shading



Kælder
1 : 100

Signaturforklaring

Rum henvisning
Kontor
50 m²

Rum beskrivelse
Areal

Dør henvisning
10M Dør bredde modul

Venderadius i HC toilet, Ø 1500 mm

Bygningsdelsbeskrivelse

Note:
Der henviser generelt til brandstrategi for brandkrav,
til lydnotat for lydkrav og til energiramme for U-værdier.

Terrændæk:
Gulvbelægning og gulvopbygning
100 mm beton
400 mm trykfast isolering

Kælder ydervægge:
Geotekstil
250 mm Isolering
300 mm beton væg

Tunge ydervægge:
3 mm metalfacadeplade på hatprofiler/
20 mm træbeklædning på afstandslister
Vindspærre klasse 1 beklædning
Præfab. træskelet element:
295 mm træskelet, udfyldt med mineraluldsisolering
200 mm beton væg

Let ydervægge:
3 mm metalfacadeplade på hatprofiler/
20 mm træbeklædning på afstandslister
Vindspærre klasse 1 beklædning
Præfab. træskelet element:
295 mm træskelet, udfyldt med mineraluldsisolering
Dampspærre/dampbremse
2 x 15,5 mm brandbeskyttelse lag iht. brandstrategi
Indvendigbeklædning (Træ/gips/cementspånplader mv.)

Let indvendigvægge:
Træskeletvægge
Evt. 60 minutter brandbeskyttelselag iht. brandstrategi
Indvendigbeklædning (Træ/gips/cementspånplader mv.)

Etagedæk over det fri:
25 mm trægulv på strø og opklodsning
250 mm isolering
220 mm betondæk
100 mm vindtæt mineraluldsisolering
Evt. nedhængtloft og ophængningsssystem i klasse A materiale, f.eks. træbeton

Etagedæk:
25 mm trægulv
30 mm gulvgips
30 mm blød træfiberplade
15 mm trykfordelingsplade, som træfiberplade
350 mm Præfab. trædæk element: iht lyd og brand

Tag:
Over og under pap som SBS
360 mm gens. mineraluldsisolering
- Faldopbygning 1:40 og modfaldskiler
Interimsluknin, tagpap som SBS
22 mm vandfast tag krydsfiner
350 mm Præfab. trædæk element: iht lyd og brand

Tagterrasse:
22 mm terrassebrædder
145 mm træ strøer c/c 600 mm på terrasse fødder
Over og under pap som SBS
360 mm gens. mineraluldsisolering
- Faldopbygning 1:40 og modfaldskiler
Interimsluknin, tagpap som SBS
22 mm vandfast tag krydsfiner
350 mm Præfab. trædæk element: iht lyd og brand

TEGN.NR:
A1.205

TRIFORK Smart Building - Dyssen

Bygherre:	Trifork A/S	Borgergade 24B	1300 København K	Tlf: 43 24 12 12	E-mail: info@trifork.com

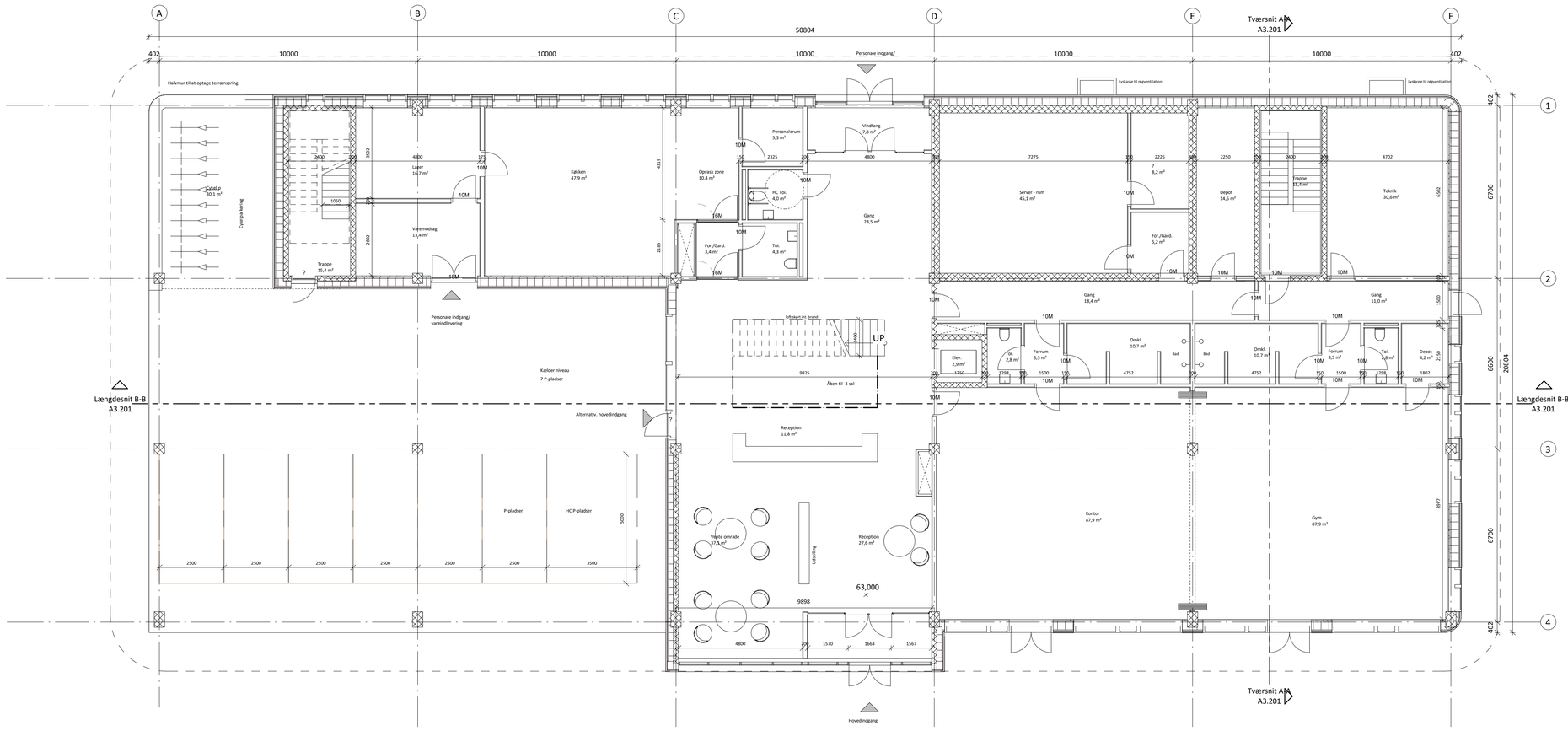
FASE: Myndighedsprojekt
EMNE: Kælder

AART / architects

TEGN.NR: A1.205 REV:

SAGS.NR.: 2019068 DATO: 10/08/20 MÅL: 1 : 100 INIT: Author KONTR: Checker GODK:Approver

● Arkitekt:	**Aart Architects**	Mariane Thomsens Gade 1C, 9. sal	8000 Aarhus C	Tlf: 87 30 32 86	E-mail: aart@aart.dk
○ Ingeniør:	Arne Elkjær a/s	Bredskifte Allé 7	8210 Aarhus V	Tlf: 86 16 47 55	E-mail: post@arneelkjaer.dk



TEGN.NR:

A1.200

Signaturforklaring

Rum henvisning
Kontor
50 m²

Rum beskrivelse
Areal

Dør henvisning
10M Dør bredde modul

Venderadius i HC toilet, Ø 1500 mm

Bygningsdelsbeskrivelse

Note:
Der henviser generelt til brandstrategi for brandkrav,
til lydnotat for lydkrav og til energiramme for U-værdier.

Terrændæk:
Gulvbelægning og gulvopbygning
100 mm beton
400 mm trykfast isolering

Kælder ydervægge:
Geotekstil
250 mm isolering
300 mm beton væg

Tunge ydervægge:
3 mm metalfacadeplade på hatprofiler/
20 mm træbeklædning på afstandslister
Vindspærre klasse 1 beklædning
Præfab. træskelet element:
295 mm træskelet, udfyldt med mineraluldsisolering
200 mm beton væg

Let ydervægge:
3 mm metalfacadeplade på hatprofiler/
20 mm træbeklædning på afstandslister
Vindspærre klasse 1 beklædning
Præfab. træskelet element:
295 mm træskelet, udfyldt med mineraluldsisolering
Dampspærre/dampbremse
2 x 15,5 mm brandbeskyttelse lag iht. brandstrategi
Indvendigbeklædning (Træ/gips/cementspånplader mv.)

Let indvendigvægge:
Træskeletvægge
Evt. 60 minutter brandbeskyttelselag iht. brandstrategi
Indvendigbeklædning (Træ/gips/cementspånplader mv.)

Etagedæk over det fri:
25 mm trægulv på strø og opklodsning
250 mm isolering
220 mm betondæk
100 mm vindtæt mineraluldsisolering
Evt. nedhængtloft og ophængningssystem i klasse A materiale, f.eks. træbeton

Etagedæk:
25 mm trægulv
30 mm gulvgips
30 mm blød træfiberplade
15 mm trykfordelingsplade, som træfiberplade
350 mm Præfab. trædæk element: iht lyd og brand

Tag:
Over og under pap som SBS
360 mm gens. mineraluldsisolering
- Faldopbygning 1:40 og modfaldskiler
Interimslukning, tagpap som SBS
22 mm vandfast tag krydsfiner
350 mm Præfab. trædæk element: iht lyd og brand

Tagterrasse:
22 mm terrassebrædder
145 mm træ strøer c/c 600 mm på terrasse fødder
Over og under pap som SBS
360 mm gens. mineraluldsisolering
- Faldopbygning 1:40 og modfaldskiler
Interimslukning, tagpap som SBS
22 mm vandfast tag krydsfiner
350 mm Præfab. trædæk element: iht lyd og brand

TRIFORK Smart Building - Dyssen

Bygherre:	Trifork A/S	Borgergade 24B	1300 København K	Tlf: 43 24 12 12	E-mail: info@trifork.com

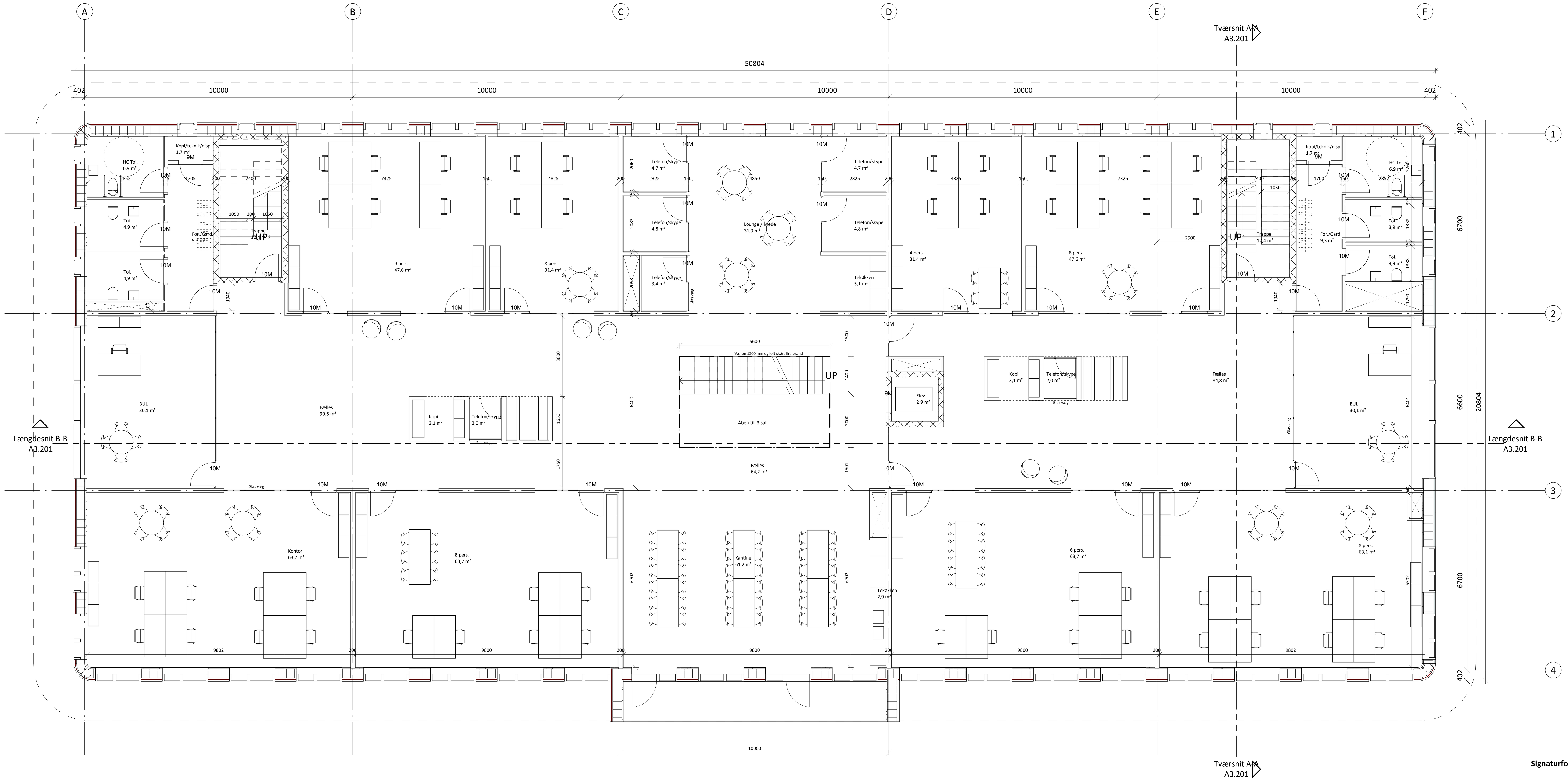
FASE: Myndighedsprojekt
EMNE: Stueplan

AART / architects

TEGN.NR: A1.200 REV:

SAGS.NR.: 2019068 DATO: 2020.08.28 MÅL: 1 : 100 INIT: CBT KONTR: RLA GODK:ATY

● Arkitekt: **Aart Architects** Mariane Thomsens Gade 1C, 9. sal 8000 Aarhus C Tlf: 87 30 32 86 E-mail: aart@aart.dk
○ Ingeniør: Arne Elkjær a/s Bredskifte Allé 7 8210 Aarhus V Tlf: 86 16 47 55 E-mail: post@arneelkjaer.dk



TEGN.NR:

A1.201

Signaturforklaring

Rum henvisning
Kontor — Rum beskrivelse
50 m² — Areal

Dør henvisning
10M — Dør bredde modul

Venderadius i HC toilet, Ø 1500 mm

Bygningsdelsbeskrivelse

Note:
Der henviser generelt til brandstrategi for brandkrav,
til lydnotat for lydkrav og til energiramme for U-værdier.

Terrændæk:
Gulvbelægning og gulvopbygning
100 mm beton
400 mm trykfast isolering

Kælder ydervægge:
Geotekstil
250 mm Isolering
300 mm beton væg

Tunge ydervægge:
3 mm metalfacadeplade på hatprofiler/
20 mm træbeklædning på afstandslister
Vindspærre klasse 1 beklædning
Præfab. træskelet element:
295 mm træskelet, udfyldt med mineraluldsisolering
200 mm beton væg

Let ydervægge:
3 mm metalfacadeplade på hatprofiler/
20 mm træbeklædning på afstandslister
Vindspærre klasse 1 beklædning
Præfab. træskelet element:
295 mm træskelet, udfyldt med mineraluldsisolering
Dampspærre/dampbremse
2 x 15,5 mm brandbeskyttelse lag iht. brandstrategi
Indvendigbeklædning (Træ/gips/cementspånplader mv.)

Let indvendigvægge:
Træskeletvægge
Evt. 60 minutter brandbeskyttelselag iht. brandstrategi
Indvendigbeklædning (Træ/gips/cementspånplader mv.)

Etagedæk over det fri:
25 mm trægulv på strø og opklodsning
250 mm isolering
220 mm betondæk
100 mm vindtæt mineraluldsisolering
Evt. nedhængtloft og ophængningsssystem i klasse A materiale, f.eks. træbeton

Etagedæk:
25 mm trægulv
30 mm gulvgips
30 mm blød træfiberplade
15 mm trykfordelingsplade, som træfiberplade
350 mm Præfab. trædæk element: iht lyd og brand

Tag:
Over og under pap som SBS
360 mm gens. mineraluldsisolering
- Faldopbygning 1:40 og modfaldskiler
Interimslukning, tagpap som SBS
22 mm vandfast tag krydsfiner
350 mm Præfab. trædæk element: iht lyd og brand

Tagterrasse:
22 mm terrassebrædder
145 mm træ strøer c/c 600 mm på terrasse fødder
Over og under pap som SBS
360 mm gens. mineraluldsisolering
- Faldopbygning 1:40 og modfaldskiler
Interimslukning, tagpap som SBS
22 mm vandfast tag krydsfiner
350 mm Præfab. trædæk element: iht lyd og brand

TRIFORK Smart Building - Dyssen

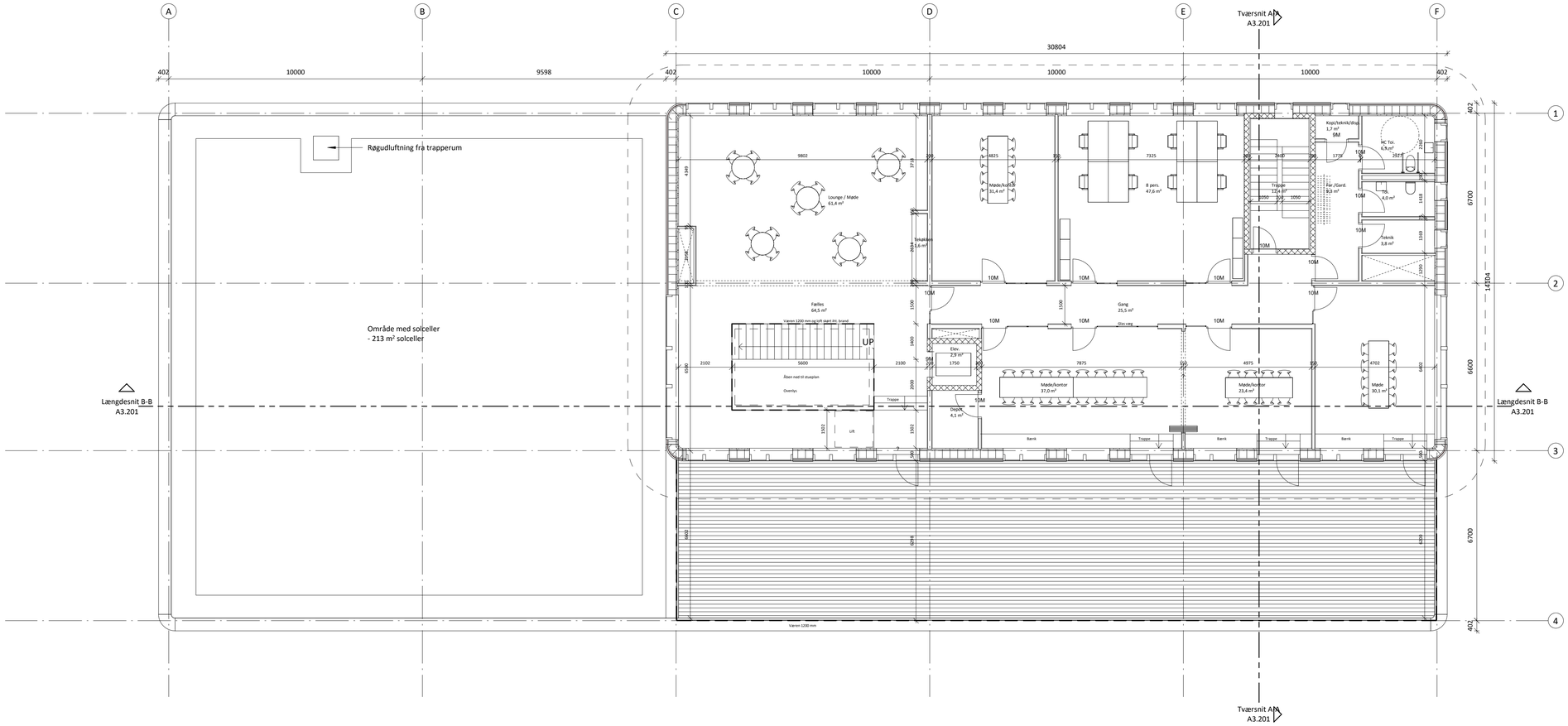
Bygherre: Trifork A/S — Borgergade 24B — 1300 København K — Tlf: 43 24 12 12 — E-mail: info@trifork.com

FASE: Myndighedsprojekt
EMNE: 1 salplan

TEGN.NR: A1.201 — REV:

SAGS.NR.: 2019068 — DATO: 2020.08.28 — MÅL: 1 : 100 — INIT: CBT — KONTR: RLA — GODK:ATY

● Arkitekt: **Aart Architects** — Mariane Thomsens Gade 1C, 9. sal — 8000 Aarhus C — Tlf: 87 30 32 86 — E-mail: aart@aart.dk
○ Ingeniør: Arne Elkjær a/s — Bredskifte Allé 7 — 8210 Aarhus V — Tlf: 86 16 47 55 — E-mail: post@arneelkjaer.dk



Tværsnit A-A
A3.201

TEGN.NR:

A1.203

Signaturforklaring

Rum henvisning
Kontor
50 m²
Rum beskrivelse
Areal

Dør henvisning
10M Dør bredde modul

Venderadius i HC toilet, Ø 1500 mm

Bygningsdelsbeskrivelse

Note:
Der henviser generelt til brandstrategi for brandkrav,
til lydnotat for lydkrav og til energiramme for U-værdier.

Terrændæk:
Gulvbelægning og gulvopbygning
100 mm beton
400 mm trykfast isolering

Kælder ydervægge:
Geotekstil
250 mm isolering
300 mm beton væg

Tunge ydervægge:
3 mm metalfacadeplade på hatprofiler/
20 mm træbeklædning på afstandslister
Vindspærre klasse 1 beklædning
Præfab. træskelet element:
295 mm træskelet, udfyldt med mineraluldsisolering
200 mm beton væg

Let ydervægge:
3 mm metalfacadeplade på hatprofiler/
20 mm træbeklædning på afstandslister
Vindspærre klasse 1 beklædning
Præfab. træskelet element:
295 mm træskelet, udfyldt med mineraluldsisolering
Dampspærre/dampbremse
2 x 15,5 mm brandbeskyttelse lag iht. brandstrategi
Indvendigbeklædning (Træ/gips/cementspånplader mv.)

Let indvendigvægge:
Træskeletvægge
Evt. 60 minutter brandbeskyttelselag iht. brandstrategi
Indvendigbeklædning (Træ/gips/cementspånplader mv.)

Etagedæk over det fri:
25 mm trægulv på strø og opklodsning
250 mm isolering
220 mm betondæk
100 mm vindtæt mineraluldsisolering
Evt. nedhængtloft og ophængningssystem i klasse A materiale, f.eks. træbeton

Etagedæk:
25 mm trægulv
30 mm gulvgips
30 mm blød trafiberplade
15 mm trykfordelingsplade, som trafiberplade
350 mm Præfab. trædæk element: iht lyd og brand

Tag:
Over og under pap som SBS
360 mm gens. mineraluldsisolering
- Faldopbygning 1:40 og modfaldskiler
Interimslukning, tagpap som SBS
22 mm vandfast tag krydsfiner
350 mm Præfab. trædæk element: iht lyd og brand

Tagterrasse:
22 mm terrassebrædder
145 mm træ strøer c/c 600 mm på terrasse fødder
Over og under pap som SBS
360 mm gens. mineraluldsisolering
- Faldopbygning 1:40 og modfaldskiler
Interimslukning, tagpap som SBS
22 mm vandfast tag krydsfiner
350 mm Præfab. trædæk element: iht lyd og brand

TRIFORK Smart Building - Dyssen

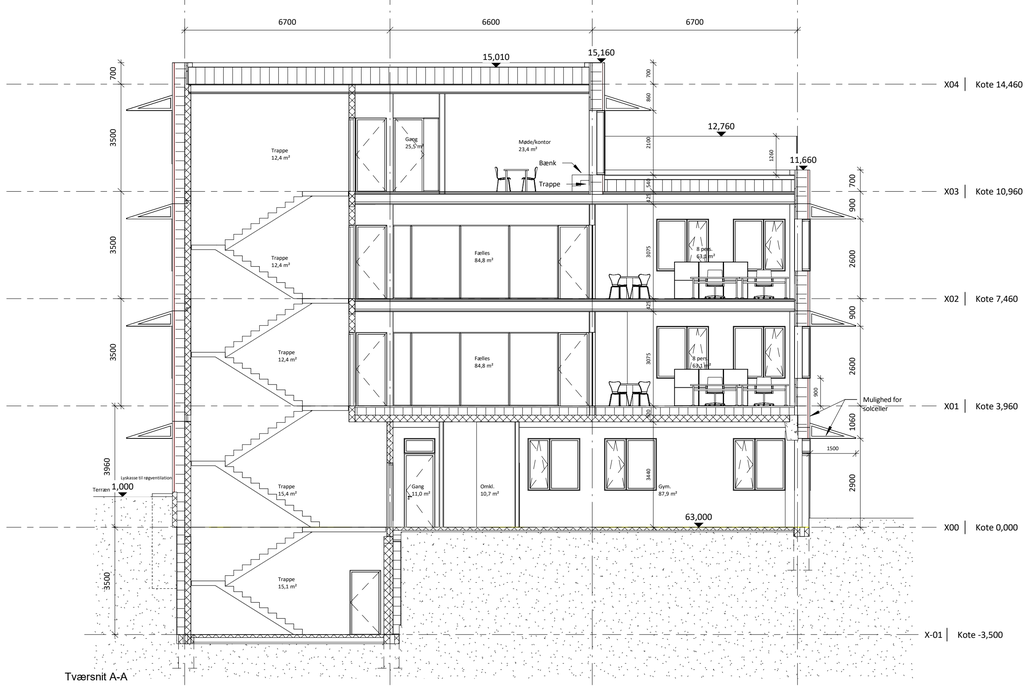
Bygherre:	Trifork A/S	Borgergade 24B	1300 København K	Tlf: 43 24 12 12	E-mail: info@trifork.com

FASE:	Myndighedsprojekt	AART / architects	TEGN.NR:	REV:
EMNE:	3 salsplan		A1.203	

SAGS.NR.: 2019068	DATO: 2020.08.28	MÅL: 1 : 100	INIT: CBT	KONTR: RLA	GODK:ATY

● Arkitekt: **Aart Architects**	Mariane Thomsens Gade 1C, 9. sal	8000 Aarhus C	Tlf: 87 30 32 86	E-mail: aart@aart.dk
○ Ingeniør: Arne Elkjær a/s	Bredskifte Allé 7	8210 Aarhus V	Tlf: 86 16 47 55	E-mail: post@arneelkjaer.dk

Længdesnit B-B



Tværsnit A-A

TEGN.NR:

A3.201

Bygningsdelsbeskrivelse

Note:
Der henviser generelt til brandstrategi for brandkrav,
til lydnotat for lydkrav og til energiramme for U-værdier.

Terrændæk:
Gulvbelægning og gulvopbygning
100 mm beton
400 mm trykfast isolering

Kælder ydervægge:
Geotekstil
250 mm isolering
300 mm beton væg

Tunge ydervægge:
3 mm metalfacadeplade på hatprofiler/
20 mm træbeklædning på afstandslister
Vindspærre klasse 1 beklædning
Præfab. træskelet element:
295 mm træskelet, udfyldt med mineraluldsisolering
200 mm beton væg

Let ydervægge:
3 mm metalfacadeplade på hatprofiler/
20 mm træbeklædning på afstandslister
Vindspærre klasse 1 beklædning
Præfab. træskelet element:
295 mm træskelet, udfyldt med mineraluldsisolering
Dampspærre/dampbremse
2 x 15,5 mm brandbeskyttelse lag iht. brandstrategi
Indvendigbeklædning (Træ/gips/cementspånplader mv.)

Let indvendigvægge:
Træskeletvægge
Evt. 60 minutter brandbeskyttelselag iht. brandstrategi
Indvendigbeklædning (Træ/gips/cementspånplader mv.)

Etagedæk over det fri:
25 mm træguly på strø og opklodsning
250 mm isolering
220 mm betondæk
100 mm vindtæt mineraluldsisolering
Evt. nedhængtloft og ophængningssystem i klasse A materiale, f.eks. træbeton

Etagedæk:
25 mm træguly
30 mm gulvgips
30 mm blød træfiberplade
15 mm trykfordelingsplade, som træfiberplade
350 mm Præfab. trædæk element: iht lyd og brand

Tag:
Over og under pap som SBS
360 mm gens. mineraluldsisolering
- Faldopbygning 1:40 og modfaldskiler
Interimslukning, tagpap som SBS
22 mm vandfast tag krydsfiner
350 mm Præfab. trædæk element: iht lyd og brand

Tagterrasse:
22 mm terrassebrædder
145 mm træ strøer c/c 600 mm på terrasse fødder
Over og under pap som SBS
360 mm gens. mineraluldsisolering
- Faldopbygning 1:40 og modfaldskiler
Interimslukning, tagpap som SBS
22 mm vandfast tag krydsfiner
350 mm Præfab. trædæk element: iht lyd og brand

TRIFORK Smart Building - Dyssen

Bygherre:	Trifork A/S	Borgergade 24B	1300 København K	Tlf: 43 24 12 12	E-mail: info@trifork.com

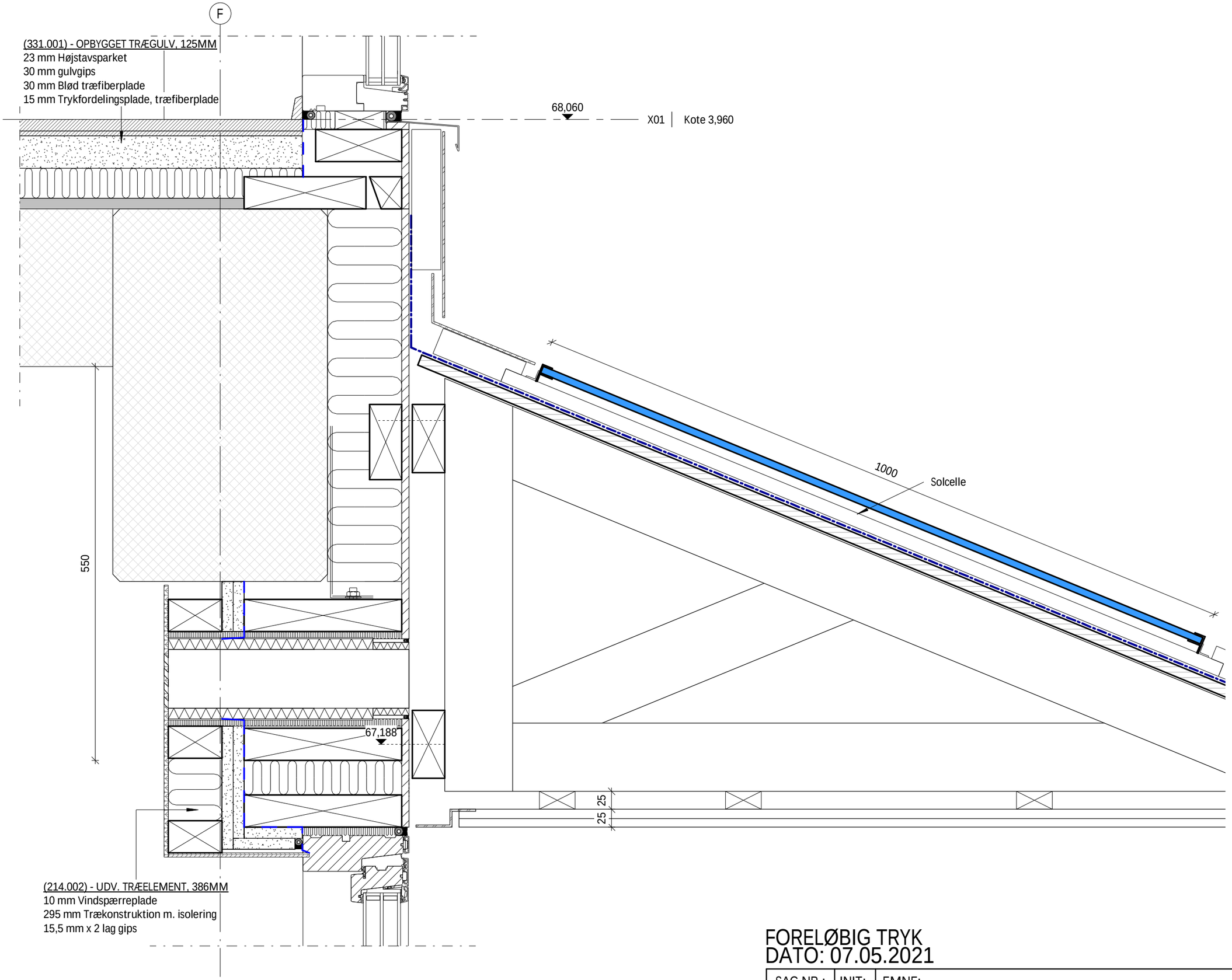
FASE: Myndighedsprojekt
EMNE: Længde og tværsnit

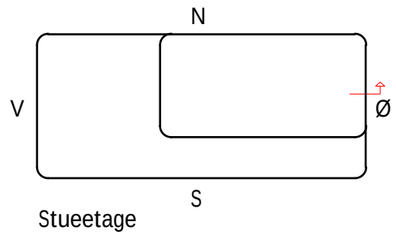
AART / architects

TEGN.NR: A3.201 REV:

SAGS.NR.: 2019068 DATO: 2020.08.28 MÅL: 1 : 100 INIT: CBT KONTR: RLA GODK:ATY

● Arkitekt: **Aart Architects** Mariane Thomsens Gade 1C, 9. sal 8000 Aarhus C Tlf: 87 30 32 86 E-mail: aart@aart.dk
○ Ingeniør: Arne Elkjær a/s Bredskifte Allé 7 8210 Aarhus V Tlf: 86 16 47 55 E-mail: post@arneelkjaer.dk





FORELØBIG TRYK
DATO: 07.05.2021

SAG NR.:	INIT:	EMNE:	MÅL:	TEGN. NR:	REV:
2019068	EBR	Detalje, lodret, Solafskærmning over stueetage, mikrovent/vindue - Gavl	1 : 5	A5.141	