Politecnico di Torino

ScuDo
Scuola di Dottorato ~ Doctoral School
WHAT YOU ARE, TAKES YOU FAR

Doctoral Dissertation
Doctoral Program in Energetics (34^{th}cycle)

Deep Reinforcement Learning-based Control Strategies for Enhancing Energy Management in HVAC Systems

By

Silvio Brandi

******

Supervisor(s):
Prof. Alfonso Capozzoli, Supervisor

Doctoral Examination Committee:
Asst. Prof. Adrian Chong, National University of Singapore
Prof. Vincenzo Corrado, Politecnico di Torino
Dr. Massimo Fiorentini, EMPA - Swiss Federal Laboratories for Materials Science and Technology
Prof. Antonio Rosato, Università degli studi di Napoli Federico II
Asst. Prof Walter Zhe Wang, The Hong Kong University of Science and Technology

Politecnico di Torino
2022

Declaration

I hereby declare that, the contents and organization of this dissertation constitute my own original work and does not compromise in any way the rights of third parties, including those relating to the security of personal data. Parts of this Ph.D. dissertation were also previously published in international Journals, also reported in Appendix A of this thesis.

The present Ph.D. scholarship at Politecnico di Torino was funded by Enerbrain s.r.l.

Silvio Brandi
2022

* This dissertation is presented in partial fulfillment of the requirements for **Ph.D. degree** in the Graduate School of Politecnico di Torino (ScuDo).

Acknowledgements

I would like to begin my thanks with the person without whom this journey would never have begun. I truly thank Prof. Alfonso Capozzoli for believing in me and in my potential, for always pushing me to do better and for the burning passion he put in trying to turn a young engineer into a scientific researcher.

I acknowledge Marco Savino Piscitelli for sharing this adventure with me side by side, for inspiring and supporting me.

My deepest and most sincere thanks to all the members of the BAEDA lab, you are fantastic colleagues but most of all good friends (Yes Giuseppe, including you!).

I want to thank my parents, all things considered you did a good job raising me. I couldn't have had better than you.

Lastly I want to thank Giulia, for reminding me that life is not only research and work but it is also and above all love.

Abstract

The widespread adoption of advanced metering infrastructures based on Internet of Things (IoT) could enable the development of Energy Management and Information Systems (EMIS) capable to leverage useful knowledge extracted from building related data. This dissertation focuses on a specific category of EMIS technologies called Automatic System Optimization (ASO). The purpose of ASO tools is to actively manage the control strategies responsible for the operations of building energy systems with the aim of enhancing energy usage. Among building subsystems, Heating Ventilation and Air Conditioning (HVAC) systems are rated among the most energy-intensive end-uses. The non-linear and stochastic nature of these systems makes the definition of robust and effective control strategies particularly challenging. In the current paradigm of smart buildings, building managers and owners can leverage ASO tools to automatically optimize the performance of their systems. However, the management of HVAC systems is mainly based on classical approaches characterized by different drawbacks including a reactive approach, lack of an optimization process and impossibility to handle multiple objectives at the same time. To overcome these limitations the application of advanced control strategies based on predictive and adaptive approaches represents a promising direction. In this dissertation four different applications of deep reinforcement learning based control strategies were conceived and tested. Deep Reinforcement Learning (DRL) is a model-free approach in which a control agent leveraging deep neural networks directly learns an optimal policy from interacting with the controlled environment. The developed applications were carried out in a co-simulation environment combining Python and EnergyPlus specifically developed in the context of this dissertation. Each application was designed to address different challenges and questions related to the application of DRL controllers to HVAC systems. In the first application, DRL is implemented to control the supply water temperature setpoint to terminal units of a heating system. The performance of the agent is evaluated against a reference

controller that implements a combination of rule-based and climatic-based logics. As a result, when the set of variables are adequately selected a heating energy saving ranging between 5% and 12% is obtained with an enhanced indoor temperature control with both static and dynamic deployment. In the second application a DRL agent was trained employing a data-driven model of the building dynamics. The trained agent was statically deployed on a calibrated Eplus model of the building to evaluate its performance. The agent was conceived to control the supply water temperature setpoint of the heating system of an office building achieving a reduction in the energy consumption of 18% while improving indoor temperature control of 5% with respect to a baseline rule-based controller. In the third application, was investigated the potentialities of DRL strategies for the management of integrated energy systems in buildings with on-site electricity generation and storage technologies. The controller is tested considering various configurations of battery energy storage system capacities, and thermal energy storage sizes. Results show that the proposed control strategy leads to a reduction of operational energy costs respect to a rule-based controller ranging from 39.5% and 84.3% among different configurations. The last application introduces a comparison between an online and offline DRL with a Model Predictive Control (MPC) architecture for energy management of a cold-water buffer tank linking an office building and a chiller subject to time-varying energy prices, with the objective of minimizing operating costs. Simulation results showed that the online-trained DRL agent, while requiring an initial 4 weeks adjustment period achieving a relatively poor performance (160% higher cost), it converged to a control policy almost as effective as the model-based strategies (3.6% higher cost in the last month). Findings and outcomes of the present research study are discussed providing a robust reasoning about the application of DRL control strategies to HVAC systems. Eventually, a wide overview on the lessons learned throughout this research study is proposed to outline the future opportunities and barriers to the adoption of advanced control strategies in the energy and building sector.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Building sector accounts for more than 40% of global energy consumption, playing a pivotal role in the energy transition and global warming mitigation processes [1]. Thanks to the complicity of incentive programs (such as *"20-20-20"*) [2], the progressive introduction of renewable energy sources and storage technologies to support the effort for decarbonisation raised several challenges for the definition of cost-effective energy management strategies in buildings.

In this context, the widespread adoption of Advanced Metering Infrastructures (AMI) based on Internet of Things (IoT) and Information Communication Technologies (ICT) could enable the development of Energy Management and Information Systems (EMIS) [3] capable to leverage useful knowledge extracted from building related data [4]. EMIS can be categorized into three main families: Energy Information Systems (EIS), Fault Detection and Diagnosis (FDD) systems and Automatic System Optimization (ASO) tools. EIS and FDD systems operate passively, employing data-driven and Artificial Intelligence (AI) techniques to provide actionable insights to the end users, highlighting anomalous energy behaviours and alerting about their potential causes [5]. The effectiveness of these tools strongly depends on the engagement and responsiveness of the users to rapidly act once information are delivered. On the other hand, ASO tools are designed to directly act on the control strategies responsible for the management of building energy systems, automatically enhancing their performance during operation.

Among building sub-systems, Heating Ventilation and Air Conditioning (HVAC) systems are rated among the most energy-intensive end-uses. In non-residential facil-

ities they account for more than an half of the energy demand of the whole building. The non-linear and stochastic nature of HVAC systems makes the definition of robust and effective control strategies particularly challenging. The main purpose of an HVAC system is to guarantee adequate levels of micro-climate conditions within a building. An advanced control strategy should seek to meet indoor environment requirements while maximizing, at the same time, different and often contrasting objectives such as the reduction of energy consumption and the minimization of energy-related costs. Moreover, the interaction of the building occupant and the influence of electrical grid requirements by means of Demand Response (DR) programs furtherly increase the complexity of the whole system.

In this perspective, the energy flexibility of building and HVAC systems has been recognized as a key resource to be exploited [6]. The flexibility has been defined as the ability to manage a building according to grid requirements, climate conditions and occupant needs [6, 7]. Advanced control strategies for HVAC systems should be capable to leverage building features (i.e. thermal mass) and equipment (i.e. renewable energy sources and storage solutions) to enhance the flexibility potential during operation while dynamically adapting to evolving conditions of external forcing variables (i.e. weather and electricity prices). Eventually, modern controls should be capable to take into account human feedback in their control logic [8].

Figure 1.1 shows the hierarchical structure of control strategies for HVAC systems. The first layer includes monitoring infrastructure which allows the collection of:

- **External disturbances**: comprise weather variables like air temperature, solar radiation, humidity and wind speed. Moreover, external disturbances may include external factors such as energy prices which strongly affect system performance.
- **Indoor variables**: comprise all the variables related to the quality of indoor environment such as temperature, humidity and pollutant concentrations. Moreover, indoor variables include information about occupant presence, behavior and, eventually, occupant feed-backs.
- **Plant variables**: comprise variables monitored on the different components of HVAC systems including temperatures, flow rates, pressures and energy consumption. Plant variables like energy consumption or power demand can

be collected at different levels of aggregation (i.e. component, system or whole building level).

The information collected by the monitoring infrastructure are forwarded to the different layers in order to enable the decision and automation process.

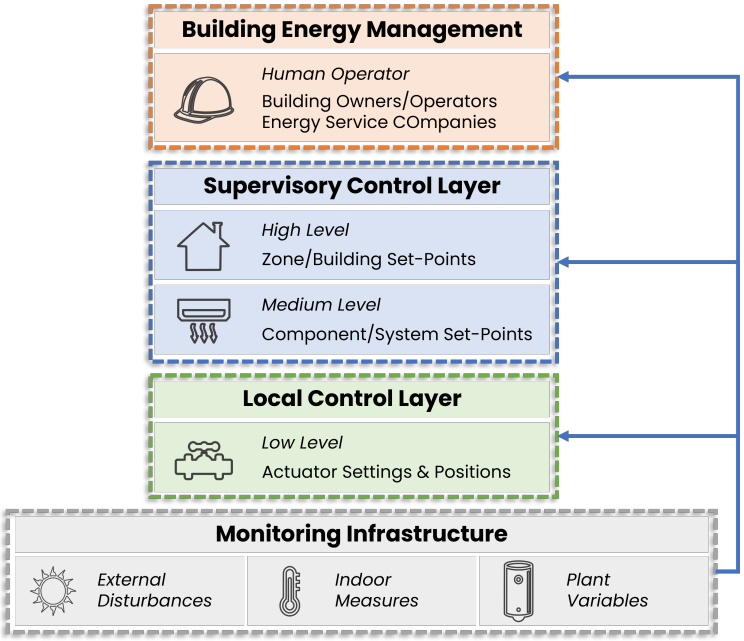


Fig. 1.1 Hierarchical structure of HVAC control.

The first layer placed at the bottom of the hierarchical structure is the local control layer. The aim of local controllers is to manage the positioning of low-level actuators responsible for the correct operation of the HVAC system. [9]. Examples of local controllers are the strategies employed to adjust fan or pump speeds and damper or valve positions to ensure that the heat carrier fluid (water or air) meet a desired set-point. Above local control layer is placed the supervisory control layer. The objective of supervisory controllers is to perform optimal management strategies fulfilling predefined goals defined at system or whole building level. Supervisory control can be performed at medium or high level and commonly involves the definition of operational set-points successively employed as a reference by low level controllers. Medium level strategies refers to supervisory controllers regulating set-points on component or system level. Typical implementations of medium level controllers are employed to manage settings of supply water/air temperatures and mass/volume flow rates. On the other hand, high level strategies are employed to

manage set-points (i.e. temperature and humidity) directly at zone or building level [8]. A substantial difference between local and supervisory controllers is the number on input on which these solutions base their decisions. While local controllers usually observe one input relative to a specific subsystem or process, supervisory controllers integrates a more comprehensive view of the HVAC system collecting information from multiple processes or subsystems. The final layer of the hierarchical structure of HVAC control showed in 1.1 is building energy management performed by an human operator (i.e. energy managers, energy service companies and building professionals). Building operators can design and tune local and supervisory control strategies based on their expertise and information opportunely gathered through the monitoring infrastructure.

Local controllers represent the first and essential layer of HVAC control systems. As a consequence, many efforts have been made in the previous years to develop cost-effective solutions for local control [10, 11], which have rapidly become the standard at industry level. On the other hand, the design of supervisory control strategies is a complex and time consuming task which is commonly performed directly by human operators based on domain expertise. This process often results in the implementation of standard and static rules systems based on typical schedules or operating patterns, without the support of technologies that could enable an automated, dynamic and optimized processing of this task.

In recent years, the development of advanced controllers based on forecasting and online analytics was supported by the recent advancements in Artificial Intelligence (AI), algorithm design and cloud computing technologies. However, the implementation of advanced control strategies for HVAC systems is still limited due to a distrust of building professionals and industry which prefer to stick with the application of more traditional solutions [12]. This is mainly due to the lack of guidelines and case studies which are capable to prove the effectiveness of advanced control strategies and to provide clear frameworks for their cost-effective implementation.

This dissertation aims at analyzing the potential benefits provided by the implementation of advanced control strategies to enhance the energy flexibility in HVAC systems. The main objective is to identify promising directions and potential barriers for the applicability in real-world context of these techniques. In particular, the developed controllers were mainly applied to a supervisory level. This choice was motivated considering different aspects. As previously introduced, supervisory con-

trol base their actions on multiple and heterogeneous inputs collected from different sources. As consequence, the adoption of AI-based algorithms capable to automatically process complex data can results in a consistent increase of the performance which could compensate the greater implementation effort and cost. Furthermore, the complexity and the potential failures brought by the adoption of advanced techniques can be handled more effectively in the supervisory layer through the implementation of safety constrains ensuring the operation of low level controllers.

1.1 Motivations of the research

In the current paradigm of smart buildings, building managers and owners can leverage ASO tools to automatically optimize the performance of their systems. However, the management of HVAC systems is mainly based on classical approaches such as Rule-Based Control (RBC) and Proportional-Integrative-Derivative (PID) control applied to both local and supervisory level. The main drawbacks of these strategies lie in their reactive approach, lack of an optimization process and impossibility to handle multiple objectives at the same time [12, 13]. These controllers are reactive since they act only on past observations of a controlled variable, adjusting the control signal to track a pre-defined set-point. Generally, the settings and parameters characterizing reactive controllers are not the result of an optimization process. As a consequence, the implemented control policy may achieve sub-optimal actions in the whole system perspective. Eventually, PID and RBC approaches usually do not comprises methods and processes to automatically adapt to evolving conditions of the forcing variables or to modifications in the controlled environment. Therefore, the performance of these controllers are strongly affected by the initial tuning conditions [14]. Moreover, the manual tuning of PID and RBC strategies based on domain expertise may be a laborious and cost-intensive task [6].

To overcome these limitations the application of advanced control strategies based on predictive and adaptive approaches has become an interesting topic to be explored by the current scientific literature. In the last few years, different classifications of HVAC control strategies have been published in numerous papers and textbooks [12, 13, 15, 16]. The two main typologies of advanced control approaches for HVAC systems can be identified as model-based and model-free methods which can be described as follow:

Model-based methods: the main components of model-based control systems are a model of the controlled environment and an optimizer as shown in Figure 1.2. The model of the system is employed along with forecast of external disturbances to predict the future evolution of the system given a certain control policy. The optimizer identifies the optimal policy given a certain objective function [17].

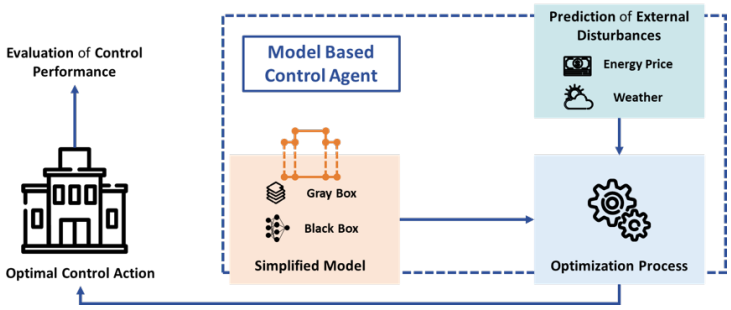


Fig. 1.2 Flowchart of a model-based control agent.

Model-free methods: model-free control methods do not require a model of the controlled environment as illustrated in Figure 1.3. Instead, they directly learns a near-optimal control policy directly interacting with the target system through a trial-and-error process [8] based on experience.

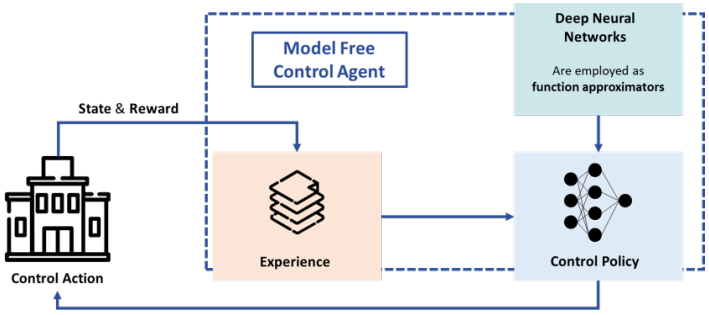


Fig. 1.3 Flowchart of a model-free control agent.

This classification may be not too general since the boundaries between the two approaches to advanced control are not perfectly clear. However, this classification can provide an helpful basis for identifying the advantages and disadvantages of the different frameworks. Different control strategies can be classified as hybrid methods employing a combination of model-based and model-free methods to achieve an optimal control policy. For example, adaptive controllers estimating unknown parameters in real time through a parameter estimator, which provides to

the controller the capability to adapt to time-varying disturbances and to account for uncertainty [18, 19].

Among model-based control methods, Model Predictive Control (MPC) aims at facing the main challenges of HVAC system control such as non-linear and time-varying dynamics and disturbances through an optimisation process performed over a receding time horizon [17, 20]. However, the complexity of the model chosen affects the type of optimization method that has to be employed to formulate the MPC controller, as well as the required computational time. Moreover, one major drawback of Model Predictive Control implementations is the labour intensive process necessary to build the model of the controlled system. This is particularly relevant for HVAC systems, since each building is a quite unique entity, the required control-oriented modelling of their envelope and energy systems is challenging, as the model built for one would most likely not fit another one directly. As a consequence, despite its robustness and advantages, MPC is still not widely adopted in the building industry [21].

Model-free control methods aims at overcoming the intrinsic limitations of model-based approaches. Reinforcement Learning (RL) is an interesting technique belonging to this family which popularity have rapidly grown in the last few years among HVAC control researchers. The RL framework is a branch of machine learning in which a control agent directly learns an optimal (or near-optimal) policy from its interactions with the environment through a delayed reward mechanism without any prior knowledge of the environment [22]. The growing interest in this technique was also supported by the evolution in the sector of artificial intelligence which offers a multitude of effective algorithms capable to automatically extract complex patterns from monitored data. In particular, a specific family of algorithms identified as Deep Reinforcement Learning (DRL), which employ Deep Neural Networks (DNN) as function approximators of the control policy, has been recently developed and applied to solve extremely complex control problems with nearly-human performances [23].

The ability to automatically improve system operations by considering multiple objectives and autonomously adapting to mutable conditions, while requiring minimal human intervention, is an highly desirable feature for HVAC controllers. In their formulations, reinforcement and deep reinforcement learning frameworks stand among the best candidates to fulfill these requirements. However, the exploration of

these control approaches is still in its infancy and effectiveness and limitations in energy and buildings applications need to be further explored.

Due to these challenging opportunities, advanced control strategies for HVAC systems based on model-free methods and, in particularly, reinforcement learning frameworks are investigated throughout this research study.

1.2 Research Outline

In order to demonstrate the capabilities of model-free frameworks applied to HVAC system control different case studies were investigated. As a preliminary step, an innovative co-simulation environment combining Pyhton and EnergyPlus was developed. This environment allows the user to simulate the effect on the building system of any controller (e.g. RBC, MPC, RL) without being limited by EnergyPlus capabilities. This environment was employed to deeply investigate the application of DRL-based control strategies to HVAC systems. Figure 1.4 shows the flowchart of a reinforcement learning control agent highlighting, in the side-panels, the different aspects investigated through this dissertation.

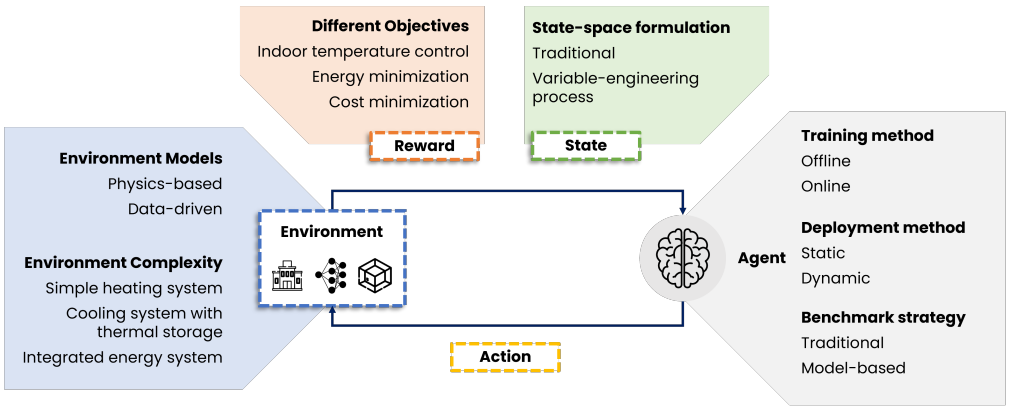


Fig. 1.4 Outline of the aspects investigated in the developed applications with reference to the reinforcement learning framework.

In the left panel the figure show the different features related to the controlled environment that were investigated in this dissertation. As reported, both physics-based and data-driven models of the controlled environment were tested. Moreover, different levels of complexity (ranging from a simple heating system to an integrated

energy system) were included in the investigation process. In the top-left panel the figure reports the control objectives (i.e. reward) considered in the developed applications. These objectives include optimization of indoor air temperature control, energy minimization and cost minimization. The top-right panel shows the analyzed features regarding the design of control inputs (i.e. state). A variable engineering process was designed and implemented to enhance adaptability properties of DRL controllers. This approach was tested against traditional frameworks in which variables are provided to DRL agents without performing pre-processing in advance. Eventually, the right panel lists the aspects related to the training, deployment and benchmark processes of DRL control agents that were analyzed through this dissertation.

These aspects were investigated through four main applications in which an DRL controller was conceived and tested. In particular, the developed case studies dealt with the following tasks related to the application of the reinforcement learning framework to HVAC system control:

- **Optimization of indoor temperature control and energy consumption in heating systems**. Water based heating system powered by gas fired boilers is a common configuration in the Italian building stock. This control problem is relatively simple as the only two features of the building that could be exploited for optimization purposes are the building thermal mass and the temperature acceptability range. Different sets of input variables (i.e. traditional and variable-engineering) along with different and deployment scenarios and methods (i.e. static and dynamic) were investigated and discussed to analyze the adaptability capabilities of the DRL controller. The development of this DRL controller is discussed in section 4.1.

- **Effective pretraining a of DRL agent by means of data-driven models to control HVAC systems in buildings.** DRL agents have to perform several interactions with the controlled environment before converging to the optimal control policy. In this context, it is common practice to pre-train a DRL agent offline in simulation environments based on engineering models of the real building. Nonetheless, the development of physics-based models requires a considerable effort beside an extensive domain expertise. Pre-training a DRL agent on a data-driven model of the building can overcome this issues. This training strategy is discussed in section 4.2.

- **Optimization of the management of integrated energy systems in buildings with Deep Reinforcement Learning**. The management of integrated energy systems in buildings is a challenging task that classical control approaches usually fail to address. In this context, DRL can achieve reduction of energy cost through a comprehensive view of the whole integrated energy system. The development of this controller is discussed in section 4.3.
- **Comparison of DRL with MPC for thermal energy management**. Although the scientific literature is particularly prolific with respect to applications of RL and DRL control techniques to HVAC system, the benefits brought by these solution are frequently presented with respect to traditional control techniques. In this dissertation a robust comparison between reinforcement learning and MPC controller was conceived and discussed in section 5.1 with the aim of analyzing strengths and weaknesses of the two approaches.

All the developed applications leveraged deep reinforcement learning frameworks based for the effective implementation of advanced control strategies in HVAC systems to analyze and discuss their effective adoption in the energy and building field.

To this purpose, the developed methodological frameworks were conceived following the perspective of an energy and building engineer providing more effort on the definition of the control problem and objectives rather than algorithmic features. In this way the result of the analysis can be translated into useful guidelines and case studies for future researchers and building professional aiming at increasing the performance of their system through the adoption of advanced control strategies.

1.3 Objectives of the thesis and novelty

As introduced in the previous sections reinforcement learning is a branch of machine learning which have proved to be very effective in solving various control problems. It owns interesting features, such as adaptability potential and self-learning properties implying minimal human intervention, making it suitable as advanced controller for HVAC systems. However, it presents some major drawbacks which are emphasized by the intrinsic slowness of the building dynamics. In particular, reinforcement learning algorithms require a considerable amount of time before converging to near

optimal solutions. These aspect make their deployment in real buildings extremely challenging. In this sense, the combination between building physics expertise and artificial intelligence can support the development of more cost-effective and robust DRL-based solutions. In this perspective the main objectives of the thesis can be summarized as follows:

- Demonstrate the necessity of evolving from traditional, reactive control approaches leveraging the opportunity provided by advanced control strategies based on predictive and adaptive paradigms. Beside introducing the benefits of advanced control strategies, this process aims to highlight the features of case studies and control problems for which the application of these techniques is more advantageous.
- Demonstrate the fundamental role of building physics expertise. Domain knowledge is a crucial aspect to consider for the definition of the control problem, the identification of the control objectives and the selection of the variables involved in the decision making process.
- Critically analyze the different development steps defining a reinforcement learning agent for HVAC system control. These steps comprise the correct tuning of hyper-parameters, the design of training strategies and the effective deployment of the controller.
- Address the need of defining proper benchmarks. Producing robust comparisons between model-free and model-based control strategies highlighting their relatives strengths and weaknesses in order to guide future researchers and practitioners to the approach best suited to their needs.
- Rationalize and discuss the concept and the meanings behind the model-free nature of reinforcement learning frameworks applied to HVAC systems control.

The main objective of this research study is to demonstrate the effectiveness of advanced model-free strategies applied to HVAC systems control. To this aim, different case studies were conceived and the performance of the proposed controllers were properly benchmarked against both traditional solutions and among each other. The novelty of this research work is not related to the development of novel control algorithms which are all taken from existing scientific literature, but it is associated

on how these approaches can be effectively implemented in HVAC systems from the perspective of energy engineers. In all the applications presented the domain expertise has been used as a reference to derive innovative approaches in the selection, design and implementation of advanced control strategies.

1.4 Organization of the thesis

The whole dissertation is divided into 6 chapters organized as shown in Figure 1.5. The main content of each chapter is summarized as follows.

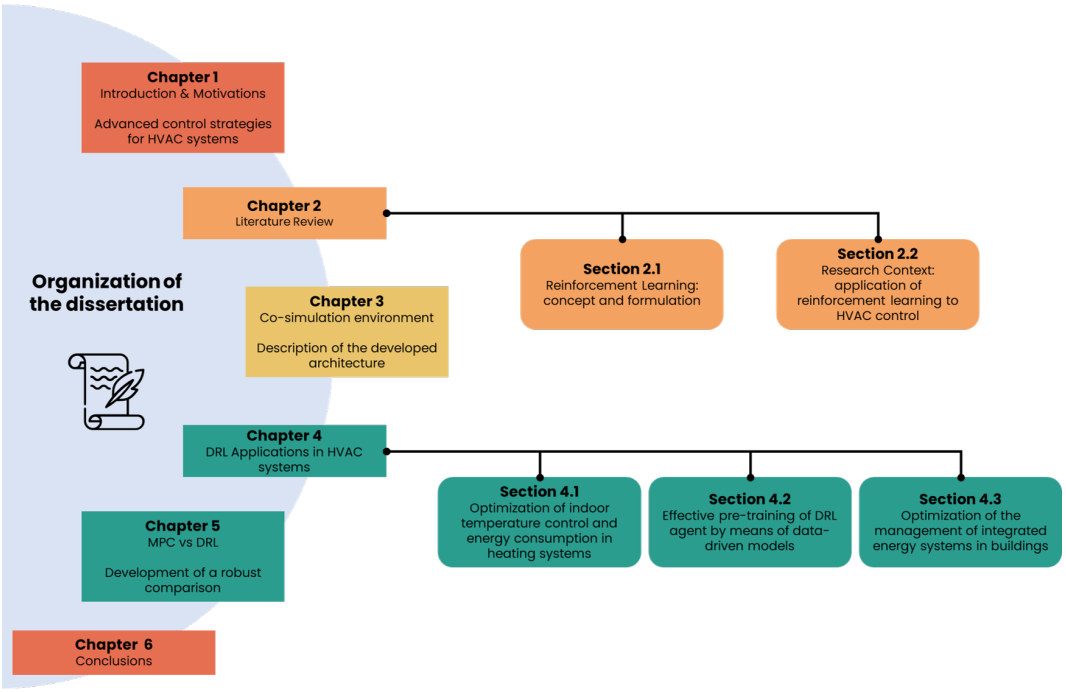


Fig. 1.5 Conceptual organization of the thesis.

Chapter 1 presents the motivation, the outline and the organization of the thesis.

Chapter 2 presents the literature review. The chapter is organized into two main sections. The section 2.1 introduces the reinforcement learning framework describing the main algorithms and approaches employed in this work. The section 2.2 reviews the applications of reinforcement learning control to HVAC systems.

Chapter 3 presents the architecture of the co-simulation environment developed in the context of this dissertation and employed to carry out the presented experiments.

Chapter 4 presents the developed applications of DRL control in HVAC systems. In particular section 4.1 presents and discusses the development of a DRL agent for controlling the heating system of an office building. Section 4.2 presents and discusses the applicability of data-driven models to pre-train a DRL control agent. Section 4.3 presents and discusses the application of DRL control to manage integrated energy systems in buildings.

Chapter 5 presents and discusses an application where DRL was tested and benchmarked against MPC.

Eventually Chapter 6 summarizes the work presented in this dissertation and gives an overview about opportunities and future research directions of reinforcement learning applied to HVAC system control.

Chapter 2

Literature Review

The scope of the present chapter is to investigate the findings achieved so far in the scientific literature about the use of reinforcement learning frameworks and their application to HVAC systems control. This chapter provides an extensive overview on reinforcement learning and deep reinforcement learning approaches applied in the context of building energy management. The chapter is organized in two main sections. On one hand, section 2.1 presents and discusses the main concepts behind reinforcement learning algorithms employed in this work. On the other hand, section 2.2 reviews all the applications of reinforcement learning for HVAC system control.

Portions of the present Chapter were already published in the following scientific papers:

- Brandi S., Piscitelli M.S., Martellacci M., Capozzoli A. 2020. *Deep reinforcement learning to optimise indoor temperature control and heating energy consumption in buildings*. Energy and Buildings 224, 110225. [24]
- Coraci D., Brandi S., Piscitelli M.S., Capozzoli A. 2021. *Online Implementation of a Soft Actor-Critic Agent to Enhance Indoor Temperature Control and Energy Efficiency in Buildings*. Energies 14, 997. [25]
- Brandi S., Gallo A., Capozzoli A. 2022. *A predictive and adaptive control strategy to optimize the management of integrated energy systems in buildings*. Energy Reports 8, pp: 1550-1567. [26]

- Brandi S., Fiorentini M., Capozzoli A. 2022. *Comparison of online and offline deep reinforcement learning with model predictive control for thermal energy management*. Automation in Construction 135, 104128. [27]

2.1 Reinforcement Learning: Concept and Formulation

Reinforcement Learning (RL) is a branch of machine learning conceived to solve control and sequential decision making processes. RL can be mathematically formalized as a Markov Decision Process (MDP) characterized by a 4-values tuple including [22]:

- **State (S)**: The state is a mathematical representation of the controlled environment which includes the set of features (defined as *observation*) that a control agent receives in order to determine a control action. If the observation is a subset of the state, this results in a Partially Observable Markov Decision Process (POMDP). For the sake o simplicity, in this dissertation the term state is adopted to indicate also observation since being a POMDP is quite a common feature among problems involving HVAC system control. In the context of HVAC system control typical examples of state variables are the indoor air temperature or the temperature of the outside environment.
- **Action (A)**: the action is the decision performed by the control agent. In the context of HVAC systems control the action could be represented by the set-point of supply water/air temperatures or pump/fan speeds.
- **Reward (R)**: the reward is the feedback received by the control agent for taking a specific action a_t in certain state s_t. The reward is calculated through a function which depends by the objectives of the specific control problem. In the context of HVAC system control the reward could be represented by a combination between energy consumption and thermal comfort evaluation.
- **Transition Probabilities (P)**: the transition probabilities describe how the environment will evolve after taking action a_t at state s_t. In the context of HVAC systems control transition probabilities are generally unknown since this process will require the development of a detailed model of the controlled environment.

Figure 2.1 shows the flowchart of the RL framework. The figure depicts the interactions between the control agent and the controlled environment taking place

through the four elements of the MDP. In each interaction, the control agent observes the current state of the environment and picks a control action. The control action induces a change in the environment which moves towards a new state. The goodness of this change is evaluated through the reward which is successively forwarded to the control agent along with the information about the new state of the environment.

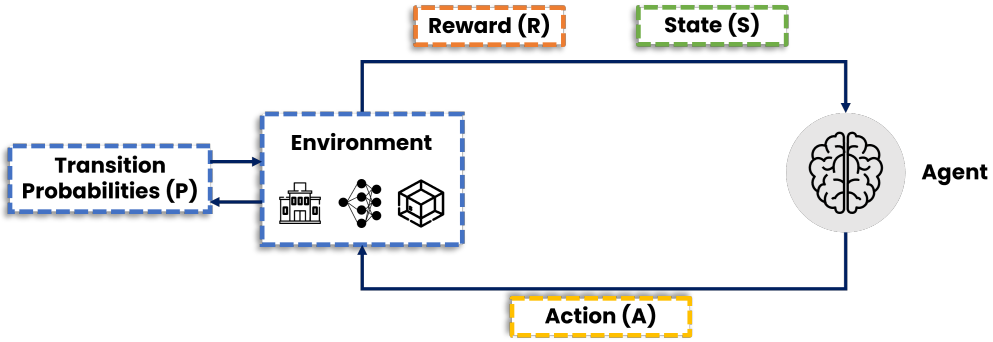


Fig. 2.1 Flowchart of the Reinforcement Learning framework.

In the reinforcement learning framework the control agent directly learns the optimal control policy (π) by interacting with the controlled environment through the previously described trial-and-error process. The policy represents a mapping between states and actions and is the core of a reinforcement learning agent [22]. The optimal control policy is the mapping between states and actions which maximizes the expected discounted return (i.e. cumulative discounted sum of future rewards). The *state-value function* evaluates the expected return obtained by the agent when starting from state s and following policy π:

$$v_\pi(s) = \sum_a \pi(a|s) \sum_{s',r} p(s',r|s,a)[r + \gamma v_\pi(s')] \tag{2.1}$$

where r is the immediate reward received by the agent while transitioning from state s to its successor state s' after taking action a. γ [0,1] is the discount factor for future rewards. An agent employing a discount factor equal to 1 will give greater importance to rewards that can be obtained in the future. Whereas, an agent implementing a discount factor of 0 will assign higher values to states that lead to high immediate rewards.

The optimal policy can be identified through different approaches. If the transition probabilities p and the rewards r are known, the solution can be found through direct approaches such as policy or value iteration [28]. However, this occurrence rarely arises for HVAC systems due to their intrinsic stochastic nature. RL can be applied also in the case in which the dynamics of the environment are unknown. In the model-based RL approach transition probabilities and rewards are firstly learnt by means of a model and then employed to learn the optimal control policy. In this formulation RL is very similar to MPC strategy since both approaches can make use of physics-based and data-driven methods [8, 17]. In the model-free RL approach the agent can learn the optimal control policy without explicitly identifying transition probabilities.

There are two methods in the model-free RL approach to identify the optimal control policy: *value-based* and *policy-based*. Value-based methods aim at learning the value function which estimates the goodness of taking a specific action a starting from state s. Policy-based methods do not employ the value function as a proxy and directly try to learn the optimal control policy π [29]. In general, value-based methods are more sample efficient while policy-based methods have better convergence properties and are capable to handle continuous problems characterized by high stochasticity.

Another aspect which characterizes RL algorithms is the difference between *on-policy* and *off-policy* methods. On-policy RL algorithms directly try to improve the policy that is used by the agent to generate decisions. Off-policy methods evaluate a policy that is different from the one used to select actions allowing them to learn from historical data and previous experience [22]. On-policy training is particularly challenging to be carried out in a real buildings since it is not feasible to let an RL agent to explore sub-optimal policies which may lead to undesired conditions. However, on-policy learning is much more effective in converging to the optimal solution since the state-action space can be better explored [8].

The scientific literature in this field is particularly prolific and innovative RL algorithms are constantly introduced. In the following subsections the algorithms employed in this work are briefly presented and described.

2.1.1 Q-learning

Q-learning is one of the most widely applied RL algorithms owing its popularity to its simplicity [30]. Q-learning is a value-based and off-policy method aiming at estimating *state-action values* (or Q-values) which represent the expected cumulative discounted reward obtained by the agent for taking action a while the environment is in state s [31]. In its most simple formulation Q-learning stores the Q-values into a tabular structure (Tabular Q-learning). The Q-values are constantly updated during agent training according to the following equation:

$$Q_{s,a} \leftarrow Q_{s,a} + \alpha[r(s,a) + \gamma max_a Q(s',a) - Q(s,a)] \tag{2.2}$$

Where α [0,1] is the learning rate which determines with which extension new knowledge overrides old knowledge. When α is equal to 1 new knowledge completely substitutes old knowledge, instead, when, α is set equal 0 no learning happens and new knowledge is not employed to update the control policy. The higher the estimation of the Q-value for a specific state-action tuple (s,a) the higher is the expected reward of the agent for taking that specific action a in the state s.

Figure 2.2 shows the flowchart of the framework of the Tabular Q-Learning algorithm. As previously introduced, Q values relative to state-action tuples are stored into tabular structures. At each interaction, the agent observes the actual state s of the environment and selects an action a based on the Q values stored in the table relative to the same state s. This action is forwarded to the environment which moves to a new state s' sending this information to the agent along with the reward signal r. These information are employed to update the Q value relative to state s and action a according to equation 2.2.

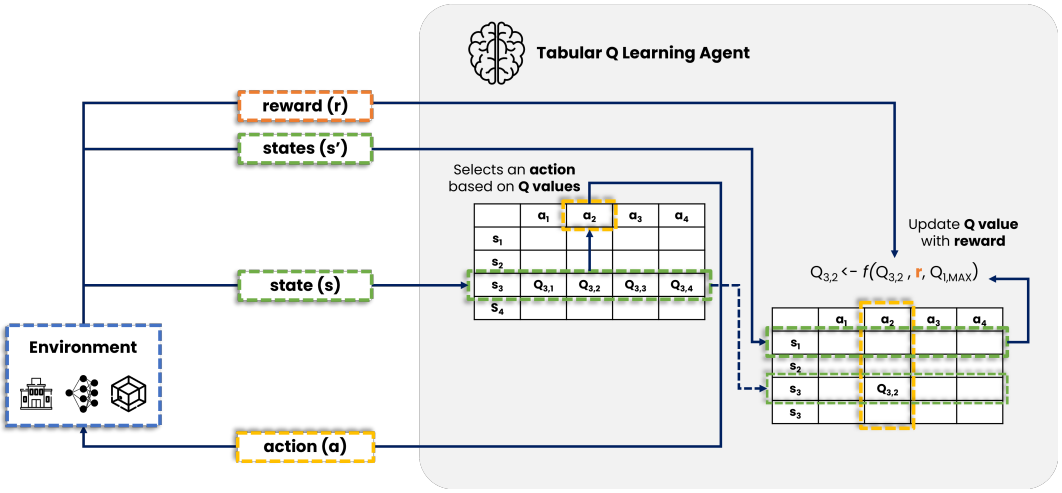


Fig. 2.2 Flowchart of Tabular Q-Learning framework.

A key-aspect to consider is the strategy employed to select the action. In this context, the identification of the optimal trade-off between exploration and exploitation in order to guarantee the convergence of the Q-values. To maximize the rewards stream, an agent must select actions previously tried that have been found to be effective in obtaining high rewards (exploitation). However, to identify such actions an agent is forced to pick actions never tried before (exploration). Two of the most frequently used methods to select actions balancing exploration and exploitation are the ε-greedy and the soft-max methods.

According to the ε-greedy method the agent acts greedy for most of the time favoring exploitation by selecting actions characterized by the highest Q-values given a certain state. The agent explores selecting a random action with a probability ε which is generally a small probability [22]. ε-greedy assigns equal probabilities to all non-optimal actions leading to poor results in some circumstances [32].

Contrarily to ε-greedy exploration in which all the actions are considered equal in the Soft-max exploration, also known as Boltzmann exploration, actions are picked according to Boltzmann distribution calculated as follows:

$$P_a = \frac{e^{Q_{s,a}/\tau}}{\sum_i e^{Q_{s,i}/\tau}} \tag{2.3}$$

Where τ is the Boltzmann temperature constant. Soft-max exploration method has shown different problems in converging to optimal control policy [32].

Eventually, the Max-Boltzmann exploration method combines the two previously mentioned approaches. According to this method, the agent acts almost deterministically when the estimations of the Q-values are not ambiguous (i.e. the Q-value associated with the best performing action significantly differs from the others), while it allows wider exploration in the region of the state-action space where the Q-values estimations are more ambiguous [32].

2.1.2 Deep Q Learning

In its classical formulation, Q-learning algorithm employs lookup tables to store and retrieve state-action values where each entry represents a state-action tuple (s,a). However, adopting a tabular representation may be unfeasible in practical problems where the state and action spaces are very large. A solution to this problem is to represent Q-values through a function approximator that allows state-action values to be represented by employing only a fixed amount of memory which depends only by the function used to approximate the problem. In particular, Deep-Neural-Networks (DNNs) have gained popularity due to their capacity to build an effective representation of the problem through their hidden layer structure. RL frameworks employing DNN as function approximators are known as Deep Reinforcement Learning (DRL) algorithms. The first work implementing Q-learning and DNNs was developed by Minh et al. [23]. In Deep-Q-Networks (DQN) the Q-value function is parametrized by θ, where θ are the weights of the network. The number of neurons in the input layer of the network is equal to the number of variables from which a state is composed, while the output layer has many neurons as the number of actions that the agent may take at each control interaction with the environment. Through this structure, the network is used to learn the relation between states and the Q-value for each action. However, in the RL paradigm, the true Q-value for each state-action pair is not known a-priori but it is learnt over successive interaction with the controlled environment. At each control step, the Q-values are updated according to Equation 2.2 and used as targets to retrain the deep neural network.

Some improvements were introduced in literature in order to improve the DQN formulation as shown in Figure 2.3. The figure shows the flowchart of the structure of a Double Deep Q-Learning agent with Memory Replay.

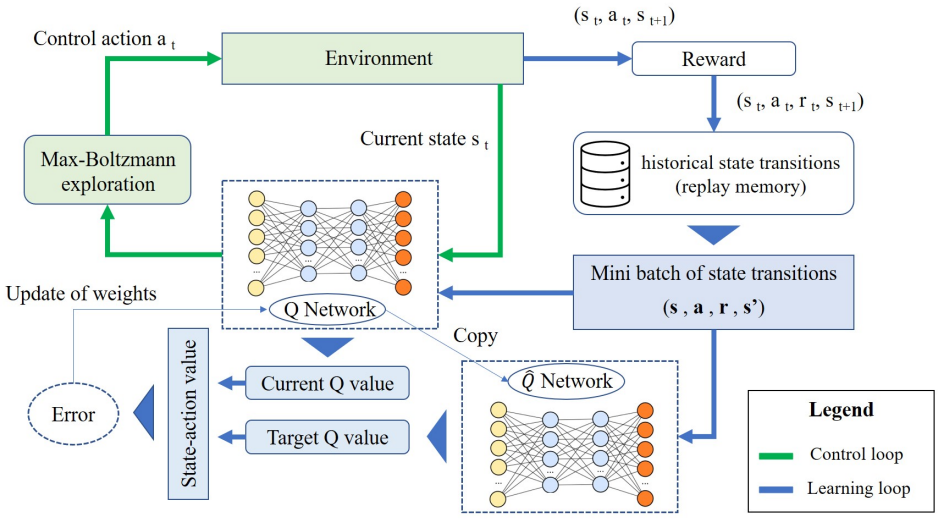


Fig. 2.3 Flowchart of the structure of a Double Deep Q-Learning agent with Memory Replay.

The first improvement involves the introduction of the replay memory to store previous experience obtained by the agent. In the optimization process of the network weights a random mini batch is extracted from the replay memory and used to fit DNN-regression using as targets the Q-values updated according to Equation 2.2. This process enables the re-utilization of previous experience collected by the agent and overcome the problem of correlated observations while performing weights optimization. The second improvement, which refers to Double-Deep Q Networks (DDQN), involves the employment of two neural networks [33]. The first one, called online network, is constantly updated and directly used in the interaction with the environment; the second one, called target network, is updated after N iterations and used to predict target values. The target network is an exact copy of the online network and during the update the weights of the online network are simply copied into the target network.

2.1.3 Soft-Actor Critic

Actor -critic frameworks are characterized by distinct memory structures to map state-action and state-value spaces [34]. One of the main advantages of actor-critic methods is that they can learn stochastic policies through a direct approach which

represent an important advantage for stochastic processes such as HVAC control [35].

In the scientific literature have been proposed several algorithms that belong to the family of actor-critic methods, one of these, recently introduced, is the Soft-Actor-Critic (SAC). Differently from other actor-critic methods, SAC is an off-policy DRL algorithm which showed excellent performance in solving several control tasks [36]. Differently from DQN methods SAC is capable to handle continuous action spaces.

The Actor-Critic architecture employs two function approximators. The Actor has the aim to determine the optimal action for a given specific state of the controlled environment (policy-based), while the Critic evaluates the decisions made by the actor (value-based). The actor and the critic are parametrized as DNN. The actor is employed in both the control loop and learning loop while the critic is employed only during learning. This framework is generally coupled with an off-policy implementation, enabling the re-utilization of the previous experience collected by the agent in order to improve the control policy (i.e. replay memory). Moreover, the SAC policy is trained to maximize the expected sum of future rewards and the expected entropy of the policy at the same time, as defined in Eq.2.4:

$$\pi^* = argmax_{\pi_\phi} E[\sum_{t=0}^{\infty} \gamma^t (r_t + \alpha H_t^\pi)] \tag{2.4}$$

Where H_t^π is the Shannon entropy term, which is a constant term which associates to each state a probability distribution over the possible actions. Through this approach, the agent has the possibility to explore during the training phase, while, during the deployment phase, the mean value of the distribution is used to select deterministic actions, ensuring a robust control policy. α is the entropy regularization coefficient which indicates the relative importance of the entropy term with respect to reward term. γ represents the discount factor for future rewards and r_t is the reward obtained by the agent at the time-step t.

Thanks to the previously introduced features, SAC showed an higher efficiency in exploring state-action spaces compared to other algorithms such as Deep Deterministic Policy Gradient (DDPG) which is very sensible to seed initialization and explores deterministic policies and Trust-Region Policy Optimization (TRPO) which is characterized by sample inefficiencies.

A modified version of SAC was recently introduced in order to handle discrete action spaces [37]. The framework of this algorithm is shown in Figure 2.4.

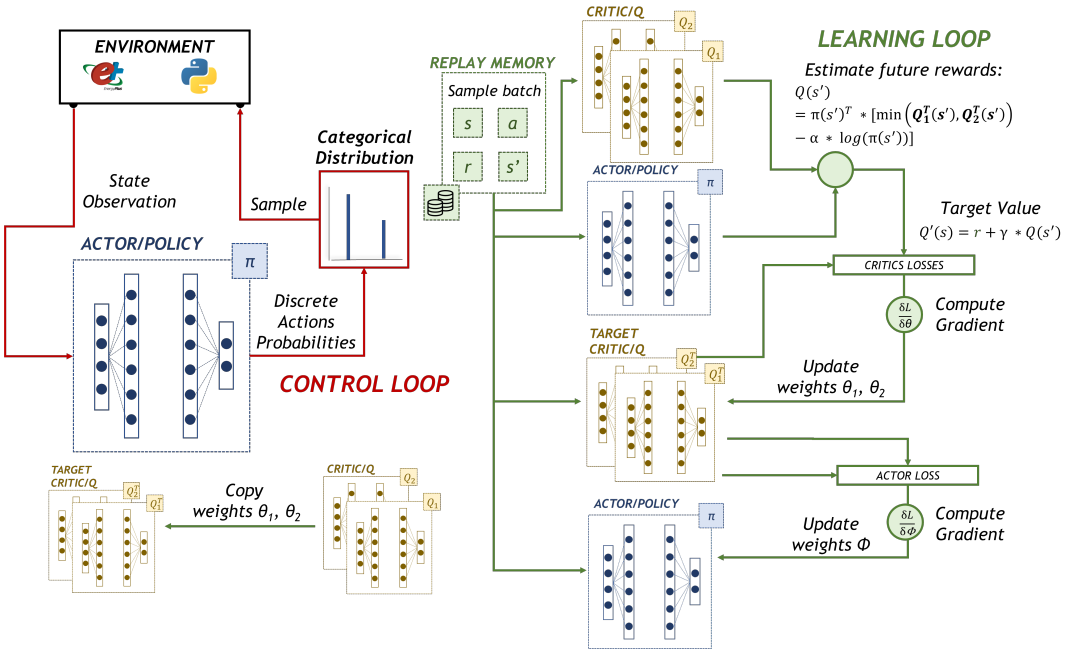


Fig. 2.4 Discrete SAC structure.

In the modified version of the SAC algorithm the critic network, also called soft-Q network, outputs directly the Q-value of each possible action. The parameters of the critic network are updated in order to minimize the error J_Q expressed as follows:

$$J_Q(\theta) = E_{(s_t,a_t)\sim D}[\frac{1}{2}(Q_\theta(s_t,a_t) - (r(s_t,a_t) + \gamma E_{s_{t+1}\sim p(s_t,a_t)}[V_{\overline{\theta}}(s_{t+1}])))^2] \quad (2.5)$$

where D is the replay buffer and $V_{\overline{\theta}}s_{t+1}$ is estimated by means of a target network. In practice two different critic networks are employed and the minimum of their two outputs is employed to compute the above objective. The actor network, also called policy network, directly outputs the action probabilities. The losses employed to update the policy network are calculated according to the following formula:

$$J_\pi(\phi) = E_{s_t\sim D}[\pi_t(s_t)^T[\alpha log(\pi_\phi(s_t)) - Q_\theta(s_t)]] \quad (2.6)$$

2.1.4 Training and deployment strategies for RL agents in HVAC systems

In ideal conditions, a model-free reinforcement learning agent should be directly implemented in the physical system gradually learning the optimal control policy. However, the learning of the optimal control policy is a process that may take a considerable amount of time leading to poor control performance in the initial implementation period.

The *offline training* framework, showed in Figure 2.5, was conceived to overcome this problem.

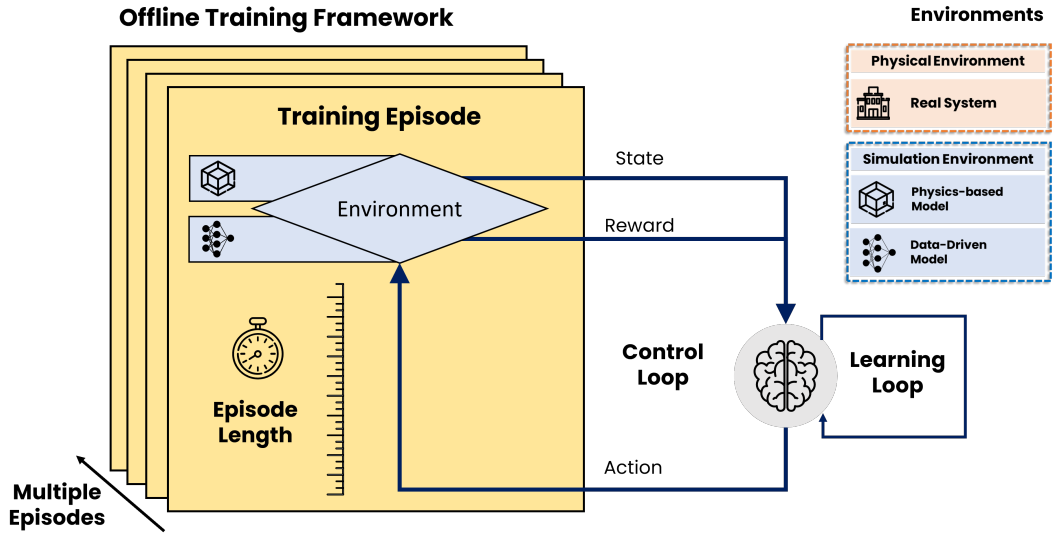


Fig. 2.5 Offline training framework of RL control agents.

According to this approach, the control agent learns the optimal control policy by interacting with the environment during multiple episodes. An *episode* is a period of time characterized by a fixed length which is representative of the control problem being analyzed (e.g., a thermal season). For example, if the control problem involves the optimization of the operation of a chiller unit, the episode will be defined to include the cooling season of the selected case study. An episode is presented multiple times to the RL algorithm that in this way is able to refine its control policy towards the optimal solution. This refinement process is carried out through two loops: the control loop and the learning loop. The control loop is responsible of selecting an action given a state according to the control policy. The learning loop is

responsible of updating control policy parameters (e.g. DNN weights) according to reward signal and previous experience. It is unfeasible, for HVAC control problems, to carry out the offline training framework directly in a physical environment since it may take many time before obtaining a trained agent. Moreover, it would be unsafe from a performance point of view to let the agent to explore different policies during training. For these reasons, this process is carried out in simulation environments where time and low performance are not an immediate issue. These environments are based on models of the system dynamics which are mainly based on a physics-based or data-driven approaches. Further details about simulation environments and models of the system dynamics are provided in the following section.

Once the training of the RL agent is completed, it can be deployed according to two different strategies as illustrated in Figure 2.6:

- **Static deployment**: In the static deployment approach, the agent is implemented as a static entity, meaning that the control policy is no longer updated, and any learning goes on. As a consequence, as showed in the left panel of Figure 2.6, the learning loop is not performed and the reward signal is not actively employed by the control agent. The advantages of such approach are the limited computational cost and the relative stability provided by a static control policy. The disadvantage is that the agent is unable to automatically adapt in the case key-features of the controlled system change (e.g. revamping intervention) and may need to be retrained.

- **Dynamic deployment**: In the dynamic deployment approach, the controller continuously learns from experience constantly updating the parameters of the control policy. As a consequence, as showed in the right panel of Figure 2.6, the learning loop is continuously performed and the reward signal is actively employed by the control agent. Following this approach a RL agent can adapt to a changing system at the expense of higher computational cost and with the risk of stability issues for the control policy [38].

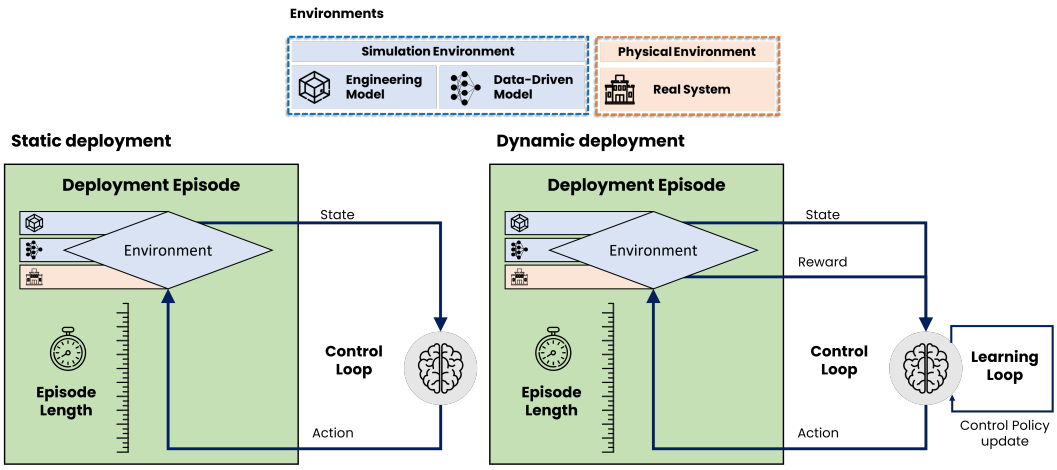


Fig. 2.6 Static deployment and dynamic deployment frameworks of RL control agents.

Both static and dynamic deployment can be carried out in physical or simulation environments. When the deployment is carried out in simulation environments the episode (i.e. the period of time in which the simulation is carried out) may differ from the episode employed during training.

Despite being successful and widely applied, offline training requires a considerable effort in developing the surrogate model of the controlled environment undermining the complete model-free nature of the RL approach.

An alternative is represented by the *online training* strategy where control agent is directly deployed on the controlled HVAC system. The framework of this approach can be represented similarly to the dynamic deployment framework with the difference that the employed control agent has not obtained prior knowledge of the dynamics of the controlled environment through a pre-training process. The control agent is forced to learn the parameters of the optimal control policy while actively controlling the system. This strategy perfectly represents a model-free controller meaning that no previously built model is employed to perform offline training. This procedure can be performed as well in simulation environment in order to study the capabilities of RL algorithms to rapidly converge to an acceptable solution. Moreover, since this procedure is extremely delicate it is necessary to previously analyze its applicability before effectively move to a testing phase in physical environments.

2.1.5 Hyper-parameters characterizing reinforcement learning frameworks

Tuning of hyper-parameters is a crucial task to be addressed during the development of RL and DRL controllers [39]. This section summarizes the main hyper-parameters characterizing the control frameworks employed in this dissertation. Part of these elements were already introduced in the previous sections. However, due to their importance, it is functional to provide a synthetic recap of their properties. Hyper-parameters can be organized according to the following classification:

- **General RL hyper-parameters**: these hyper-parameters are shared among all RL frameworks.
- **Specific RL hyper-parameters**: these hyper-parameters are specific of different RL frameworks characterizing their behavior and converging properties.
- **DNN hyper-parameters**: these hyper-parameters characterize the architecture of DNN employed by DRL algorithms as function approximators.
- **Environment hyper-parameters**: these hyper-parameters characterize the controlled environment and can strongly influence stability and convergence of implemented control agents.

One of the main general RL hyper-parameters which is shared among all RL frameworks is the *discount factor (γ)* for future rewards. The discount factor assumes a value included between 0 and 1. It is a mathematical object introduced to prevent the cumulative sum of future rewards going to infinite ensuring the convergence of the algorithm. Values close to 1 gives greater importance to rewards obtained far in the future with respect to the moment in which the control action is taken. Values close to 0 gives greater importance to immediate rewards obtained after taking a certain action.

Three other important hyper-parameters characterizing off-policy DRL frameworks (like DQN, DDQN and SAC) are the *Replay Memory Size*, the *Batch Size* and the *Number of Gradient Steps*. Replay memory stores the results of previous interactions of the agents with the controlled environment. The size of this memory determines the amount of previous knowledge that can be leveraged by the algorithm

to refine the control policy. Batch size regulates the amount of elements drawn from Replay Memory during the learning phase. Small values of the batch size can guarantee faster convergence properties with the risk of being stuck in near-optimal solutions. Higher values of the batch size may result in slower convergence properties with the benefit of mitigating the risk of learning sub-optimal policies [40]. The number of gradient steps is an hyper-parameter that regulates the number of batches randomly drawn from memory buffer on which gradient update is performed at each control time-step. Typically, this hyper-parameter is set equal to 1, but in particular cases this value can be increased in order to encourage faster learning.

Specific RL hyper-parameters depends on the specific RL of DRL algorithms being implemented (explained among brackets in the following list):

- *Learning Rate (α)* (Tabular Q-Learning, DQN, DDQN): the learning rate can take a value included between 0 and 1 and determines the rate at which new knowledge overrides old knowledge while updating Q-values. Typically, during training this value is set equal to 1.
- *Exploration Rate (ε)* (Tabular Q-Learning, DQN, DDQN): determines the probability of the agent of taking a random action. It can be set at high values at the beginning of the training phase (i.e. close to 1) and be gradually reduced while learning progress.
- *Boltzmann Temperature (τ)* (Tabular Q-Learning, DQN, DDQN): determines the degree of randomness in the choice of action. When high values are implemented the actions are taken with almost the same probability. When low values are implemented the actions with higher Q-values are more probable to be chosen.
- *Entropy Coefficient (α)* (SAC): is the temperature parameter that determines the relative importance of the entropy term versus the reward, and thus controls the stochasticity of the optimal policy.
- *Target Model Update Frequency* (DDQN, SAC): determines the frequency at which the parameters of the online network are copied into the target network.

DNNs are the most widely applied function approximators thanks to their excellent properties in successfully mapping non linear patterns and relationships.

However, DNNs are characterized by several hyper-parameters adding a further degree of complexity to the development of RL controllers:

- *Neural Network Structure*: The most widely applied neural network architecture is the Multi-Layer Perceptron (MLP). The number of *Hidden Layers* and *Neurons* for each hidden layer are the hyper-parameters determining this architecture.
- *Activation Function*: The choice of the activation function may influence convergence of DRL algorithms. The most widely applied functions are the REctified Linear Unit (RELU)[41] and Hyperbolic Tangent (tanh).
- *Optimizer*: The choice of the optimizer may influence convergence and performance of DRL algorithms. The most widely applied optimizers are Adam [42] and RMSprop [43].
- *Optimizer Learning Rate*: The learning rate of the optimizer implemented in DNNs is an hyper-parameter that controls the degree of change of the network in response to the estimated error each time the weights are updated. Increasing the value of the learning rate may be useful in some circumstances to speed-up the learning process.

Eventually, the controlled environment is characterized by a series of hyper-parameters that requires careful tuning:

- *Episode Length*: The length of the episode depends on the specific control problem being studied. It can ranges from few weeks to an entire year.
- *Number of training episodes*: The number of training episodes must be tuned in order to provide to the agent a sufficient amount of experience to identify the optimal control policy.
- *Reward coefficients*: As previously introduced the reward function combines in a mathematical expression the different objectives that an agent seeks to maximize (or minimize). The relative importance of these different objectives is commonly managed through the introduction of weight factors. The tuning of the weight factors plays a key role in the definition of a robust reward function.

As demonstrated by several research [39, 24], the choice of hyper-parameter values can affect sensibly the performance of RL and DRL controllers. The tuning process of these values may result in counter-intuitive solutions that can be easily discarded in the design phase if the tuning task is not approached with a rigorous and robust method. The results obtained from the applications presented in this dissertation support this claim also providing guidelines and indications to future researchers and practitioner approaching RL control for building energy systems.

2.1.6 Software and programming languages for developing RL and DRL controllers

RL and DRL algorithms are usually developed through high-level programming languages such as Python and MATLAB [44] which provide useful tools and libraries for developing these control frameworks. In the current scientific literature, the most widely applied tools are the following:

- **Tensorflow**: Tensorflow is an open source machine learning and artificial intelligence library developed by Google Brain [45]. In Tensorflow computations are handled through stateful dataflow graphs. Tensorflow is employed to build and train deep neural networks models which are the foundation of DRL control strategies.
- **Pytorch**: PyTorch is an open source machine learning (ML) framework based on the Torch library and the Python programming language. It's one of the most popular deep learning research platforms. The framework was created to expedite the transition from research prototyping to implementation [46].
- **keras-rl**: keras-rl is an open source library developed for Python implementing state-of-the-art DRL algorithms [47]. This library seamlessly integrates with the deep learning library Keras.
- **Stable Baselines**: similarly to keras-rl, Stable Baselines is a open source library developed in Python. Stable Baselines integrates a wider range of algorithms compared to keras-rl and employs as back-end engines both Tensorflow and PyTorch [48].

- **Reinforcement-Learning Toolbox**: The Reinforcement-Learning Toolbox provides functions and a Simulink block for training reinforcement learning algorithms through interactions with environments modeled in MATLAB or Simulink. The Toolbox represents policies and value functions using deep neural networks or look-up tables and provides implementation of multi-agent controllers.

Pytorch and Tensorflow are real machine learning libraries that do not provide any pre-set implementation of reinforcement learning algorithms. Using these tools, it is necessary to build the DRL algorithms from scratch. This solution is more complex for a novice user, but it allows to better understand the functioning of the algorithms as well as an high level of flexibility. On the contrary keras-rl, stable baselines and the toolbox are tools in which state of the art implementations of the most famous DRL algorithms are already provided to the user. The use of these tools allows a simplified approach to the application of this control technique. However, the possibility of making modifications according to specific user requirements is limited.

2.2 Research context: Application of reinforcement learning control to HVAC systems

The Section 2.2 introduces the research context of the dissertation. While in the previous section reinforcement learning framework was discussed, in the following the main applications of this advanced methodology for HVAC control are reviewed. These applications represent the state-of-art of the implementation of RL and DRL strategies for HVAC systems control which is the focus of this thesis. Applications of RL and DRL can be analyzed and categorized following several patterns. Figure 2.7 shows a conceptual scheme based on Figure 2.1 reporting the five different criteria related to the application of RL to HVAC system control on which is based the proposed analysis of the literature.

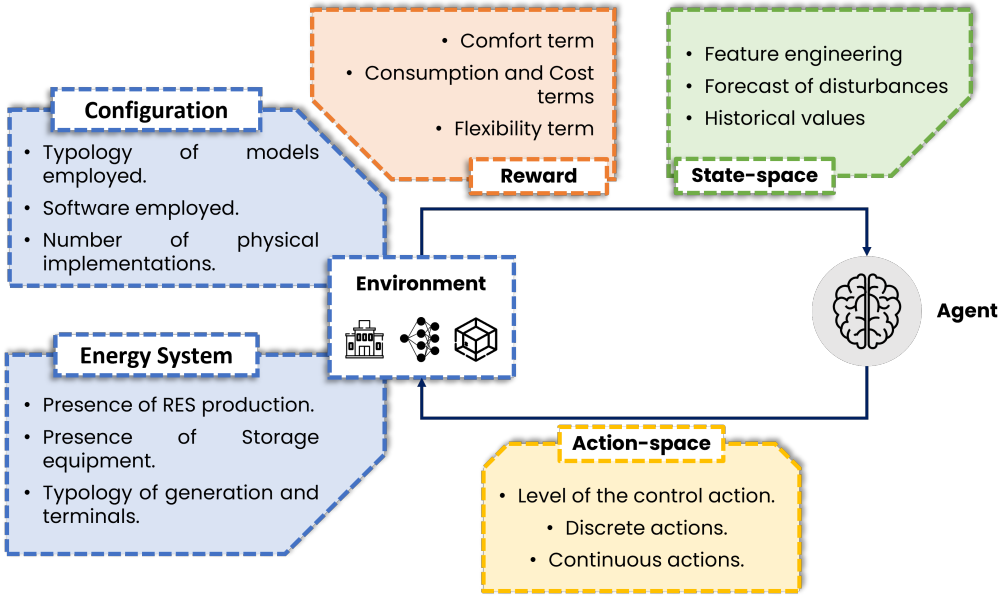


Fig. 2.7 Conceptual scheme of the features of RL application to HVAC system control.

The core of every RL application is the controlled environment. As illustrated in the top-right panel of Figure 2.7, the first criteria employed to analyze the current scientific literature is the configuration of the controlled environment. The controlled environment can be of two typologies: i)virtual-simulative or ii)physical-real. The applications in the literature were reviewed highlighting which of the two types was investigated, whether both were used, and what tools the researchers employed in

the implementation process. A second criteria, showed in the bottom-right panel of Figure 2.7, involves the features and characteristics of the building and the energy system. The scale and complexity of these systems affect the definition of the controller and provide a valid benchmark for different applications. Beside the controlled environment, the criteria on which the analysis of the literature was based included the definition of the action-space, reward signal and state-space illustrated respectively in the bottom, top-left and top-right panels of Figure 2.7. The action-space represents the output of the controller. This output can be provided at a high, medium, or low level of the control hierarchy and can be of continuous or discrete type. The reward represents the objective of the controller and it is mathematically encoded within a function. This function can include one or more objectives that are often contrasting. The state-space represents the input of the controller. A careful selection of these variables is desirable since it can promote a faster convergence and a greater robustness of the learned control policy.

It was decided to not classify the applications in function of the different RL and DRL frameworks and their algorithmic differences. The following sections review RL and DRL applications according to the five classification criteria introduced in Figure 2.7.

In particular, Section 2.2.1 reviews the applications in function of the nature of the controlled environment. The main software employed to simulate HVAC system are introduced along with an analysis of implementations in physical test-beds.

Section 2.2.2 analyzes and discusses the feature of the HVAC system, highlighting the presence of flexibility sources such as RES and storage equipment. Moreover, the analysis focuses on the typology of the most typical configurations of generation systems, distribution systems and terminal units.

Section 2.2.3 presents and discusses the various applications of RL and DRL to HVAC system from a control action perspective. The level (high, medium, and low) at which the control action is provided along with an analysis on the different configurations (continuous and discrete).

Section 2.2.4 discusses the different control goals applied in the literature and how they have been combined within reward functions.

Section 2.2.5 discusses the control inputs as state-space formulation in applications of RL and DRL for HVAC system control by highlighting different approaches that include variable engineering and the use of system disturbance forecasts.

The final aim of this chapter is then to discuss a wide research context, for better pointing out broader challenges and opportunities related to the application of RL and DRL for HVAC system control.

2.2.1 Controlled environment: configuration

The first criterion used to classify reviewed applications considers whether the case study employed is physical or simulative. RL and DRL are frameworks that are still in their infancy and their applicability for HVAC system control must be firstly tested within safe and enclosed environments. To this purpose, researchers worldwide developed different simulation environments employing surrogate models of the building dynamics characterizing different control problems. These surrogate models can be developed according to different approaches:

- **Physics-based models**: physics-based models, also identified as white-box or engineering models, use physical knowledge to describe the dynamics of the building and of the HVAC system. They are based on the principles of heat transfer and energy and mass conservation [49]. The parameters defining these models are physically significant and can be retrieved from technical documentation available if the case study is a real system or from standards and guidelines in case of prototype buildings. On one hand, physics-based models suffer from modeling inaccuracies due to the vast amount of parameters required for their definition and they usually require a considerable amount of time to be developed [50]. On the other hand, if correctly tuned they are capable to correctly emulate the physical properties and dynamics of the building system. Physics-based models include also simplified solutions such as first or second order models that can be used to perform a simplified simulation of system dynamics.
- **Black-box models**: black-box models are developed starting from monitored or surrogate data without prior assumptions regarding physical relationships. The main advantage provided by these models is the limited amount of information required to their development. The main disadvantage relies in the fact

that these models have limited generalization capabilities and are unreliable in predicting building dynamics that fall outside their training range [51].

- **Gray-box models**: The gray-box category encompasses a wide range of models that include simplified physical relationships but also necessitate parameter estimation using measurable data. In most gray-box models, the physics is reduced through state space dimensionality reduction or linearization. The RC analogy, which characterizes every model by its affinity with a resistor-capacitor electrical circuit, is a common gray-box concept [49]. Theoretically, gray-box models can overcome the limitations of both physics-based and purely data-driven approaches. Since part of the knowledge regarding the physics of the system is already present in the model structure, gray-box are more likely to perform correctly outside the calibration range [52]. Moreover, they require less information that white-box models to be developed. In practice, the main drawbacks is related to the necessity of a robust parameter identification method.

In the current scientific literature, physics-based approach was the most widely implemented with the purpose of RL and DRL development. Typically, physics-based modeling was performed through programs and software allowing the simulation of energy and mass flows within the building also considering the interaction of the system with the surroundings (e.g. weather). In this context, the most popular simulation software are the following:

- **EnergyPlus** [53]: is an open-source, comprehensive building simulation program capable to model energy consumption for heating, cooling, ventilation, plug and process loads.
- **TRNSYS** [54]: is a flexible proprietary software environment used to simulate transient systems. The standard library provides more than 150 models of different equipment that can be modified by the user to enhance simulation capabilities.
- **Modelica** [55]: is an open-source, object-oriented language for modeling heterogeneous physical systems. In particular, the Modelica Buildings Library [56] provides powerful models to simulate buildings and district energy control systems.

Since RL and DRL frameworks are developed in programming languages such as Python and MATLAB, simulation software, which are usually employed for design purposes, are embedded into architectures relying on specific co-simulation tools such as Building Control Virtual Test Bed (BCVTB) [57] and Functional Mockup Interface (FMI) [58].

Among the previously mentioned software, EnergyPlus is the most widely applied. It was employed to analyze the effect of DRL control strategies for a variety of case studies ranging from residential [59] end use to commercial [24, 60] end use considering both physical sites [24, 61, 62] and reference buildings [63, 64] taken from different guidelines.

Compared to EnergyPlus, the application of TRNSYS [65] and Modelica [66, 67] is still limited. One barrier to the adoption of TRNSYS could be related to its proprietary nature which confine its applicability in research environments. Conversely, the trend of adoption of Modelica is rising also thanks to the recent release of Spawn Of Energy Plus (SOEP) project. SOEP combines EnergyPlus and Modelica leveraging the first for weather, envelope, lights and load models and the second for state-based models of HVAC system. This approach allows to overcome implicit, load-based modeling of HVAC loads and controls of EnergyPlus and has the potentialities to become a new standard in the building energy simulation community. In this context, Touzani et al. [68] and Lee et al. [69] recently employed a simulation environment combining EnergyPlus and Modelica to evaluate DRL control strategies applied to HVAC systems.

Still in the context of physics-based modeling, Chen et al.[70] and Jiang et al.[71] employed in the simulation environment simplified first order model and second order model of system dynamics respectively. Despite their computational lightness, these models can hardly represent the complexity of HVAC systems. For this reason, the application of RL and DRL control techniques on this type of models represents more of an academic exercise than a real (or at least realistic) demonstration of the capabilities of these frameworks for the management of HVAC systems.

The black-box approach to building models for implementing co-simulation environments has been less widely used than the physics-based approach. Zou et al. [72] employed recurrent neural networks trained with monitored data to build the models of the dynamics of two air-handling units of a commercial building. Through the paper, the authors demonstrated how a DRL agent can be effectively

trained employing a black-box model of the controlled system. This approach is very interesting because exploiting data-driven models for pre-training DRL agents would greatly reduce the time and complexity of the learning phase before their actual deployment in the physical world. However, in this work the authors performed the deployment phase of the control agent on the same data-driven model on which the agent was trained. Therefore, this approach does not provide an accurate indication of the performance of the same agent on the original system.

Grey-box models, being widely used in the MPC framework, were employed in early applications proposing controllers based on RL techniques [73]. They have been gradually replaced in the scientific literature by physics-based models but are still used when available for certain case studies [74].

Most of the experiments proposed in the literature have been limited to the implementation of DRL agents in a simulation environment, however, some authors have succeeded in bringing the developed controllers from simulation to the physical world. Touzani et al. [68] pre-trained in a simulation environment a control agent for 4 years. This agent was successively deployed for 7 days in a real residential building achieving 39.6 % of cost savings with respect to a baseline controller. Valladares et al. [75] implemented a DRL agent pre-trained for 10 years in a real classroom and laboratory during a period of almost 6 months. They results showed a reduction in the energy consumption of 4-5 % and increased PMV compared to the baseline. Chen et al. [76] deployed an agent pre-trained through imitation learning in a campus building achieving a 17 % saving of cooling energy. Zhang et al [77] designed a physical test-bed for demand response scenarios providing exhausting details about the technology implemented to deploy the RL agent. In [78] the authors implemented their controller in six residential buildings increasing significantly the self-consumption of PV system compared to baseline thermostat control.

The benefits from the energy management perspective of implementing these control techniques is well illustrated by these works. What remains to be investigated and discussed in detail is the implementation cost of these solutions. Implementing DRL techniques in the real world requires specific tools and knowledge, the cost of which is rarely taken into account in scientific publications. For this reason, their industry-wide adoption and scalability are still challenging.

2.2.2 Controlled environment: building and energy system

A second useful criterion for classifying RL and DRL applications is provided by the analysis of the building and of the HVAC system. In the current scientific literature, RL and DRL were applied to both residential and commercial case studies.

Residential sector plays a key role in the application of demand side management strategies aimed at increasing demand flexibility [79]. Thermostatically controlled loads such as heat pumps, electric water heaters and air conditioning represent example of flexibility sources which are interesting for demand response purposes [80]. In this context, RL and DRL algorithms have been applied to optimize the management of thermostatically controlled loads. Ruelens et al. applied batch-Q-learning demonstrating its effectiveness in managing heat pumps and electric water heaters in both open-loop and closed-loop form. Liang et al. [81] proposed a Q-learning controller to manage flexible demand in a residential context. Residential case studies have also been used to study the management of water heating systems powered by gas boilers, demonstrating their effectiveness in reducing consumption while ensuring comfort conditions [59]. However, residential systems are very challenging to control mainly due to the high stochasticity of occupancy. Moreover, in absolute terms, the cost savings provided by advanced control techniques are rather limited. Greater advantages in this sense are obtained when more residential systems are aggregated and controlled in an effective and coordinated way [82].

Conversely, in commercial buildings (such as schools, university campuses, shopping malls and offices) HVAC systems account for 40%-50% of overall consumption [1]. The complexity of the buildings belonging to this sector leaves ample room for improvement in terms of management and control policies. In this context, DRL was applied to large office buildings characterized by complex heating systems formed by multiple gas-fired boilers, circulation pumps and radiators serving multiple zones with different schedules and requests [24].DRL was also employed to manage an innovative radiant-based heating system served by district heating [38, 76]. Moreover, DRL was successfully applied to cooling water systems which are a fundamental subsystem of an HVAC characterized by circulation pumps, cooling towers, chiller condensers and economizers [83]. The purpose of cooling water systems [65] is to release the heat rejected by chillers. The chiller COP (coefficient of performance), which determines the energy consumption of the overall HVAC system, is strongly dependent on the operation of the cooling water system [84]. DRL was employed to

manage simpler case studies as variable refrigerant flow system [85, 86] and fan-coil units serving a small commercial zone [87].

A distinctive feature of several case studies is the presence of energy storage technologies. In particular, Thermal Energy Storage (TES) proved to be a sustainable solution making HVAC systems more flexible to time-varying electricity prices improving capabilities of the system to shift its demand patterns [88, 89]. Ruelens et al. [90] achieved 30 % cost savings applying a RL controller to a residential system characterized by an electric water heater and an hot-water TES. Liu et al. [91] developed one of the first application in the field in which a RL controller was employed to manage a building equipped with chiller, a cold-water TES and VAV terminal units. Pinto et al. [92], Canteli et al. [93] and Kathirgamanathan et al. [94] analyzed the same case study implementing different DRL controllers with both centralized and de-centralized architectures to coordinate a district of buildings equipped with reversible heat pumps and both hot and cold water TES. The authors of these works achieved both cost savings in the range of 3-10 % and peak reduction between 20% and 30%.

In addition to TES, the presence of on-site PV production systems is another interesting feature to consider. PV systems have been widespread adopted and promoted to sustain the growing energy demand in buildings [95]. Due to PV weather-dependent nature, electrical storage solutions have been introduced to increase self-consumption providing benefits to both end-users and grid operators [96]. However, Battery Energy Storage Systems (BESS) (e.g., Lead-acid and Li-ion batteries) are characterized by high investment cost making their adoption unfeasible for many applications [97] if incentives provided by policymakers are not foreseen [98]. Nevertheless, BESS improves PV energy utilization, by addressing the problem of the low flexibility of solar energy. In this context, a DRL agent was used by Sanaye et al. [99] to control the operation of a Combined Heat and Power (CHP) generation unit and of a gas-fired boiler in an hybrid system with PV panels, solar collectors, wind turbines, a hot water storage tank and batteries. This control strategy reduced the operational cost of a residential complex with respect to two different RBC strategies. Chenxiao et al. [100] developed an RL controller to manage a residential energy storage module based on partial knowledge of the controlled system achieving up to 50% cost savings. Anvari-Moghaddam et al. [101] proposed an energy management strategy based on a multi-agent system for IES in a microgrid to reduce operation cost and to ensure user's needs. Particularly, a Bayesian

Reinforcement Learning (BRL) controller was used for the battery operation, which was coordinated with other agents in charge of collecting and sharing information, making predictions of renewable generation and providing computation services. Zsembinszki et al. [102] applied DRL to an innovative hybrid energy storage system increasing the performance by about 50% compared to baseline strategy. Raman et al. [103] developed an RL controller to manage a residential integrated energy system comprising of a PV system and batteries achieving similar performance to an MPC.

From the point of view of electrical energy storage, a very interesting technology that is spreading in recent years is represented by electric vehicles. Considering the building and the mobility sector at the same time results in more efficient control systems [104]. For example, when the electricity price is low, a controller may choose to provide space heating/cooling, charge the EV, or store energy in a stationary battery for later use. On the other hand, this union poses some difficulties. Charging electric vehicles increases a building's total – and possibly peak – energy usage. Furthermore, once an EV is attached, most EV chargers begin charging at full power. As a result, if numerous EVs in a neighborhood are charged at the same time, the aggregated demand might be very high, potentially causing concerns with energy dispatching and grid stability [105, 106]. Mocanu et al [107] employed a DQN agent to manage multiple residential buildings in a simulation experiments achieving 20% cost savings. Svetozarevic et al. [106] developed a DRL controller to manage a residential building equipped with an heat pump and an electric vehicle. The controller was tested in-field for two weeks achieving 42% cost saving with respect to the baseline.

Following these considerations, what emerges from the analysis of the different applications is a greater need for advanced control techniques such as RL and DRL in case studies characterized by a high complexity of the energy system. The greater is the complexity, brought for example by the presence of renewable energy sources and storage systems, the greater is the necessity to integrate a predictive and adaptive energy management strategy. In this context, integrated energy systems in buildings represent a promising case study to focus the application of advanced control strategies. Figure 2.8 shows a schema of the features characterizing an integrated energy system in buildings.

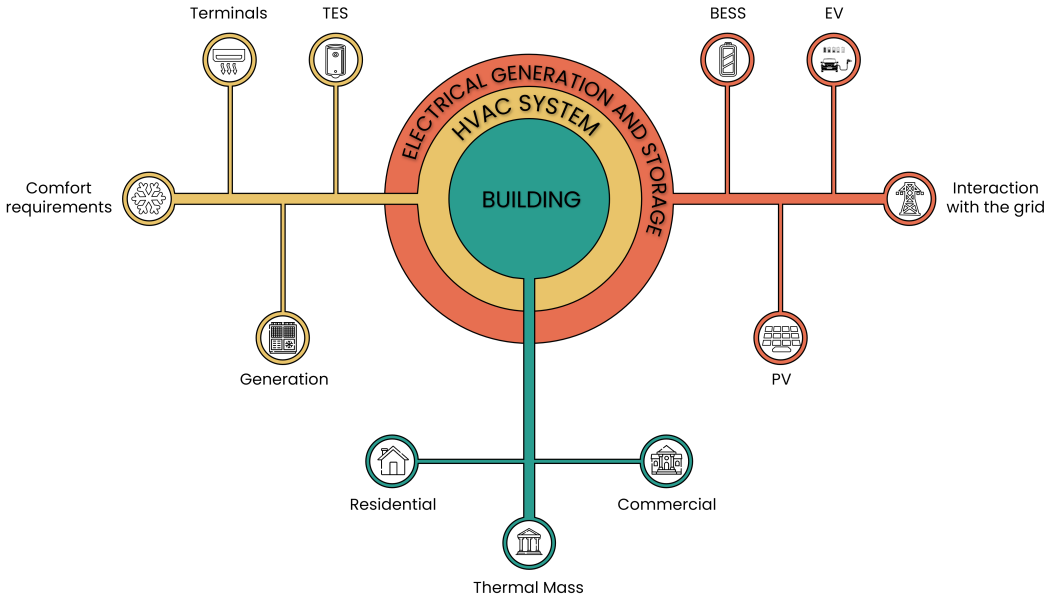


Fig. 2.8 Schema of the features of an integrated energy system in buildings.

The building can be either commercial or residential and its thermal mass can be regarded as one of the main flexibility sources of the system. The HVAC system is characterized by different components. Generation equipment may include renewable sources such as solar water heaters and the thermostatically controlled loads such as chillers and heat pumps. As previously introduced, TES represents one of the key flexibility sources since it provides the opportunity to shift thermal load of the building. The nature of the terminal units plays a significant role in the defining the applicability of advanced controllers. Complex systems characterized by several subsystems such as Air Handling Units (AHU) are ideal candidates for the application of advanced control techniques. Such systems are often characterized by several operational inefficiencies and can greatly benefit from improved management strategies. Eventually, the electrical system, which directly integrates with the grid, can include different flexibility sources such as PV production, BESS and EVs. These features contribute to increase the flexibility potential of the building energy system. However, this potential have to be correctly managed in order to ensure the optimal operation of the system.

2.2.3 Control outputs: action-space

The control action that represents the controller output is one of the first components to be designed when developing an RL or DRL agent. The control action is closely related to the type of the considered building or facility. However, the choice of the output is not always obvious. For example, Figure 2.9 shows the different levels at which a control action can be performed for a simple heating system formed by a gas-fired boiler, a circulation valve and a radiator serving a building thermal zone.

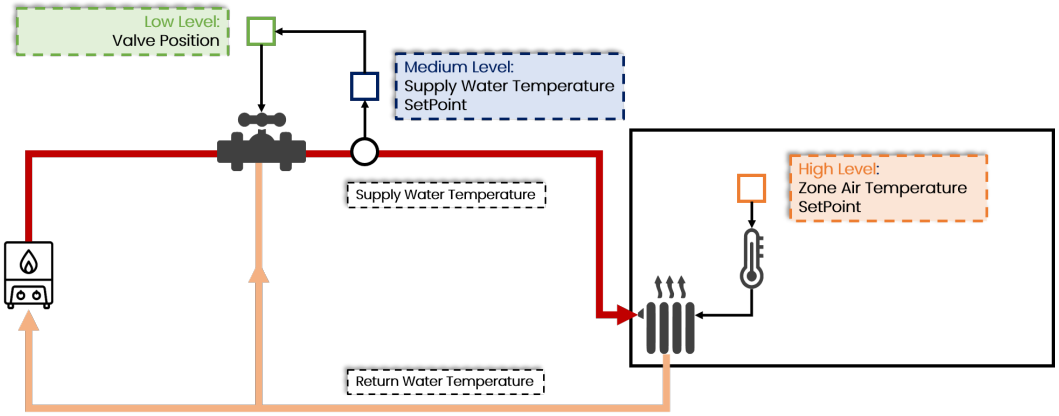


Fig. 2.9 Example of the different levels of control actions for a simple heating system.

As can be seen, although the system is very simple there are several outputs that an advanced controller can consider to manage the system. Starting at the high-level, the controller might be designed to directly manage the ambient air temperature set-point. This set-point would indirectly adjust the emission terminals, radiators in this case, to provide more or less heat to the thermal zone by adjusting the circulated flow rate. Acting at this level enables the effective control of the pre-heating phase of the building[8]. However, this control is often carried out by the occupants, in residential buildings, or by the building manager, in commercial buildings. Therefore, an high-level controller have to account for occupant interaction which can possibly by-pass the choices made by the agent, further complicating the decision-making process. Conversely, a medium-level controller can be designed to regulate the supply water temperature set-point. This value is employed by a low-level controller managing the opening position of the circulation valve of the circuit. Acting on low-level controllers can be challenging since they represent the first line of management for HVAC systems. Any failure or malfunction of these controllers can have serious

consequences in terms of performance. Moreover, traditional control strategies, such as PID, already represent a robust solution for effectively tracking set-points. Acting on high-level or medium-level implies the presence of traditional controllers, at low-level to manage the actuators tracking the desired set-points. An advanced DRL controller might be able to manage all set-points and actuators simultaneously acting on high, medium and low level. However, even DRL controllers suffer from the curse of dimensionality [108]. By increasing the number of outputs, and inputs, the convergence time and controller instabilities are significantly larger. For these reasons, it is critical to carefully design the optimal output signal of the control agent for each specific application.

A great amount of the reviewed applications employed high-level HVAC controllers managing set-points at building or zone level. The most frequently controlled variable was indoor air temperature [109, 110], but some work also focused on relative humidity [111] as the HVAC system considered allowed for latent heat control within the building. Other studies focused on medium-level control acting on supply water temperature set-point [24, 38], supply air temperature set-point of AHU systems [112, 75], supply air temperature flow rate [113] and TES temperature set-point [114, 115]. Eventually, few works developed low-level controllers to manage control signals such as air damper opening position [116, 72], humidifier operations [85] and valve position [106].

Another fundamental aspect relating to the choice of control action concerns its discrete or continuous nature. The most widely applied DRL technique for discrete action spaces is DQN previously introduced in this chapter. There are several DRL algorithms such as SAC and Deep Deterministic Policy Gradient (DDPG) [117] allowing for the implementation of continuous action-spaces. Often, especially in physical test-beds, it is not possible to implement a completely continuous control on HVAC systems due to technological limitations. Even the most sensitive control system can still provide in output a discrete signal. As a consequence, the continuous output of certain algorithms would require to be discretized once the signal is sent to the involved actuators. In addition, in the case of the implementation of a medium-level or high-level controller the output of the controller will be employed as a reference for traditional controllers. As a consequence, an excessively fine control over these values could be counter-effective for low-level actuation. However, when the physical range of control actions is very large, such as in the case of flow temperatures of a heating system, a discrete action-space formulation may

risk including an excessive amount of distinct values. In this case even the most sophisticated algorithms designed for discrete action-spaces may struggle to converge to the optimal control policy being outperformed by their continuous action-spaces counterparts. Since HVAC control should deal with both discrete and continuous actions Li et al. [118] proposed a Trust Region Policy Optimization (TRPO) based approach capable to handle simultaneously the two cases.

Eventually, a feature that is worth to careful consider when designing action spaces and, more in general, RL and DRL agents is the frequency of the control action. In a simulative context, it is relatively simple to coordinate the exchange of information between the agent and the controlled environment. In the RL framework, at each control step, the agent receives information about the state of the environment and the reward obtained for taking a certain action in the previous step. In the period between two control steps, the environment should have enough time to evolve towards the new state determined by the control action chosen by the agent. Through this process is possible to correctly assign the values of the reward function. Coordinating the timings of these interaction in a physical environment is a daunting task especially if the time between two control steps becomes shorter. The typical frequencies for medium-level and high-level controllers ranges from 15 minutes to 1 hour while for low-level controller this value can reduce down to 5 minutes.

2.2.4 Control objectives: reward

The reward function defines through a mathematical formulation the control goals of the agent. In the analysis of the scientific literature were found four major goals in HVAC system control:

- **Energy conservation**: this goal aims at minimizing both thermal and electrical energy consumption of the controlled system. Energy conservation can be achieved not only through retrofit intervention introducing new and better performing equipment but also through the implementation of effective control strategies capable to enhance the efficiency of existing technologies [119].
- **Cost reduction**: this goal aims at minimizing operating cost of the controlled system. It is strongly influenced by energy price schedules and by the presence of flexibility sources and RES production which are capable to shift the demand to low-price periods [120].

- **Flexibility**: this goal is strictly related to cost reduction and is becoming a prerogative of systems using electricity as the primary energy carrier. It can target different objectives including the reduction of peak absorption from the electrical grid and the optimization of the operations according to Demand Response (DR) programs [121, 122]. In this context it is particularly relevant since in DR scenarios the introduction of price-based programs could lead to new undesirable peaks of demand [123].

- **Comfort**: this is a primary goal for HVAC systems and thus for advanced control strategies like RL and DRL. The satisfaction of occupant comfort along with appliances use is responsible of 80% of energy consumption in commercial buildings [124]. Moreover, maintaining comfort is a key-aspect to ensure morale, working efficiency and productivity of the occupants [125]. In particular, HVAC systems are responsible for thermal comfort and Indoor Air Quality (IAQ). Thermal comfort is challenging to be evaluated, especially in physical implementations. The most common metrics to evaluate thermal comfort are Predicted Mean Vote (PMV) and Predicted Percentage of Dissatisfied (PPD) based on Fanger's theory [126]. However, given the dependency of these metrics from many variables which collection is not a trivial task most application relies on indoor air temperature measurements to evaluate thermal comfort. Moreover, the performance of this metrics for an effective evaluation of thermal comfort status of building occupants remains an open issue. A recent study, demonstrated that these metrics are accurate only for the 33% of the time considering a wide and robust dataset [127]. For these reasons, researchers are exploring new approaches based on occupant feedback and spatio-temporal analysis to evaluate thermal comfort within control frameworks [128]. Eventually, IAQ is evaluated trough the measurement of pollutants in the indoor air such as CO_2.

The challenge in the reward design process relies in the definition of the optimal strategy to combine multiple goals. Figure 2.10 shows a schema of the different terms, related to the previously introduced goals, eventually composing the reward function of an RL or DRL control agent.

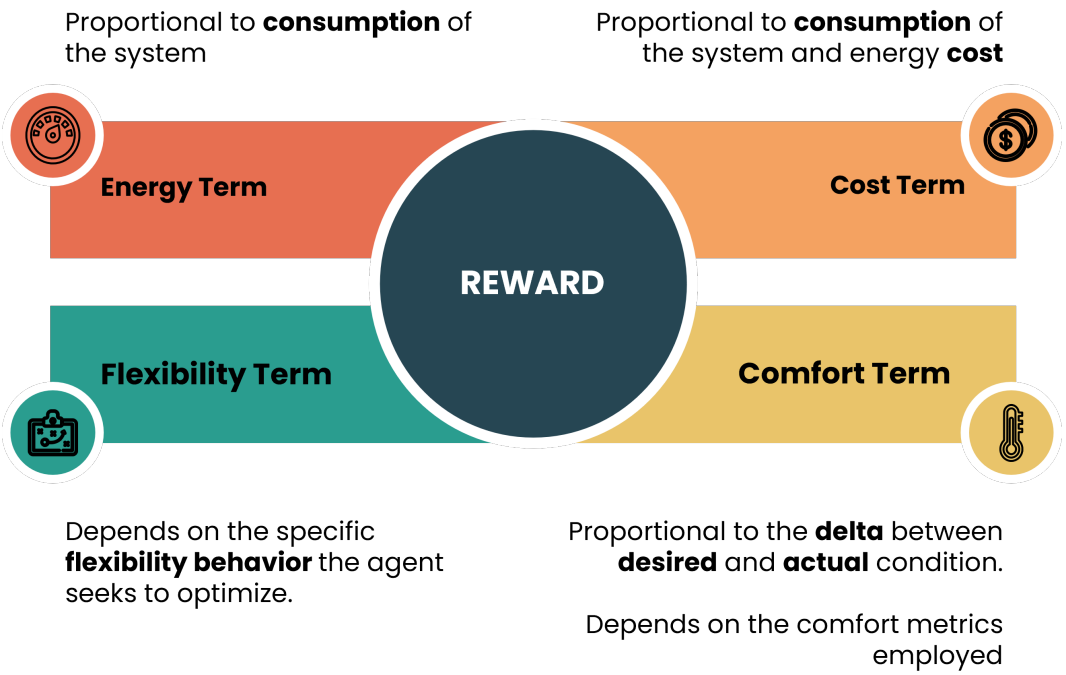


Fig. 2.10 Schema of the different terms composing the reward function.

In building energy systems control field rewards are commonly expressed as penalties meaning that each term of the function as a negative impact on the overall value. The energy term is commonly defined proportional to the energy consumption of the controlled system and it is expressed as follow:

$$r_{energy}(t) = -E_{system}(t) \tag{2.7}$$

The minus sign on the right side of the equation is introduced to penalize the agent proportionally to the magnitude of the energy consumption. The cost term is defined similarly to the energy term combining the consumption of the controlled system with the purchase price of the energy:

$$r_{cost}(t) = -price(t) * E_{system}(t) \tag{2.8}$$

The minus sign is commonly adopted also in this case to penalize the agent proportionally with the cost achieved. The flexibility term depends on the specific behavior that the agent seeks to maximize. A frequent scenario explored in the

literature is the reduction of the peak load of the system. In this case the flexibility term can be defined as follow:

$$r_{flex}(t) = -price_{peak} * P_{system,max} \tag{2.9}$$

Where $price_{peak}$ is a specific tariff defined for peak load and $P_{system,max}$ represents the peak load. Eventually, the comfort term is commonly defined as the deviation of actual value of a specific metric from the desired value:

$$r_{flex}(t) = -|x - x_{set\,point}| \tag{2.10}$$

Where x is the employed metric (e.g. indoor air temperature, CO_2 concentration) and $x_{set\,point}$ is the desired set-point. The minus sign is introduce to increase the penalty received by the agent with the increase of the delta between actual and desired values.

The most widely approach is to employ a weighted sum of the different goals. An example is provided in the following equation:

$$r = a_1 * r_{comf} + a_2 * r_{cost} \tag{2.11}$$

Different combinations of goals have been introduced in the scientific literature. Baghaee et al. [129] designed a reward function to optimize energy consumption, thermal comfort and IAQ. In [130] the authors combined energy sold and energy withdrawn from the grid in order to minimize the operational cost. Claessens et al. [131] designed a reward function to simultaneously optimize energy arbitrage, which is function of an external price, and peak shaving/valley filling evaluated on a daily basis.

In a different approach positive and negative terms are added to the reward function besides the main optimization goals to encourage or discourage particular behaviors. In [132] the authors introduced an exploration bonus to promote improvements in the model formulation by the agent. Brandi et at. [27] added a penalty term to discourage states in which the temperature of a TES rises above a safety threshold.

The identification of the reward function is one of the time-consuming processes in the design of RL and DRL controllers. The optimal configuration depends

from optimization goals, HVAC system features and characteristics of the control problem. As a consequence, reward functions may significantly change for different applications. It is recommended to professionals and researchers approaching to RL and DRL control for HVAC systems to employ the current scientific literature as a guideline for reducing the design time of reward functions given the features of their case studies.

2.2.5 Control inputs: state-space

The selection of the variables forming the state-space is another crucial task in RL and DRL development. The state-space must include all the variables required the agent to learn the optimal control policy. Without important variables it is impossible to converge to the optimal solution regardless of the robustness of the algorithm. However, including too many variables is counterproductive as the agent suffers from the curse of dimensionality [8].

The variables included in the state space may not refer only to the current control step. For example, introducing past values of certain variables is preparatory to provide the agent with correct information regarding the building dynamics and the thermal inertia of the system due to the thermal mass. Fuselli et al. [133] included state values lagged by two time-steps in the past significantly increasing the number of outputs. Brandi et al. [27] considered the four past values of storage tank temperature to provide information about the inertia of the system to the control agent. In [134] the authors directly employed Recurrent Neural Networks (RNN) as function approximators for actor and critic networks. In particular, Long-Short Term Memory (LSTM) networks were employed for their capacity to effectively map sequence-based highly non-linear patterns. However, adding historical states increase the number of inputs and the risk to incur in the curse of dimensionality. To overcome this issue, Claessens et al. [135] used a Convolutional Neural Network (CNN) model to compress the previous ten temperature values reducing state-space dimensions.

Beside historical values, prediction of certain variables represents a key-information to be forwarded to RL and DRL algorithms. Forecasts can be added to the state-space to provide to control agents information about the future evolution of external forcing variables such as weather and energy prices. In [27] the authors used predicted values

for the next 24 hour of electricity prices and outdoor air temperature to optimize the operation of an integrated energy system. Deltetto et al. [122] used in addition to outdoor air temperature and prices also solar radiation predictions to optimize a system characterized by PV production. However, the accuracy of predictions strongly influences the performance of the controller. Thus, grater attention must be devoted in developing accurate forecasting models.

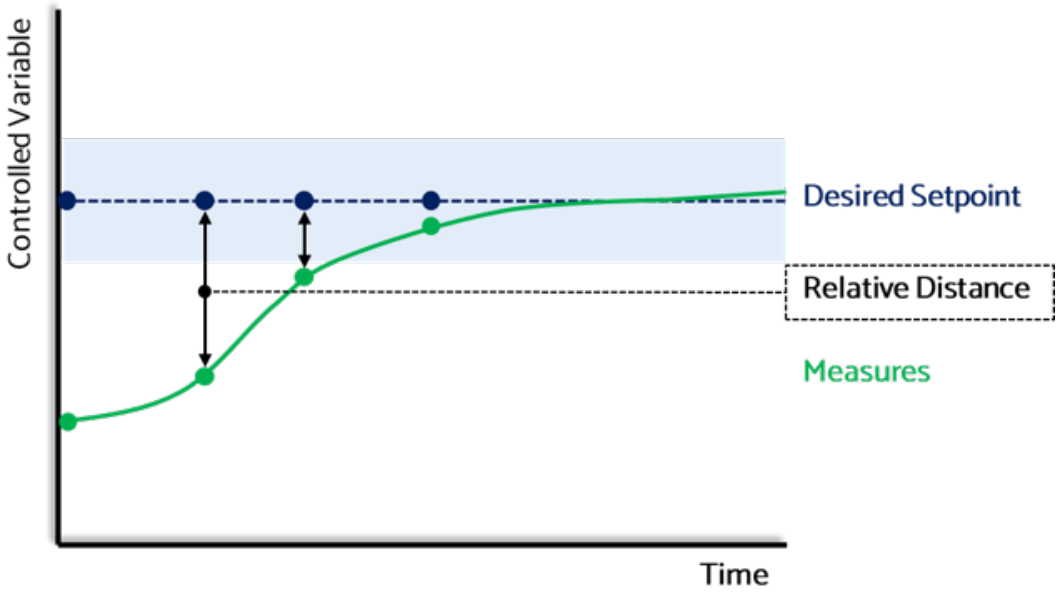


Fig. 2.11 Example of a variable-engineering process.

An important aspect to consider in the definition of the state-space is variable-engineering. Variable-engineering processes are required by traditional RL algorithms such as Q-table since these frameworks accept only discrete values in input. In DRL frameworks, neural networks allows the usage of continuous variable as they are collected in the controlled environment. Despite neural networks capabilities in mapping complex patterns among data a variable-engineering process may result effective in providing more robustness to the trained agent. Figure 2.11 shows an example of this process. Instead of providing to the agent both the measures of the controlled variable and the desired set-point at each time-step the variable-engineering process passes to the agent only information about the relative distance between these two values. Through this method is possible to reduce the number of variables in the state-space and to provide to the agent the capability to adapt to evolving conditions of the desired set-point.

2.3 Discussion of the literature review

The application of RL and DRL techniques represents a powerful opportunity to enhance HVAC systems operations considering different objectives. The applications introduced and discussed in the previous sections demonstrated the effectiveness of these control frameworks in different applications considering almost any typology of system.

Model-free controllers are capable to automatically learn an optimal policy considering different objectives through a predictive and adaptive approach. However, given the considerable amount of time required to converge, the scalability of these methods in the building industry is still an open issue. Moreover, the majority of the reviewed applications benchmarked RL and DRL controllers against traditional and conventional approaches which suffer from well-known issues. Few works introduced a comparison of model-free and model-based techniques such as MPC. Despite its scalability issues in the energy and buildings field, MPC has been successfully applied to several applications from low-level to supervisory control, such as zone temperature control in multi-zone buildings [136, 137], charging and discharging of ice-storage systems [138] and management of radiant heating systems [139]. MPC has proven to be particularly effective in managing renewable sources and energy storage systems, because of its ability to use predictions of future intermittent renewable generation [140, 141], as well as managing bi-directional energy exchange with the grid and variable energy tariffs [142]. In the current scientific literature only a few studies implemented model-based benchmarks [103, 143, 144].

In this context, despite the great interest aroused by techniques based on the RL framework this approach is still in an exploration and research phase with limited adoption in physical case studies. The application of RL frameworks in a real-world context presents several challenges with respect to a simulation environment. An issue not found in simulation environments is related to the quality of the monitored data. RL agents, and DRL agents in particular, are highly dependent on the quality of the monitored data to perform the update of the control policy. In this regard, it is fundamental to design and apply robust data preprocessing procedures to ensure high data quality. Data quality involves not only the treatment of missing values or outliers, but also the time alignment between different quantities monitored in field. Different sensors may be characterized by different sampling rates. In this context, it is important to properly organize the inputs provided to the agent. Another issue

is related to monitoring the variables necessary for the agent to learn an optimal control policy. While electrical energy and indoor air temperatures are easy to collect, operational temperatures of heat carrier fluids, flow rates, and thermal energies are not always monitored. Eventually, the selection of the optimal algorithm, of the state and action spaces along with the reward function requires a considerable amount of expertise in both building physics and artificial intelligence. Thus, extending the existing body of knowledge on RL-based control strategies for HVAC system providing new evidences and innovative perspectives on the implementation these techniques could pave the way for the progressive introduction of advanced control strategies as an industry standard in the field.

This dissertation seeks to fulfill this goal through the development of four innovative DRL applications considering different HVAC systems and control objectives leveraging both building physics and artificial intelligence expertise. Figure 2.12 shows a flowchart of the steps followed in the definition of the DRL control agents of the four different applications developed through this dissertation.

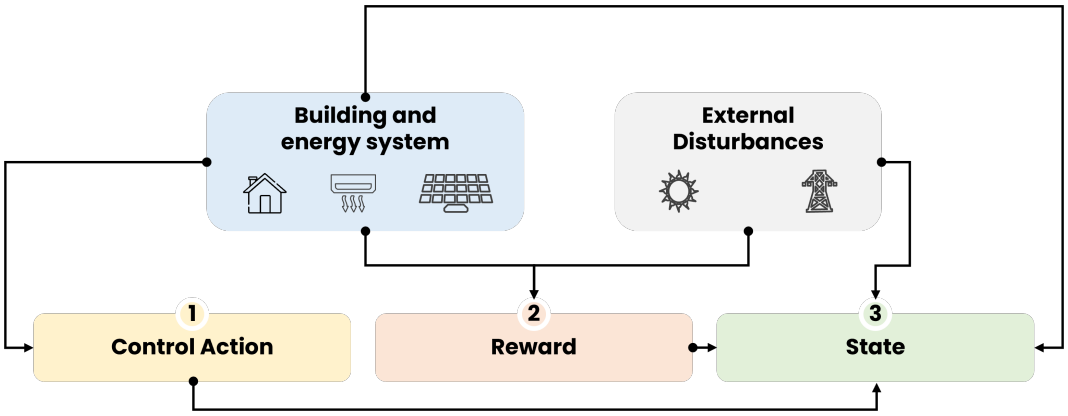


Fig. 2.12 Flowchart reporting the steps followed in the definition of RL control problems applied to HVAC systems control.

Once defined the controlled environment characterized by the building and the external disturbances (i.e. weather and interaction with the grid) the first step involved the definition of the control action. The control action can significantly influence the development of the DRL controller. The action is closely related to the energy system since it depends on the actuators and the set-points that can be actively managed. The reward function is defined in the second step. The reward depends on both energy system features and external disturbances. The third step

involved the definition of the states. Since states represent the inputs of the agent they must carry all the necessary information required to effectively map the optimal control policy. It would be impossible to design the inputs without correctly defining the outputs and the goal of the controller.

Chapter 3

Co-simulation environment

This chapter introduces in detail the co-simulation environment developed in the framework of this dissertation.

Portions of the present Chapter were already published in the following scientific papers:

- Brandi S., Piscitelli M.S., Martellacci M., Capozzoli A. 2020. *Deep reinforcement learning to optimise indoor temperature control and heating energy consumption in buildings*. Energy and Buildings 224, 110225. [24]
- Coraci D., Brandi S., Piscitelli M.S., Capozzoli A. 2021. *Online Implementation of a Soft Actor-Critic Agent to Enhance Indoor Temperature Control and Energy Efficiency in Buildings*. Energies 14, 997. [25]
- Brandi S., Gallo A., Capozzoli A. 2022. *A predictive and adaptive control strategy to optimize the management of integrated energy systems in buildings*. Energy Reports 8, pp: 1550-1567. [26]
- Brandi S., Fiorentini M., Capozzoli A. 2022. *Comparison of online and offline deep reinforcement learning with model predictive control for thermal energy management*. Automation in Construction 135, 104128. [27]

3.1 Development of the co-simulation environment

The co-simulation environment described in this section was developed during the first half of 2019 and combines Python and EnergyPlus. This latter is one of the most widely applied energy simulation software worldwide by both researchers and building professionals. EnergyPlus is completely open-source.

In the development process of a physics-based model of a building system through EnergyPlus one of the most challenging parts is the construction of the geometric model. To this purpose different graphical interfaces can be employed to reduce the development time. The two most widely applied graphical interfaces for EnergyPlus are OpenStudio [145] and DesignBuilder [146]. Moreover, these interfaces allow a rapid definition of the HVAC equipment connections. Once the building and HVAC model is defined the EnergyPlus input data file (idf) can be extracted and employed.

One of the main challenges to the simulation of advanced control strategies in EnergyPlus is its limited capability to integrate user defined logic and the impossibility to rely on external modules. To overcome this limitation EnergyPlus can be interfaced with other software such as Python. In this implementation the interface between these two software is managed by BCVTB that was introduced in the previous section. BCVTB relies on the *ExternalInterface* object of EnergyPlus. Through this object is possible to set both the inputs received from BCVTB and the outputs that are send to the interface at each time-step. The developed co-simulation environment relies on the pyEp [147] library which enables the utilization of the ExternalInterface in Python. Moreover, it provides an EnergyPlus-OPC bridge service that exposes EnergyPlus simulation variables as an OPC tree tag structure. The pyEp library allowed to wrap up the building model in a Python class based on OpenAI Gym [148].

The interaction between the two software is dynamic, and during a simulation a continuous exchange of data take place. The data flow is characterized by the following temporal features:

- *Control time-step*: it represents the frequency at which control actions are forwarded from Python to EnergyPlus. With same frequency, EnergyPlus outputs are provided to python in order to determine the next control action.

- *Simulation time-step*: it is defined in the EnergyPlus environment and it is not directly linked to control time-step. For example, if the simulation time-step is set equal to 5 minutes and the control time-step is set equal to 15 minutes, as a result, a control action occurs every 3 simulation time steps. In this case, the same control action is repeated for multiple simulation time-step. This procedure can be useful in order to let the EnergyPlus model to gradually converge to the new state with an higher accuracy.

- *Episode*: it is a simulation time period performed by EnergyPlus. One episode (or one simulation) is repeated multiple times during the training phase of the agent in order to allow the exploration of different trajectories. Conversely, an episode in the deployment phase is performed once in order to simulate the deployment of a trained control agent. Training and deployment episodes may differ, for example an agent can be trained on a heating season relative to one year and deployed in the heating season of the successive year.

Figure 3.1 shows the information flow that occurs during a simulation of DRL control interacting with the EnergyPlus simulation model. The entire process is handled through a *main* Python file which import the various components required to run the co-simulation environment.

The first module being imported in the main script is the Python class including the EnergyPlus simulation model and interface. The initialization procedure of this class is handled through the *init()* method which is invoked at the beginning of the *main* script. The arguments of this method can include any hyper-parameter that the user desire to set for the simulation process. The list of arguments may include the length of an episode expressed as the number of control time-steps, reward function weight coefficients, maximum and minimum values employed to re-scale variables included in the state-space before being fed to the neural network.

Before effectively running the simulation, the *main* file initializes the control agent. This agent can be built using different approaches as introduced in section 2.2.

The co-simulation process starts with the *reset()* method. This method initializes one EnergyPlus simulation procedure which includes the warp-up period and returns the initial state of the environment. This method is automatically invoked at the end of each episode in order to re-start the simulation from the same starting point. The

state returned by this method, since it is directly returned by EnergyPlus, is defined as physical quantities and must be processed before they are provided to the DNN of the DRL agent.

The process continue by forwarding the action picked by the DRL agent back to the simulation model. This procedure is carried out through the *step(a)* method. The only argument of this method is the action *a* selected by the control policy. Within this method, the action value which is in an encoded form is translated into a physical control action. This latter value is forwarded to EnergyPlus models which simulates the successive time-step. The new state is processed through the method and the reward value is calculated. These two values represent the output of the method which are passed to the *main* script and the DRL agent. It is important to remember that state variables may include not only actual values collected directly from EnergyPlus but also historical values and forecasts of external disturbances. This aspect was integrated within the co-simulation environment in the *step(a)* method. The possibility to integrate forecasts as Python class attributes was provided in the *init()* method while historical values can be stored directly in the *step(a)* method. The process continues until the end of an episode is reached. It is worth remembering that the length of an episode can be arbitrarily chosen, and it is defined within EnergyPlus model. The green lines in the figure highlight the flow of data exchanged between Python and EnergyPlus that is handled through BCVTB.

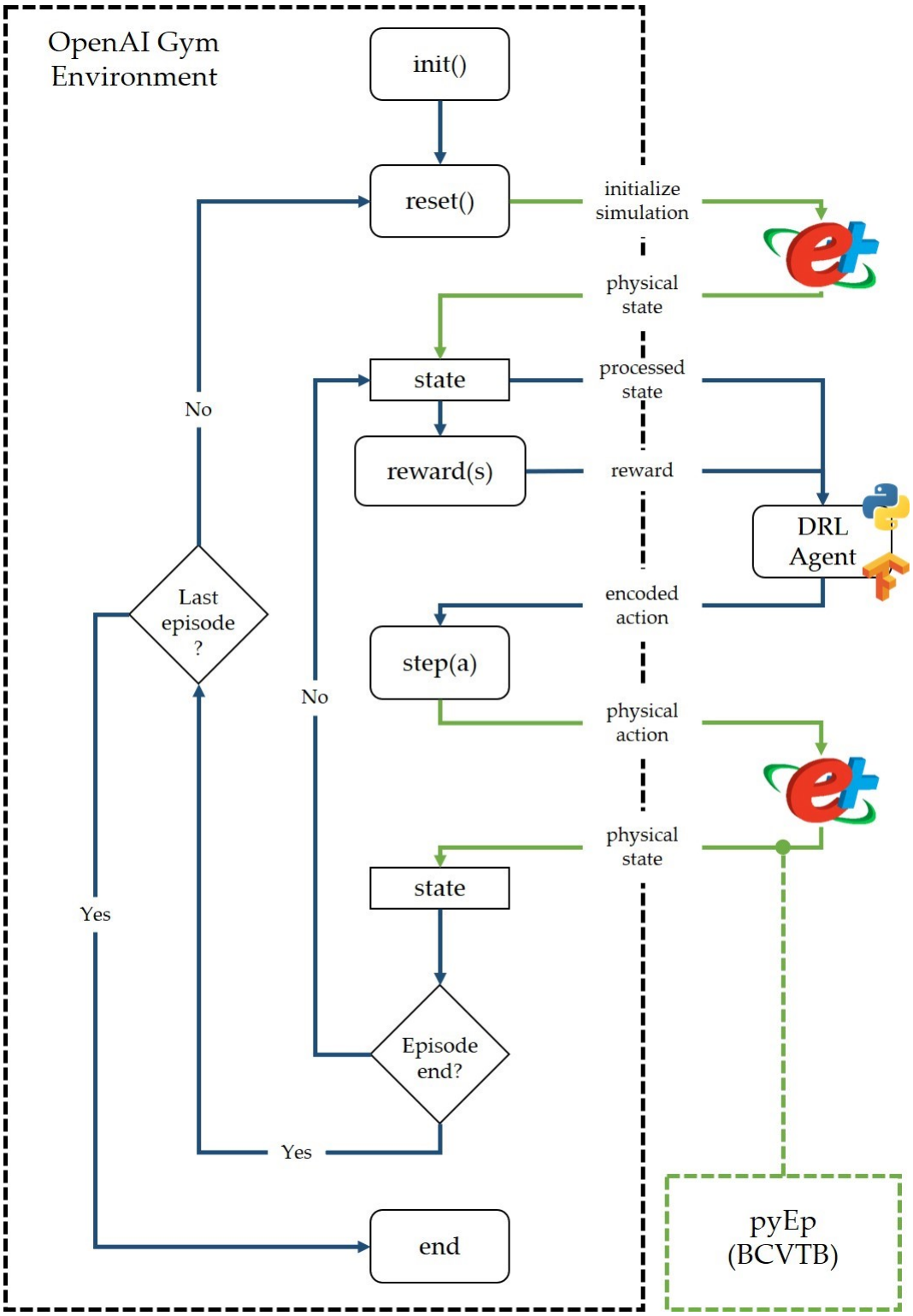


Fig. 3.1 Architecture of the co-simulation environment for RL control in HVAC systems [24].

3.2 Co-simulation environments from the current scientific literature

As introduced in the previous section the simulation environment presented in the framework of this dissertation was developed in the first half of 2019 and was used to implement most of the applications that will be presented in the next chapters.Several environments and co-simulation libraries for the analysis of advanced control strategies in buildings have been publicly released during this period.

In order to guide and facilitate the reader, the environments and tools identified in current literature are listed below:

- **BOPTEST** [149]: is a building operations testing framework which includes several building models developed in Modelica with different HVAC system configurations for different climatic zones. BOPTEST allow the interaction of control algorithms with the models through a pre-defined API system. Moreover, it includes a series of key performance indicator (KPI) of the performance of control strategies and forecasts of external disturbances.
- **Energym** [150]: is a Python-based library that includes 11 simulation models developed with both Modelica and EnergyPlus and ranging from residential to office case studies. The aim of Energym is to provide a standardized environment to test climatic control and energy management strategies.
- **AlphaBuilding** [151]: is a simulation test-bed for a medium size office wrapped up in a OpenAI Gym interface. The building model is taken from DOE commercial reference building type. The environment comes with several implementations of DRL algorithms.
- **CityLearn** [152]: is an OpenAI Gym environment developed for testing multi-agent DRL agent for the coordinated energy management of districts and cities. At the present time, CityLearn includes simplified building models developed in Python employing pre-defined building loads. The models include domestic hot water, heat pumps, chilled water and PV.
- **Advanced Controls Test Bed (ACTB)** [153]: is a test-bed which enables the interface between external controllers and high-fidelity Spawn of EnergyPlus

models. The two libraries used to interface control strategies are *do-mpc* for model predictive controllers and OpenAI Gym for reinforcement learning controllers (RLC). Spawn of EnergyPlus is a model-exchange framework that allows the simulation of building envelope and internal gains models in EnergyPlus, and their HVAC systems and controls in Modelica. The ACTB is based on the BOPTEST framework.

Chapter 4

DRL applications in HVAC systems

This chapter discusses in detail the development of deep reinforcement learning applications to HVAC system control. The focus is on the definition of the control problem for different configuration of the HVAC system and of the training and deployment strategies of control agents.

Portions of the present Chapter were already published in the following scientific papers:

- Brandi S., Piscitelli M.S., Martellacci M., Capozzoli A. 2020. *Deep reinforcement learning to optimise indoor temperature control and heating energy consumption in buildings*. Energy and Buildings 224, 110225. [24]
- Coraci D., Brandi S., Piscitelli M.S., Capozzoli A. 2021. *Online Implementation of a Soft Actor-Critic Agent to Enhance Indoor Temperature Control and Energy Efficiency in Buildings*. Energies 14, 997. [25]
- Brandi S., Gallo A., Capozzoli A. 2022. *A predictive and adaptive control strategy to optimize the management of integrated energy systems in buildings*. Energy Reports 8, pp: 1550-1567. [26]
- Brandi S., Fiorentini M., Capozzoli A. 2022. *Comparison of online and offline deep reinforcement learning with model predictive control for thermal energy management*. Automation in Construction 135, 104128. [27]

4.1 Optimization of indoor temperature control and energy consumption in heating systems

DRL has recently gained popularity among RL algorithms due to its ability to adapt to very complex control problems characterized by a high dimensionality and contrasting objectives. DRL employs deep neural networks in the control agent due to their high capacity in describing complex and non-linear relationship of the controlled environment. Despite the advantages provided by the implementation of DRL as a control method for HVAC systems, some major drawbacks in the design and the training process of the DRL agent need to be further explored.

The next section presents the main research challenges analyzed in this application and introduces the motivations and novelty of the proposed methodological approach.

4.1.1 Motivations and novelty of the proposed approach

A DRL agent is characterized by a number of hyper-parameters that need to be carefully tuned depending on the specific case study and objective functions [39]. As a consequence, despite its model-free nature, DRL requires a sort of modeling effort in its initial state to find the set of hyper-parameters which may lead to the learning of a control policy close to the optimum in less time as possible and with an acceptable uncertainty [75]. In the existing literature an analysis on the effect of the hyper-parameters settings on the performance of the control strategy was poorly investigated. Moreover, two opposite approaches can be followed when deploying a DRL agent previously trained offline: static deployment and dynamic deployment as introduced in Chapter 2. Moreover, in the design of the DRL a proper selection of the variable set which describe the environment is particularly important, considering it represents the environment as it is observed by the control agent. The effect of variable section on the adaptability capability of the DRL controller need to be further explored respect to the exiting literature.

The present application focuses on the development of a DRL agent to control the set-point of supply water temperature to heating terminal units system serving a thermal zone of an office building. The main scope of the application is to extensively test the operation of a robust agent by exploring its adaptability to the variation of

forcing variables such as weather conditions, occupant presence patterns and different indoor temperature set-point requirements. The analyses were conducted considering both a static and dynamic deployment with the aim of underlining limitations and opportunities. Moreover, two different sets of input variables (with an adaptive and non-adaptive approach respectively) were analyzed for assessing the impact of variable selection process on the adaptability capabilities of the RL controller.

On the basis of the literature review on RL and DRL control in HVAC systems presented in Chapter 2 the main innovative contributions that this application intends to provide can be summarized as follows:

- The control performance of a DRL agent was analyzed both in terms of indoor temperature control and energy consumption against a baseline controller implementing a climatic-based logic of supply water temperature set-point and a rule-based control of heating system operation.
- The design of a DRL agent was conducted performing a tuning of the hyper-parameters which may strongly affect the control performance of the agent.
- A proper variable selection was proposed to prevent the agent from learning an overfitted control policy. When a DRL agent is developed, in most of the cases the input variables describing the controlled environment are not defined to provide information to the agent in an adaptable manner with respect to control objectives. To this purpose, the variable selection process was performed both with adaptive and non-adaptive approach in order to produce an effective comparison.
- The two approaches of DRL deployment, static and dynamic, were compared in four different deployment scenarios to assess the adaptability of the agent to the variation of forcing variables such as weather conditions, occupancy patterns and different indoor temperature set-point requirements.

The rest of the section is organized as follows. Section 4.1.2 presents the methodological framework adopted to test the DRL controller. Section 4.1.3 introduces the case study and the control problem. Section 4.1.5 presents the results obtained for the analyzed case study. Section 4.2.6 discuss the results and their implications.

4.1.2 Methodological framework of the application

In this section the methodological framework is presented with the aim of introducing each stage of the DRL control agent development. The present framework unfolds over three different stages as shown in Figure 4.1.

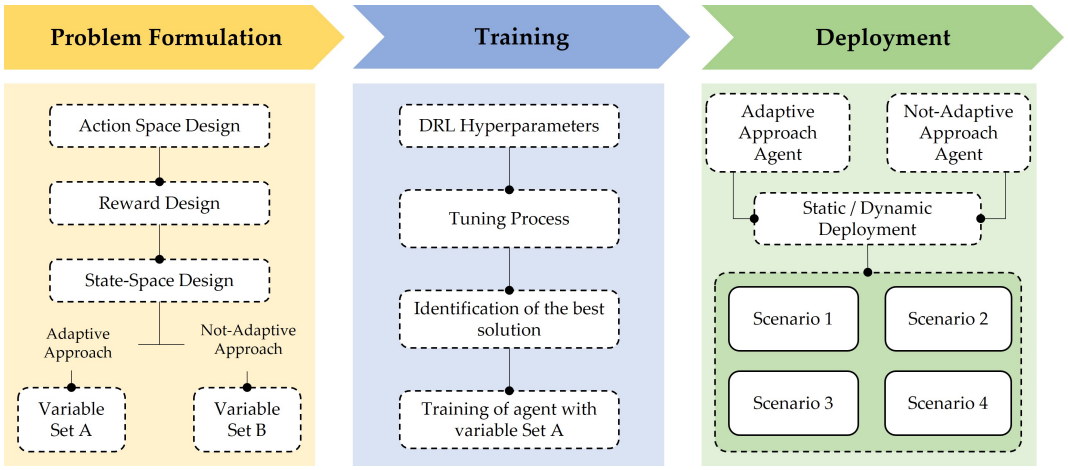


Fig. 4.1 Framework of the application of DRL control [24].

Problem formulation: the first stage of the framework was aimed at defining the main components of the reinforcement learning control problem. The action-space includes all the possible control actions that can be taken by the control agent. Considering that a Deep-Q-learning was implemented, the action space is discrete. The reward is a function which describes the performance of the control agent with respect to the control objectives. Finally, the state-space is a set of variables related to the controlled environment which are fed to the agent in order to learn the optimal control policy which maximizes the reward function. The state-space was formalized following two approaches. In the first approach (*Adaptive*), the variables were selected in order to make them flexible to possible changes in the controlled environment (*Variable Set A*). In the second approach (*Non-Adaptive*), the selected variables are equally representative of the state of the environment but do not follow an adaptability paradigm (*Variable Set B*).

Training: in the second stage of the process the DRL agent was trained. As introduced in section 3 reinforcement learning agents are characterized by many hyper-parameters which require appropriate tuning. In this stage, a tuning was carried out on some of the most important hyper-parameters by training the agent

with different configurations, in order to analyze the variations in the results obtained. The training process was implemented in an offline fashion using a training episode (i.e. a time period representative of the specific control problem) multiple times to constantly refine agent's control policy.The training episode was expressly chosen in order to not completely cover the full state-action space of the present control problem. In this way, it was possible to evaluate the adaptability of the agent to climatic conditions never explored during training during the deployment phase.The hyper-parameter tuning process was performed for an agent implementing the variable set A. The best configuration of hyper-parameters resulting from the analysis was successively employed to train the agent with variable set B.

Deployment: the resulting agents, one trained on adaptive approach (using variable set A) and the other one trained with non-adaptive approach (using variable set B), were tested in the last stage. Both agents were tested through a static and dynamic deployment in one episode which includes a different period (i.e. weather conditions) from the training episode. Moreover, the deployment was performed in four different scenarios including different occupant presence patterns and indoor temperature requirements from the training stage. Eventually, a comparison of the performance obtained with the different approaches was proposed.

4.1.3 Description of the case study

The DQN algorithm described in section 2.1.2 was implemented to control the water supply temperature of a heating system for a simulated office building. In the following sub-sections, a description of the case study together with the formulation of the control problem are provided.

Description of the building

The simulated building is representative of a huge portion of the Italian building stock in terms of both heating system configuration and building construction features. It is a six-level mixed-use building with a net heated surface of 9300 m^2 located in Turin, Italy. The indoor environment is heated through water terminal units (i.e., radiators). The building is composed of three thermal zones served by different hot-water circuits and was built between 1930 and 1960. The average transmittance values of the opaque and transparent envelope components are respectively 1.084

and 2.921 W/m^2K. The ratio between heat transfer surface and gross volume (i.e, aspect ratio) is equal to 0.25 m^{-1}. The implementation of the DRL controller is tested for one thermal zone which includes only office rooms. This zone is composed of four-levels with a net heated surface of 7000 m^2 and a net heated volume of 33000 m^3. The remaining zones are occupied by the local police department and the warden of the whole building. Figure 4.2 shows a picture of the real building and highlights the thermal zone modelled in this application.

Fig. 4.2 Office case study located in Torino, Italy. Detail of the office zone modelled in this application [24].

Heating system and control objectives

The heating system installed in the real building is quite complex. It is composed by two hot water loops connected by a heat exchanger. The primary loop includes four gas-fired boilers with a total nominal capacity of 1300 kW. The secondary loop includes three zone-loops served by different pumping systems. The three zone-loops withdraw hot water from the same water collector. The control of the supply water temperature is achieved through three-way mixing valves. However, EnergyPlus does not reach this level of complexity in the definition of the HVAC system and some simplifications were introduced to model the building.

In the present case study, the control problem focuses on the regulation the supply water temperature (T_{SUPP}) to heating terminal units of a single thermal zone.

The heating system was modeled in EnergyPlus with a single hot water loop. The supply side includes a single gas fired boiler (*Boiler:HotWater*) and a constant speed pump (*Pump:ConstantSpeed*). The supply water temperature set-point (SP_{TSUPP}) was managed through a *SetPointManager:Scheduled* which directly receives inputs from Python through the *ExternalInterface*. The demand side includes one thermal zone and its relative bypass branch. The goal of the control policy is to reduce the amount of thermal energy provided to the supply water while maintaining indoor air temperature within an acceptability range during occupied periods. This application, even being developed in a simulation environment in which every thermal comfort-based parameter can be easily evaluated, considers only the zone air temperature (T_{ZONE}). In fact, other comfort related-variables are not monitored in the real building. Moreover, the water terminal units can control only the sensible part of the thermal load. If the zone air temperature value falls between upper and lower threshold of a pre-defined acceptability range, then indoor temperature requirements are satisfied. In this application the acceptability range was defined in the interval [-1,1] °C from the desired indoor temperature set-point (SP_{TZONE}). The application focuses on the energy supplied for heating the carrier fluid (Q_{SUPP}) regardless the type of the generation system serving the building. Technically, in real life implementations, the regulation of supply water temperature can be achieved through different solutions such as three-way mixing valves or by modulating boiler or heat pumps. The control policy developed through the presented approach could be then employed independently by the actual generation system installed. Figure 5 provides a simplified scheme of the heating system and of the control problem formulation.

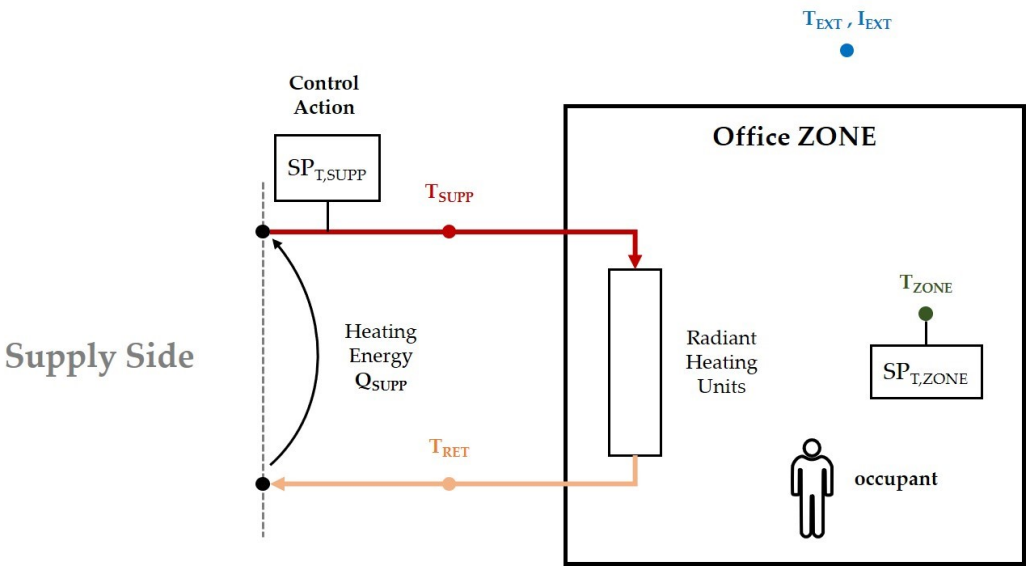


Fig. 4.3 Schematic of the heating system analysed [24].

Baseline control strategy

The performance of the DRL control was evaluated against a baseline control logic implementing a combination of rule-based and climatic-based logics for the control of the supply water temperature. The starting time of the heating system was determined according to the value of indoor temperature and the amount of time before the occupant's arrival. The controller is enabled to turn on the heating system up to four hours before the arrival of the occupants if the difference between the actual indoor temperature and the low threshold of the acceptability range is higher than 3 °C, or up to three hours before if that difference is higher than 2 °C. In any other case the controller turns on the heating two hours before occupant's arrival if the zone temperature is lower that the low threshold of the acceptability range. When the zone reaches the upper threshold of the acceptability range the heating system is turned off. If the zone temperature falls below the lower threshold the heating system is turned on again. This control strategy is operated until two hours before occupants leave the building, when the heating system is turned off to exploit thermal inertia until the next day. The supply temperature value is linearly interpolated between a maximum value of 70 °C when the outdoor air temperature falls below -5 °C and a minimum value of 40 °C when the outdoor air temperature is over 12 °C. These

values were selected according to the control logic of the supply temperature actually implemented in the Energy Management System of the real building.

4.1.4 Design of the DRL controller

The DRL control algorithm described in section 2.1.2 was trained and tested in the developed simulation environment. In the next sub-sections, the design of the action space and of the reward function are discussed along with the configuration of the training and deployment phases.

Design of the action-space

At each control time-step the agent selects a value of supply temperature set-point (SP_{TSUPP}). Considering that the DQN was chosen as control agent the action-space is expressed in a discrete space. The space includes the following actions related to the supply water temperature in °C:

$$A_t = [20, 40, 50, 60, 70] \tag{4.1}$$

These values where selected in order to provide to the DRL agent the same range of supply water temperature set-point as the baseline controller. At the same time, the values were selected to limit the actions to only five values in order to not over-complicate the control problem formulation. Given the inertia of the water-based heating system intermediate values of supply water temperature can be reached by the agent switching between available control actions during system operation. The introduction of intermediate values of set-point supply water temperature in the present action-space (e.g. 45 °C, 55 °C, 65 °C) would have only increased the complexity of the calculations performed by the neural network model [154] without effectively producing an improvement on the learned control policy. The simulation environment was set in order to shut down circulation pump when the supply water temperature value falls below 20 °C.

Design of the reward function

The reward that the agent receives after taking an action at each control time-step depends by two competing terms: the energy and temperature-related terms. The energy-related term is proportional to the energy provided to supply water to reach the desired set-point. Unlike other applications where the energy-related term is purely intensive, in this study this term was normalized with respect to the temperature difference between zone temperature set-point and outdoor air temperature. This formulation was found effective in accelerating the convergence of the developed agent to a near-optimal solution. Through this approach the agent is not excessively penalized for taking energy-intensive actions when the outdoor temperature is very low and vice-versa. Moreover, this formulation can represent a robust approach to the development of DRL control agents since it is less sensitive to extreme weather conditions and modifications of indoor temperature set-point.

The temperature-related term is quadratically proportional to the distance between zone air temperature set-point and its actual value. This formulation was found to be effective in speeding up the learning process, making the agent able to easily avoid the exploration of states characterized by unacceptable conditions of the indoor environment from the very beginning of the training phase. The formulation of the reward function is expressed by the following equation:

$$R = -\beta * \frac{Q_{SUPP}}{SP_{TZONE} - T_{EXT}} - \rho * |(SP_{TZONE} - T_{ZONE})^2|_{OCC=1} \tag{4.2}$$

The coefficients ρ and β were introduced to weight the importance of the two terms of the reward function.

Design of the state-space

The state represents the environment as it is observed by the control agent. The agent, at each control time-step, choses among the available actions according to the values assumed by the state. In this application, two different state-space were designed as introduced in section 4. The first one includes a set of input variables (variable set A) selected in order to guarantee the maximum adaptability of the learned control policy. Part of these variables were obtained by applying a variable-engineering process as illustrated in section 2.2.5. The second state-space, instead, is composed

by a set of input variables (variable set B) which do not follow an adaptive approach. These variables are provided in input to the control agent as they are collected from the environment without performing any variable-engineering process. In both cases the variables were selected according to the following criteria:

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

The two variable sets are reported in Table 4.1 and Table 4.2 respectively. Overall, the adaptive set (*variable set A*) includes 11 variables while the not-adaptive set includes 13 variables (*variable set B*).

Table 4.1 Variables included in the variable set A conceived with an adaptive approach [24].

Variable	Min Value	Max Value	Unit	Time-step
ΔT Indoor set-point – external air	6	31	°C	t
Direct solar radiation	0	720	W/m^2	t
Supplied heating energy	0	125	kWh	t
Supply water temperature	10	80	°C	t
Return water temperature	10	80	°C	t
Time to occupancy start	0	36	h	t
Time to occupancy end	0	12	h	t
ΔT Indoor set-point – indoor air	-3	10	°C	t,t-1,t-2,t-3

Table 4.2 Variables included in the variable set B conceived with a non-adaptive approach [24].

Variable	Min Value	Max Value	Unit	Time-step
Time of the day	0	24	h	t
Day of the week	1	7	-	t
External air temperature	-12	26	°C	t
Direct solar radiation	0	720	W/m2	t
Supplied heating energy	0	125	kWh	t
Supply water temperature	10	80	°C	t
Return water temperature	10	80	°C	t
Occupants' presence status	0	1	-	t
Indoor set-point	13	25	°C	t
Indoor air temperature	13	25	°C	t,t-1,t-2,t-3

External air temperature and direct solar radiation were both included in variable set B, as they are the most influencing ambient variables affecting building heating energy consumption and indoor temperature. On the contrary, in the feature set A, external air temperature was substituted by the temperature difference between indoor set-point and external air since it is directly related to the formulated reward function. This formulation was found to be effective in removing the dependency of the learnt control policy from a fixed value of indoor temperature set-point which could limit agent adaptability.

The supplied heating energy was selected considering that it is proportional to the energy-related term in the reward function and it represents a key information that has to be provided to the agent. Moreover, the heat supplied to the water depends by the supply water temperature and by the return water temperature. These variables, which represent the main operational parameters of heating system, were included in both the variable sets.

Information about the presence of occupants in the zone, from which depends the temperature-related term in the reward function, is provided through three different variables. The occupants' presence status, added in the set built following not-adaptive approach, indicates if, in a certain control time-step, the zone is occupied or not (it depends only by the occupancy schedule) and it is expressed in the range [0,1]. However, this information alone is not comprehensive. It would be desirable

for the agent to learn when it is convenient to pre-heat the zone so as to ensure an adequate indoor air temperature during occupancy period. A common approach to this problem in the literature, implemented in the non-adaptive set, is to select as variables time-of-the-day and day-of-the-week. However, following this procedure, the agent may learn to fit only to a specific occupancy-schedule provided during the training process. To overcome this issue, the variables time to occupancy start and time to occupancy end were introduced in the variable set A to define the time left for the subsequent change in the occupancy pattern. When the building is not occupied, time to occupancy start represent the number of hours left before occupants' arrival time, during occupancy periods this variable is equal to 0. Conversely, when the building is occupied, time to occupancy end represent the number of hours to occupants' leaving time, during off-occupancy periods this variable is equal to 0.

Eventually, the agent needs information about the zone air temperature which is directly connected with the temperature-related term of the reward function. This information was straightforwardly added to the variable set B along with its 3 lagged values in the past (15, 30 and 45 minutes lag respectively) and the indoor set-point. Contrarily, in variable set A, this information was provided indirectly introducing as variable the difference between the zone air temperature and indoor set-point along with its 3 lagged values in the past (15, 30 and 45 minutes lag respectively).

The relative humidity was not included in the two set of variables considering that the heating system based on water radiators is capable to control only the sensible part of the heating load.

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

Setting of the training phase

The Reinforcement Learning framework is characterized by a number of hyper-parameters that strongly affect the behavior of the control agent. In order to analyze their impact on the performance of the control agent, different configurations of the most interesting hyper-parameters were tested and compared in this study. The configurations implemented for the training of the DRL agent are described in the following tables.

The hyper-parameter tuning process was performed only with the agent implementing the state space built following adaptive approach (variable set A). In Table 4.3 are listed the values of the hyper-parameters kept unchanged during the training.

Table 4.3 Fixed hyper-parameters of the DRL agent training [24].

Hyper-parameter	Value
DNN architecture	4 layers
Neurons per hidden layer	512
DNN optimizer	RMSprop
Optimizer learning rate	0.0001
DQN batch size	32
Episode length	5856 control steps (61 days)
Sequential memory size	5 episodes
Target model update	672 control steps (7 days)
Number of episodes	50
Boltzmann temperature (τ)	1
ε start	1
ε end	0.1
Energy related term weight factor (β)	1

Although hyper-parameters such as neural network architecture and optimizer learning rate can influence the learning capabilities of a DRL agent, in this application these values were selected according to the experience and guidelines provided in the current scientific literature for similar applications.

The two hyper-parameters involved in the tuning process are the discount factor and the weight factor of the temperature-related term (ρ). The discount factor determines the importance of future rewards over immediate rewards and directly affects the magnitude of Q-values. The weight factor of the temperature-related term of the reward function (ρ) defines the relative importance of indoor temperature requirements with respect to energy consumption. Lower values may result in a control policy which guarantees higher energy saving at the expense of higher temperature violations and vice-versa. Table 4.4 reports the details of each hyper-parameter configuration implemented for the tuning process.

Table 4.4 Different hyper-parameter configurations implemented in the training phase [24].

run	Discount Factor γ	Weight Factor ρ
1,2,3	0.9	10
4,5,6	0.95	10
7,8,9	0.99	10
10,11,12	0.9	20
13,14,15	0.95	20
16,17,18	0.99	20
19,20,21	0.9	1
22,23,24	0.95	1
25,26,27	0.99	1

The performance of Deep Reinforcement Learning is affected by the stochastic behavior that is intrinsic in both deep neural networks and controlled environments. In order to account for this aspect, each configuration has been ran three times employing multiple random seeds in order to ensure consistency according to [155]. Successively, the hyper-parameters of the run leading to the best performance in terms of both energy savings and temperature control were selected to train also the agent implementing variable set B.

As stated in section 4.1.2, a training episode was selected to be representative of the present control problem. At the same time the training episode did not include a full state-action space in order to test the adaptability of the proposed agents during deployment. In particular, a training episode includes 2 months, from 1^{st} of November to 31^{st} of December (5856 control steps, one every 15 minutes). The weather file used in this application is the reference weather file (*ITA_TORINO-CASELLE_IGDG.epw*) available in EnergyPlus for Torino, Italy. The same weather file from the 1^{st} of January to 31^{st} of March was used for the deployment phase. As reported in Table 4.3 each training episode was repeated 50 times for each hyper-parameter configuration in order to let the agent explore several control strategies. On average one episode took 3 minutes to be simulated on a machine with an Intel Core i78550 CPU 1.80GHz processor and 16.0 GB RAM. An entire training period (including 50 episodes) for each hyper-parameter configuration took on average 150 minutes to be simulated.

Figure 4.4 shows the patterns of outdoor air temperature and direct solar radiation in the two periods (i.e. training and deployment period). For the sake of legibility, the solar radiation values include only daylight period. The training period was selected for its wide range of temperature values spanning between -8 °C and 17 °C while the direct solar radiation is higher during the deployment period. However, this latter aspect allows to test the adaptability of DRL agent different climatic patterns from those used for the training. In the training phase occupancy was simulated between 07:00 and 19:00 from Monday to Saturday. The required indoor set-point was set equal to 21 °C and the temperature acceptability range between 20 °C and 22 °C.

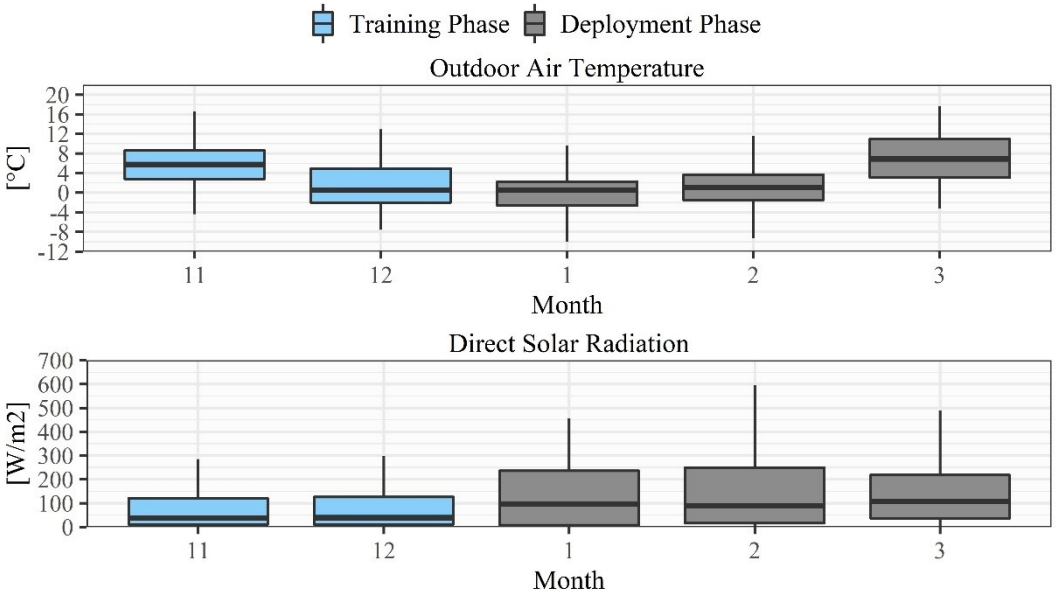


Fig. 4.4 Outdoor Air Temperature patterns during training and deployment periods [24].

Deployment phase

In the last phase of the process the two agents were deployed in four different scenarios in order to assess the adaptability capabilities of the learned control policy to different configurations related to the controlled environment. Each agent was deployed for one episode including the period between 1^{st} January and 31^{st} March. The four different scenarios are:

- Scenario S1: this is the base case where no changes in the controlled environment were implemented. The goal is to test the adaptability of the DRL

controller only to patterns of outdoor conditions (i.e. air temperature and solar radiation) never observed during the training phase.

- Scenario S2 & S3: in these scenarios the zone temperature set-point was increased to 22 °C and decreased to 20 °C respectively in order to assess the performance of the agent in satisfying temperature requirements that differ from the ones assumed in the training.
- Scenario S4: in this case the zone occupancy schedule was modified as shown in Figure 7 maintaining unchanged the zone temperature set-point respect to the training conditions. The lighting and electric appliances schedules were also changed according to the new occupancy schedule.

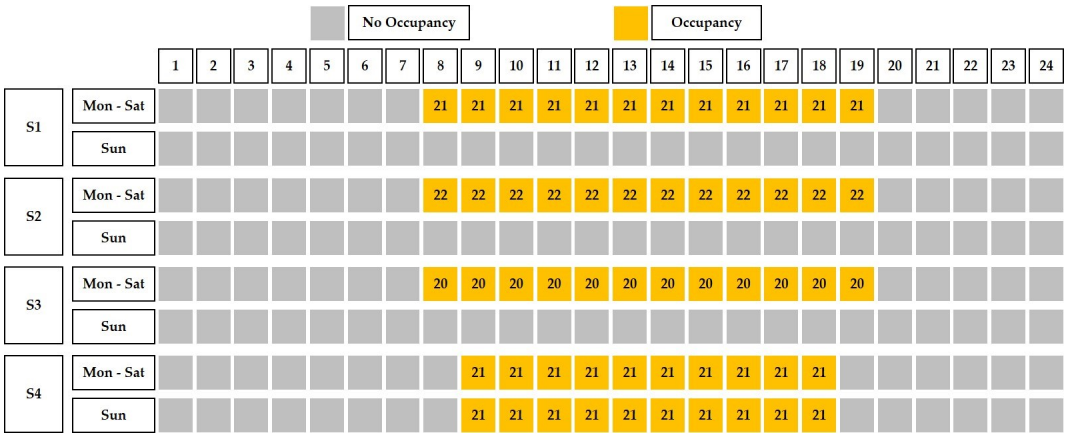


Fig. 4.5 Occupancy schedules and indoor set-point in different design conditions [24].

The trained control agents were deployed in each testing scenario in both static and dynamic configuration. In the static configuration the control policy was not updated during the deployment of the agent. Contrarily, dynamically deployed agents constantly leverage new experience obtained interacting with the environment to adjust their control policy. In particular in this configuration, the agents leveraged a new replay buffer (i.e. including tuples collected only during the deployment episode) while the update frequency of the target model of the DQN framework was lowered to 288 control time-steps (i.e. 3 days). The control policy was updated every control time-step while the values of the other hyper-parameters were left unchanged with respect to the values showed in Table 4.3. The dynamic deployment configuration, despite providing greater adaptability, requires additional computational cost and may cause instabilities in the learned control policy.

4.1.5 Results obtained

The framework presented in section 4.1.2 was implemented in the integrated simulation environment. The results are presented in this section in order to compare the performance of different DRL control agents (trained with different input variable sets and deployed following different approaches) and the baseline control of supply water temperature to terminal units of a heating system.

Results of the training process

As introduced in section 4.1.2, in the first step of the training phase a tuning process was carried out on two DRL hyper-parameters to highlight their influence on the performance of the control algorithm. The variable set based on adaptive approach introduced in section 4.1.4 was implemented for this tuning process.

A useful indicator to assess the goodness of the learning process of a DRL agent is represented by the evolution of the cumulative reward per episode. The reward, which has not a direct physical meaning, takes into consideration both the energy consumption and indoor temperature values and combines them in a single value. Higher values of the reward correspond to a better performance obtained by the control agent. It is important to supervise if the reward converges to a stable value. A non-convergent trend in the reward may be caused by an agent that failed in achieving an optimal control policy. To this purpose, the convergence of the different configurations of the agent were analyzed in the episode-reward plot showed in Figure 4.6. The figure is split into two main panels representing the evolution of the energy-related term and temperature-related term respectively. Each main panel is furtherly organized in a grid in which each sub-panel represents a specific configuration of the hyper-parameters. Each sub-panel shows the evolution of the relative term of the reward function during the training episode. The solid line shows the average value per episode of the three different runs performed for each configuration, while the grey area was drawn between maximum and minimum value per episode. In all the configurations the agent starts exploring high values of the energy-related term and extremely low values of the temperature-related term. Across the different runs, the agent firstly learns how to correctly maintain indoor temperature during the first 20 episodes; this fact can be observed by analyzing the increase of the temperature-related term values and the relative decrease of

the energy-related term. From this stage (i.e. 20^{th} episode) the agent begins to learn how to reduce energy consumption while keeping indoor temperature in the range it previously learned. In fact, the values of the temperature-related term are quite stable while the values of the energy-related term increase. Agents that were initialized with a discount factor γ equal to 0.99 represent an exception, showing highest variance in terms of temperature control performance. The training runs performed with this specific configuration ($\gamma = 0.99$) seek to obtain higher rewards in a longer time horizon compared to other agents generating an instability in the objective function. This aspect is particularly clear observing the evolution of the temperature-related term of the agent implementing a discount factor of 0.99 and a weight of the temperature-related term equal to 20. On the other hand, agents applying a discount factor equal to 0.9 shows the higher stability among all the training configurations due to the shorter time horizon considered.

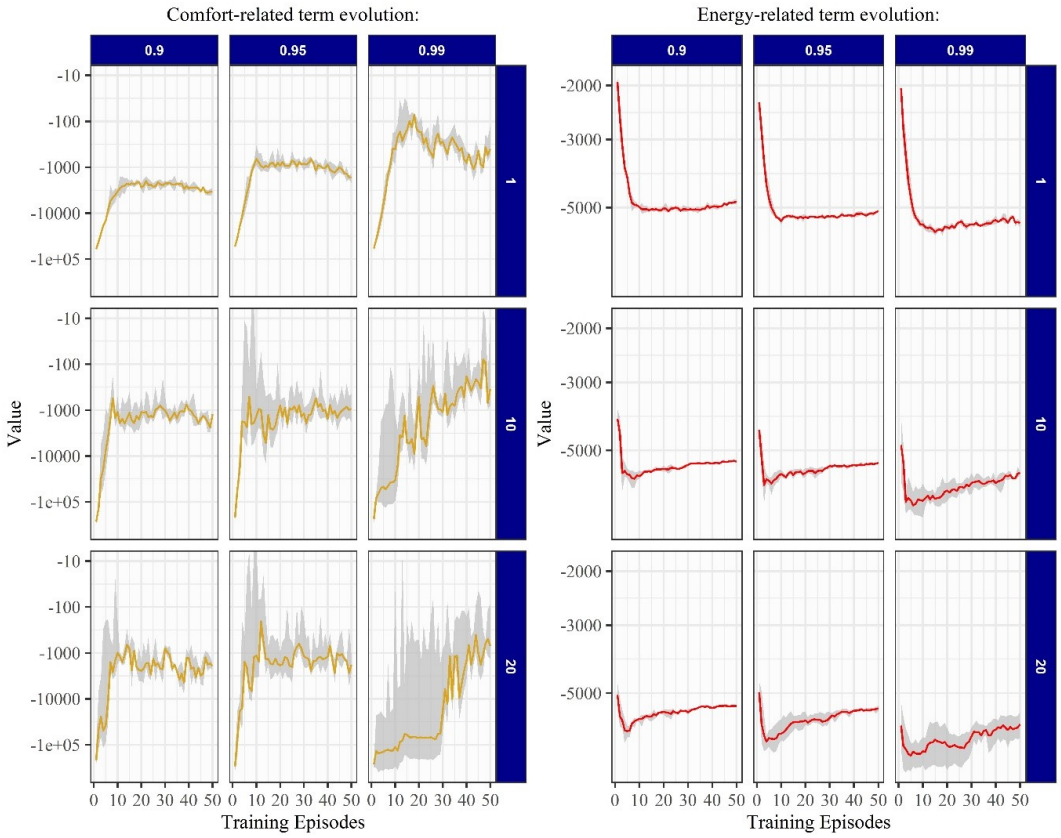


Fig. 4.6 Evolution of energy-related and temperature-related term of the reward function during training phase [24].

In this application the reward function is the weighted sum of supplied heating energy to water and temperature control performance. Therefore, the reward value alone cannot directly provide a straightforward metric to evaluate the overall performance of DRL control.

While the energy performance can be straightforwardly evaluated comparing the amount of heating energy supplied to the water, the temperature control performance requires the definition of an appropriate metric. In the present application, the indoor temperature control performance was evaluated by calculating the cumulative sum of temperature violations during occupancy hours. A temperature violation occurs when the building is occupied, and the indoor temperature falls outside the acceptability range. The magnitude of the temperature violation is then calculated as the absolute difference between actual indoor temperature and desired set point value at each simulation step. The cumulative value of this quantity over an entire episode returns the performance of the control algorithm expressed in °C.

Figure 4.7 shows, in a four-quadrant visualization, the cumulative sum of temperature violations during occupancy periods, as a function of the heating energy saving with respect to climatic-based control baseline for the different hyper-parameter configurations reported in Table 4.4.

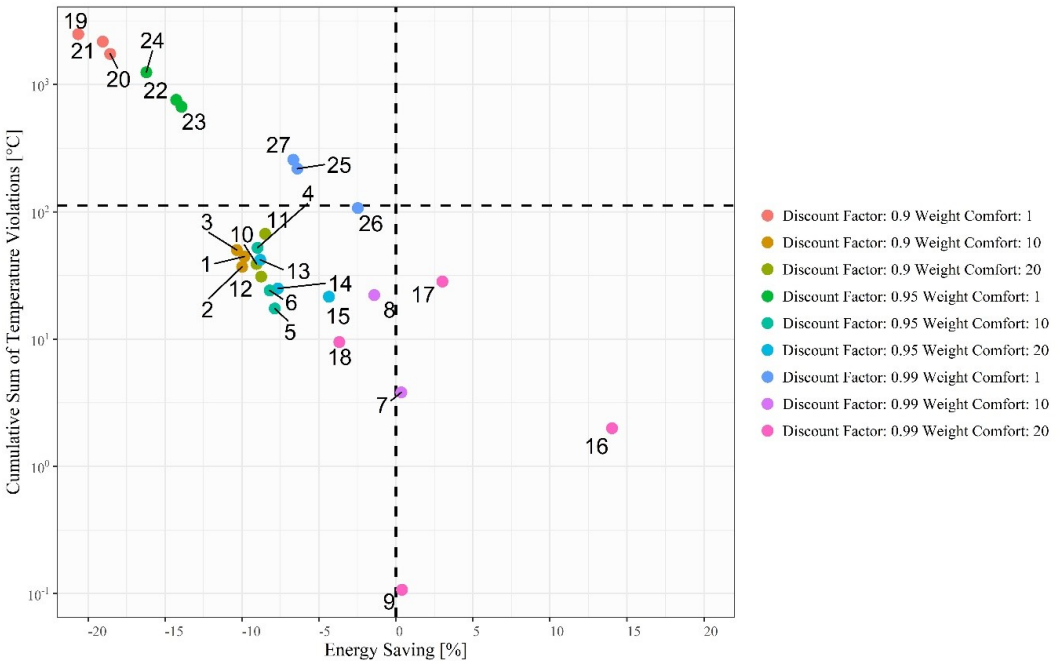


Fig. 4.7 DRL control performance in the last episode of the training phase. Each point refers to a different training runs as reported in Table 4.4 [24].

The figure reports the results obtained in the last episode (50^{th}) of the training process. For the sake of legibility of the plot the y-axis was defined on a logarithmic scale. The black-dashed lines indicate the performance achieved by the baseline controller. The left-bottom quadrant includes all the solutions that have performed better than the baseline both in terms of indoor temperature control and energy consumption. Worst solutions, corresponding to higher energy consumption and temperature violations than the baseline, should be displaced in the right-top quadrant. None of the training runs produced results that fall within this latter region. In particular, solutions with a discount factor (γ) of 0.99 and a weight of temperature-related term (ρ) of 10 (runs 7, 8 and 9) and 20 (runs 16, 17 and 18) show the highest variability. Agents trained with discount factors (γ) of 0.9 and 0.95 and a weight (ρ) of 10 or 20 lead to the best trade-off solution achieving, at the same time, energy saving and temperature control improvement. In particular, the setting of the discount factor equal to 0.9 (run 1, 2 and 3) produced the less scattered solutions. This aspect can be interpreted as an indicator of the consistency of the control policy learned by such agents. As can be expected, agents implementing a weight factor of the temperature-related term equal to 1 achieved greater energy savings at the cost of worse temperature control. Following these considerations, the agent number 2, with

a discount factor of 0.9 and a weight factor ρ of 10, was selected as best solution among configurations explored in the hyper-parameter tuning process.

In order to furtherly characterize the results of the training phase, the performance of the different solutions was analyzed on daily scale.

In Figure 4.8 are compared three agents implementing different values of the discount factor γ. The comparison is proposed for the same working day of the training episode. The figure shows the behavior of the agent when the discount factor changes while the weight factor is kept constant (ρ = 10) for the same day of the training period. Overall, in the three training runs, the agent has learnt to maintain the indoor temperature between lower and upper thresholds of the temperature acceptability range as can be observed from the central panels of the figure. However, in the solution obtained considering a discount factor equal to 0.9, the agent learnt to better maintain the indoor temperature across lower threshold of the acceptability range. As can be observed from the left figure, the run performed with a discount factor of 0.99 considerably anticipated the start-up phase resulting in higher energy consumption compared to other solutions. Given the higher discount factor, this agent learnt how to optimise the rewards stream in a longer horizon causing higher instability. The agent implementing a discount factor of 0.9 selected higher values of the supply water temperature during the first hours of the morning. As a result, the zone air temperature reached exactly the lower threshold of the acceptability range (20 °C) at the beginning of the occupied period (07:00). This agent led to a heating energy saving of about 100 kWh in comparison with the agent implementing a discount factor of 0.95 that shows a similar pattern of indoor air temperature.

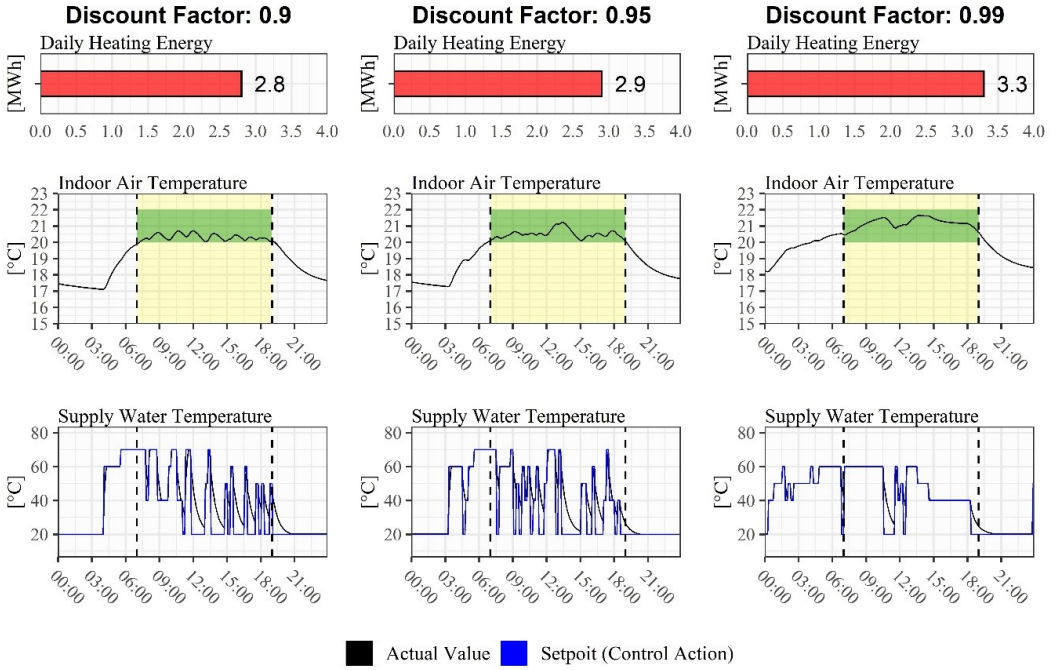


Fig. 4.8 Comparison between agents implementing different discount factors during a training day [24].

Figure 4.9 reports the performance of the trained agents considering different values of the weight factor ρ and a constant discount factor ($\gamma = 0.9$). It is possible to notice the relative importance given to temperature violations obtained in the three different solutions. In detail, the agent trained with a weight factor equal to 1 sacrificed indoor temperature control at the beginning and ending of the occupancy period. However, this agent obtained a further daily energy saving of about 100 kWh, respect to the previously discussed solution ($\rho = 10$, $\gamma = 0.9$), at the cost of keeping indoor air temperature 1°C below the lower threshold of the acceptability range at 07:00 and 19:00.

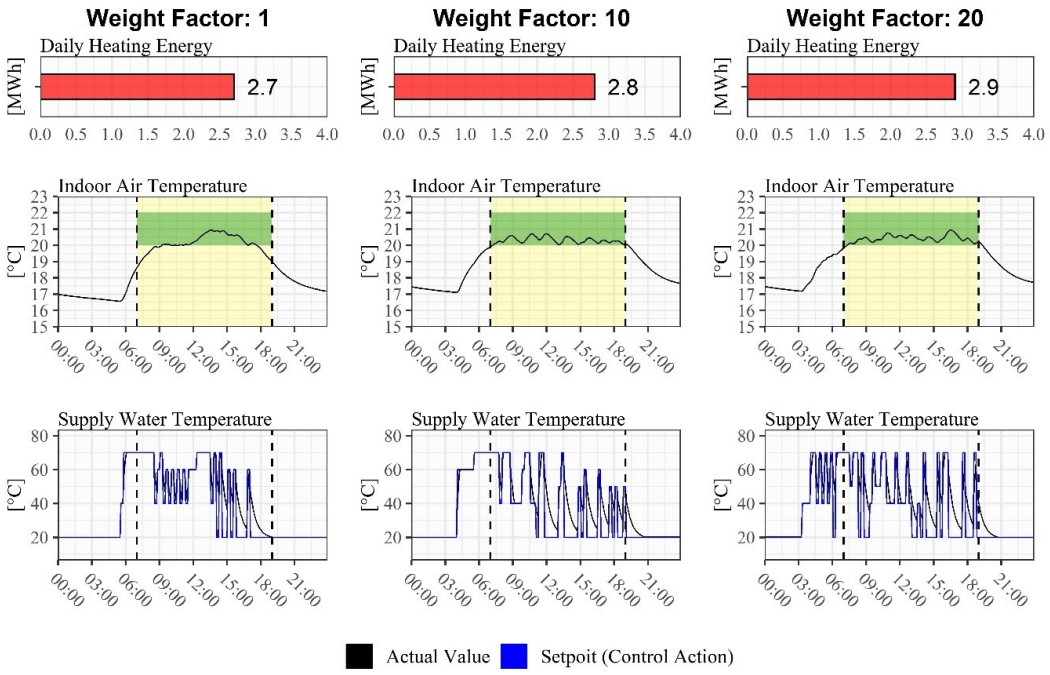


Fig. 4.9 Comparison between agents implementing different weight factors of the temperature-related term during a training day [24].

At the end of the training phase, the same hyper-parameter configurations of the best solution resulting from tuning process (i.e., discount factor $\gamma = 0.9$ and weight factor $\rho = 10$) were employed to train a second agent with the variables of the state-space selected following the non-adaptive approach (variable set B). Table 4.5 report the performances of the two agents relative to the last (50^{th}) training episode which lasts for 2 months between the 1^{st} of November and 31^{st} December.

Table 4.5 Performance comparison at the end of the training phase between agents implementing adaptive and non-adaptive variable set in the definition of the state-space (γ=0.9, ρ=10) [24].

Var. Set	**DRL Control**			**Climatic-Based Control**			**Saving**
	Cons. [MWh]	Temp. viol. Cum. [°C]	Occ.-rate [%]	Cons. [MWh]	Temp.viol. Cum. [°C]	Occ.-rate [%]	[%]
A	101	37	2.8	113	112	3.3	**-10.0**
B	102	96	5.7				**-9.92**

The table reports for each set of variables the performance in terms of total consumption (Cons.) and temperature violations. The temperature violations during occupancy were expressed both in terms of cumulative value of violations (Cum.) and occurrence rate (Occ.-rate). As a reference, a temperature violation with an occurrence rate of 5% means that the indoor temperature is out of range for the 5% of the total simulation steps included in the occupied periods of the building. Eventually, the table shows in the last column the energy savings expressed in percentage achieved by DRL controller with respect to climatic-based control.

As can be observed the two agents show similar performance in terms of energy saving obtained compared to baseline. Despite both agents improved the indoor temperature control and reduced heating energy consumption respect to the baseline, the agent trained with variable set A performed slightly better especially in terms of indoor temperature control. This aspect suggest that this agent was capable to better exploit internal and external heat gains, improving temperature control and, at the same time, increasing energy saving.

Results of the deployment phase

In this last section are analyzed the results of the deployment of the two agents (trained with variable set A and B and considering $\rho = 10$ and $\gamma = 0.9$) in the four different scenarios introduced in section 4.1.4. The deployment of each agent was simulated both in a static and dynamic way for one episode. As previously introduced, the deployment episode is 3 months long, including January, February and March, and the climatic data employed in the simulation are gathered from the reference weather file referred to Torino (ITA_TORINOCASELLE_IGDGėpw).

Figure 4.10 summarizes the performance obtained in terms of supplied heating energy and cumulative sum of temperature violations for all the possible configurations resulting from the combination of the four scenarios, two variable sets, and two deployment processes (16 configurations) including also the baseline configuration. The performance of the agent trained with the variable set A did not produce always with dynamic deployment configuration an improvement with respect to static deployment across the four scenarios (azure and blue bars in the Figure 12). In particular, in scenarios S2 and S3 the dynamically deployed agent achieved a lower energy saving compared to its statically deployed counterpart. In scenario S2 this led to a slight improvement of temperature control performance while in

scenario S3 the temperature control was performed with less accuracy compared to statically deployed agent. Even without updating its control policy the agent trained with the variable set A is capable to adapt to the different requirements in the different scenarios achieving better performance than the baseline controller. The agent based on variable set B, instead, shows opposite behavior and the effect of dynamic deployment over static deployment is particularly significant (yellow and orange bars in the figure). For example, in the scenario S2, which considers an increased temperature set-point compared to training condition, the statically deployed agent obtained the lowest consumption (yellow bar in the first panel of the bottom figure) but an extremely high value of the cumulative sum of temperature violations (yellow bar in the second panel of the bottom figure) meaning that the control policy was not able to adapt to the new indoor temperature requirements. On the contrary, the dynamically deployed agent in the same scenario achieved an overall performance comparable with agent implementing the variable set A conceived with an adaptive approach.

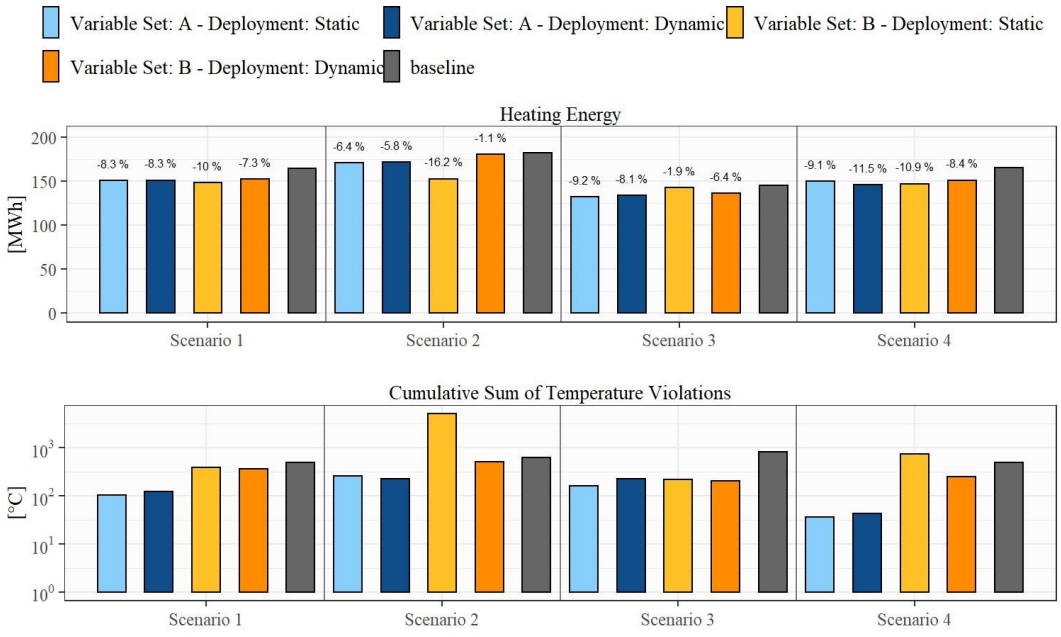


Fig. 4.10 Heating energy supplied and cumulative sum of temperature violations for agents trained with both variable sets in four different scenarios under static and dynamic deployment configuration. In the upper part of the figure are reported on the bars the heating energy saving respect to the baseline [24].

A similar condition occurred also for the fourth scenario, which considers the presence of the occupants during Sunday (contrarily the training period) where the dynamic deployment drastically improved the indoor temperature control performances of the agent trained with variable set B. The same agent (trained with variable set B) shows a different pattern in the third scenario. In this case, in which the desired indoor set-point was reduced from 21 °C to 20 °C, the statically deployed solution was capable to achieve satisfying temperature control performance (yellow bar in the third panel of the bottom figure), but it obtained lower energy saving. On the contrary, the dynamically deployed solution achieved almost the same temperature control performance (orange bar in the third panel of the bottom figure) but increased the energy savings obtained from 1.9% to 6.4%. Also in this case the dynamic deployment was found to be effective in improving performance of the agent by means of continuous refinement of the control policy during the deployment episode. However, as the Figure 4.10 clearly shows, even in the dynamic deployment configuration the agent trained with variable set B was not able to achieve the performance of the agent trained with variable set A across all the four scenarios.

The successive figures (from Figure 4.11 to Figure 4.13) provide details about some configurations that are of particular interest for supporting the discussion.

Figure 4.11 shows a comparison between statically deployed agent trained with variable set A, and the baseline controller during a week of the deployment period. The plot shows the indoor air temperature patterns generated by the two controllers along with supply water temperature, outdoor air temperature and direct solar radiation profiles. The DRL agent was able to exploit solar heat gains reducing supply water temperature and, consequently, save energy. This aspect is particularly relevant during the third and sixth day when solar radiation is higher.

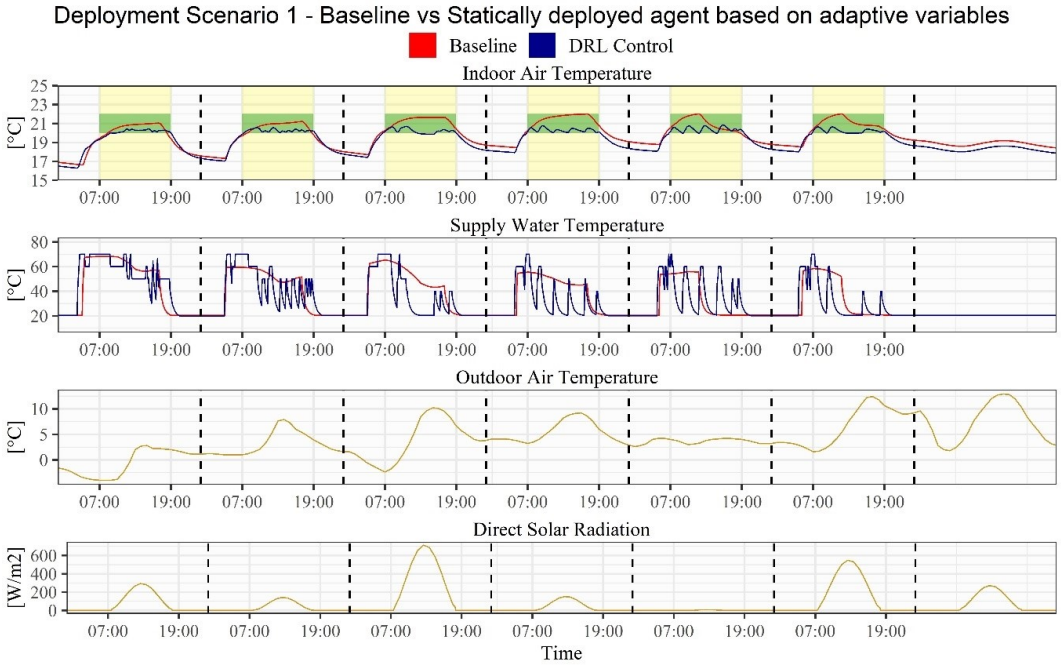


Fig. 4.11 Comparison between statically deployed agents trained with variable set A and variable set B in terms of daily indoor temperature profiles during Tuesdays in the scenario S2 [24].

Figure 4.12 highlights the differences between agent trained with variable set A (red lines) and agent trained variable set B (blue lines). The plot shows for different weeks and the same working day (Tuesday), the daily indoor temperature profiles in the scenario S2, which implements an increased indoor set-point (22 °C) compared to the training phase (21 °C). As can be observed the agent based on adaptive variables (variable set A) was promptly able to adapt to the change of indoor temperature requirements maintaining satisfying conditions within the zone despite any learning goes on during static deployment. On the other hand, the agent trained with non-adaptive variables (variable set B) was not capable to adapt without relying on dynamic deployment.

Fig. 4.12 Comparison between statically deployed agents trained with variable set A and variable set B in terms of daily indoor temperature profiles during Tuesdays in the scenario S2 [24].

Figure 4.13 compares the effect of a static and a dynamic deployment for the agent trained with variables selected according to the non-adaptive approach (variable set B). This detail is particularly interesting considering that, as can be observed in Figure 4.10, the differences between the two deployment strategies are more emphasized for the agent trained with the variable set B.

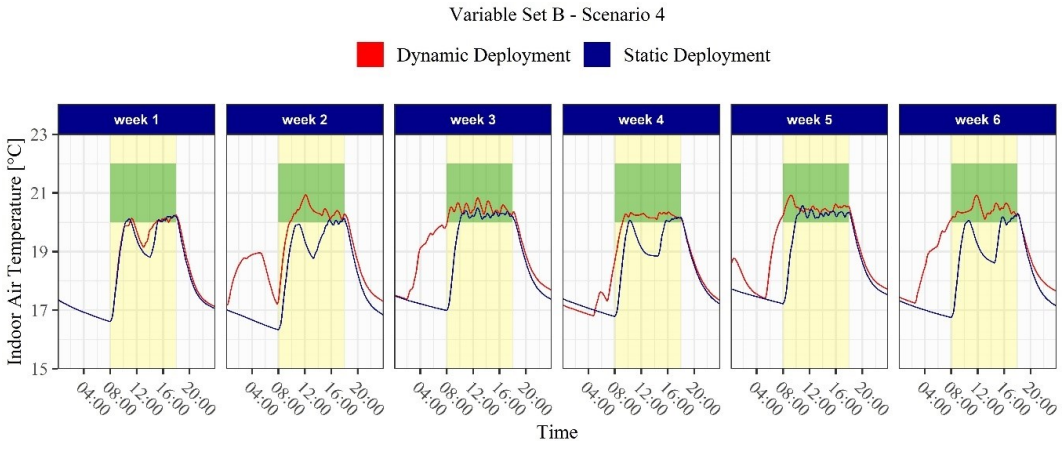


Fig. 4.13 Comparison between dynamically and statically deployed agent trained with variable set B in terms of daily indoor temperature profiles during Sundays in scenario S4 [24].

The figure shows the results obtained during the first 6 Sundays in deployment scenario S4. This scenario is particularly interesting because, differently from the

training conditions, implements the presence of occupants during Sundays. The plot shows, for the first 6 weeks, the daily indoor temperature profiles generated by the two agents. It is interesting to notice that the divergence between the profiles increases over time suggesting that the two agents have different adaptability capabilities. During the first week the two agents generated almost the same pattern which clearly do not satisfy the indoor temperature requirements. The larger temperature violation is localized during the first hours of the day since both the agents were not able to anticipate occupants' arrival. A second temperature violation region is localized in the middle part of the day, when, during training, the agent correctly learnt to exploit solar heat gains in order to reduce supply water temperature. However, the reduction of supply water temperature caused the occurrence of temperature violation condition since the agent did not performed a sufficient pre-heating of the zone in order to reach the acceptability range of the indoor temperature. This pattern was replicated by the statically deployed agent among the six weeks demonstrating its lack in adapting to the modified occupancy schedule. On the contrary, the dynamically deployed agent was capable to learn from experience and it was able to achieve satisfying temperature conditions starting from the third week of deployment.

4.1.6 Discussion

The presented application focuses on the development of a DRL controller of supply water temperature set-point to terminal units of a heating system. The developed controller was trained and deployed in a simulation environment which combines EnergyPlus and Python. The controller aims at optimizing both energy consumption and indoor temperature control trying to identify the best trade-off between the two contrasting functions. The control problem analyzed in this application was relatively simple, not involving elements such as renewable energy sources or storage which may effectively require an optimized controller to be fully exploited. Although the only two features of the building that could be exploited in the considered optimization process were the building thermal mass and the temperature acceptability range, the DRL controller led to good performance improvements in comparison to the baseline controller.

In DRL algorithms hyper-parameters tuning and reward design play a key role in identifying the optimal configuration of DRL controller. In this application, a tuning process was carried out on some of the main hyper-parameters to highlight their

influence on the final performance of the developed controller. Given this strong dependence it seems necessary for reinforcement learning applications in HVAC systems to rely on simulated environments, at least in the initial stage of training. As a consequence, despite the model-free nature of reinforcement learning control, a modelling effort needs to be accounted.

The effect of adaptive variables defining the state-space was analyzed. A variable set designed to enhance adaptability and flexibility of a DRL agent with respect to variable requirements of the indoor environment (i.e. indoor temperature set-point and occupancy schedule) was introduced. A DRL agent based on adaptive variables was compared with an agent trained with more classic non-adaptive variables. The comparison was performed by simulating the deployment of the two agents in four different scenarios. Moreover, the deployment of the agents was simulated both in static and dynamic configuration. The agent trained with adaptive variable set was capable to adapt to each scenario performing better than the baseline controller even if statically deployed. The dynamic deployment of the same agent did not produce significant improvements on the overall performance, showing slight poorer performance compared to static deployment case.

On the contrary when the variables were selected with a non-adaptive approach the dynamic deployment performed better that the static deployment in all the scenarios analyzed. These results proved that the proposed variable selection process was useful in providing to the agent the capability to adapt itself to changes that may occur in the controlled environment. This analysis suggests that a DRL controller with a carefully designed state-space is capable to provide the necessary flexibility and adaptability to changing indoor requirements even in a static deployment configuration. Through this approach is possible to leverage the advantages provided by static deployment (i.e. lower computational costs and higher stability) without sacrificing adaptability. However, the adoption of an adaptive approach in the design of the state space may not be enough to guarantee a good control performance in the case of retrofit on the HVAC system or other building components. In such cases thermal dynamics of the controlled environment may change requiring DRL controller to update its policy through a dynamic deployment. The implementation of the proposed controller in a physical test-bed requires the monitoring of a few variables that can be easily collected through low-cost solution already available in the market. An outdoor ambient sensor is required to monitor outdoor air temperature and solar radiation. Alternatively, those data can be easily obtained by an external

weather data provider. Many of those services requires no fees for a limited number of data requests and already implement Application Program Interfaces (APIs) which enable the streaming of data. Low-cost solutions are available also for what concerns indoor air temperature monitoring. Supply and return water temperature are usually collected by the Building Management System (BMS) and thermocouples must be installed in the relative pipes. The most challenging quantity to be monitored is the supplied heating energy. This variable can be indirectly calculated from supply and return water temperature if the water mass flow rate through the system is known and collected through an appropriate sensor or directly by installing a non-invasive heat meter. Since the considered case study is an office building the variables time to occupancy start and time to occupancy end included in the variable set based on adaptive approach can be easily obtained through working timetables. The most challenging aspect is to design an infrastructure capable to manage the stream of data from different sources in order to provide to the controller the required input information. The static or dynamic deployment can be achieved in-situ if the BEMS allows the running python scripts otherwise all the operations can be performed in a cloud server.

4.2 Effective pre-training of DRL agent by means of data-driven models

DRL agents have to perform several interactions with the controlled environment before converging to the optimal control policy. In this context, it is common practice to pre-train an RL agent offline in simulation environments relying on physics-based models of the real building. Nonetheless, the development of physics-based models requires a considerable effort and expertise beside a huge amount of input data. The application introduced in this section aims to evaluate the effectiveness of pre-training a DRL agent on a data-driven model of a building based on Long Short-Term Memory (LSTM) neural networks.

The next section presents the main research challenges analyzed in this application and introduces the motivations and novelty of the proposed methodological approach.

4.2.1 Motivations and novelty of the proposed approach

When detailed monitored data of the analyzed building are available, data-driven models can be employed to pre-train DRL agents. Data-driven models require significantly less input data than their physics-based counterparts resulting in reduced development times. Following this approach, Zou et al. [72] demonstrated how a reinforcement learning agent could learn an optimal control policy by interacting with data-driven models based on LSTM architecture of an air handling unit system built from monitored data. However, as the authors clearly states in their work, the performance of the trained agent were not evaluated in field but on the same data-driven models employed for training. The drawback of data-driven models is their inability to map patterns that were not present in the training data. For example, if during training on a data-driven model the DRL agent explores trajectories that deviates from the normal behavior of the physical building, the response of the model may be significantly different from the real behavior of the system. As a consequence, the control policy learned by the DRL agent may be sub-optimal.

Based on the previous reasoning the application presented in this section aims at evaluating the performance of DRL agent pre-trained by means of a data-driven model of the original system. Given the unavailability of a physical system to test the

approach a EnergyPlus model was used a surrogate of the real building. Despite this simplification the primary purpose of the present application was not undermined.

4.2.2 Case study and control problem

The control problem focuses on the regulation of the heating power delivered from a gas-fired boiler to the water serving an office building equipped with radiators as terminal units. The regulation of the plant is performed by adjusting the supply water temperature set-point while the water mass flow rate is constant. The objective of the controller is to reduce the amount of thermal energy provided to the supply water while maintaining indoor air temperature within the desired acceptability range ([+1 °C,-1 °C] with respect to set-point value) during occupancy periods. Since the analyzed system is an all-water system, heating terminals are not able to control the relative humidity. Thus, comfort analysis have been focused only on the analysis of indoor air temperature deviation with respect to the desired set-point. The control action was taken with a frequency of 15 minutes.

4.2.3 Methodology

In this section the methodological framework is presented with the aim of introducing each phase of the DRL control agent development. The methodological framework of the present application unfolds through three different phases as illustrated in Figure 4.14. Moreover, two different weather data files have been employed across the different phases.

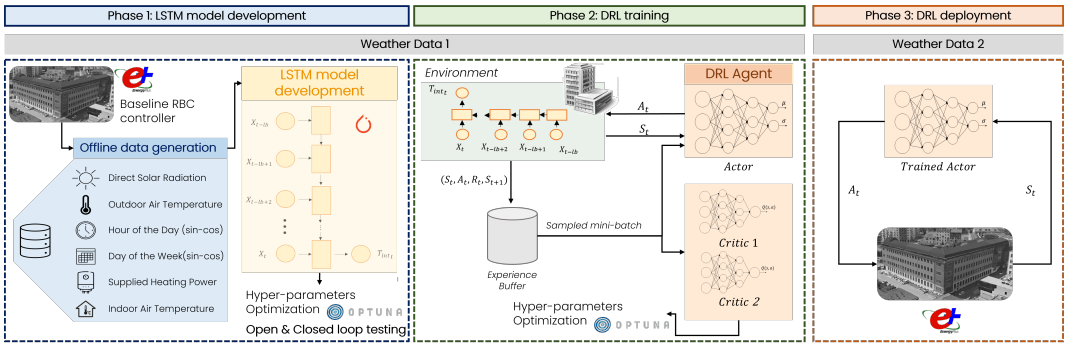


Fig. 4.14 Methodological framework of the application of DRL control pre-trained with data-driven models.

LSTM model development: the first phase of the framework is aimed at the development and training of a robust data-driven model of the building dynamics. An EnergyPlus simulation of the case study was performed employing a rule-based algorithm as a control strategy. This process was conceived to replicate the case in which, for the considered case study, historical monitored data were available to be employed for the training of the data-driven model. An LSTM neural network was designed to predict the evolution of indoor air temperature for the next time-step given a series of input variables. The model was trained performing an optimization of the hyper-parameters values in order to identify the best configuration. Eventually, the performance of the model were evaluated for both open-loop and closed-loop configuration.

DRL training: in the second phase the trained data-driven model was wrapped up in an OpenAI Gym interface and employed to train a DRL controller. The agent was trained for several episodes considering the same weather data input of the previous phase. Also in this phase an optimization of the hyper-parameters was carried out. The proposed controller was based on a SAC architecture for continuous action spaces described in Chapter 2. SAC was chosen among different DRL frameworks considering its capacity to handle continuous action-spaces, its off-policy evaluation mechanism and the capability to learn stochastic control policies.

DRL deployment: In the third phase, after the training process, the agent was statically deployed in an OpenAI Gym environment implementing the EnergyPlus model of the building as described in Chapter 3. In this phase, a new weather file was implemented in order to simulate the deployment of the trained agent during a new heating season as it was implemented in a physical case study. The performance of the proposed controller were compared to the rule-based baseline employed to generate the offline dataset in the first phase of the framework. This aspect represent the only exception of the methodology to the exact emulation of a physical case studies since it would had not been possible to deploy both baseline strategy and DRL in the real system for the same period.

4.2.4 Implementation of the proposed methodology

The test facility analyzed in this application consists of a six-level mixed-use building. This is the same facility described for the previous application and introduced in sec-

tion 4.1.3. The following subsections introduce in detail the different implementation steps of the proposed methodology. Figure 4.15 shows the distribution of outdoor air temperature, relative humidity and direct solar radiation for the two weather files employed in this application.

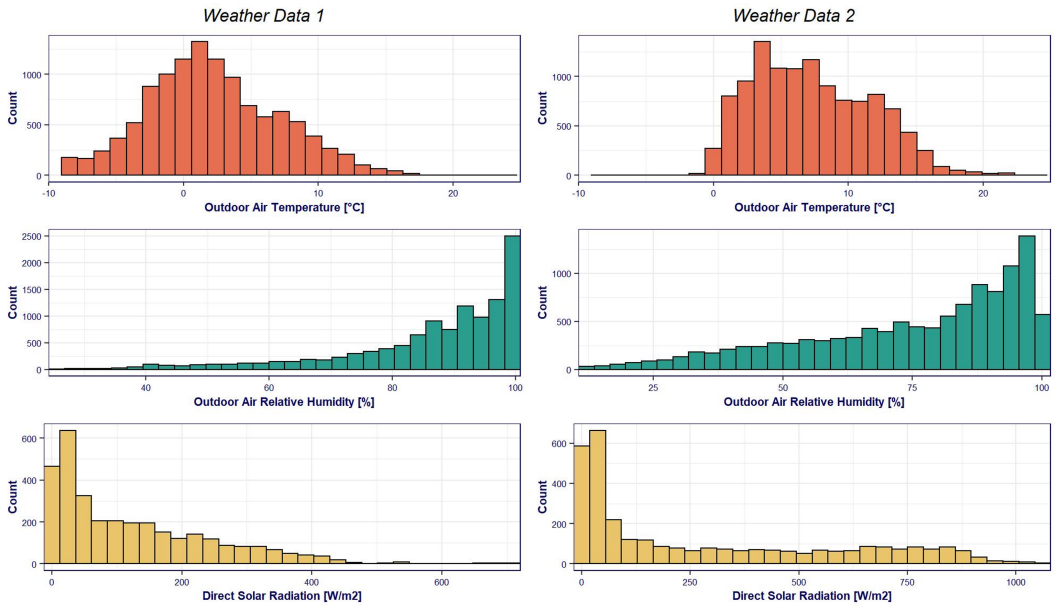


Fig. 4.15 Distribution of outdoor air temperature, relative humidity and direct solar radiation for both weather data 1 (left) and weather data 2(right).

Implementation of the baseline controller

The baseline control strategy is based on the combination of a rule-based strategy managing the on and off status of the system, and a climatic regulation determining the supply water temperature set-point. The system is switched on 4 hours before the arrival of the occupants if the difference between indoor air temperature and the lower threshold of the acceptability range is greater than 3°C. Similarly, the system is switched on 3 hours before the arrival of the occupants if the difference is greater than 2 °C. Eventually, the system is switched on 2 hours before the arrival if the previous two conditions were not met. During occupancy periods when the indoor air temperature reaches the upper threshold of acceptability range the heating system is turned off. The heating system is turned on again if the zone temperature falls below the lower threshold. The climatic regulation is based on linear function that spans from 70 °C to 40 °C when the outdoor temperature goes from -5 °C to 12 °C.

Implementation of the LSTM model

As introduced in the previous section, the LSTM model was trained with data generated from a EnergyPlus simulation implementing the baseline controller for a period of 4 months between 1^{st} November and 28^{th} February with a timestep of 15 minutes. The weather data used in this phase and identified in figure 4.14 as *Weather Data 1* is is the reference weather file (*ITA_TORINOCASELLE_IGDG.epw*) available in EnergyPlus for Torino, Italy.

This process emulates the collection of building related data in a physical building managed through a traditional control strategies. Given a set of predictor attributes, the objective of the LSTM model is to estimate the evolution of indoor air temperature for the successive time-step. Predictor attributes were organized into 48 look-back sequence. Each sequence included the variables reported in table 4.6.

Table 4.6 Variables included in a input sequence of the LSTM model

Variable	Min Value	Max Value	Unit
Outdoor air temperature	-12.0	26.0	°C
Direct solar radiation	0.0	720	W/m^2
Hour of the day	0	23	-
Day of the week	1	7	kW
Supplied heating power	0.0	522.0	kW
Indoor air temperature (previous time-step)	13.0	25.0	°C

Predictor sequences include variables which are easily available in physical environment. Temporal information such as the hour of the day and day of the week are necessary to provide knowledge to the model about usage patterns of the building which determines endogenous loads. Outdoor air temperature and direct solar radiation were included as the weather variables which affect the considered control problem determining exogenous loads. Indoor air temperature value provides information to the model about the actual state of the building. Eventually, supplied heating power during the time-step is a key information along with endogenous loads, endogenous loads and actual temperature to determine the temperature of system in the successive time-step. It is not trivial to obtain supplied heating power

in a real-world context. However, this value can be calculated knowing supply/return water temperatures and mass flow rates to the considered building/zone.

As any other machine learning algorithm, LSTM networks are characterized by several hyper-parameters. In this application, part of these values were arbitrarily fixed and part were the result of an optimization process. The fixed hyper-parameters were the number of training epochs chosen equal to 30 and the Adam optimizer of the neural network. The other hyper-parameters, reported in table 4.7, were tuned employing the Optuna optimization library [156]. Optuna employs a Tree-structured Parzen Estimator (TPE) to search the best configuration of hyper-parameter values. Both Mean Absolute Percentage Error (MAPE) and Root Mean Squared Error (RMSE) were chosen as optimization metrics for values search. Table 4.7 reports the minimum and maximum hyper-parameters values along with their type.

Table 4.7 Variable hyper-parameters of the LSTM network.

Hyper-parameter	Min	Max	Type
Batch size	80	120	Integer
Optimizer learning rate	0.0001	0.05	Float
Hidden layers	1	3	Integer
Neurons per layer	10	30	Integer

For the sake of hyper-parameter tuning, the performance of the network were evaluated for the open-loop configuration. These performance represents what is commonly identified as training performance in supervised learning frameworks. Once identified the best configuration, the performance of the network were evaluated also for the closed-loop configuration. According to this strategy only for the first sequences of the indoor air temperature values were collected from the training dataset (i.e. 48 sequences for this application). For the successive time-step the network predictions were recursively fed as input data to the model while other inputs were left unchanged with respect to training dataset. Through this approach it was possible to evaluate potential deviations in model estimations.

Testing performance evaluated on a specific dataset were not included in this application for two reasons. The first relates to the limited amount of data available for training the model. Although this data comes from a simulation and it would have been possible to easily generate more, it was preferred to have limited amount of

information since it was considered more adherent to a real case study. The second reason relates to the structure of the LSTM networks. These models, specifically designed to model time series data, receive input structured in sequential architectures whose length is variable. Consequently, it is impossible to carry out a classic random sampling with 70% of training data and 30% of testing data. Moreover, examining these aspects it has been considered counterproductive to execute a sampling on temporal base (employing a period for the training and another for the testing) since this procedure could have invalidated the goodness of the training process. Having available few months, each of these provides information on the behavior of the system in different climatic conditions that otherwise would be excluded from the training set.

Design and training of the DRL agent

The DRL control algorithm was firstly trained offline exploiting the LSTM model. A SAC control agent handling continuous action-spaces as described in Chapter 2 was implemented. The following paragraphs describes in details the different features of the proposed control strategy.

Design of the action-space The action space includes the set of possible control actions that can be performed by the agent. Since SAC as DRL algorithm was selected the action space is continuous limited between 0 and 1. The action correspond to the fraction of the nominal supplied heating power (i.e. 522 kW) to the system at each time-step.

This quantity is not easily controllable in physical environments. However, the training of both DRL agent and LSTM model was found more effective employing this variable. In the next subsection is illustrated the strategy to convert this action value in a control signal that can be handled by the considered building system.

Design of the reward function The reward function includes two terms and it is described by the following equation:

$$R = -\delta * Q_{heating} + \beta * C_{Temp} \tag{4.3}$$

The first term is proportional to the power supplied by the heating system and it was introduced to minimize energy consumption. The second term is proportional to temperature control (C_{temp}) and it was introduced with the aim to maintain the indoor air temperature within an acceptability range of +/- 1 °C from the desired set-point of 21 °C during occupancy periods. δ and β are the weight factors of the two terms and are two hyper-parameters characterizing the DRL agent. The temperature term was evaluated only when the building was occupied and it is characterized by the following values:

- if $T\_int < 20$ °C: $-(21 - T\_int)^3$
- if $T\_int > 22$ °C: $-(T\_int - 21)^3$
- if $21 < T\_int \leq 22$ °C: $-(T\_int - 21)$
- if $20 \leq T\_int \leq 21$: $+\theta$

The temperature term was conceived to encourage the controller to maintain indoor temperature value as close as possible to the set point value. The reward function has a positive value, equal to the hyper-parameter θ, when the temperature value falls in the lower range of acceptability range in order to incentivize the exploration of this condition.

Design of the state-space The state-space was conceived following a similar approach to the feature selection process for the LSTM architecture. Variables have been chosen following an adaptive approach as demonstrated in the previous application described in section 4.1. The table 4.8 reports the variables included in the state-space for this application.

Table 4.8 Variables included in the state-space.

Variable	Min Value	Max Value	Unit	Time-step
Outdoor air temperature	6	31	°C	t
Direct solar radiation	0	720	W/m^2	t
Time to occupancy start	0	36	h	t
Time to occupancy end	0	12	h	t
ΔT Indoor set-point – indoor air	-3	10	°C	t,t-1,t-2,t-3

Outdoor air temperature and direct solar radiation were included due to their influence on the heating load. Moreover, time to occupancy start and time to occupancy end are two engineered occupants-related variables that were included. Eventually, the difference between the indoor air temperature and the desired set point evaluated at the current time-step and with 15, 30, 45 minutes lags were included. Each state variable was re-scaled between 0 and 1 using a min-max normalization.

Training The SAC agent was trained in an environment implementing the LSTM network as model of the building dynamics. The environment was initialized providing to the network the first 48 sequence from the training set of the LSTM model. The simulation continues employing the same weather data and time variables while the input supplied heating power is defined by the DRL agent control action and the indoor air temperature is recursively determined by the neural network model.

The Optuna library was employed also for this phase of the analysis since the DRL agent is characterized by several hyper-parameters. Part of these values were arbitrarily fixed due to computational limitations. These hyper-parameters include the number of hidden layer of actor and critic networks fixed equal to 4 and the number of neurons per hidden layer taken equal to 64. Moreover, the batch size was assumed equal to 128 and the total number of training episodes for each configuration was set to 20. Table 4.9 reports the hyper-parameters optimized and the relative search ranges.

Table 4.9 Variable hyper-parameters of the SAC control agent.

Hyper-parameter	Min	Max	Type
Weight factor energy term (δ)	0.0001	0.001	Float (Step 0.0001)
Weight factor temperature term (β)	1.0	8.0	Float (Step 0.5)
Temperature prize (θ)	0.005	0.05	Float (Step 0.05)
Discount factor (γ)	0.9	0.99	Float

After 20 training episode the DRL agent was statically deployed in the same environment implementing the LSTM network. Its performance were evaluated in terms of total heating energy supplied to the system and total amount of temperature violations. A temperature violation occurs when the building indoor air temperature falls above or below the acceptability range defined from the desired set-point of

21 °C during occupancy periods. The agent implementing the configuration of hyper-parameters leading to the best performance in terms of heating energy and temperature violations was chosen as a candidate to be effectively deployed on the system.

Deployment phase

In the last phase of the process the trained agent was deployed in the environment implementing the EnergyPlus model in order to assess the performance of the control policy learned through the interaction with the LSTM model. The deployment period last for one episode of four months from the 1^{st} November to the 28^{th} February. Climatic data employed in this phase and identified in figure 4.14 as *Weather Data 2* were real data collected for Torino between 2018 and 2019.

As described in the previous section the action taken by the DRL agent is the normalized supplied heating power that during training was provided to the LSTM network to predict indoor air temperature evolution. However, this process is not suited for being implemented on the EnergyPlus model (that, it is worth remembering, in this application acts as the real building) since it is impossible to directly regulate the supplied heating energy to the system. In the EnergyPlus environment the available control output is the set-point of supply water temperature to the building. In order to correctly match the output of the trained DRL agent and the actuator available in EnergyPlus, the control action was converted into a supply water temperature set-point according to the following formula:

$$SP_{T,SUPP}(t) = \frac{a(t) * P_{Heating,MAX}}{C} + T_{RET}(t-1) \tag{4.4}$$

Where $a(t)$ is the control action from the DRL agent, $P_{Heating,MAX}$ is the maximum supplied heating power to the building equal to 522 kW and C is a constant which depends from the mass flow rate and it is expressed in [kW/°C]. $T_{RET}(t-1)$ is the value of the return water from the building at the previous time-step.

4.2.5 Results obtained

This section reports the results obtained implementing the proposed methodology. A data-driven model of the building dynamics was built from EnergyPlus simulation data. These model was employed to train a DRL control agent that was successively deployed to interact with the EnergyPlus environment with new weather conditions. The entire process was conceived to emulate the implementation of a DRL controller in a physical test-bed employing the EnergyPlus model as the real building.

Table 4.10 reports the values of the best configuration of variable hyper-parameters of the LSTM network as obtained from Optuna.

Table 4.10 Values of variable hyper-parameters of the LSTM network obtained from Optuna.

Hyper-parameter	Value
Batch size	108
Optimizer learning rate	0.0004
Hidden layers	2
Neurons per layer	17

Figure 4.16 shows in two subplots the distributions of the prediction errors for both open-loop (left) and closed-loop (right) configurations. The errors were calculated between LSTM predictions and the values of indoor air temperature in the training dataset. The values of supplied heating power to the system and weather forcing variables provided to the LSTM models in both configurations are the same resulting from the EnergyPlus simulation implementing baseline rule-based strategy and *Weather Data 1*. The first subplot show how the distribution of errors in the open-loop case is narrower compared to the closed-loop case. In absolute terms, closed-loop resulted in an errors distribution with a maximum value around 0.5 °C compared to open-loop which maximum value is in the range between 1 °C and 1.2 °C. This situation was expected since while in open-loop the model received in input exactly the values taken from the training dataset in closed-loop it employed its own predictions to provide the next values risking to propagate errors.

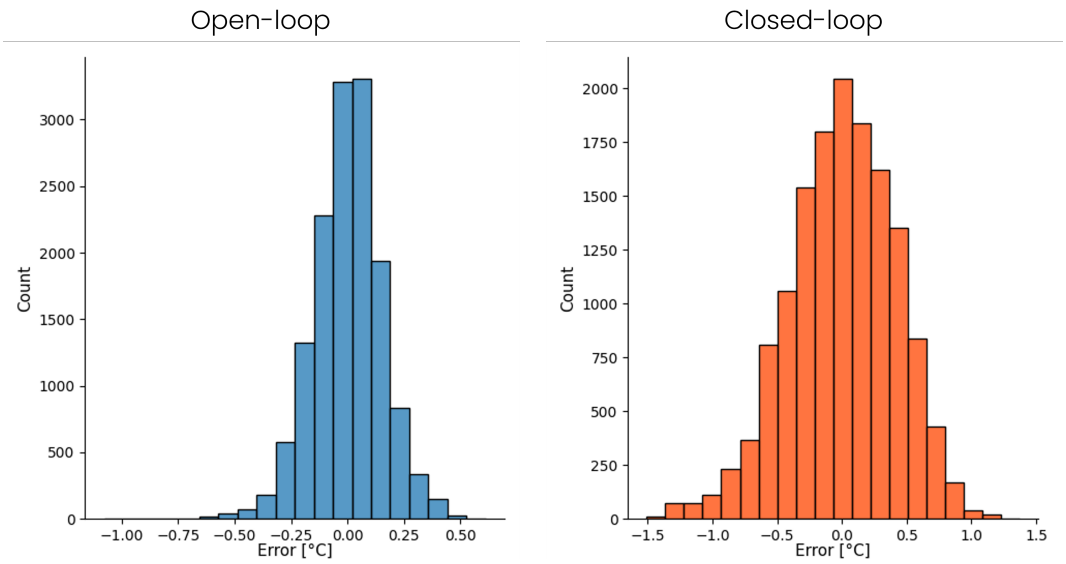


Fig. 4.16 Error distribution of the LSTM network implementing the best configuration of hyper-parameters for both open-loop (left) and closed-loop (right) conditions.

Table 4.11 reports MAPE and RMSE values obtained by the LSTM network implementing the best configuration of hyper-parameters for both open-loop and closed-loop conditions. As expected, closed-loop performance are slightly worse compared to open loop. However, MAPE value below 2% and RMSE equal 0.412 °C with respect to the training dataset proved the robustness of the training process.

Table 4.11 MAPE and RMSE obtained by the LSTM network implementing the best configuration of hyper-parameters for both open-loop and closed-loop conditions.

Configuration	MAPE [%]	RMSE [°C]
Open-loop	0.665	0.152
Closed-loop	1.891	0.412

Figure 4.17 shows the temperature profiles of the ground-truth, open-loop prediction and closed-loop prediction obtained by the developed LSTM model for the first 4 weeks of the deployment period. These time series were obtained by applying the baseline RBC strategy control signal. As can be observed, the error of the closed loop prediction was mainly localized during nights and weekends.

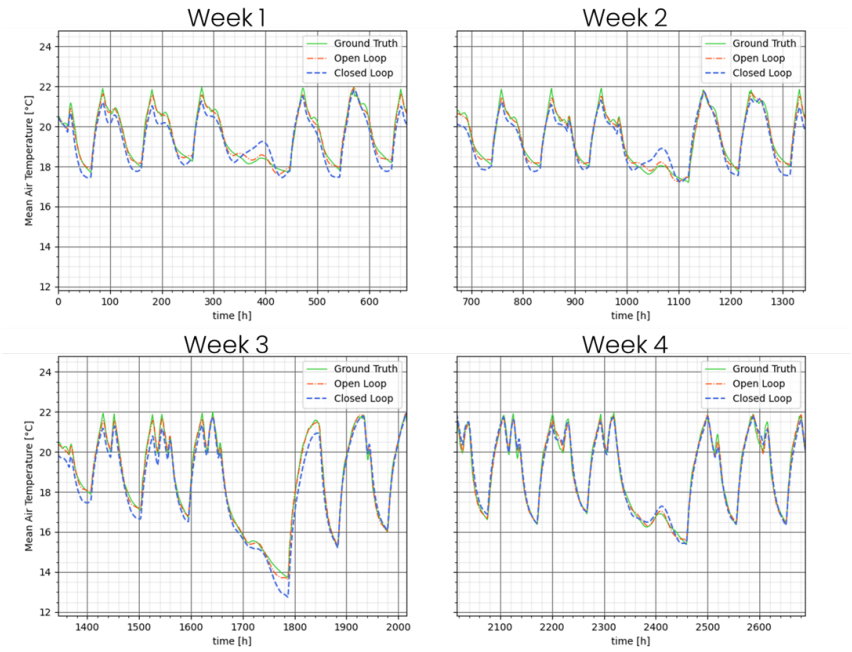


Fig. 4.17 Temperature profiles of the ground-truth, open-loop prediction and closed-loop prediction for the first 4 weeks of the deployment period (i.e. Weather 2).

Table 4.12 reports the values of the best configuration of variable hyper-parameters of the DRL agent as obtained from Optuna. The choice of this configuration was based on the performance achieved by the agent statically deployed in the environment employing the LSTM network as model of the building dynamics.

Table 4.12 Values of variable hyper-parameters of the DRL agent obtained from Optuna.

Hyper-parameter	Value
Weight factor energy term (δ)	0.007
Weight factor temperature term (β)	2
Temperature prize (θ)	0.005
Discount factor (γ)	0.95

It is interesting to notice that, contrarily to what is suggested by the majority of the applications found in the literature, the best discount factor γ for this application was found equal to 0.95 instead of 0.99. One explanation to this behavior may rely in the fact that the model of the building dynamics employed to train the DRL agent (i.e. the LSTM model) lacks in accuracy compared to engineering models commonly

Table 4.13 Performance comparison between DRL and RBC in the deployment period considering heating energy supplied, cumulative temperature violations and average violation magnitude.

Metric	**RBC**	**DRL**	Difference
Heating energy supplied [MWh]	318	261	18.3%
Cumulative temperature violations [°C]	164	52	68.5%
Average violation magnitude [°C]	0.33	0.10	69.6%

employed to accomplish this task. As a consequence, it was found effective to focus on most immediate rewards (i.e. reducing the discount factor) since they can be estimated with an higher accuracy.

Successively, The DRL agent trained with the configuration of hyper-parameters reported in table 4.12 was statically deployed for one episode in the environment employing the EnergyPlus model. As previously introduced, the deployment episode lasts four months between the 1^{st} of November and 28^{th} February and the climatic data employed comes from real-world measurement collected for Torino in the years 2018 and 2019. The performance of the DRL controller were compared to the baseline control strategy implemented in the same environment with identical weather conditions. Table 4.13 summarises the performance obtained in terms of supplied heating energy, cumulative sum of temperature violations and average violation magnitude for the proposed DRL controller and RBC baseline during the deployment period.

The proposed controller outperformed the baseline RBC considering every metric. The DRL agent reduced the supplied heating energy provided to the building by 18.3% from 318 MWh to 261 MWh in the deployment period. Concurrently, it was capable to limit the cumulative sum of temperature violations of the acceptability range during occupancy periods to 52 °C. This metric was evaluated with same frequency of the control action, thus every 15 minutes. Eventually, the magnitude of the average temperature violation was reduced by 69.9% from 0.33 °C to 0.10 °C.

Figure 4.18 shows the heating load curves obtained by implementing DRL and RBC control strategies during the deployment periods. The two curves were calculated from supplied heating power values and the area between the two represents the amount of heating energy saved.

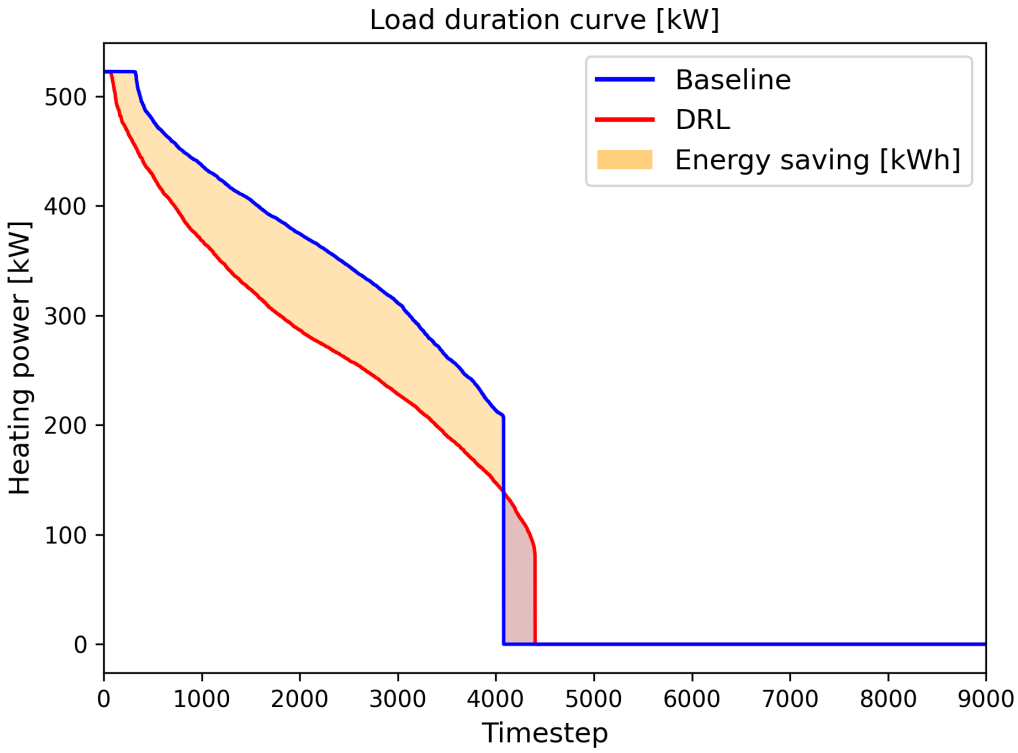


Fig. 4.18 Heating load duration curves achieved by DRL and RBC during the deployment period.

The figure shows how both strategies provided heating power to the building for around 50% of the period length. Moreover, it can be observed how DRL provided heating power for a greater amount of time but with a lower magnitude. Figure 4.19 is organized into two subplots representing the indoor air temperature distributions achieved by baseline (left) and proposed (right) control strategies during occupancy periods. The black dotted line represent the desired set-point while the red dotted line the lower and the upper limits of the acceptability range.

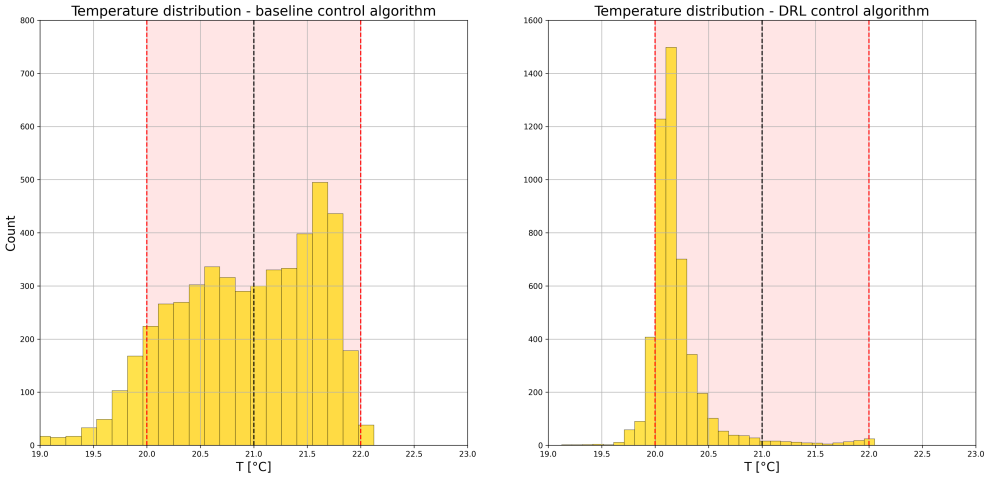


Fig. 4.19 Indoor air temperature distributions obtained by baseline (left) and proposed (right) control strategies during occupancy periods.

The figure highlights the differences between the two strategies. The DRL strategy maintained temperature values as close as possible to the lower threshold of the acceptability range. The baseline controller, instead, resulted in values more distributed around the desired set-point. Considering this aspect, DRL was more effective in leveraging one of the few flexibility sources provided by this control problem to achieve both energy saving and acceptable levels of indoor air temperature. Figure 4.20 shows a comparison between the DRL agent and the baseline RBC controller during a representative week of the deployment period. The plot reports in two subplots the indoor air temperature patterns generated by the two controllers along with supply water temperature profiles.

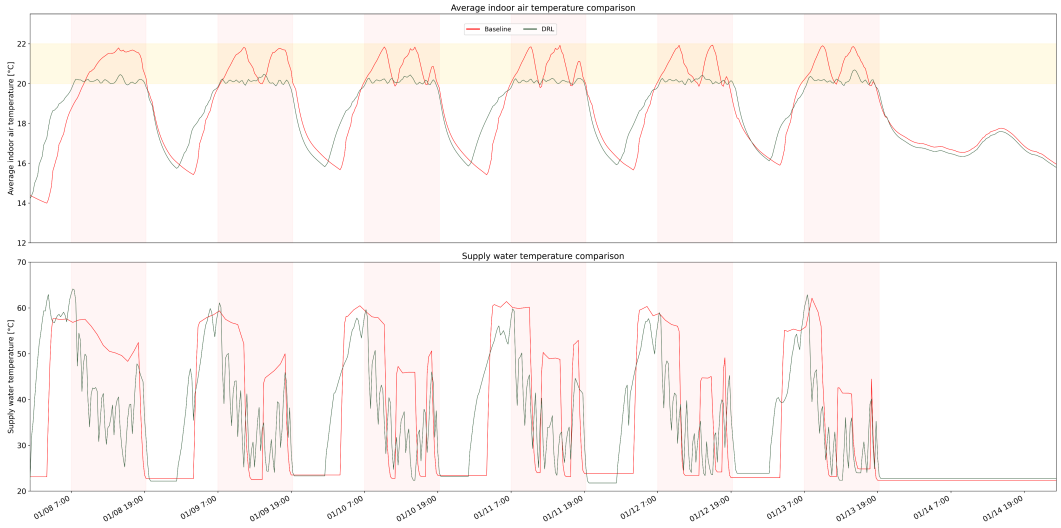


Fig. 4.20 Comparison between DRL agent and baseline RBC controller during a week of the deployment period.

At the beginning of the week, the control policy learned by the DRL agent pre-heated the building correctly before occupants arrival. During, the other days DRL correctly managed the indoor air temperature around the lower threshold of the acceptability range also exploiting higher heat gains during central hours of the day to reduce supply water temperature values. Despite, being trained on a simplified model of the building dynamics the learned control policy was capable to correctly manage the original system under different weather conditions.

4.2.6 Discussion

The presented application focuses on the development of a DRL controller pre-trained by means of a data-driven model of the building system. The developed approach relies on an EnergyPlus simulation model as a proxy of a physical building to unfold the proposed methodology. EnergyPlus was first employed to generate "synthetic" monitoring data of the building implementing baseline control logic. This dataset was successively used to train an LSTM network to estimate building dynamics. These model was integrated within a simulation environment to train a DRL agent. The hyper-parameters of both LSTM model and DRL controller were tuned through an open-source library available in Python. Eventually the trained agent was implemented to control the EnergyPlus simulation of the building in

order to assess its performance. In this application the use of EnergyPlus simulation was conceived exclusively to emulate a real building system. However, in order to effectively evaluate the performance of the DRL controller, a comparison with the baseline control strategy implemented in the same environment with identical weather conditions was introduced. LSTM networks, as any other supervised machine learning algorithm, lacks in generalizability when employed to perform predictions in conditions differing from training process. This aspect is particularly dangerous if machine learning models are developed to predict building dynamics and employed to train a reinforcement learning control agent. Since the agent, especially in the initial period of the training, tends to explore the state-action space the response of the machine learning model could significantly deviate from the expected values. This deviation can be generate from different sources such as weather conditions, building utilization and operational patterns. Despite these limitations, the approach presented in this chapter the DRL agent pre-trained by means of an LSTM model was able to converge to an acceptable control policy. This result can be motivated by the fact that during deployment weather was the only forcing variable that was modified, while operating conditions and building utilization were left unchanged.

However the present application suggests that the adoption of data-driven models to pre-train DRL agents could represent a key-point for the scalability of this control strategy in the energy and building sector. Data-driven models requires significantly less input data than physics-based models and their development process can be easily standardized and reproduced in an automatic fashion. Moreover, reducing the complexity and the time required for the development of the model enable the introduction of optimization techniques for the robust identification of hyper-parameters values which strongly affect DRL agent performance.

Eventually, an agent pre-trained by means of data-driven models could benefit from a dynamic deployment strategy in order to furtherly adapt to modifications in the controlled environment with respect to the conditions in which the data-driven model was trained.

4.3 Optimization of the management of integrated energy systems in buildings with Deep Reinforcement Learning

The widespread adoption of RES production system to sustain the decarbonization introduced the paradigm of Integrated Energy Systems (IES) in the building sector. Energy storage technologies both thermal and electrical plays a pivotal role in this process. In this context, the identification of optimal management strategies capable to increase the profitability of storage systems is a key aspect to address. The optimal operation of storage systems in buildings with IES is affected by exogenous factors such as weather, energy demand patterns and electricity prices which all vary over time. Classical control strategies are usually not able to consider trade-offs between multiple and contrasting objectives, such as thermal comfort, energy consumption, energy flexibility and Self-Sufficiency (SS), and are not capable to adapt to an evolving system characterized by dynamic boundary conditions, including grid requirements, and constraints [6].

To overcome these limitations, researchers worldwide have recently focused their efforts in the development and implementation of advanced control strategies to improve the management of IES in buildings based on predictive architectures or optimization processes. Comodi et al. [157] assessed the viability of introducing a Cold Thermal Energy Storages (CTES) for demand side management strategies into an existing cooling system of an institutional building under a Time of Use (ToU) pricing scheme. The storage was charged during night time to exploit higher chiller Coefficient of Performance (COP) and lower electricity price. It was demonstrated that a CTES could increase the overall energy efficiency and decrease the energy cost by being charged during off-peak hours with a payback period between 8.9 and 16 years. Arteconi et al. [158] analyzed a factory building equipped with Heat Pump (HP) and TES. The TES was charged during low price periods to cover the whole cooling demand during occupancy periods. This strategy was able to save about 54% of the electricity cost related to the cooling process. Ioli et al. [159] proposed a novel convex constrained optimization to optimize the operational cost of cooling system coupled with a TES into a single zone office building by controlling the storage operation and zone temperature. The proposed approach achieved 14.8% cost saving and 6.5% energy saving with respect to strategy where zone temperature is fixed.

Other strategies have been developed recently, as in Ren et al. [160] that analyzed an IES with an HVAC assisted by a photovoltaic thermal hybrid collector and a TES. The results showed that using the PV panels to power the heat pump to charge the TES provided additional energy flexibility respect to the use of only Demand Side Management (DSM) strategies. Comodi et al. [161]managed the integration of electrical and thermal storage into a nearly Zero Energy Building (nZEB). Thermal flows were optimized by a Mixed-Integer Linear Programming (MILP) algorithm to reduce the grid exchange electricity, while BESS were managed by a RBC. In that way the building achieved an SS level of 100% even though the cost of electrical storage did not justify the investment. A fuzzy rule control logic was developed by Dimitroulis et al. [162] for the charging scheduling of a BESS within an IES with renewable generation and Electric Vehicle (EV). The results showed a reduction of the monthly bill as compared to the linear optimize, and to an RBC. Biyik et al. [163] proposed a Model Predictive Control (MPC) for an IES with HVAC system, renewable generation and BESS to reduce the peak load. The controller provided an average reduction of 23% in peak electrical demand compared to a baseline where indoor temperature is kept fixed. Predictive management for energy supply networks using PV, HP and battery units was developed by Wakui et al. [164] by combining two-stage stochastic schedule programming and RBC to reduce operating cost. The proposed approach performed better than the management based on the deterministic schedule planning and the rule-based management without schedule planning.

In this context, RL and DRL control strategies can prove their effectiveness. The next sections present the implementation of a reinforcement learning-based control strategy in an office building characterized by integrated energy systems with on-site electricity generation and storage technologies. The proposed controller was tested considering various configurations of battery energy storage system capacities, and thermal energy storage sizes.

4.3.1 Motivations and novelty of the proposed approach

The management of storage systems is a key factor to consider in buildings with IES to enhance energy flexibility and reduce operational costs. Traditional controls may behave sub-optimally due to their lack of adaptability and their reactive approach. The design and implementation of storage solutions, including BESS and TES, is usually performed by different actors which are also responsible of the definition of

their control logic. Failure to consider proper control strategies in the design stage may result in oversized storage systems and consequently higher investment costs [165, 166].

The introduction of advanced control strategies based on a predictive and adaptive approach can enable a better management of multiple storage technologies in buildings. These controllers, thanks to their predictive and adaptive nature, can increase the effectiveness of storage equipment during building operation making competitive also solutions characterized by relatively low sizes and capacities. Accounting for the effect of advanced control strategies in the design stage can limit investment costs by adopting storage solutions that otherwise could be considered not suitable. For instance, in [167] the authors highlighted that most approaches to storage system sizing do not take into account storage daily performance which could contribute to determine appropriate sizes and capacities of storage equipment.

With this in mind, the application presented in this section aims to analyze the performance of a DRL strategy coupled with a RBC against a fully RBC to manage the operation of a chiller system coupled with a cold-water storage tank for an office building with on-site electricity generation and battery system. The analysis was carried out for multiple configurations of the energy systems including different sizes of TES and different capacities of BESS. The main contributions of the application presented in this section can be summarized as follows:

- Demonstrate the energy and cost benefits of adopting advanced DRL-based control strategies in IES characterized by BESS and TES equipment over classical RBC approaches.
- Evaluate the flexibility potential and the storage management which can be achieved with the adoption of advanced control strategies integrating a comprehensive management of the whole IES.
- Analyze the effectiveness of advanced control strategies with the variation of sizes of TES and capacities of BESS equipment highlighting the impact of the control also on the selection of storage in buildings.
- Adopting a novel formulation of the SAC algorithm specifically designed for discrete control actions, as described in Chapter 2, differently from the commonly implemented DQN framework.

The rest of the section is organized as follows. Section 4.3.2 introduces the case study and the control problem, Section 4.3.3 describes the methodological framework and provides information about DRL control, Section 4.3.4 reports implementation details of the different control strategies and configuration of the energy system. Section 4.3.5 reports the results obtained while Section 4.3.6 includes the discussion.

4.3.2 Formulation of the control problem

In this application, the effect of the adoption of advanced control strategies on the operation of IES in buildings considering different configurations of storage was evaluated for an office building located in Turin, Italy. The building is equipped with a TES system (i.e., a cold water storage tank) that is operated as a buffer between the building and an air-to-water chiller. The IES also includes a mono-crystalline silicon PV module and a lithium-ion electrical battery (i.e. BESS). Further, technical specifications of the components are provided in section 4.3.4.

Figure 4.21 shows a simplified schema of the electrical and cooling systems of the analyzed case study. The building electrical load (P_{dem}) is determined by the electrical demand of the chiller and circulation pump. The electrical system is formed by a DC bus and AC bus interfaced by a mono directional AC/DC inverter. On the DC bus a PV system and a BESS are installed. The PV and the battery are connected to the DC bus by a DC/DC converter. Grid is not allowed to charge the BESS according to the normative of many European Countries, but it is used to assist in matching electricity demand of the building and renewable power generation at each time-step [168]. At each step if local RES production is not zero the PV injects energy into the system according to the following priority: i) building, ii) BESS, iii) grid.

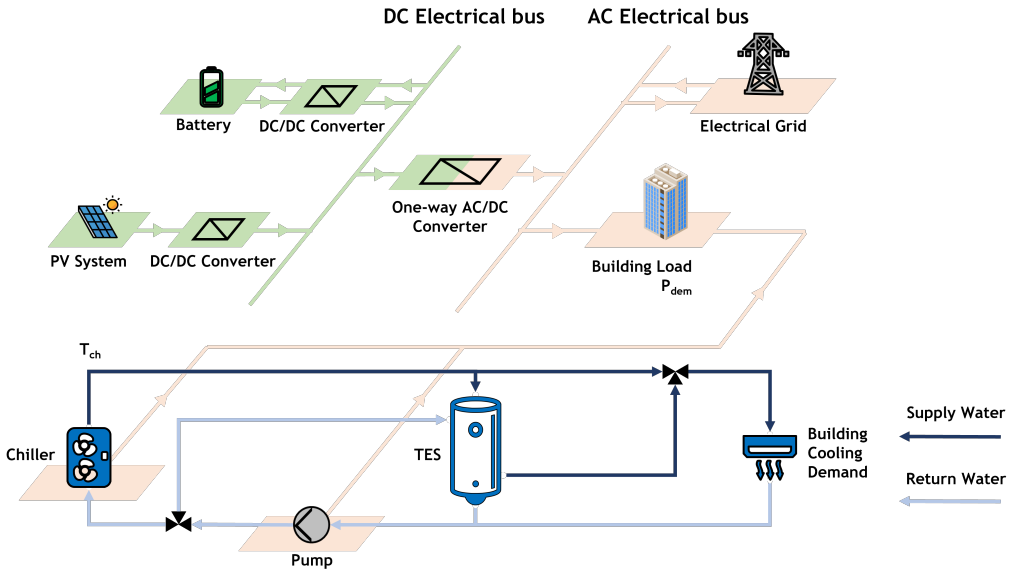


Fig. 4.21 Schematics of the electrical and cooling systems of the analyzed case study [26].

The electric chiller supplies cold water at constant set-point value (T_{ch}). The thermal storage can be operated in a temperature range between $T_{s,min}$ and $T_{s,max}$. The thermostatic control of the building was not considered in this application as the building cooling demand is considered as an external disturbance of the system along with weather conditions and electricity prices. To this purpose building cooling demand is evaluated in advance to maintain fixed conditions of indoor air temperature and relative humidity given the influence of weather and occupancy schedules.

The aim of the controller is to minimize the electricity cost of the chiller and circulation pump by managing three different cooling operation modes and BESS operation at each time-step.

The three different cooling operation modes showed in Figure 4.22 are i) charging mode, where cooling energy is provided to both storage tank and building (if requested) simultaneously, ii) discharging mode, where cooling energy is provided to the building to meet the demand only through the storage and iii) chiller cooling mode, where cooling energy is provided to the building exclusively through the electric chiller.

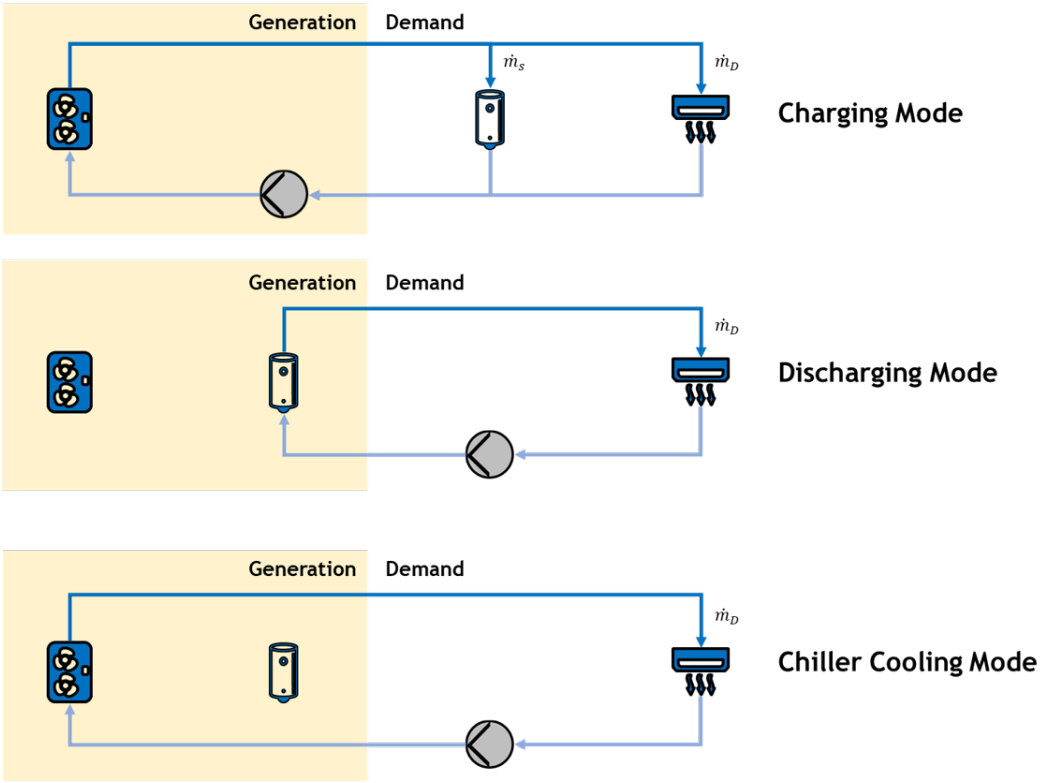


Fig. 4.22 Schematics of the three different modes of the cooling system analyzed [26].

Discharging mode and chiller cooling mode were introduced considering that the system configuration was not conceived to provide cooling to the building via two separate sources at the same time. However the two modes were introduced to allow the control agent to select during building operation at each control step the one that is optimal according to boundary conditions (i.e. employ chiller also during high price periods due to high PV production).

The proposed control strategy couples DRL to manage the cooling system operation with RBC which is employed to manage the BESS. Conversely, the baseline employs a fully RBC strategy to manage both BESS and cooling system operation. The case study was designed to assess the effect of adopting advanced control strategies also considering the performance for different sizes and capacities of TES and BESS, respectively.

4.3.3 Methodology

This section describes the methodological steps and the main methods adopted in the present application. The case study introduced in section 4.3.2 was used as a test-bed to assess the effectiveness of an advanced control strategy consisting of a DRL coupled with a RBC for an office building with IES

A DRL control agent was developed and trained in order to identify the optimal control policy for the management of the cooling modes. The performances of the proposed control strategy was evaluated against a baseline consisting of a fully RBC for different configurations of storage systems. A different DRL control agent was trained for each configuration resulting from the combination of BESS capacity and TES size.

Design of baseline and proposed control strategies

As introduced in the previous section the baseline controller employs a fully RBC approach. This strategy was conceived to simulate the performance of classical control approaches applied to manage TES and BESS as two distinct system. Two RBC strategies were separately designed to control the cooling operation modes and BESS without sharing mutual information between the two. This hypothesis was deemed legitimate since BESS and TES equipment are usually implemented by different stakeholders.

On the other hand, the proposed controller employs an approach where DRL control agent was coupled with an RBC strategy. The BESS system was managed by the same RBC strategy employed by the baseline controller. Conversely, the management of cooling operation modes which involves TES was implemented through an advanced DRL controller which exploits also information on PV production and BESS status. The reason behind the choice to couple DRL with an RBC controller is that this latter strategy is very effective in managing BESS considering building demand, electricity price, and PV production [169, 170]. However, the management of cooling modes requires an advanced controller capable of considering also the boundary conditions determined by the PV system and the BESS in selecting the optimal action. Thanks to this approach, the proposed controller operates with a comprehensive perspective of the whole energy system.

The discrete SAC control algorithm described in section 2.1.2 was employed as DRL control strategy. This control framework was chosen over the classical DQN approach for its sample efficiency and considering the difficulties of the DQN algorithm in balancing exploration and exploitation. In the next sub-sections, the design of the action-space, state-space and reward function are discussed along with the configuration of the training phase.

Design of the action-space

The control action determines the operation mode of the cooling system at each time-step. Since three operation modes were defined for the proposed case study, the action-space was designed as a discrete space as follows:

$$A_{(t)} = [0, 1, 2] \tag{4.5}$$

where 0 correspond to discharging mode, 1 to chiller cooling mode and 2 to charging mode as described in section 4.3.2.

Design of the reward function

The reward measures the performance of the controller after selecting an action at each time-step. The controller operates with the aim to minimize the energy cost related to the energy exchanged between the electrical grid and the system (E_{grid}). Electrical energy can be imported from the grid when there is no PV power generation and the BESS system is out of charge. Electrical energy is injected to the grid when PV power generation excesses the building electrical demand and the BESS system is fully charged. The electrical energy exchanged with the grid was defined as negative when it is imported from the grid and positive when injected. The reward function was defined as follows:

$$\begin{cases} r(t) = \beta E_{grid}(t) \cdot C_{buy}(t) & if E_{grid}(t) < 0 \\ r(t) = \beta E_{grid}(t) \cdot C_{sell}(t) & if E_{grid}(t) > 0 \end{cases} \tag{4.6}$$

Where $C_{buy}(t)$ and $C_{sell}(t)$ are defined according to the schedule price for buying and selling electricity and β is a factor introduced to weight the magnitude of

the reward, namely reward scale, and it is considered an hyper-parameter of the algorithm.

Design of the state-space

The state-space includes all the variables employed by the SAC control agent to determine at each time-step the optimal control action capable to maximize the stream of future rewards. Moreover, the state-space may include information relative to historical values of the variables describing the behavior of the system and future values of external disturbances. In this application, information about historical values were introduced to account for slow-responsive thermal dynamics of the components of the controlled system. At the same time, future values of external disturbances were introduced since they can provide crucial information that the agent can leverage to optimally solve the control problem. In the present application perfect predictions of external disturbance were employed.

More detailed information on the variables included within the state-space are provided in section 4.3.4.

Setting of the training phase

The control policy of the SAC agent was trained on a model of the proposed case study described in section 4.3.2. During the training process a specific period called episode was presented multiple times to the control agent in order to gradually improve its control policy by enabling the exploration of different trajectories. At the end of this process the trained agent was statically deployed on the same episode in order to evaluate its control performance. The static deployment of a SAC agent was achieved by stopping the update of the parameters determining the control policy and employing the actor network to select the optimal control actions given the state of the environment.

Design of BESS and TES configurations

Different configurations consisting in the combination of various volumes of the cold-water storage tank and nominal capacities of the BESS were investigated. The aim is to find out how the proposed advanced control strategy can improve

the performance with respect to a classical control strategy while implementing storage equipment with various sizes and capacities. The objective is to evaluate if the introduction of advanced control strategies could support the introduction of equipment characterized by smaller sizes and capacities. Thus, reducing the initial investment cost which is decisive to guarantee the spread of the storage technologies.

4.3.4 Implementation of the proposed methodology

The test facility analyzed in this application consists of two study rooms, one control room and a technical room. The technical room is not served by the air-conditioning system and the storage tank is placed within it.

The facility is a prefabricated building with a rectangular layout. The floor area is 196.3 m^2 (11.25 × 17.45 m). The interior gross floor conditioned area is around 96.8 m^2. The ceiling height is 2.8 m at the minimum and 3.7 m at the maximum above the floor level, due to the different tilt angles of the roof, which are 13.4° on SE side and 15° on NW side. The features of the building envelope are reported in Table 4.14.

Table 4.14 Features of the building envelope [26].

Feature	Value
Conditioned floor area	96.8 m^2
Conditioned volume	501 m^3
Envelope surface/conditioned volume ratio	0.85 m^{-1}
Transparent/opaque envelope surface ratio	6.6%
Opaque envelope surface	400 m^2
$\hat{U}_{op}$	0.16 W/m^2K
$\hat{U}_{tr}$	0.55 W/m^2K

The chiller has a reference capacity Q_{cap} of 12 kW and reference COP of 2.67. The reference COP is provided by the employed chiller model provided by EnergyPlus and it is calculated considering a reference leaving chilled water temperature of 6.67 °C and a reference entering condenser fluid temperature of 35 °C. The design water mass flow rate during charging phase ($\dot{m}_S$) is 0.2 kg/s while during discharging phase ($\dot{m}_S$) is 0.35 kg/s. This latter value corresponds to the sum

of the design mass flow rates of the three air-conditioned zones. The supply water temperature at the outlet of the chiller was set equal to 7 °C. The TES operates in the range between 10°C and 18°C which correspond to a state-of-charge (SOC_T) of 1 and 0, respectively.

The HVAC system serving the building can meet the cooling demand through the electric chiller or the TES. The building cooling demand was considered as an external disturbance of the system and was calculated through EnergyPlus considering an indoor air temperature of 26 °C and a relative humidity of 55% during occupancy periods which occur between 09:00 and 18:00 from Monday to Friday. During these periods, the zones were supposed to be occupied at their maximum capacity (i.e. 3 people for the control room and 10 people for the two study rooms). No regular occupancy was expected for the technical room. The air infiltration rate was set to 0.15 h^{-1}, a typical value for office buildings. The air ventilation rate for the control room and the study rooms was set to 10 L/s per person according to Italian standard UNI10339, resulting in 30 L/s and 100 L/s, respectively.

The price of the electrical energy drawn from the grid to operate the chiller unit and auxiliary equipment is based on a Time-Of-Use (TOU) tariff structure commonly implemented in Italy. The weekly period is divided into low price, medium price and high price periods, corresponding to 0.03 €/kWh, 0.165 €/kWh and 0.3 €/kWh respectively. The tariff rates of the electricity were designed in order to discriminate the values for the optimization application starting from a real value of the high price period. This approach has been found to be effective in ensuring better discrimination of time periods of the day based on the price of electricity providing the agent with faster convergence to the optimal control policy. Specifically the low and medium price values were chosen to be respectively 1/10 and 1/2 of the highest one. Table 4.15 reports a summary of electricity prices used in this application.

Table 4.15 Details of electricity prices used in this application in €/kWh [26].

Day	**Hour of the Day**				
	00:00-07:00	07:00-08:00	08:00-19:00	19:00-23:00	23:00-24:00
Mon-Fri	0.03	0.165	0.3	0.165	0.03
Sat	0.03	0.165			0.03
Sun	0.03				

The price of the electrical energy sold to the grid from the PV overproduction was assumed equal to 0.01 €/kWh according to data extracted from the Italian regulator .

The weather file used is the reference weather file (ITA_TORINOCASELLE_IGDG.epw) available in EnergyPlus for Torino, Italy. Considering that the system under investigation involves the optimization of a cooling system the simulation period was limited from June to August. Both the control and simulation time-steps were set equal to 1 hour.

The efficiency of mono-directional DC/AC was assumed to be equal to 90% and the efficiency of DC/DC converters to 95%.

The experiments were carried out in a co-simulation environment described in Chapter 3. Building dynamics and the cooling system were implemented in EnergyPlus while the electrical system including PV and BESS was developed in Python along with the different control strategies.

Modeling of the PV system

The model of the PV system was implemented through a Python class. Solar position was imported from the pvlib package [171]. A commercial mono-crystalline silicon photo-voltaic module was modeled in the proposed environment. The selected module has a specific power of about 80 W/m^2 and an efficiency (η) of 15% under standard conditions (solar irradiance G_{STC} = 1000 W/m^2, cell temperature T_{STC} = 25°C, Air Mass AM_{STC} = 1.5), as described by Durisch et al.[172] and reported in Equation 4.7.

$$\eta = f(G, AM, T_{out}) \tag{4.7}$$

The PV panels tilt angle has been chosen from the world data-set provided by M.Z. Jacobson and V. Jadhav [173]. Thus, the tilt angle was set to 33°, whereas the azimuth is constrained by the orientation of the test facility. These inputs along with solar radiation and incidence angle allow to compute the PV power generation (P_{PV}) at each time-step which was calculated as the product of the efficiency and incident solar radiation. Table 4.16 recaps the parameters of the PV module.

Table 4.16 PV parameters [26].

Parameter	Value
Nominal power	3 kW
Surface	22 m^2
η_{STC}	0.15
Tilt angle	33°
Azimuth angle	116°

The nominal power of the PV system of 3 kW was chosen in order to match up to the peak power of the building total electrical demand.

Modelling of the BESS system

The battery system was simulated through a Python class. A simple and widely adopted model was implemented according to [169]. The model involves the estimation of the State-Of-Charge (SOC), which it was considered sufficiently accurate for carrying out a preliminary evaluation of the impact of BESS installation, even though the degradation of the battery is not taken into account. The calculation of the SOC at each time-step t was performed according to the set of equations reported in Equation 4.8:

$$\begin{cases} SOC_B(t) = SOC_B(t-1) + \eta_{rte}\frac{P_{B,ch}(t)*\Delta t}{C_B} & (charge) \\ \\ SOC_B(t) = SOC_B(t-1) - \frac{P_{B,dis}(t)*\Delta t}{C_B} & (discharge) \end{cases} \tag{4.8}$$

where $SOC_B(t-1)$ is the SOC at the previous time-step and η_{rte} is the round-trip efficiency. $P_{B,ch}$ and $P_{B,dis}$ are the average power exchanged in the period between two consecutive the time-steps (Δt) between the BESS and the system during charging and discharging process respectively. C_B is the battery nominal capacity. Safety constraints were introduced in order to preserve battery lifetime. Charging and discharging processes have to respect two limits defined by $P_{B,ch,max}$ and $P_{B,dis,max}$. These values are introduced in the technical specifications to avoid too rapid charging/discharging operations. Typically, maximum charging and discharging power are

different and when the power exceeds these thresholds, the controller limits it to the maximum recommended values. In order to preserve the health of the battery, the levels of the SOC were constrained by the minimum and maximum values provided by the manufacturer (i.e. $SOC_{B,min}$, $SOC_{B,max}$).

The characteristics of the BESS considered in this application were gathered from the data sheet of a modular Li-ion battery available on the market and reported in Table 4.17.

In compliance with the typical values for the lithium-ion technology the minimum SOC value ($SOC_{B,min}$) was set equal to 10% and the maximum SOC value ($SOC_{B,max}$) was set equal to 90% for a total Depth of Charge of 80% [169]. An initial SOC of 50% was imposed. The maximum charging power ($P_{B,ch,max}$) and maximum discharging power ($P_{B,dis,max}$) were set equal to 0.5 times and 1 time the nominal capacity of the battery (C_B) respectively.

Table 4.17 BESS characteristics [26].

Parameter	Value
Round-Trip Efficiency	0.96
Maximum discharging power	1C
Maximum charging power	0.5C
$SOC_{B,min}$	10%
$SOC_{B,max}$	90%

Setup of BESS and TES configurations

As introduced in Section 4.3.3 the baseline and the proposed control strategies were implemented considering different capacities of BESS and different sizes of TES.

Table 4.18 reports for each size of TES the total volume and the corresponding UA-value considered to estimate heat losses. The largest size of 10 m^3 was chosen considering 3-times the maximum daily cooling demand of the building. The smallest size of 3 m^3 was chosen considering 2-times the maximum hourly cooling demand of the building. The intermediate values were picked up according to commercial sizes between minimum and maximum values.

Table 4.18 TES configurations [26].

Volume [m^3]	UA-value [W/K]
10.0	12.0
8.0	10.3
6.0	8.5
3.0	6.0

Table 4.19 reports the features of the various configurations of the BESS. A commercial capacity for the battery unit of 2.4 kWh has been chosen as a reference. This value was selected according to the maximum value of the building electrical demand on an hourly basis. The other two capacities of BESS are supposed as obtained by connecting in series two and three units respectively.

Table 4.19 BESS configurations [26].

Capacity $[kWh]$	Max Charging Power $[kW]$	Max Discharging Power $[kW]$	Units in Series
2.4	1.2	2.4	1
4.8	2.4	4.8	2
7.2	3.6	7.2	3

Eventually, Table 4.20 summarizes all the configurations resulting from the combination of the different capacities of BESS and sizes of TES that have been tested with both baseline and proposed control strategy.

Table 4.20 Configurations simulated for the experiment [26].

Configuration	BESS capacity $[kWh]$	TES volume $[m^3]$
1	2.4	10.0
2	4.8	10.0
3	7.2	10.0
4	2.4	8.0
5	4.8	8.0
6	7.2	8.0
7	2.4	6.0
8	4.8	6.0
9	7.2	6.0
10	2.4	3.0
11	4.8	3.0
12	7.2	3.0

Implementation of the baseline fully Rule-Based Control

As introduced in section 4.3.3, the baseline strategy manages both the operational modes of the cooling system (and consequently the TES) and BESS through two different RBC strategies. The baseline RBC strategy operates the cooling system in charging mode whenever the price of electricity is low (i.e. between 11 p.m and 7 a.m during Mondays and Saturdays and between 0 a.m and 24 p.m during Sundays) and the temperature of the TES is greater than 12 °C. During these periods the storage is charged until its temperature reaches 10 °C or the price of electricity rises. The cooling system is operated in discharging mode whenever the building cooling demand is not zero until this value returns to zero or the temperature of the TES is greater than 18 °C. If the temperature of the TES is greater than 18 °C) and building cooling demand is not zero the cooling system is operated in chiller cooling mode.

A simple still effective controller inspired from previous scientific literature [170, 169] was implemented for BESS management. The BESS is charged when PV generation is greater than the building electrical demand, otherwise it is discharged. More specifically, during charging process the PV surplus is diverted to the BESS if it is allowed by the constraints on charging power ($P_{B,ch,max}$) and maximum SOC

($SOC_{B,max}$). If PV generation is greater than the sum of building electrical demand and BESS capacity the remaining overproduction is diverted to the grid. During discharging, the BESS works in parallel with the PV to meet the electrical demand. If the contribution from both PV and BESS is not sufficient to meet the building electrical load the grid is employed to meet the demand.

Implementation of the proposed control strategy based on DRL coupled with RBC

As introduced in section 4.3.3 the SAC agent manages the three cooling operation modes (i.e. charging mode, discharging mode and chiller cooling mode) while the BESS is managed by the same RBC strategy described in the section above. The SAC agent is defined through the reward function, the action space and the state space. Table 4.21 reports the variables included in the state-space.

Table 4.21 Variables included in the state space [26].

Variable	Min Value	Max Value	Unit	Time-step
Outdoor Air Temperature (T_o)	7.0	40.0	°C	t
TES SOC (SOC_T)	0.0	1.0	-	t, t-1, t-2
BESS SOC (SOC_B)	0.0	1.0	-	t
Building Cooling Demand (Q_d)	0.0	10.0	kW	t, t+1, ...,t+24
PV power generation (P_{PV})	0.0	3.0	kW	t, t+1, ...,t+24
Electricity price (C_{buy})	0.03	0.3	€/kWh	t, t+1, ...,t+24

The state-space was conceived to provide to the agent comprehensive information about the whole IES including PV production and BESS status. Observations of the storage tank including the SOC (SOC_T) evaluated at the current time-step t and up to two time-step $(t-2)$ in the past were provided to the agent. These values carry information about the amount of cooling energy actually stored and its evolution over time.

The SOC of the BESS is also a key-information provided to the agent to correctly manage the operation of the cooling system. BESS is operated to provide electricity to the chiller and the pumping system during high price periods. This value was

provided only at the current time-step t due to lower inertia of BESS compared to TES.

The electricity price is the main driver of the agent choices since it strongly influences the reward. Current value was provided to the agent along with the exact values for 24 hours ahead. The electricity price schedules were supposed to be always known.

The building cooling demand together with the PV power generation is a fundamental information to optimally manage the controlled system. Also, the values related to time-step t to time-step $t+24$ were provided to the agent. The predictions of building cooling demand and PV power generation were assumed to be perfectly known.

Eventually, information about outdoor air temperature were included in order to provide knowledge about its influence on the COP of the chiller unit. Despite being a key information, the solar irradiation was not included in the state-space since the PV power generation is directly related to this variable.

Table 4.21 reports the maximum and the minimum values that were employed to re-scale the state space through a min-max normalization before providing the variables to the neural network models.

Besides the definition of state-space, action-space and reward function, the SAC algorithm is characterized by a series of hyper-parameters. The settings of these hyper-parameters adopted in this application are reported in Table 4.22.

Table 4.22 Hyperparameters of the SAC control agent [26].

Hyperparameter	Value
Discount factor (γ)	0.99
Optimizer learning rate	0.001
Boltzmann temperature coefficient (α)	0.2
Number of hidden layers	2
Number of neurons per hidden layer	256
Activation Function	ReLu
Optimizer	Adam
Batch size	32
Number of training episodes	30
Reward magnitude weight-factor (β)	100

Each episode (i.e. one cooling season lasting from June to August) is presented to the SAC control agent 30 times in order to train the control policy for each configuration. At the end of the training process the SAC agent was statically deployed for one single deployment episode corresponding to the same cooling season as the training episode. When the SAC agent is statically deployed the control policy is determined by the weights resulting from the last update (i.e., the last control step of the last training episode) of the training phase. For this reason, as common practice, the static deployment of the SAC agent was performed on the same period (i.e., from June to August) of the training. In fact, during the deployment process the performance of the agent during the training period could provide a good indication of the stability of the learned control policy.

Co-simulation details

The experiments were carried out in the co-simulation described in Chapter 3. Figure 4.23 provides further detail the architecture of the co-simulation environment for this application. The architecture is organized in two sides.

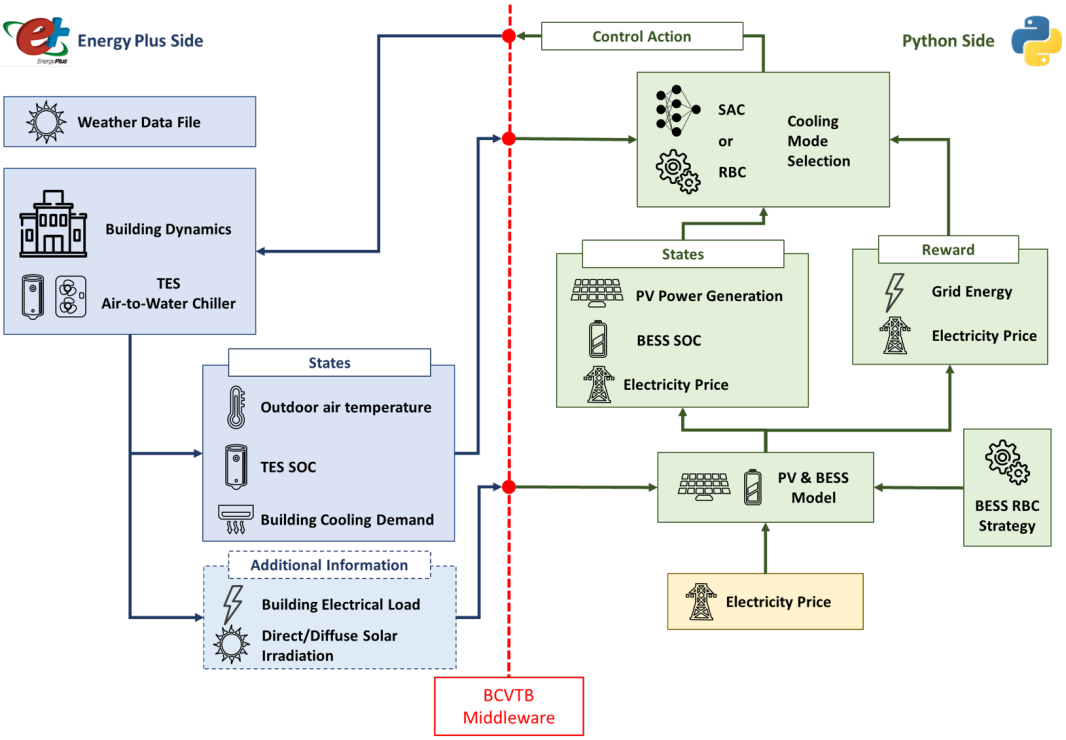


Fig. 4.23 Architecture of the co-simulation environment.

The EnergyPlus side is formed by a model of the building dynamics and of the cooling system (comprising of the air-to-water chiller and TES) receiving at each time-step information from the weather data file and a controller which selects the cooling mode. This model provides in output the state variables (i.e. outdoor air temperature, TES SOC and Building Cooling Demand) employed by SAC agent evaluated at each time-step. Moreover, the Energy Plus model produces additional information such as building electrical load and direct and diffuse solar irradiation employed by the PV and BESS models.

The Python side of the co-simulation environment is formed by the PV and BESS models and by the control strategies employed to manage the integrated energy system. The PV model employs solar irradiation to calculate the PV power generation which is one of the state variables provided to the SAC agent. The BESS SOC is evaluated through the BESS model which receives information to whether charge or discharge the battery from the BESS RBC strategy described in the previous section. This strategy manages the BESS according to building electrical load (provided by EnergyPlus and determined by chiller and pump operation), PV

power generation and electricity price (provided to Python through a csv file). The electricity price is furtherly forwarded as a state variable employed by both the SAC agent and the RBC strategy to select the cooling mode at each time-step. Once the BESS operation is evaluated, the environment evaluates the energy exchanged with the electrical grid which determines the reward obtained by the SAC agent along with the electricity price. The final component of the Python side is represented by either the SAC agent or the RBC strategy employed to select the cooling mode. The SAC strategy makes use of all the information included within the state space and the reward function to learn the optimal control policy. The RBC strategy employs only TES SOC and electricity price to determine the control action. Eventually, the control action is forwarded to EnergyPlus in order to advance the simulation to next time-step.

4.3.5 Results obtained

This section reports the results of the implementation of the methodology introduced in section 4.3.3.

A SAC control agent coupled with RBC was simulated together with a baseline fully RBC strategy during the cooling season in the period ranging from June to August for different sizes and capacities of TES and BESS, respectively. For the sake of simplicity, in the following sections the proposed controller which couples SAC with RBC is indicated as SAC, while the baseline fully RBC strategy is simply indicated as RBC.

Table 4.23 reports both electrical energy imported from and sold to the grid together with the electricity costs achieved by implementing SAC and RBC strategies for each configuration during the whole simulation period. The last column of the table reports the monetary savings achieved through the implementation of SAC strategy.

Table 4.23 Energy imported from grid ($E_{grid,buy}$), energy sold to grid ($E_{grid,sell}[kWh]$), Cost of electricity and economic savings obtained from the implementation of SAC agent and RBC strategy [26].

Config	$E_{grid,buy}[kWh]$		$E_{grid,sell}[kWh]$		Cost [€]		Cost Savings
	SAC	RBC	SAC	RBC	SAC	RBC	[%]
1	314.70	871.40	380.90	919.10	6.0	16.9	64.7
2	223.70	749.10	274.60	776.80	6.5	14.7	55.8
3	172.40	628.60	222.30	636.50	3.9	12.5	68.8
4	292.20	872.70	357.50	928.10	6.9	16.9	59.2
5	310.60	750.90	355.90	786.40	8.9	14.7	39.5
6	147.90	632.00	193.70	648.00	3.4	12.5	72.8
7	355.40	861.10	420.80	928.90	8.2	18.1	54.7
8	231.10	747.20	281.90	796.20	5.2	14.9	65.1
9	188.20	636.60	230.40	667.30	5.3	12.5	57.3
10	281.20	797.00	358.50	862.00	7.7	49.2	84.3
11	209.10	693.00	271.70	740.70	4.9	24.5	80.0
12	178.00	591.50	233.20	622.40	6.1	12.5	51.2

The results in Table 4.23 show that SAC control policy learnt to minimize the interactions with the electrical grid with respect to RBC strategy. Across all configurations the energy imported from grid and energy sold to grid were on average 67% and 61% lower for SAC strategy compared to RBC strategy. RBC performance in terms of operational cost improved with the increasing of BESS size. A cost reduction between 26.1% and 74.3% was achieved by the baseline strategy when nominal capacity was increased from 2.4 kWh to 7.2 kWh.

Independently from TES size, RBC achieved the best performance with a BESS capacity of 7.2 kWh (i.e. configurations 3, 6, 9 and 12). The increase of TES size beyond 6 m^3 did not lead to significant improvements in terms of operational costs of RBC strategy for the configurations implementing the same BESS capacity (i.e. configurations from 1 to 6).

Similarly to RBC, the operational cost with the SAC control agents decreased with the increase of BESS capacity. However, due to their intrinsic stochastic nature in the training process and initialization of the neural network policy their performance did not show a linear pattern.

SAC strategy led to the best performances with the configurations implementing 8 m^3 and 10 m^3 leading to a monetary expense of 3.4€, and 3.9€, respectively.

The SAC control agents led to a better performance than the RBC with an economic savings ranging from 39.5% to 84.3%. The highest difference between the two control strategies were achieved for configuration 10 implementing both TES and BESS with the lowest sizes.

Table 4.24 reports the building electrical consumption over the simulation period (E_{dem}) along with the percentages indicating the contribution of each source by implementing SAC and RBC strategies. PV_{frac}, $BESS_{frac}$ and $Grid_{frac}$ indicate the percentage of electrical demand satisfied by PV generation directly provided to the building, by BESS and through the grid, respectively.

Table 4.24 Contribution of the different sources (PV, BESS and Grid) to the building electrical demand (E_{dem}) obtained by SAC and RBC strategy for the different configurations [26].

Config	$E_{dem}[kWh]$		PV_{frac}		$BESS_{frac}$		$Grid_{frac}$	
	SAC	RBC	SAC	RBC	SAC	RBC	SAC	RBC
1	1070.5		0.56		0.14	0.12	0.30	0.80
2	1075.4	1090.7	0.58	0.08	0.21	0.23	0.21	0.69
3	1064.2		0.55		0.29	0.34	0.16	0.58
4	1073.10		0.60		0.13	0.12	0.27	0.80
5	1078.70	1083.0	0.50	0.08	0.21	0.23	0.29	0.69
6	1069.10		0.58		0.28	0.34	0.14	0.58
7	1070.20		0.52		0.15	0.11	0.33	0.80
8	1064.30	1072.2	0.53	0.09	0.25	0.22	0.22	0.69
9	1063.60		0.51		0.31	0.32	0.18	0.59
10	1055.70		0.57		0.16	0.11	0.27	0.74
11	1053.30	1075.1	0.54	0.15	0.26	0.20	0.20	0.65
12	1058.00		0.54		0.29	0.30	0.17	0.55

In the case of RBC strategy, independently from TES size, the implementation of different BESS capacities had no influence on the percentage contribution of PV generation directly feeding the building and the electrical energy demand as can be seen for the configurations 1-3, 4-6, 7-9 and 10-12, respectively. Generally, SAC led to lower energy consumption compared to RBC, as shown by second and third column (i.e. E_{dem}) suggesting that SAC learnt a better management strategy.

Moreover, as shown by column PV_{frac}, the SAC strategy was capable to better exploit PV generation to feed the building with respect to RBC. In the case of baseline controller the percentage contribution of PV generation directly feeding the building ranges between 8 % and 15 % increasing with the reduction of TES size. SAC outperformed RBC exploiting the PV production in a range between 50 % and 60 % across all configurations.

With the increasing of the BESS capacity RBC was capable to shift the contribution from the grid to the BESS. SAC and RBC showed similar utilization of the BESS system among all configurations.

Considering the configurations implementing the smallest BESS capacity of 2.4 kWh (i.e. configuration 1, 4, 7, 10) the configuration 10 is the one which led to the highest operational cost despite the lowest percentage of electricity drawn from the grid with respect to configurations 1, 4 and 7. This pattern suggests that in that case the RBC controller was forced to rely on electrical grid to operate the chiller during high-price periods due to not enough thermal or electrical energy stored.

Key indicators to assess the performance of PV-BESS systems are the Self-Sufficiency (SS) and the Self-Consumption (SC), the former describing the amount of the demand which is satisfied by the local generation, the latter the amount of the local generation which is consumed in place. SC also indicates the economic viability of the PV systems which is usually increased through the introduction of BESS. Since the BESS is charged only through PV, the value of PV generation employed to calculate SS and SC comprises the PV generation directly feeding the building and the electricity provided to the building by the BESS.

Figure 4.24 shows the SS and SC resulted from the implementation of SAC and RBC strategies for all the configurations of storage analyzed.

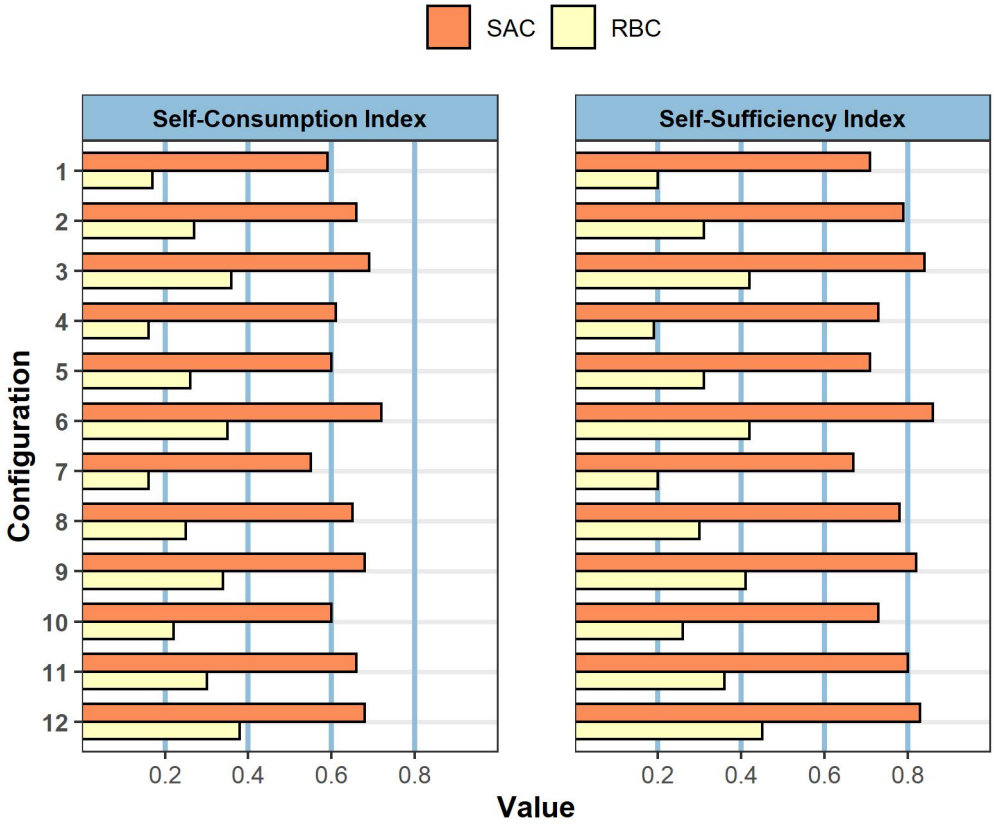


Fig. 4.24 SS and SC indices obtained by implementing SAC and RBC for all configurations of TES and BESS analyzed [26].

The results show that TES volume did not significantly affect SS and SC values. SAC performed significantly better than RBC, increasing SS and SC with an average value of 40 % considering all the configurations. Moreover, RBC performance was affected by BESS capacity both in terms of SS and SC, whereas SAC managed to maintain their values almost constant among the configurations.

Table 4.25 reports the TES operation in terms of thermal energy charged (*Charge*) and discharged (*Discharge*) along with the percentage of the building cooling demand (*Demand*) satisfied through storage discharging by implementing SAC and RBC strategies.

Table 4.25 Thermal energy exchanged by the TES during charging (Charge) and discharging (Discharge) phases and percentage of building cooling demand satisfied (Demand) by implementing the different control strategies [26].

Config	Charge$[kWh_{th}]$		Discharge$[kWh_{th}]$		Demand $[\%]$	
	SAC	RBC	SAC	RBC	SAC	RBC
1	1825.0		1703.0		54.30	
2	1630.1	3307.4	1494.3	3132.8	47.65	99.96
3	1594.4		1488.2		47.43	
4	1643.3		1518.6		48.40	
5	1975.9	3281.0	1829.3	3129.0	58.34	99.77
6	1622.1		1508.6		48.06	
7	1895.2		1775.4		56.57	
8	1649.4	3160.2	1538.1	3046.3	48.99	97.06
9	1548.0		1435.6		45.74	
10	1429.2		1351.4		43.00	
11	1444.8	2234.8	1372.6	2131.9	43.68	67.88
12	1420.9		1339.9		42.65	

The results show that the operation of the thermal storage was not influenced by the capacity of BESS when the RBC is employed. On the other hand, SAC managed the system by charging less the TES when the capacity of the BESS is higher. Moreover, while the RBC almost fully met the building cooling demand demand through TES discharging for the configurations implementing a TES size greater than 6 m^3, SAC met only the 48.7% on average among all configurations.

These patterns along with the results presented in Tables 4.23 and 4.24 suggest that SAC learnt to optimally manage the cooling system and the thermal storage in coordination with local PV production and BESS.

Figures 4.25 and 4.26 report the SOC profiles for both BESS and TES resulted from RBC and SAC implementation for configuration 3 and 10 respectively during the month of August. The black dotted lines indicates the beginning of a different week (i.e. from Monday to Sunday).

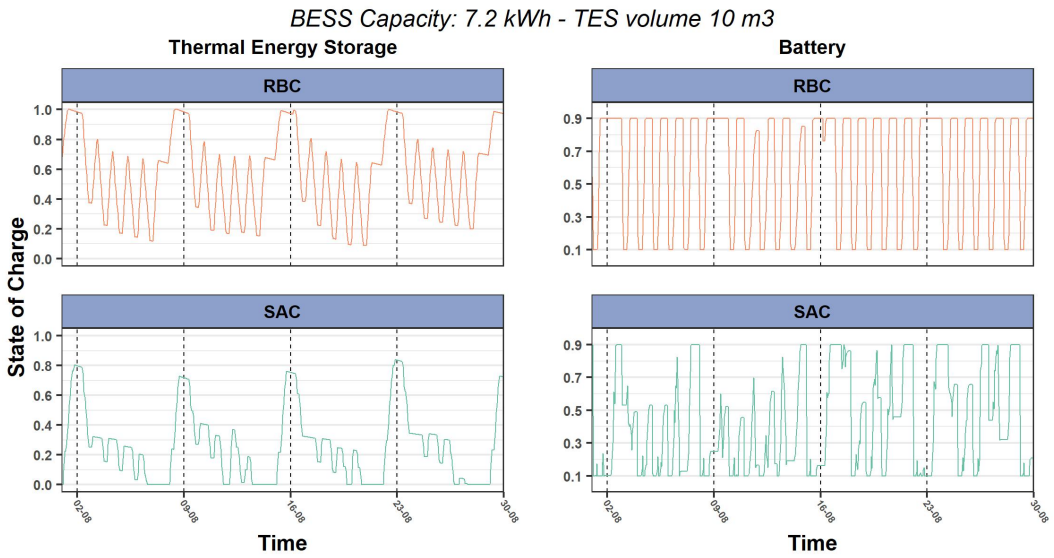


Fig. 4.25 TES and BESS SOC resulted by SAC and RBC implementation during the whole simulation period for system configuration 3 [26].

Configuration 3 implements the highest sizes for both TES and BESS (ie. 10 m^3 and 7.2 kWh). It can be observed that SAC learnt to manage the thermal storage to maintain in average a lower SOC of the system compared to RBC. In particular, the SAC agent charged the TES at the beginning of the week and gradually released this energy during the first days of the week. Despite the controllers directly act only on the operational state of the cooling system, the control strategies affected also the operation of the BESS. The BESS was charged and discharged more frequently when the SAC strategy is adopted compared to the case implementing RBC strategy.

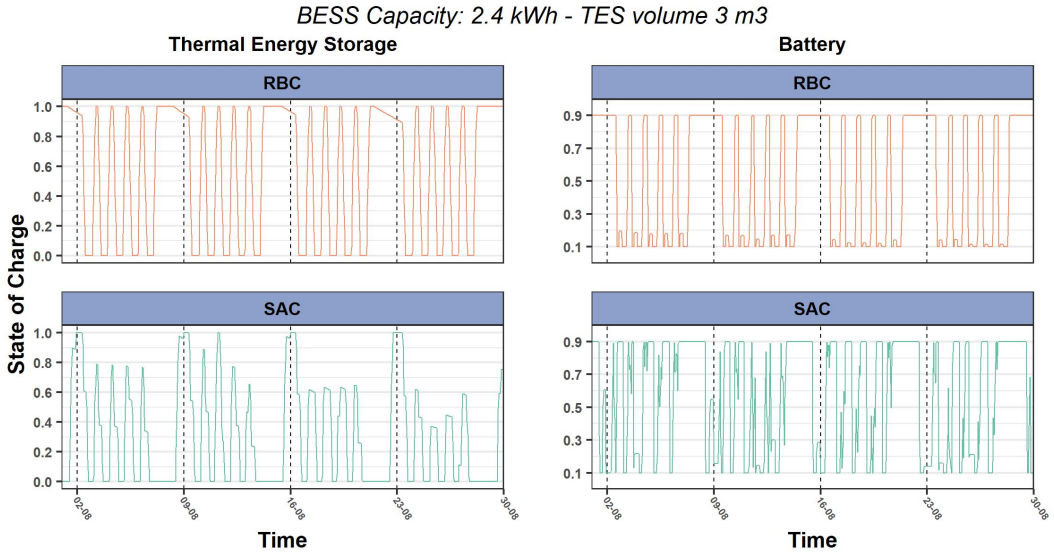


Fig. 4.26 TES and BESS SOC obtained by SAC and RBC during the whole simulation period for system configuration 10 [26].

The variation of the SOC of BESS and TES for configuration 10 which implements the lowest sizes for both TES and BESS (ie. 3 m^3 and 2.4 kWh) is reported in Figure 4.26. Also in this case SAC managed the cooling system in order to maintain the SOC of the thermal storage as low as possible. This pattern is particularly evident during weekends in which RBC maintained a SOC close to 1 while SAC maintained it close to zero until the beginning of the successive week. Also for this configuration SAC showed a more variable use of the BESS system than RBC strategy.

Figures 4.27 and 4.28 better depict how the different management strategies of the cooling modes affected the behavior of the whole energy system. The figures show in three subplots the trend of several variables on hourly basis for five days of the simulation period (i.e. between Friday 14-08 and Tuesday 18-08). For the sake of simplicity, only the results obtained for configuration 10 implementing a TES size of 3 m^3 and a BESS capacity of 2.4 kWh are presented. This configuration was chosen since it resulted as particularly representative of the difference between SAC and RBC strategies. The top subplot reports the building total electrical load and the sources through which it is met. Moreover, the subplot reports the PV power production and its dispatchment. The central subplot shows the building cooling demand and the sources employed to meet it along with the cooling energy provided by the chiller to charge the TES. The bottom subplot depicts the trend of

SOC for TES and BESS along with electricity price values scaled with a min-max normalization.

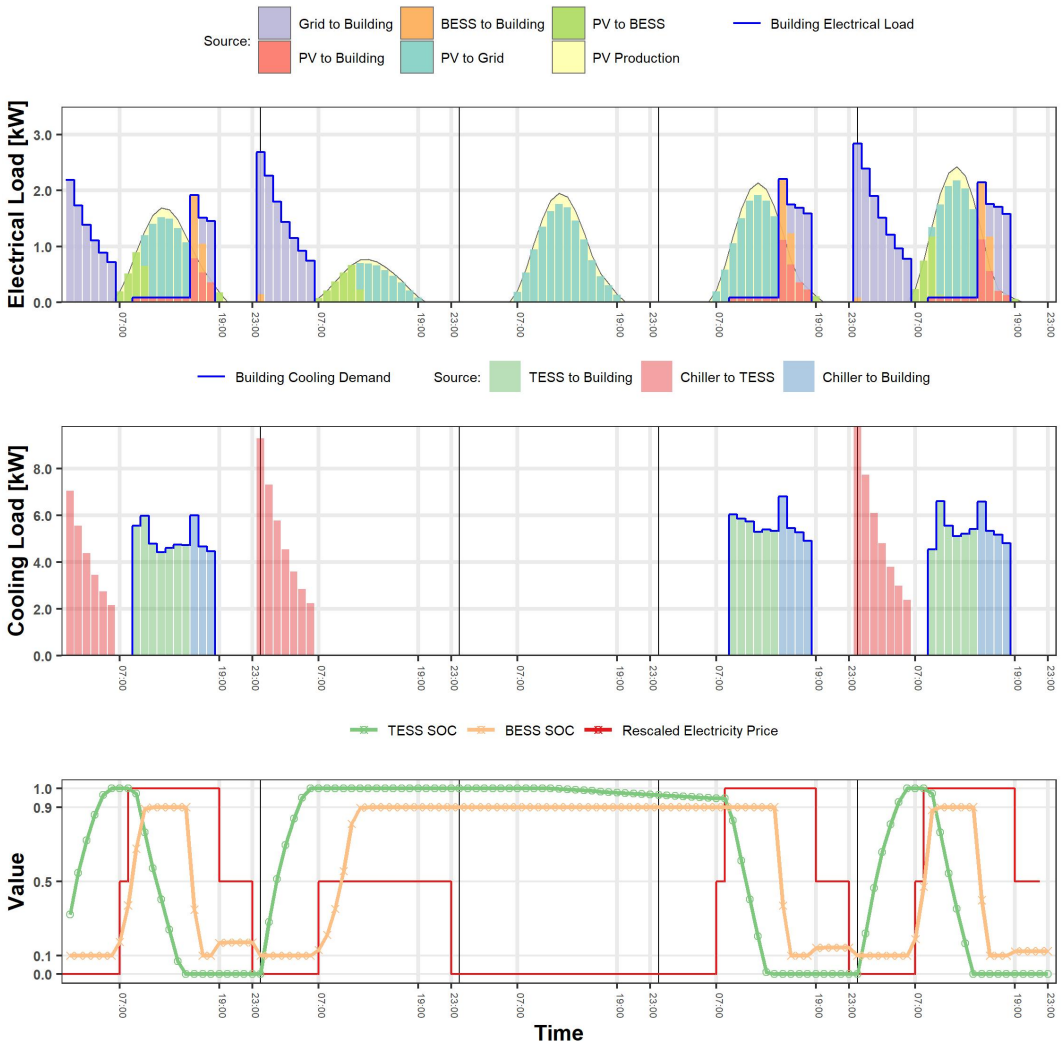


Fig. 4.27 Trends of the electrical load, cooling load and SOC obtained by RBC strategy between Friday 14-08 and Tuesday 18-08 for configuration 10 [26].

Figure 4.27 presents the results with reference to RBC strategy. According to this strategy the TES is charged whenever the price of electricity drop to its minimum value. This behavior generated an electricity demand due to chiller operations mainly during night hours when the PV production is null. As a consequence, the system was forced to import energy from the grid during low-price periods. Until 09:00 AM there is no electrical demand from the building and the PV fed the BESS. When the building is occupied, the TES was discharged to meet the cooling demand while

the building electrical load is determined only by circulation pumps which were powered by PV production. Through this approach, the import of electricity from the grid during high-price periods was avoided. When the BESS was fully charged the PV overproduction was sold to the grid. Since for configuration 10 the BESS capacity is relatively small, the amount of energy sold to the grid during this period is considerable. During the last hours of the day the thermal energy stored within the TES is exhausted and the systems was forced to use the chiller to meet the cooling load. The PV generation was not sufficient to meet the electrical load, and as a consequence, BESS and grid were employed during high-price periods as shown in the bottom subplot. Moreover, it can be noticed that at the beginning of the weekend the RBC strategy immediately charged the TES due to the occurrence of a low price period. The TES was fully charged after few hours and was not discharged until the beginning of the next week. In these periods TES lost part of its thermal energy to the ambient, resulting in a sub-optimal management of the system.

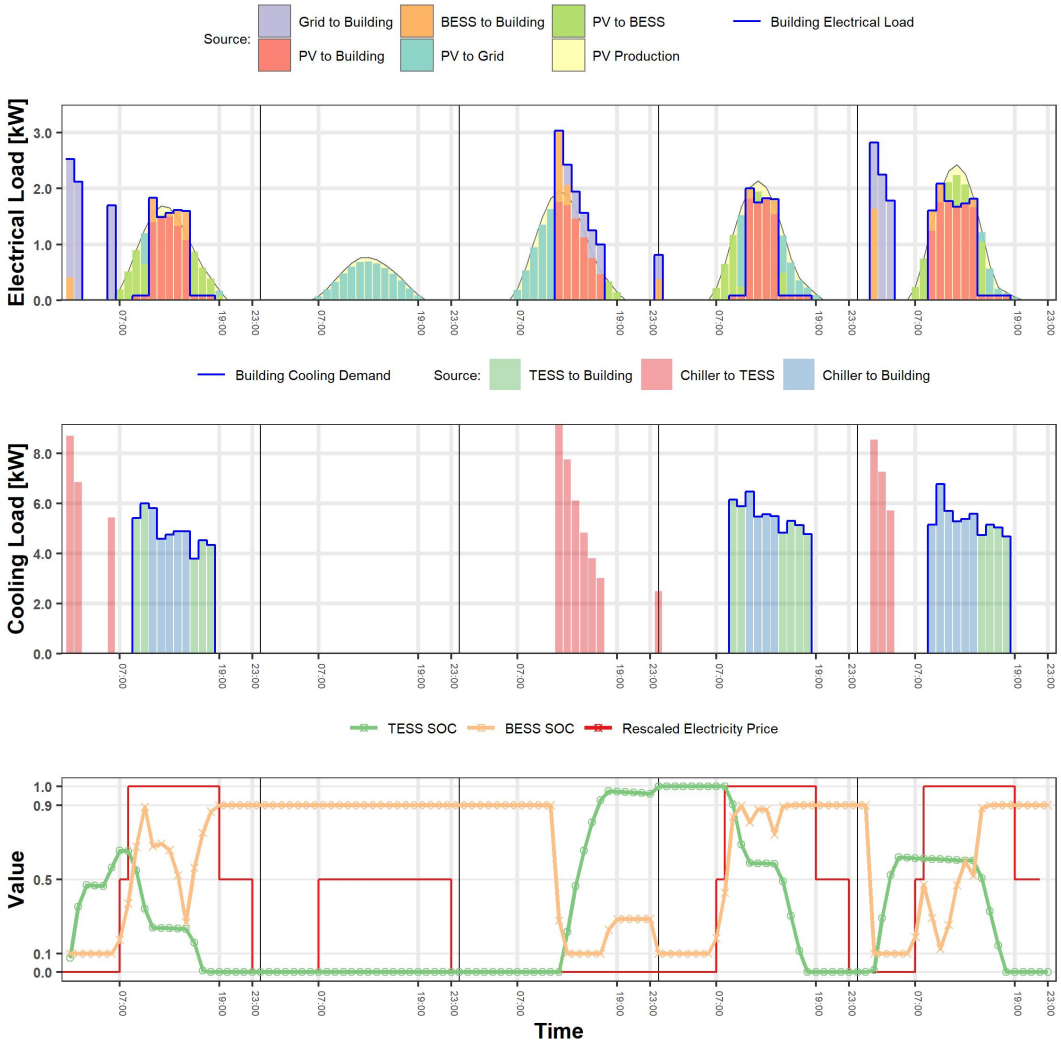


Fig. 4.28 Trends of the electrical load, cooling load and SOC obtained by SAC control strategy between Friday 14-08 and Tuesday 18-08 for configuration 10 [26].

Figure 4.28 shows the results obtained by SAC control strategy. The agent tried to charge the thermal storage during low-price periods close to arrival time of occupants in order to minimize heat losses to the ambient due to storage inactivity. Through this approach, the SAC strategy was capable to reduce electrical energy consumption due to TES charging and, consequently, to consume less electrical energy than RBC as reported in Table 4.24. During the first hours of occupancy in working days the SAC agent followed a similar policy to RBC powering circulation auxiliaries through PV production and charging the BESS at the same time. However, during the central hours of working days, the control policy learnt by the SAC agent is

completely different from RBC. The agent switched the system in chiller cooling mode in order to leverage PV production to feed the chiller avoiding to sell renewable energy to the grid and maximizing SC. When the PV production was not sufficient, the BESS previously charged was employed. During the last hours of the occupancy period which are still characterized by high electricity prices, the cooling system was switched again to discharging mode since the PV and BESS could not meet the electrical load of the chiller. In this period the PV production was employed to operate the circulation pumps and charge the BESS while the excess of energy was sold to the grid. Moreover, SAC control strategy during weekend awaits Sundays to charge the TES in order to minimize electricity cost even during low-price periods and maximizing SC. Through this approach the SAC agent was also capable to limit TES heat losses compared to RBC strategy.

4.3.6 Discussion

The results obtained by applying RBC and SAC strategies for an IES of an office building provided interesting information about the impact of an advanced control strategy on the sizing and operation of energy storage solutions.

SAC was capable to outperform RBC in terms of operating cost for all the configurations of TES and BESS tested. RBC proved to be very sensitive to storage capacities resulting in a huge impact on the operational cost. This aspect is particularly relevant for the BESS capacity.

On the other hand, SAC strategy was able to achieve considerable economic savings also with small capacities, but as the storage capacities increase, the improvement achieved was lower than those achieved by RBC. SAC did not show a clear dependency of the operating cost from the capacities of the storage systems, rather it learnt effective control policies for each configuration. Larger BESS helped SAC in reducing the TES utilization while a similar pattern was not observed for RBC.

BESS is largely considered as the best way to increase SC. However, the operating cost decrease as long as the PV production is sold to the grid leaving no room of improvement of SC levels. Advanced control strategies such as SAC proved to be a viable solution to increase SS and SC levels also with relatively low capacity of the BESS. This is an important aspect to consider given that BESS has a great impact on the total investment cost of energy systems. Reducing the energy exchanged

with the electrical grid results in higher profitability of storage technologies and higher flexibility of the building IES. When PV production is sold to the grid the performance of the system in terms of SC degrades. For this reason, SAC aimed at matching PV production and chiller operation as much as possible. In this way, SAC not only avoided unnecessary BESS operations, which would have involved electrical losses due to the round-trip efficiency and converter efficiency, but it also managed effectively the system with smaller capacity of BESS.

Eventually the energy consumption and PV contribution to building electrical demand was not affected by the capacity of BESS when RBC was employed. The reason might be that RBC strategy employed two distinctive s for both BESS and TES. These controllers were responsible only for their relative system and did not share information between each other. As a consequence, the RBC controller managing the cooling system operation was not aware of PV production and BESS SOC and vice versa. This aspect strongly limited the capability of the RBC to optimally control the proposed system despite the reasonable control rules implemented. This fact clearly shows the limitations of traditional control approaches.

Infact, the RBC strategy implemented in this application was developed making the hypothesis that in classical control approaches the different storage solutions are managed by control laws unaware of other systems. This assumption was deemed reasonable considering that BESS and TES are usually implemented in existing buildings by different stakeholders in different periods of time. Moreover, installers and maintainers usually lack of competences to design an integrated control system capable to coordinate multiple storage equipment.

Conversely, SAC based its decision process on a set of information including those relative to PV production and BESS status. This approach provided to the agent with a comprehensive view of the operation of the whole IES, enabling the identification of a better control policy compared to RBC. Moreover, SAC leveraged predictions of external disturbances to furtherly optimize the decision process.

SAC outperformed the RBC in managing the PV-BESS system, even though the control action does not directly act on the battery. High levels of SS and SC make the use of electricity storage technologies much more desirable from the point of view of the building flexibility. Moreover, SAC was capable to achieve appreciable levels of SS and SC with configurations implementing small sizes of the storage systems.

These results suggest that advanced control strategies are necessary elements to be integrated in a system where the number of connections and prosumers is drastically increasing.

Chapter 5

Robust comparison between model-based and model-free strategies for HVAC systems

This chapter discusses in detail the comparison between model-free deep reinforcement learning controllers and model-based model predictive control strategies. The comparison process presented in this chapter was defined as "robust" since it considers different boundary conditions and different configurations of control agents. The focus is on the identification of the relative strengths and weakness of the two approaches when applied to HVAC system control.

This content is developed as a work published in the Elsevier journal "Automation in Construction":

- Brandi S., Fiorentini M., Capozzoli A. 2022. *Comparison of online and offline deep reinforcement learning with model predictive control for thermal energy management*. Automation in Construction 135, 104128. [27]

5.1 Comparison of offline and online DRL with MPC for thermal energy management

MPC is a well-established model-based method for controlling complex interacting dynamical systems. Firstly developed in the process industry, it has recently been receiving wide attention from the building industry as it is capable of considering the physical behavior and dynamics of the controlled systems, its constraints, and a prediction of the future disturbances to minimize a cost function solved with different optimization methods [17]. Despite its low adoption in the building industry, MPC is still one of the most promising advanced control techniques for HVAC systems control given its mature stability, feasibility and robustness along with an inherent constraint handling capacity [174].

On the contrary DRL shows interesting adaptability properties together with a low deployment complexity. However, in ideal conditions, a model-free DRL agent should be directly employed in the controlled environment to gradually learn the optimal control policy. However, this process may take a considerable amount of time leading to poor control performance in its first implementation period. Moreover, besides the necessity of pre-training the control agent, DRL algorithms are characterized by a vast amount of hyper-parameters that require careful tuning in order to achieve good performance. As a consequence, despite being successful, DRL requires a considerable effort in developing the surrogate model of the controlled environment undermining the complete model-free nature of the framework.

The present chapter presents and discusses a comparison between an online and offline DRL formulation with a MPC architecture for energy management of a cold-water buffer tank linking an office building and a chiller subject to time-varying energy prices, with the objective of minimizing operating costs.

The next section presents the main research challenges analyzed in this application and introduces the motivations and novelty of proposed methodological approach.

5.1.1 Motivations and novelty of the proposed approach

This application present a robust comparison between an MPC and a DRL strategy applied to building energy management, benchmarking them against baseline classical rule-based controllers. Moreover, besides the approach typically followed in the literature in which DRL agents are commonly pre-trained offline on surrogate models of the environment, this application present an online implementation of a DRL controller which maintains the model-free nature of the algorithm.

As introduced in the literature review in Chapter 2, DRL algorithms are commonly compared to rule-based control strategies and only a few studies implemented model-based benchmarks. Raman et al. [175] implemented a DRL control strategy pre-trained offline without developing an online counterpart. A similar approach was adopted by Biagioni et al. [143] for price responsive water heaters. Outside building energy management applications, Ceusters et al. [144] compared MPC and DRL in dynamically simulated multi-energy systems where the complexity of the proposed case studies made it difficult to discern the differences between the two controllers.

For this reason, in the present application, the comparison between MPC and DRL is proposed for a simple case study in order to better analyze the performance of the implemented control strategies. The controllers have to manage the charging and discharging operations of a cold-water storage tank within an HVAC system of an office building while minimizing the cost associated with the operation of an electric chiller. The comparison was performed in a simulation environment described in Chapter 3.

The rest of the section is organized as follows. Section 5.1.2 introduces the proposed case study, Section 5.1.3 describes the control methodology proposed in this study, Section 5.1.4 reports the implementation details of the different control strategies. Section 5.1.5 presents the results obtained. Section 5.1.6 includes the discussion of the present application.

5.1.2 Case Study and Control Problem

The case study selected for the performance benchmarking of the different control approaches presented in this application consists of a cold thermal storage tank that acts as a buffer between the demand of an office building and the generation of cold

water via an air-to-water chiller. The system can operate in two modes, i) charging mode, where the cold water can be fed to the building and to the storage tank at the same time and ii) discharging mode, where the demand of the building is met only through the storage.

The controller in the charging and discharging phases can modulate the amount of heat transfer to the storage tank by adjusting the fraction of the nominal flow rate to/from the storage as shown in Figure 5.1.

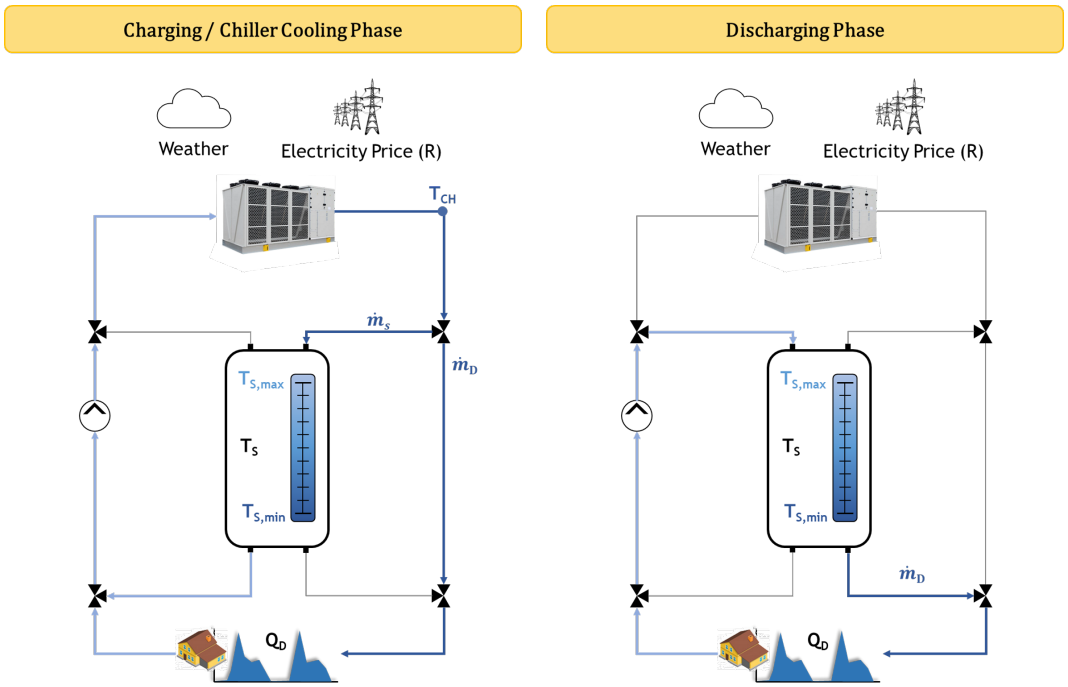


Fig. 5.1 Schematics of the cooling system analyzed [27].

In charging mode (left-hand side of Figure 5.1), if the building cooling demand (Q_d) is not zero, the chiller provides cooling to the building and the capacity of the chiller limits the amount of energy transferable to the storage. The supply water temperature (T_{ch}) is considered to be constant. The storage can be operated within a defined temperature range (between $T_{s,min}$ and $T_{s,max}$), however, these boundaries are not as considered hard constraints and may be slightly exceeded in particular situations.

In discharging mode (right-hand side of Figure 5.1) the chiller is by-passed and the building is cooled only via the cold thermal storage, with a constant design water mass flow rate m_D.

The control problem aims at optimizing the total electricity cost (R) of the energy used by the chiller by managing i) the scheduling of the two operating modes and ii) the charging/discharging power at each time-step. For simplicity, the building's thermostatic control is not considered and its cooling demand is considered as an external disturbance along with the price of the electricity and the temperature of the zone in which the storage is located.

5.1.3 Methodology

This section introduces the methodological framework of this study. As shown in Figure 5.2 the case study introduced in section 5.1.2 was used as a test-bed to benchmark the performance of three different control strategies, i) an MPC controller, ii) a DRL with offline training and iii) a DRL with online training, against two classical RBC controllers.

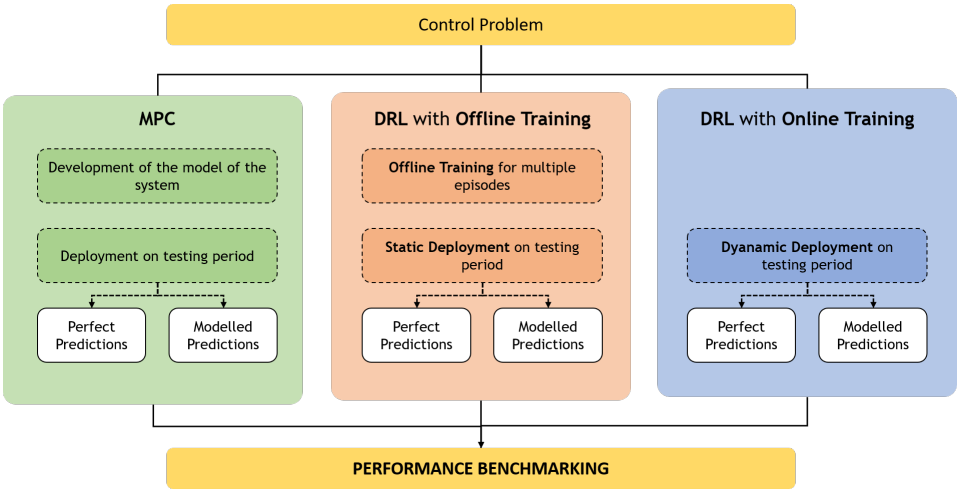


Fig. 5.2 Methodological framework of the proposed study [27].

MPC formulation

As the system is allowed to either discharge the storage to provide the cooling required by the building, or to supply cooling via the chiller to the building and/or to the storage, this results in two distinct operating modes leading to a Mixed Integer Linear programming (MILP) problem to be solved.

Model Defining a Boolean variable δ that is true when the system is in discharge mode, the storage dynamical behavior is described in Eq. 5.1:

$$\begin{cases} \frac{\Delta T_s(k)}{\Delta t_s} = \frac{Q_d(k) - UA(T_s(k) - T_a(k))}{C_s} if \delta = 1 \\ \\ \frac{\Delta T_s(k)}{\Delta t_s} = \frac{-Q_{ch}(k) - UA(T_s(k) - T_a(k))}{C_s} if \delta = 0 \end{cases} \tag{5.1}$$

Where Q_d and T_a, the building demand and the ambient temperature where the storage is located are measured disturbances; T_s , the storage temperature, is the system state; Q_{ch}, the charging power to the storage, and δ, the selection of the operating mode, are the controlled inputs; UA and C_s are the storage UA value and capacitance respectively.

Constraints At each time-step the controlled input Q_{ch} is limited by the possibility of the chiller to charge the storage. The chiller capacity, Q_{cap}, constrains the maximum thermal energy delivered to the building and the storage, as in Eq. 5.2.

$$Q_{ch}(k) + Q_d(k) \leq Q_{cap} \tag{5.2}$$

The storage temperature, since the supply temperature and maximum flow rate to the storage are fixed, limits the maximum heat transfer rate $Q_{ch,max}$, as in Eq. 5.3.

$$Q_{ch}(k) \leq Q_{ch,max}(k) \tag{5.3}$$

The maximum heat transfer rate $Q_{ch,max}$ is calculated as in Eq. 5.4.

$$Q_{ch,max}(k) = \dot{m} c_p (T_s(k) - T_{ch}) \tag{5.4}$$

The storage operation should be maintained within a reasonable temperature range. This constraint, also to help to ensure the feasibility of the optimization when the controller is deployed, was formulated as a soft constraint, as in Eq. 5.5.

$$T_{s,min}(k) - \varepsilon \leq T_s(k) \leq T_{s,max}(k) + \varepsilon \tag{5.5}$$

where $T_{s,max}$ and $T_{s,min}$ are the upper and lower temperature boundaries for the operation of the storage, and ε is the slacking variable. This latter variable is also subject to a constraint that limits the allowable temperature excess as in Eq. 5.6.

$$\varepsilon \leq T_s(k) \leq \varepsilon_{max} \tag{5.6}$$

Cost Function In this economic MPC formulation, the total cost to be minimized is the actual energy cost over the prediction horizon N, as in Eq. 5.7:

$$J = \sum_{k=1}^{N} R(k) \frac{Q_{ch}(k) + Q_d(k)}{COP} \tag{5.7}$$

Where R is the cost of the electrical energy per kWh, and the COP of the chiller is considered to be constant and equal to the equipment nominal value.

The problem is solved at each time-step k, with a control time-step Δt_s equal to 1h and the length of the horizon N equal to 48h.

The control actions of the MPC are the value of δ, which determines the operating mode, and the charging power Q_{ch}, from which the mass flow rate is determined (by dividing Q_{ch} by the known temperature difference between storage and charging flow, water density and specific heat capacity).

DRL formulation

In this application the classical implementation of SAC algorithm described in Chapter 2 was implemented. SAC methods are useful as they are capable of handling continuous action spaces. The Actor-Critic architecture employs two function approximators. The Actor has the aim to determine the optimal action for a given specific state of the controlled environment (policy-based), while the Critic evaluates the decisions made by the actor (value-based). This framework is generally coupled with an off-policy implementation, enabling the re-utilization of the previous experience collected by the agent in order to improve the control policy. The Soft-Actor-Critic algorithm originally implemented in the library Stable-Baselines [48] was employed in this application. In the following paragraphs, the design of

the action-space and of the reward function are presented along with the different training strategies employed.

Design of the action-space At each time-step, the action has to control the scheduling of the charging and discharging modes, as well as the fraction of design water mass flow rate (m_D) that should circulate through the storage. These aspects were encoded through an action space defined in the interval between -1 and 1 from which the DRL agent can select control actions. Following this approach, if the agent selects an action strictly less than 0 the system operates in discharging mode. Conversely, if the agent selects an action greater or equal than 0 the system operates in charging/chiller cooling mode and the water mass flow rate circulated to the storage is set proportional to the modulo control action value.

Safety constraints A safety constraint was introduced in order to guarantee that the cooling demand of the building is always met and to maintain the temperature of the storage within the prescribed range.

In particular, the constraint was introduced in the discharging mode (i.e. control action < 0). If the temperature of the storage tank rises above a certain value and the building cooling demand is not zero the system automatically switches to charging/chiller cooling mode in order to meet the demand regardless of the negative control action selected by the agent. This value was set equal to the upper-temperature boundary ($T_{s,max}$) plus a defined tolerance value τ. The violation of the upper-temperature boundary is penalized by associating a cost to the reward, as explained in the following subsection.

Reward function The reward function obtained by the agent after selecting an action at each time-step measures its control performance. Since the building's thermostatic control was not considered in this application, the objective of the controller is to minimize the cost of the electricity consumed by the chiller unit. The reward depends from this value as described in Eq. 5.8:

$$r(t) = -\beta * R(k) * Q_{ch,elec}(k) - P(k) \tag{5.8}$$

Where $R(k)$ is the electricity cost at each time-step k and $Q_{ch,elec}(k)$ is the chiller electricity demand in the time interval between $k-1$ and k. β is a weight factor introduced to regulate the magnitude of the reward. Moreover, the term $P(k)$ is a cost term introduced to penalize the agent of a quantity P if the temperature of the storage rises over the upper-temperature boundary ($T_{s,max}$) as introduced in the previous paragraph. In any other case (i.e. the temperature of the storage lower than upper-temperature boundary) the cost term $P(k)$ is equal to zero.

Design of the state-space The state or observation space includes all the variables which describe the environment at each time-step as it is seen by the DRL agent. In this study, the state-space does not include only information about the current time-step but also information about the recent past and the future disturbances.

Historical values were added to the state-space in order to account for the effect of thermal dynamics which characterize the present control problem.

All the variables included in the state-space are physical quantities directly extracted from the simulation output with the exception of the State of Charge (SOC) of the storage tank that was calculated according to Eq. 5.9 :

$$SOC(k) = 1 - \frac{(T_s(k) - T_{s,min})}{((T_{s,max} + \tau) - T_{s,min})} \tag{5.9}$$

More detailed information on the variables included within the state-space is provided in section 5.1.4.

DRL with offline training According to offline strategy, a DRL control agent was first trained using a calibrated model of the energy system over a training period and successively statically deployed on the same model over the deployment period. The training period, also identified as training episodes, was repeated multiple times, allowing the agent to explore different control policies in order to identify the optimal control strategy. Once the training phase was completed, the agent was statically deployed meaning that the parameters of the control policy were not updated during the process. The advantages of such an approach are the limited computational cost and the relative stability provided by a static control policy. The disadvantage is that the agent is unable to automatically adapt in the case key features of the controlled system change (e.g. revamping intervention) and may need to be retrained.

DRL with online training According to the online strategy, a DRL control agent was directly deployed on the calibrated model of the energy system. The control agent has no prior knowledge of the dynamics of the controlled environment. Thus, it was forced to learn the parameters of the optimal control policy while actively controlling the system. This strategy completely emulates a model-free controller in the sense that no previously built model was employed to pre-train the control agent.

A particular configuration of two hyper-parameters, *learning rate* and *number of gradient steps*, was adopted to directly deploy the DRL agent on the controlled system. In the offline approach was implemented a constant value of learning rate and number of gradient steps since the agent has at disposal a large amount of experience in the replay buffer, generated through multiple episodes. In the online approach the values of learning rate and the number of gradient steps vary over time according to two step functions. In particular, high values of the learning rate and number gradient steps were employed during the first period to encourage faster learning of neural networks weights. This approach is motivated by the fact that at the beginning of the deployment period the online agent has no prior knowledge of the problem and limited experience is available in the memory buffer. The learning rate and the number of gradient steps were then gradually reduced as long as learning progress to limit the risk of converging to near-optimal control policies. The step functions adopted for both the learning rate and number of the gradient steps during the simulation of online trained DRL agent are reported in section 5.1.4.

Modelled predictions of forcing variables

Since in this application both MPC and DRL implements predictions of the forcing variables to identify the optimal control policy, the performance of the three strategies was evaluated employing both perfect and modeled predictions of external disturbances.

The implementation of perfect predictions represents an ideal test scenario in which the performance of the controllers is benchmarked knowing exactly the evolution of the disturbances. The implementation of modeled predictions represents a test scenario closer to reality in which the evolution of the disturbances cannot be exactly known but can be estimated using data-driven methods. Through this approach was possible to evaluate the effect of the accuracy of the predictions on the different control strategies.

The modelled prediction was obtained by developing an LSTM network model for each disturbance. In particular, the disturbances predicted through this approach were the building cooling demand (Q_d) and the air temperature of the space in which the storage is located (T_a). The prediction of the price of electricity was always supposed to be perfectly known.

The prediction models were developed on an hourly basis with a prediction horizon of 48 hours. The inputs sequences to the two LSTM models are formed by the following common variables: day of the week, hour of the day, outdoor air temperature and outdoor solar radiation. Along with these variables, the sequences were completed with building cooling demand or air temperature of the space in which the storage is located depending on which was the target output. The sequences were provided to the LSTM models up to 48 hours in the past.

5.1.4 Implementation of the proposed methodology

The case study described in Figure 5.1 consists of an office module that is currently under construction at Politecnico di Torino, Italy. The module has an overall surface of 95 m^2 and consists of two 10-persons office rooms, one control room and a 3-persons technical room. The technical room is not served by the air-conditioning system and the storage tank is placed within it. The average transmittance value of the opaque and transparent envelope components are 0.15 and 0.6 W/m^2K respectively. The reference capacity of the chiller (Q_{cap}) is 12 kW and the reference COP is 2.67. The chiller can provide cooling energy to the building or to cold water storage which has a volume of 10 m^3. The storage was sized considering 1.5-times the maximum daily cooling demand of the building. The design water mass flow rate during charging phase is 0.2 kg/s while during discharging phase is 0.35 kg/s. This latter value corresponds to the sum of the design mass flow rates of the three air-conditioned zones. The supply water temperature at the outlet of the chiller was set equal to 7 °C. The operating range of temperature of the storage tank ranged between 10 °C ($T_{s,min}$) and 17 °C ($T_{s,max}$). The cooling demand was considered as an external disturbance of the system and was calculated within EnergyPlus in order to maintain an indoor temperature of 26 °C and a relative humidity of 55 % between 08:30 and 18:00 from Monday to Friday. In this time interval, the zones were supposed to be occupied at their maximum capacity.

The price of the electric energy drawn from the grid to operate the chiller unit is a further external disturbance of the analyzed system. A summary of the electricity prices used in this application is presented in Figure 5.3, which is based on the tariff structure commonly implemented in Italy.

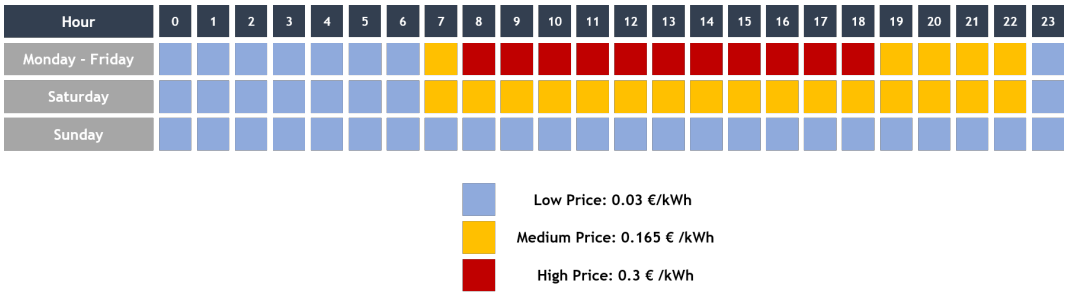


Fig. 5.3 Detail of the electricity prices used in the application [27].

Three different tariff levels were considered: i) a “High Price" level, with an electricity rate of 0.3 €/kWh, ii) a “Medium Price" level, with a rate of 0.165 €/kWh and iii) a “Low Price" level, with a rate of 0.03 €/kWh.

The price of the electricity was assumed to be relatively different from each other, to discriminate the values for the optimization application. Specifically, the low and medium price values were chosen to be respectively 1/10 and 1/2 of the higher one. The system was simulated using EnergyPlus. DRL control agents were designed and implemented in Python, the MPC controller in Matlab [44]. The weather file used in this application is the reference weather file (ITA_TORINOCASELLE_IGDG.epw) available in EnergyPlus for Torino, Italy. The cooling season was defined to last between June and August. The control time-step was set equal to 1 hour whereas the simulation time-step was set equal to 5 minutes in order to improve the precision of the simulation. As a result, a control action is defined every 12 simulation steps for which the same control action is repeated.

Implementation of RBC strategies

Two different RBC controllers were implemented in this application.

The first one, named *RBC 1*, was designed in order to charge the storage during mornings (i.e. between 0 a.m and 7 a.m) of working days when the price of electricity is low and if the temperature of the storage (T_s) is greater than 12 °C. In this mode, the

controller charges the storage at the maximum flow rate. The storage is discharged during peak-cost hours until the storage temperature reaches 17 °C or until the building cooling demand (Q_d) is null.

The second one, identified as *RBC 2*, charges the storage whenever the price of electricity is low (i.e. between 11 p.m and 7 a.m during Mondays and Saturdays and between 0 a.m and 24 p.m during Sundays) and the temperature of the storage is greater than 12 °C. In this phase, the storage is charged until its temperature reaches the lower limit of the temperature range or the price of electricity rises. The storage is switched to discharging mode whenever the building demand is not zero until this value return to zero or the temperature of the storage is greater than 17 °C.

RBC 1 was initially conceived as the only benchmark solution. However, as showed in the next sections, despite it can be considered a reasonable control law, its performance resulted extremely poor leading to the design of RBC 2. Nevertheless, RBC 1 was still considered as a baseline together with RBC 2, to show how an expert-based design of the control strategy may require different trials before converging to the optimal setup.

Implementation of MPC strategy

The MPC strategy was implemented using Matlab R2019b, the Multi Parametric Toolbox and Hysdel for the problem formulation [176], and Gurobi [177] as a solver for the MILP problem. As Hysdel was used to describe the Mixed Logical Dynamical (MLD) system, the future measured disturbances were taken into account by augmenting the prediction model with an additional linear model and treating the vector of the future references and measured disturbances as additional states [178]. For this reason in the implementation phase, the three tariffs described in the previous section were converted into discrete variables to enable switching to linear systems with a fixed energy cost equal to the rate of that time window. The parameters used for the implementation of the MPC strategy, including the system design variables (the storage UA and C_s, and the chiller COP) and MPC formulation (the control time-step Δt_s, the horizon length N, and weight and maximum value of the slacking variable ε) are reported in Table 5.1.

Table 5.1 Parameters used in the MPC controller [27].

Parameter	Value	Units
ε	0.005	€/K
ε_{max}	3	K
Δt_s	1	h
N	48	-
COP	2.67	-
C_s	11.62	kWh/K
UA	0.012	kW/K

Implementation of Deep Reinforcement Learning Control strategies

The reinforcement learning control problem is defined by action-space, by the reward function and the state-space that are summarized in Figure 5.4.

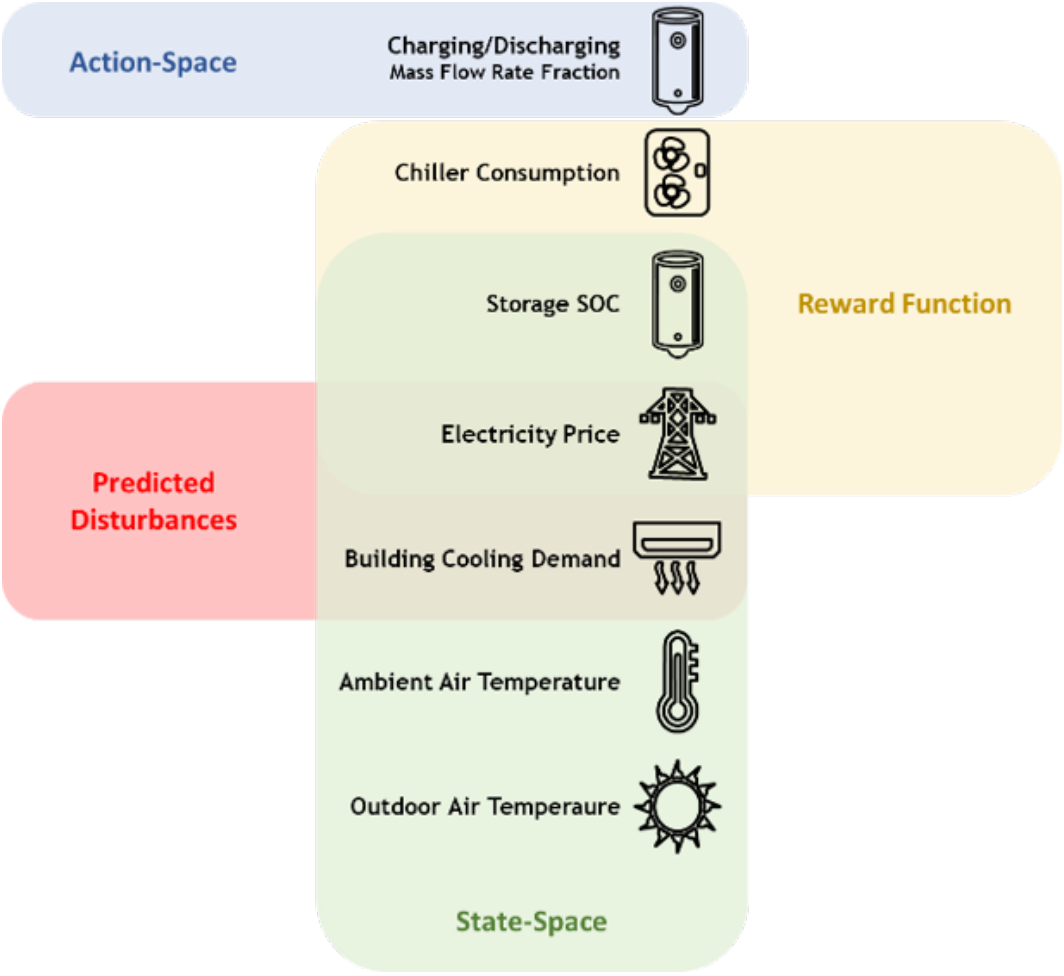


Fig. 5.4 Summary of the variables included in the state-space, action-space and employed to evaluate the reward.

The figure shows the variables included in the state-space highlighting the disturbance for which a prediction was provided (i.e. electricity price and building cooling demand). Moreover, Figure 5.4 summarizes the control action (i.e. charging/discharging mode and water mass flow rate fraction circulated from/to the storage) and the variables employed to evaluate the reward. These features are the same for both DRL trained through offline approach and DRL trained through online approach.

Table 5.2 Hyper-parameters of the reward function [27].

Variable	Value
Weight factor (β)	100
Penalty cost (P)	1
Storage temperature tolerance (τ)	1 °C

Table 5.2 shows the hyper-parameters of the reward function introduced in section 5.1.3. These values were found to be the most effective after conducting a hyper-parameter tuning process as implemented in [24]. Table 5.3 furtherly describes the variables included in the state-space along with their maximum and minimum values that were employed to re-scale the state space through a min-max normalization before providing the variables as inputs to the DNNs.

Table 5.3 Variables included in the state-space [27].

Variable	Min Value	Max Value	Unit	Time-step
State of charge (SOC)	0	1	-	k-4,..., k-1, k
Electricity Price (R)	0.03	0.3	€/kWh	k, k+1,..., k+24
Building Cooling Demand (Q_d)	0	20	kW	k, k+1,..., k+24
Ambient Air Temperature (T_a)	13	30	°C	k
Outdoor Air Temperature (T_o)	7.5	40	°C	k

The storage tank State Of Charge (SOC) at the time-step k was introduced to provide to the agent information about the amount of energy actually stored. Moreover, past values of this variable were introduced to provide information about the evolution of the temperature caused by the charging/discharging of the system up to 4 hours before the actual control time-step. These values were included to provide to the agent information about the inertia of the system evaluated at the time-step k.

The electricity price is a key-information for the agent in order to correctly plan the operations of the system. Actual value is provided along with the exact values for the 24 hours ahead. As introduced in section 5.1.3, the electricity price patterns were supposed to be always known.

The building cooling demand together with the price of electricity is a fundamental information to optimally manage the controlled system. Also the values of building cooling demand from time-step k to time-step $k+24$ were provided to the agent. The predictions of building cooling demand were assumed to be perfectly known or estimated by means of a neural network model.

Eventually, information about the air temperature of the space in which the storage is located along with information about outdoor air temperature were included. The first variable provides knowledge about the heat losses from the storage while the latter affects the COP of the chiller unit.

Besides the formulation of the reward function and of the state-space, the reinforcement learning frameworks require of a series of hyper-parameters to be set as the discount factor for future rewards (γ) and the structure of the neural networks employed as function approximators. The values of the hyper-parameters selected for this application that are the same for the two control strategies based on DRL are summarized in Table 5.4.

Table 5.4 Hyper-parameters of the DRL Agents [27].

Hyper-parameter	Value
Discount factor (γ)	0.99
Number of hidden layers	2
Number of Neurons per Hidden Layer	256
Activation Function	ReLu
Optimizer	Adam
Entropy regularization coefficient (α)	0.2
Memory size	2160

In the next following subsections the different implementations of DRL agent trained in offline and online modes are described.

Implementation details of the DRL agent with offline training

In this case the DRL control agent was pre-trained offline using the model of the system as introduced in section 5.1.3. The pre-training unfolds by repeating the same training episode in order to let the agent converge to the optimal control policy. The

selected training episode includes the month of June of the weather file implemented in this application. The training episode was repeated 15 times before obtaining an acceptable solution. The trained agent was successively deployed statically on the system for the whole cooling season (June-August) in order to assess its performance. The deployment process was performed considering also the training period to assess the stability of the learned control policy. The optimizer learning rate was set equal to 0.001 while the batch size equal to 256.

The agent was trained using only the perfect predictions of the external disturbances while the deployment was performed considering both perfect and modeled predictions. Through this approach, the stability of the control policy of the DRL agent pre-trained offline was challenged during deployment since it had no information about prediction uncertainties during training.

Implementation details of the DRL agent with online training

This DRL agent was directly implemented on the simulated case study as if it was the real system. The behavior of two hyper-parameters, the learning rate of the DNN optimizer and the number of gradient steps, was defined according to the function depicted in Figure 5.5. The batch size was set equal to 32 differently from the DRL agent trained offline due to the lower amount of data available to the agent to train the control policy in the online training fashion. Employing smaller batch size can provide a faster convergence to near-optimal solution [40] which is an extremely desirable feature for a DRL agent directly deployed on the controlled system.

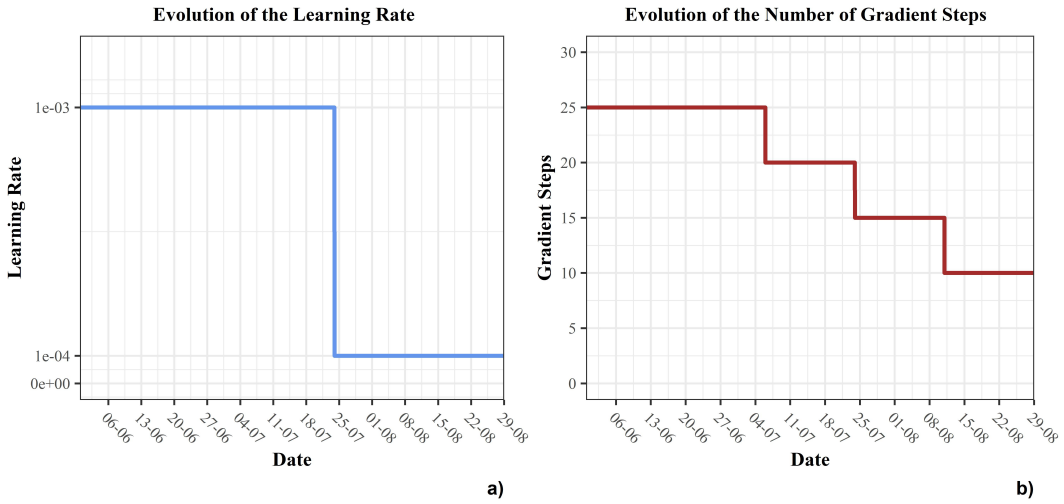


Fig. 5.5 a) Evolution of the learning rate and of b) Number of the gradient steps during the simulation of Online trained DRL agent [27].

The figure shows how the two hyper-parameters, learning rate figure (a) and number of gradient steps (b), were reduced during the simulation. This approach was found to be effective in speeding up the training process in the first weeks of deployment given the limited amount of data available to learn from. The two functions shown in Figure 5.5 were found according to a manual tuning process.

Implementation of modeled predictions of disturbances

As introduced in section 5.1.3 modeled predictions of building cooling demand and air temperature of the zone in which the storage is located were obtained by means of two LSTM models. These models were characterized by two hidden layers with 48 neurons, one LSTM layer and one dense layer with *relu* activation function. The predictor attributes employed are described in section 5.1.3. The models were trained for 300 epochs with a batch size equal to 64.

The training data-set was generated with the same simulation model considering a different weather file obtained from real world measures collected for Torino (Italy) during the year 2019. The trained LSTM models were used to predict the values of the building cooling demand and air temperature of the zone in which the storage is located considering the same weather file employed to analyze the different control strategies (i.e. reference weather file ITA_TORINO-CASELLE_IGDG.epw). The performance of the two LSTM models were evaluated in terms of Root Mean

Squared Error (RMSE). The models developed to predict building cooling load and the air temperature of the zone achieved in testing conditions an RMSE of 229.6 W and 0.84 °C respectively.

5.1.5 Results obtained

As introduced in Section 5.1.4 the main objective of this application is to compare the performance of model predictive and deep reinforcement learning control strategies. Considering that both control techniques benefit from predictions of future disturbances values for the evaluation of the optimal control sequence, a comparison of their performance considering both perfect and modeled predictions was undertaken. The two rule-based control strategies described in section 5.1.4 were used as a benchmark. The different control strategies were simulated during the cooling season in the period ranging from June to August. Table 5.5 reports the total cost and consumption of electricity obtained by implementing the different control strategies.

Table 5.5 Total operating cost and electricity consumption comparison of the system using the different control strategies. The *Pred* column refers to the type of predictions of external disturbances used (perfect *P* or *M* modeled predictions) [27].

Strategy	Pred	June		July		August	
		€	kWh	€	kWh	€	kWh
RBC 1	-	13.1	272	21.9	344	18.9	310
RBC 2	-	8.40	280	10.8	361	9.78	326
MPC	P	8.16	272	10.26	343	9.24	309
DRL Offline	P	8.28	277	10.5	351	9.6	320
DRL Online	P	21.24	269	13.1	357	10.14	326
MPC	M	8.16	272	10.26	342	9.3	310
DRL Offline	M	8.34	278	10.5	351	9.6	319
DRL Online	M	24.3	270	12.78	365	10.62	314

The results in Table 5.5 show that the MPC achieves the best performance in terms of total cost in both the implementations with perfect and modeled predictions, followed by the DRL strategy with offline pre-training. It can be noticed that RBC 1 obtained the worst performance among the employed control strategies, which was the main reason for the development of the RBC 2 controller. This latter RBC

strategy, better suited for the system, performs only 4.7% worse in terms of operating cost compared to the MPC. However, it led to the highest amount of electrical energy over the simulation period, 4.6% higher than the best performing MPC.

It is also interesting to notice that the DRL trained offline and the MPC obtained very similar results, showing that both solutions are most likely near-optimal.

The MPC and the DRL with offline training were only mildly affected by inaccurate disturbances predictions, whether the DRL controller trained online was more affected, with a reduction in performance compared to the case with perfect predictions of 7.2%.

The negative effect of modeled predictions was mitigated in the case of DRL with off-line training since this agent utilized more experience to converge to an optimal solution, relying less on the goodness of the prediction.

As expected, the DRL controller trained online did not perform well if the entire 3-months period is considered, since the agent was deployed without prior knowledge of the controlled system. However, it can be noticed how the total cost achieved through this controller gradually decreased over time. This can be seen from the monthly results in Table 5.5, and in Figure 5.6, which reports the cumulative cost (starting from July) of each strategy. Table 5.5 shows that the performance difference between the MPC approach and the online DRL in terms of total cost decreases from 160% in the first month, to 21% in the second, to only 3.6% during the last month.

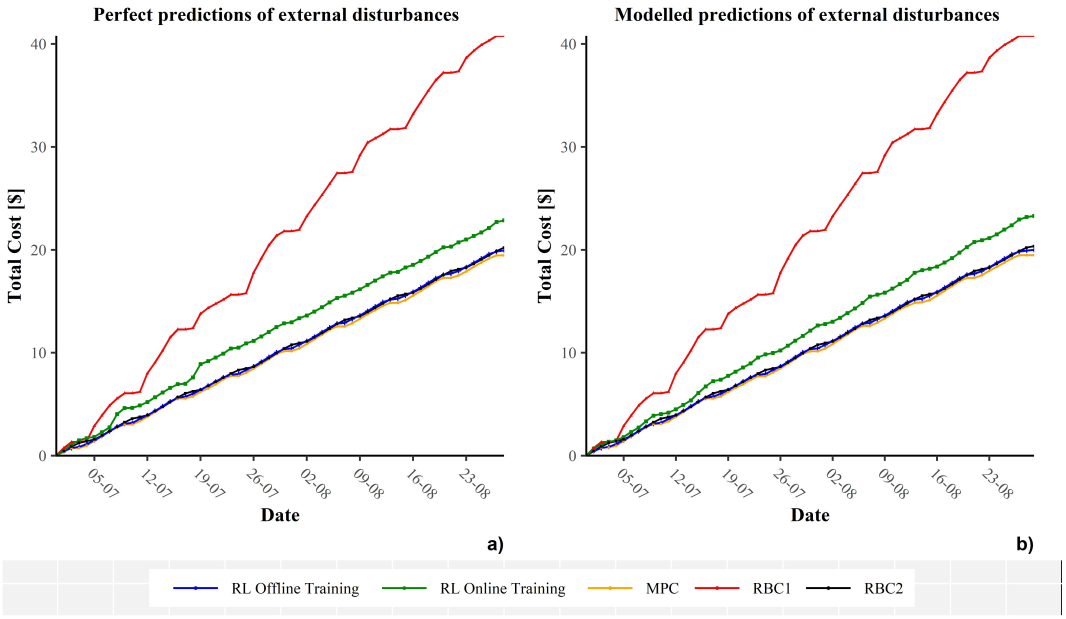


Fig. 5.6 Total electricity cost obtained by the different control strategies in the period July-August in the case of a)perfect prediction and b)modelled predictions of external forcing variables [27].

This is also clear in Figure 5.6, which shows that the cumulative cost of trained online DRL (green line) becomes parallel to the curves of the best performing solutions, indicating that these control solutions became more and more similar over time.

The higher costs of the DRL with online training were expected as, differently from the one trained offline, needed to explore behaviors in the initial period that, without having a-priori knowledge of the system, could be detrimental. In addition, the agent trained online might not react to sudden changes (e.g. the increase in the cooling demand from June to July).

The importance of flexibility of energy sources can be furtherly highlighted by reporting the costs related to an identical system without the introduction of an active thermal storage which are 86.7 €, 101.2 € and 90.4 € during June, July and August respectively.

Table 5.6 reports the use of the storage tank in terms of thermal energy charged and discharged, as well as the fraction of cooling demand satisfied, achieved by the different control strategies over the three simulated months.

Table 5.6 Thermal energy exchanged by the storage tank during charging (Ch) and discharging (Disch) phases and the fraction of cooling demand satisfied (Dem) by the different control strategies [27].

Strategy	Pred	June			July			August		
		Ch	Disch	Dem	Ch	Disch	Dem	Ch	Disch	Dem
		kWh	kWh	%	kWh	kWh	%	kWh	kWh	%
RBC 1	-	845	886	95	1014	963	89	923	877	90
RBC 2	-	924	936	100	1189	1080	100	1077	976	100
MPC	P	897	936	100	1132	1080	100	1022	976	100
DRL Off.	P	906	936	100	1142	1080	100	1042	976	100
DRL On.	P	743	797	85	1154	1068	99	1066	976	100
MPC	M	896	936	100	1131	1080	100	1024	976	100
DRL Off.	M	906	936	100	1142	1080	100	1043	976	100
DRL On.	M	729	786	84	1185	1076	99	1025	976	100

The results show that the MPC, DRL trained offline and RBC 2 strategies used the storage to satisfy always the full building cooling demand. However, RBC 2 employed an higher amount of energy to charge the system resulting in an increased storage heat losses to the ambient. It can also be seen that RBC 1 never fully met the building cooling demand through the storage, resulting in a higher cost. RBC 1 was never capable to charge the storage enough to cover the demand during off-peak time slots reaching, during discharging phase, the upper limit of storage temperature range before the demand was completely satisfied and incurring in a higher energy cost. As far as the DRL trained online is concerned, a gradual improvement of the control policy can be seen, with an increase of the percentage of cooling demand satisfied through the storage during the second and third months.

Figure 5.7 shows the temperature profile of the storage tank achieved by the different control strategies. For simplicity, only the MPC and DRL results with perfect predictions were reported.

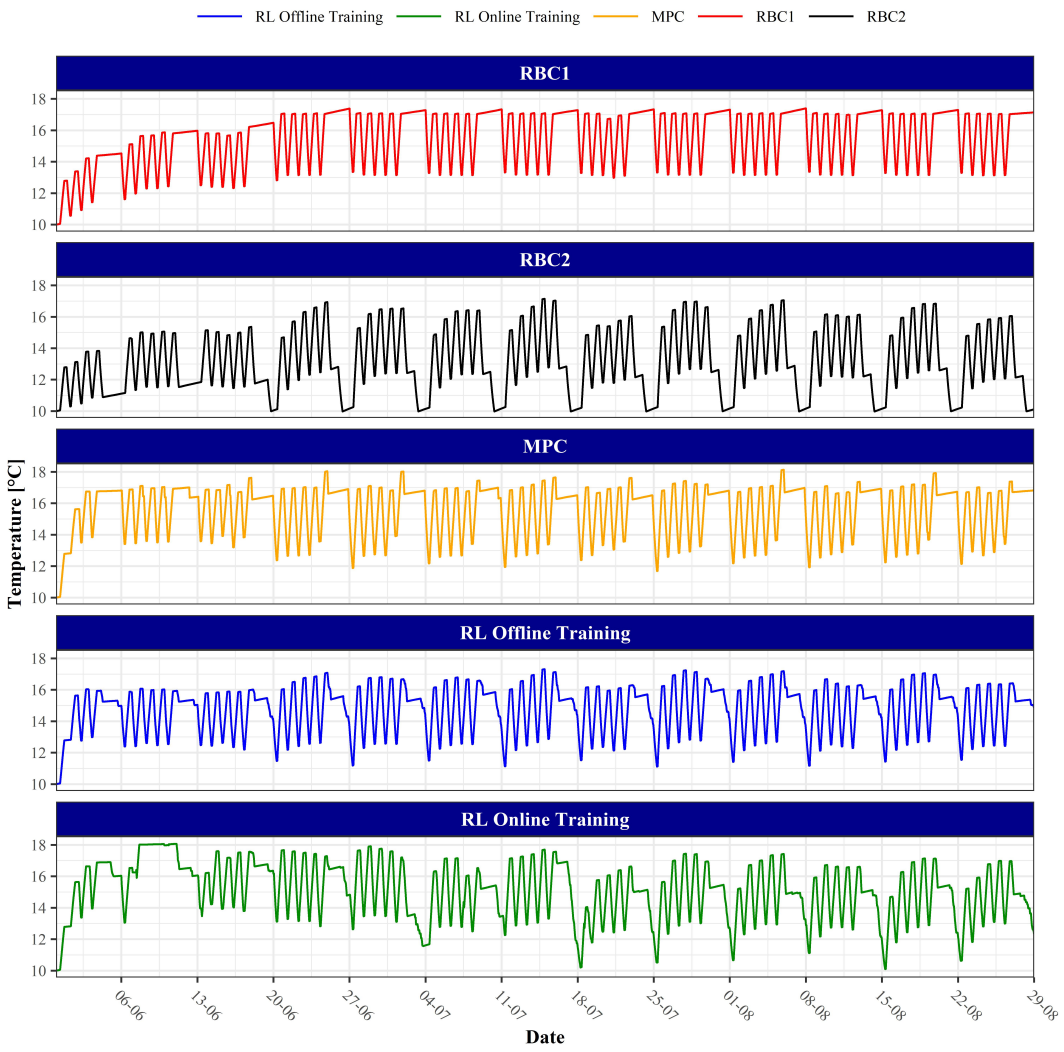


Fig. 5.7 Comparison of storage tank temperature profiles for the different control strategies (perfect predictions), June-August [27].

In this figure it can be observed that RBC 1 charged the storage tank only during morning of working days, which is insufficient to cover the building demand. As a results the storage temperature gradually increases during the first weeks to later remain stable between 13 °C and 17 °C.

RBC 2 control strategy charged the storage whenever the price of electricity was low and whenever the temperature of the storage was higher than 12 °C. Consequently, during weekends the storage was cooled to the lower temperature limit (i.e. 10 °C), resulting in a better cost performances compared to *RBC 1*, but in a higher energy losses to the ambient.

The MPC controller, which achieved the best performance, tried to maintain the temperature of the storage as close as possible to the upper limit, to provide sufficient cooling energy to satisfy the daily demand while minimizing the thermal losses. During some Fridays the MPC controller violated the soft constraint on the upper limit of storage temperature range of approximately 0.5 °C, as enabled by the constraint formulation in Eq. 5.5. This could be due to a compensation of the mismatch between the model and the system closing the loop, or the willingness of the controller to incur the cost of violating the upper temperature constraint to access charging energy at a lower price a few hours later.

The offline DRL controller achieved a very similar performance as the MPC, but with a slightly different sequence of inputs, as it charged the storage more during weekends. This additional energy was gradually released during the week allowing the controller to violate the upper constraint of storage tank temperature for shorter periods of times compared to the implemented MPC approach.

The online DRL controller clearly struggles during the first weeks to manage the storage, charging it only intermittently resulting in a higher cost due to a more extensive use of the chiller during high-price periods. However, it can be observed how the control policy gradually improves to finally converge to a control pattern very similar to the behavior described for the DRL controller pre-trained offline.

Figure 5.8, Figure 5.9 and Figure 5.10 provide more details on the behavior of the MPC and DRL control strategies during a week included between 21/08 and 2/08. The patterns observed during this week were not necessarily observed in all other weeks, however it were deemed sufficiently representative to be presented and discussed.

The top plot in these figures shows the amount of heat transfer to the storage tank, where a negative value means charging and a positive one discharging of the storage. The background color shows the electricity price, where a green background corresponds to a low price time period, a yellow one to a medium price and a red one to a high price period.

The second subplot shows the temperature profile of the storage tank together with the upper temperature limit, marked with a red line at 17 °C.

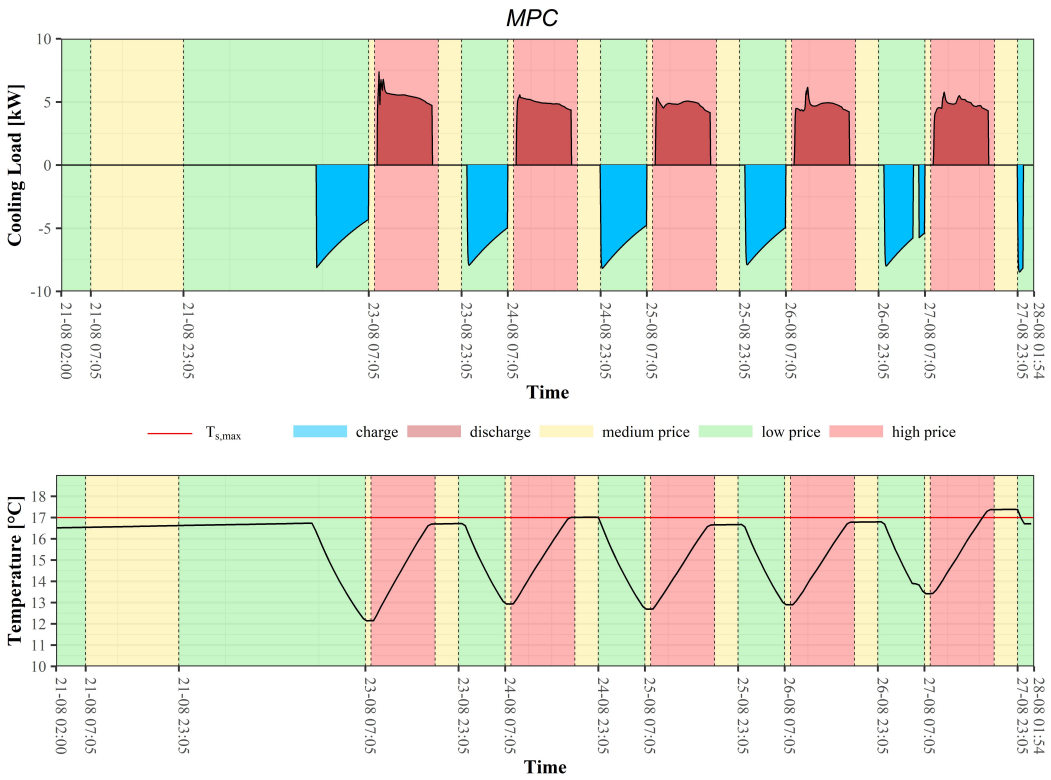


Fig. 5.8 Storage cooling load and temperature patterns for MPC control strategy in the period 21/08 - 27/08 [27].

Figure 5.8 is related to the operation of the MPC strategy. As expected, the controller charged the storage only during low-price time slots in order to match the cooling demand of the building. At the same time, the controller managed to maintain the temperature of the tank as close as possible to the upper temperature limit. During the last day of the week, (Friday 27/07 in this figure), the soft constraint on this limit was relaxed by the controller to satisfy immediately the cooling demand and return below the temperature boundary after a few hours, when the energy price lowered again.

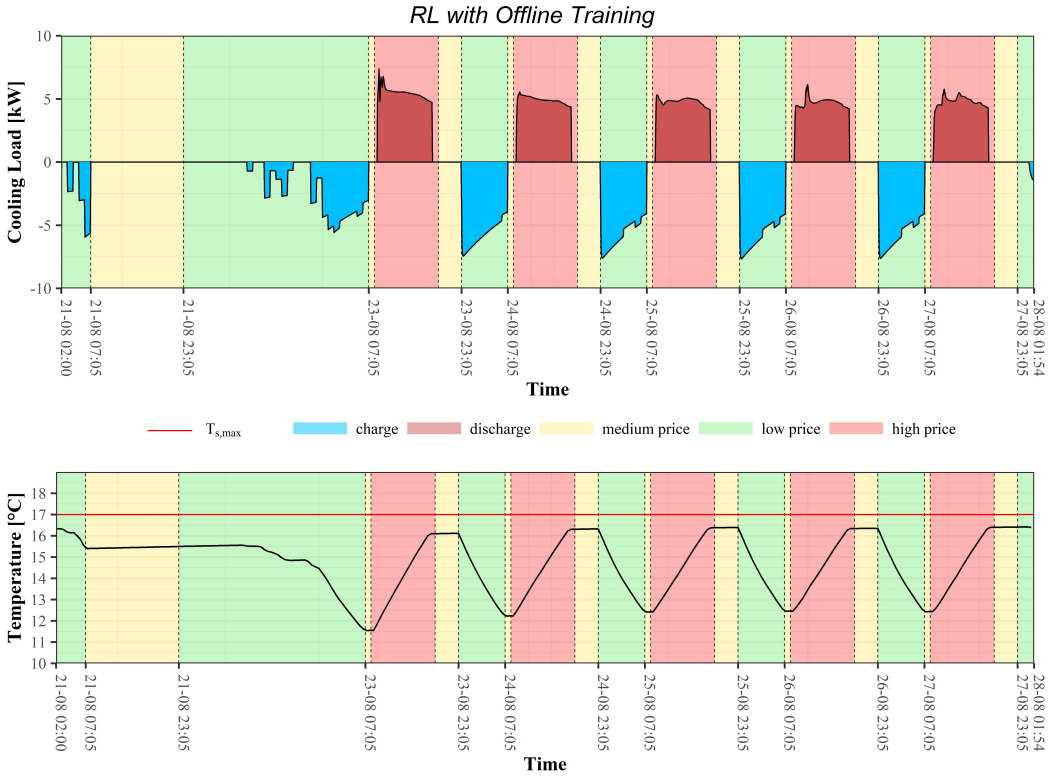


Fig. 5.9 Storage cooling load and temperature patterns for DRL with Offline Training control strategy in the period 21/08 - 27/08 [27].

Figure 5.9 shows the behavior of the DRL controller pre-trained offline. Differently from the MPC, it is interesting to observe that this controller decided to pre-charge more the tank over the weekend in preparation for the coming week.

In this region the control policy was relatively noisy, alternating between charging and free-floating. However, the storage tank was cooled at a lower temperature at the beginning of the week respect to the MPC strategy, and the upper temperature constraint was never violated during this week.

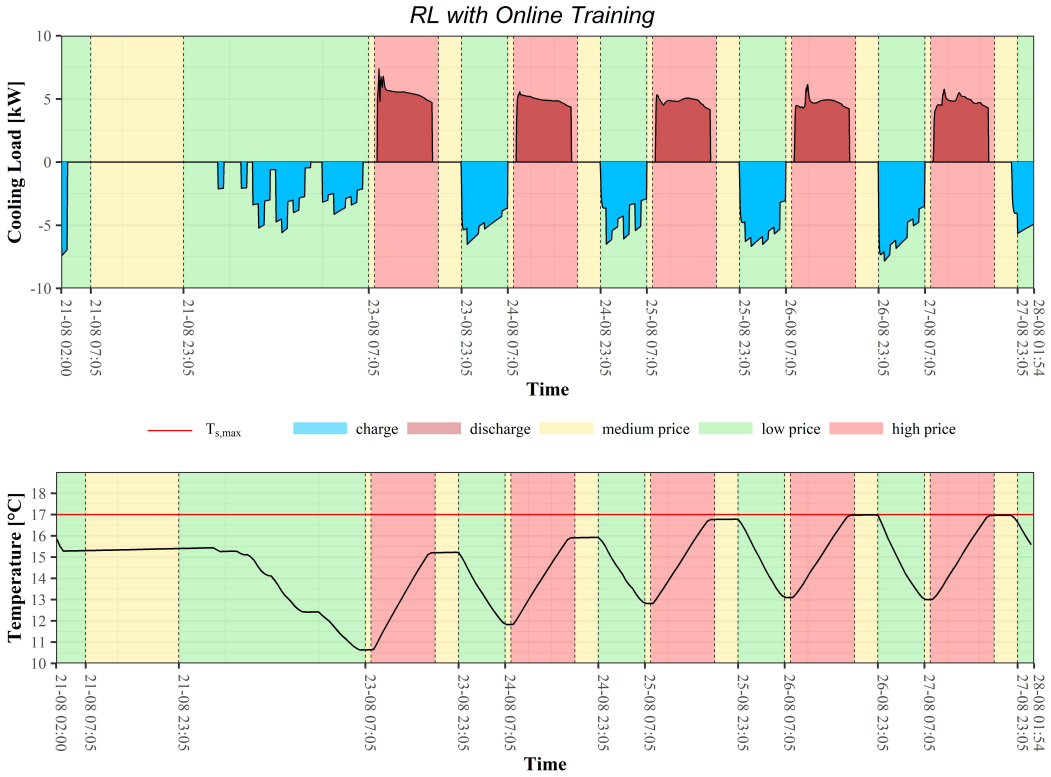


Fig. 5.10 Storage cooling load and temperature patterns for DRL with Online Training control strategy in the period 21/08 - 27/08 [27].

Figure 5.10 shows the control policy of the DRL controller trained online. This controller shows similar behavior to DRL agent pre-trained offline after six weeks of training. During Sunday-Monday morning the control policy was uncertain resulting in a more pronounced alternation between charging and free-floating. It is interesting to observe from the bottom plot how this controller gradually released the cooling energy stored in the tank during the week.

When comparing the results of the three control approaches it can be seen that, while they all charge during off-peak time slots between the working days, during the weekend they follow different policies. The MPC controller waits as long as possible before charging the storage on Monday morning, as it can be seen in Figure 5.8. As Figures 5.9 and 5.10 show, the two DRL controllers, charge the storage more and over a longer period of time, showing a more intermittent behavior. This pattern suggests that the control policies learned by the DRL agent are still uncertain and might be slightly less efficient.

However, the DRL control policies were able to capture requirements of the system beyond the 24h prediction horizon. More specifically, DRL agents were capable to extract a recurrent pattern (i.e. the alternation between weekend and weekdays) which spawns for a longer period compared to the horizon of the predictions included in the state space.

5.1.6 Discussion

The results from the comparison of a MPC, a DRL and a RBC applied to a case study featuring a cold water storage tank and an air-to-water chiller showed interesting similarities and differences between these approaches.

The different control strategies, to minimize the operational cost, had to correctly manage the storage to shift the demand to the lower energy cost time slots, ensure that the storage was sufficiently charged to satisfy the demand each day of the week, and minimize the thermal losses to the ambient.

The MPC approach achieved the best performance, managing the system in order to charge the storage tank with enough cooling energy during off-peak price periods in proximity of on-peak time slots. Thanks to this approach the controller was able to minimize thermal losses to the ambient and guarantee the building cooling demand satisfaction with the storage. In particular circumstances (e.g. at the end of each week) the MPC controller decided to soften the upper temperature boundary constraint, possibly due to the limited length of the prediction horizon (48h) that did not allow the controller to use the weekend to further cool in advance the storage.

The DRL agent pre-trained offline achieved a similar performance to the MPC approach, with a similar control pattern during the weekdays, but during weekends and early Monday mornings the DRL controller attempted to provide more cooling to the storage, in order to gradually release it during the week. This allowed the controller to violate less the upper limit on storage tank temperature. This difference can be explained by the fact that the DRL controller based its decisions on patterns such as alternation between weekend and weekdays that went beyond the 24h prediction horizon that it had available in the state-space. A similar behavior was observed for some weeks also for the DRL agent trained online. This pattern represents an advantage with respect to MPC provided by the off-policy evaluation method employed by DRL algorithms. This method leverages previous experience

to learn an effective mapping between states and actions capable to identify complex patterns thanks to DNN capabilities. However, DRL controllers charged the storage intermittently and over a longer period of time, alternating it with no-charging periods. As a result, the heat losses to the ambient were greater compared to the MPC case, possibly showing intrinsic instabilities affecting a DRL control approach.

RBC 2 achieved a similar performance in terms of cost of electricity compared to the MPC ad DRL trained offline approaches. However, it used on average 4.5 % more energy with respect to MPC due to heat losses since it was designed to charge the storage to its minimum temperature whenever the price of the electricity was low. On the other hand, RBC 1 failed at providing the storage with enough cooling energy and was forced to use the chiller to satisfy the remainder of the building cooling demand at higher electricity costs as expected.

The DRL agent trained online demonstrated to be capable to rapidly adapt to the controlled environment reaching, after one month, comparable results to the best performing solutions and outperforming the RBC 1 controller. On the other hand, during the first month, the poor performance of this agent led to high operating costs.

In this application both MPC and DRL made use of predictions of external disturbances (electricity price, the temperature of the ambient of the storage and cooling demand) to derive the optimal control policy. They were considered as either perfectly predicted or modelled through a deep neural network to make a future 48h forecast, to replicate an implementation in real-world conditions. The inputs to the disturbances models were outdoor air temperature and solar radiation for the unconditioned space, with the addition of the occupancy schedule for the cooling demand.

Alongside the predictions of the external disturbances, the MPC required a model of the controlled system and an optimizer. The definition of the model is usually addressed as one of the most time-consuming tasks in MPC development. In this application, given the relatively simple nature of the problem considered, a straightforward model of the controlled system was derived. In this application MPC demonstrated a good performance, with an implementable solution that worked well with both perfect and modelled predictions of the external disturbances.

A DRL agent pre-trained offline, despite not using a model directly in its formulation, still requires the development of a model of the controlled environment

for training, as it needs to see the same simulation period several times to learn an efficient control policy comparable to the one obtained by an MPC approach.

The DRL agent directly deployed on the controlled environment and trained online, despite an initial lower performance, was capable to converge to an acceptable though not optimal solution, demonstrating to be capable of improving the performance of the system controlled by a rule-based system without using a model of the system or supervision.

The RBC controllers developed showed the limitations of classical approaches. The first RBC controller, RBC 1, despite having a reasonable control law, resulted in an under-performing solution, which lead to the design of RBC 2 controller for benchmarking. This solution obtained satisfying performance in terms of electricity cost but led to the greatest amount of energy losses to the ambient.

The results obtained in this study highlight that, despite the simplicity of the control problem, an expert-based design of the control strategy may require time to identify the optimal control setup that is only applicable to a specific system. In this sense, rule-based controllers are not capable to adapt to the evolution of the controlled environment over time. For example modification to the patterns of building cooling demand or price of electricity may lead to a poor performance that advanced control strategies would not suffer, as they are more effective in adapting to known and changing boundary conditions.

Chapter 6

Conclusions

The present dissertation was aimed at demonstrating the applicability and effectiveness of DRL-based strategies for HVAC system control. DRL-based control strategies have the capability to automatically improve HVAC system operation considering multiple goals while adapting to evolving conditions. However, their application in the building sector is still in its infancy and it requires expertise in both artificial intelligence and building physics.

The framework in which the present dissertation was undertaken was carried out with the aim of bridging the gap between these two research fields.

To this purpose four different applications of the DRL framework for HVAC system control were conceived and tested leveraging a co-simulation environment specifically designed in the context of this dissertation. Figure 6.1 shows a summary of each application highlighting the different aspects being investigated. These aspects are related to:

- Environment models employed to train DRL control agents. Both physics-based and data-driven models were employed.
- The complexity of the controlled environment. Simple heating systems were investigated along with more complex configurations involving storage technologies and integrated energy systems.
- Control objectives. The DRL control strategies were designed to meet different goals including indoor temperature control, energy minimization and cost minimization.

- Formulations of the state-space. Input variables to DRL control agents were identified according to traditional approaches or following variable-engineering processes. Moreover, prediction of external disturbances were considered in some applications in order to provide more information to the controller.
- DRL training methods. DRL agents were trained following both offline pre-training and online training approaches.
- DRL deployment methods. DRL agents were deployed according to both static and dynamic deployment configurations.
- Benchmarking strategy employed to evaluate the performance of DRL agents. Traditional and model-based strategies were developed to provide robust benchmarks of the proposed DRL agents.

	Environment Models	Environment Complexity	Control Objectives	State-Space Formulation	Training Method	Deployment Method	Benchmark Strategy
Section 4.1 Optimization of indoor temperature control and energy consumption in heating systems	Physics-Based	Simple Heating System	Indoor Temperature Control & Energy Minimization	Traditional & Variable-engineering process	Offline	Static & Dynamic	Traditional
Section 4.2 Effective pre-training of DRL agents by means of data-driven models to control HVAC systems in buildings	Data-driven	Simple Heating System	Indoor Temperature Control & Energy Minimization	Variable-engineering process	Offline	Static	Traditional
Section 4.3 Optimization of the management of integrated energy systems in buildings with Deep Reinforcement Learning	Physics-Based	Integrated energy systems	Cost Minimization	Traditional & Forecast of Disturbances	Offline	Static	Traditional
Chapter 5 Comparison of DRL with MPC for thermal energy management	Physics-Based	Cooling system with thermal storage	Cost Minimization	Traditional & Forecast of Disturbances	Offline & Online	Static	Traditional & model-based

Fig. 6.1 Summary of the four different applications and relative aspects being investigated.

Each application was designed to address different challenges and questions related to the application of DRL controllers to building systems bringing the following innovative perspectives:

- **Optimization of indoor temperature control and energy consumption in heating systems:** Despite the simplicity of the case study, the flexibility and adaptability of the control agent to different occupancy schedules and indoor temperature requirements was tested in different scenarios showing the potentialities of the proposed solution. In this perspective, this process can be

categorized as an application of transfer learning in which a learned control policy was evaluated in different context. A proper selection of variables defining the state-space was proposed with the aim of developing a controller capable to adapt to dynamic changes of the environment. The importance of hyper-parameters selection was highlighted by analyzing the sensibility of the results for different configurations of their values. The DRL control agent with variables selected according to adaptive approach led to savings between 5% and 12% of heating energy depending by the analyzed scenario. This agent was able to achieve these performances in a static deployment configuration suggesting that a careful design of the state space may be sufficient in providing to an agent the capability to adapt to changes in the controlled environment without scarifying its stability with a dynamic deployment configuration. At the same time, the controller achieved satisfying performance in controlling indoor air temperature.

- **Effective pre-training of DRL agents by means of data-driven models to control HVAC systems in buildings:** Pre-training of DRL control agents can be effectively performed through data-driven models of the building dynamics. Through this approach a DRL controller can be effectively trained without performing the time-consuming and expensive task of developing physics-based models of the controlled environment. The risk of this approach is related to the fact that monitored data only carries information about specific usage patterns of the building limiting prediction capabilities of the data-driven model. This situation is particularly relevant during the training process in which the agent may explore a wider range of the state-action space never mapped from monitored data. The presented application showed how an LSTM network carefully tuned can be effectively exploited as data-driven model for DRL training. The result showed how the agent trained through the proposed framework can be capable to reduce energy consumption while maintaining indoor air temperature requirements for a water-based heating system of an office building.

- **Optimization of the management of integrated energy systems in buildings with Deep Reinforcement Learning:** The optimal management of storage technologies is a fundamental task to address in buildings with IES to enhance energy flexibility and reduce operational costs. In the developed application

the baseline strategy, conceived following a traditional approach, was not aware of local PV production or BESS status resulting in sub-optimal control policy especially when the capacities of TES and BESS were small. The proposed DRL strategy proved to be capable to learn better control policy compared to RBC given the same storage capacities reducing the operating cost between 39.5% and 84.3%. Traditional baseline controller resulted more sensitive to the storage size, giving greater importance to the initial design, whereas DRL achieved high savings also when smaller capacities were implemented. The advantage with respect to the baseline narrows down as the capacities were increased. For the same BESS capacity installed, DRL control strategy was capable to notably increase the levels of SS and SC, reducing the energy exchanged with the grid and increasing building energy flexibility. The results obtained highlighted the importance of implementing advanced control strategies in the design framework of IES in buildings. However, the proposed DRL control strategy despite its model-free definition is not completely independent by a modeling effort since it was trained for several episodes before converging to the final solution.

- **Comparison of DRL with MPC for thermal energy management:** MPC is a model-based solution that employs a simplified model of the controlled system to perform an optimization process over a receding horizon, using predictions of external disturbances. Similarly, DRL employs predictions of external disturbances to learn a near-optimal control policy. However, despite the model-free nature of the control algorithm, as this control approach requires a certain amount of time to converge to an acceptable solution, a common approach consists in pre-training the DRL agent offline with a simulated model of the controlled system, losing the intrinsic model-free nature of the algorithm. Conversely, a DRL controller directly deployed in the controlled environment learning the control policy online may achieve a sub-optimal performance in the first period of deployment, as shown in this study, but can converge to a near-optimal strategy in an acceptable amount of time (in the order of a few weeks as shown in the results). This approach, differently from the DRL with offline training, is model-free in the entire deployment process. These considerations open several research questions on the development of DRL algorithms. If DRL control strategies are implemented with offline training, they require a model of the system, removing this theoretical advantage in

comparison to an MPC approach. DRL has the advantage of not relying on a numerical optimization process which generally requires linearized models and and a convex problem to be formalized. This also leads to lower computational times compared to an MPC approach. On the other hand, MPC demonstrated to be a more robust and stable control approach. The flexibility shown by DRL agents is associated with the risk of temporary poor control performance. This is particularly evident when employing a DRL agent trained online, but this represents nevertheless a promising truly model-free approach. The DRL agent trained online presented in this study proved to be able to improve its control performance over time, approaching the behaviour of a near-optimal MPC strategy or the similar one of a DRL pre-trained offline. However, the possibility to really deploy such a controller in a plug-and-play fashion is still to be assessed, as the hyper-parameters and reward function, which play a key role in determining the performance of this category of controller, can require different setting depending on the system on which they are implemented.

Regardless of the specific goals of each application the main objective at the basis of the methodological development was the implementation of model-free strategies for HVAC system control following an energy engineer perspective. For this reason, within this dissertation, more emphasis has been given on how to effectively implement these methodologies considering each step, from the definition of the variables involved to the different deployment strategies, rather than the algorithmic complexity of the different approaches.

In that perspective, the developed applications significantly contributed to achieve this demanding target in the robust way as possible. Most of the findings and outcomes of the present research work were already discussed in detail in the previous chapters. Therefore, the aim of this final chapter is to provide a comprehensive overview of the lessons learned in the framework of this research work.

Advanced control strategies: an opportunity or a necessity? Through the applications developed in the context of this dissertation emerged how advanced control strategies can bring significant benefits to the management of HVAC systems in buildings. The greater the complexity of the system under consideration, the greater the advantage these techniques can provide over traditional control strategies. However, the convenience of adopting such strategies, cannot be measured only in terms

of performance during operation but also in terms of development and implementation costs. Certainly, the implementation of control strategies based on artificial intelligence algorithms has a higher economic impact than traditional strategies. However, it should also be considered the decreasing trend in the cost related to the development and maintenance of cloud services on which these techniques can be deployed. Moreover, the efficiency of plant equipment and components available in the market has almost reached its theoretical limit boosted by the technological progress and incentive programs. Considering this aspect, it is desirable that, in the near future, incentives initiatives will be mainly focused on supporting innovative strategies for the management of equipment and systems during operation rather than further technological improvements.

Domain expertise still matters? In a research environment dominated by data scientists and complex mathematical algorithms promising to agnostically extract complex information from any type of data, building physics knowledge still plays a key role. As demonstrated in several applications an effective advancement in this area is only possible through a perfect combination of knowledge about algorithm development and physics laws governing the control problem under investigation.

Model-based versus model-free: is this a dilemma? Model-free controllers are often presented as the panacea to problems and limitations of their model-based counterparts. As also shown in the application introduced in section 5.1 both have strengths and weaknesses. While model-based controllers showed greater robustness and stability, model-free controllers are more adaptable and are able to learn complex behaviors. The questions to which scientific research in this area will hopefully provide answers are the following:

- For which control problems and configuration of HVAC system is it more convenient to use one or the other strategy?
- Is it possible to combine the strengths of the two approaches in order to obtain a strategy that is both robust and capable to adapt?

In this context, innovative algorithmic approaches merging the relative benefits of MPC and RL techniques are emerging from the scientific literature [179].

Is model-free really free from models? The literature review and the applications developed in the present dissertation showed how DRL agents employed to manage HVAC system heavily relies on models of the controlled environment. Given the considerable amount of interactions required to converge to an acceptable control policy, it seems unfeasible to directly implement model-free controllers in real buildings without performing a pre-training phase on simulation models. The application presented in section 4.2 demonstrated the applicability of data-driven modeling of building dynamics to pre-train DRL agents. In that case a model of the environment is still required for the implementation. However, its development has a greater potential of standardization and automation compared purely model-based frameworks. Conversely, the application presented in section 5.1 introduced a purely model-free agent which was directly deployed on the analyzed case study. What emerged after carrying out the research of the present dissertation is that the term "model-free" could result misleading to the inexperienced reader or practitioner.

Without trasferability there will be no scalability The trend emerging from the current scientific literature highlights the dependency of advanced control strategies from machine learning and deep learning models. Both model-based and model-free frameworks increasingly employ these techniques to effectively map system dynamics or control policies. In this context, a promising methodology to scale-up the application of machine learning models in real-world environments is transfer learning [180]. This technique aims at transferring a model trained for one system or task to another similar system or task minimizing the modeling effort in the process. Transfer learning could enable the re-utilization of models of building dynamics or control policies exploiting knowledge previously learned increasing the scalability of advanced control strategies and reducing their implementation cost.

References

[1] IEA. World energy outlook 2019, iea, paris. *IEA*, 2019.

[2] IEA. Directive 2018/844/eu of the european parliament and of the council of 30 may 2018, amending directives 2010/31/eu on the energy performance of buildings and directive 2012/27/eu on energy efficiency. *Eur. Commun*, 156:75–91, 2018.

[3] Zhun Yu, Benjamin C. M. Fung, and Fariborz Haghighat. Extracting knowledge from building-related data — a data mining framework. *Building Simulation*, 6(2):207–222, Jun 2013.

[4] A. Capozzoli, T. Cerquitelli, and M.S. Piscitelli. Chapter 11 - enhancing energy efficiency in buildings through innovative data analytics technologiesa. In Ciprian Dobre and Fatos Xhafa, editors, *Pervasive Computing*, Intelligent Data-Centric Systems, pages 353–389. Academic Press, Boston, 2016.

[5] Alfonso Capozzoli, Marco Savino Piscitelli, Silvio Brandi, Daniele Grassi, and Gianfranco Chicco. Automated load pattern learning and anomaly detection for enhancing energy management in smart buildings. *Energy*, 157:336 – 352, 2018.

[6] Christian Finck, Paul Beagon, John Clauß, Thibault Péan, Pierre Vogler-Finck, Kun Zhang, and Hussain Kazmi. Review of applied and tested control possibilities for energy flexibility in buildings - a technical report from iea ebc annex 67 energy flexible buildings. 05 2018.

[7] John Clauß, Christian Finck, Pierre Jacques Camille Vogler-Finck, and Paul Beagon. Control strategies for building energy systems to unlock demand side flexibility: A review. In *Proceedings of the 15th IBPSA Conference San Francisco, CA, USA, Aug. 7-9, 2017*, volume 15 of *Building Simulation Conference proceedings*, pages 1750–1759. IBPSA, 2017. 15th IBPSA Conference : Building Simulation 2017 ; Conference date: 07-08-2017 Through 09-08-2017.

[8] Zhe Wang and Tianzhen Hong. Reinforcement learning for building controls: The opportunities and challenges. *Applied Energy*, 269:115036, 2020.

[9] Shengwei Wang and Zhenjun Ma. Supervisory and optimal control of building hvac systems: A review. *HVAC & R Research*, 14:3–32, 01 2008.

[10] Q. Zhang, Y. W. Wong, S. C. Fok, and T. Y. Bong. Neural-based air-handling unit for indoor relative humidity and temperature control. *ASHRAE Transactions*, 111 PART 1:63–70, 2005. American Society of Heating, Refrigerating and Air-Conditioning Engineers ASHRAE 2005 Winter Meeting ; Conference date: 05-02-2005 Through 09-02-2005.

[11] J.B. Rishel. Control of variable speed pumps for hvac water systems. *ASHRAE Transactions*, 109:380–389, 01 2003.

[12] Timothy I. Salsbury. A survey of control technologies in the building automation industry. *IFAC Proceedings Volumes*, 38(1):90–100, 2005. 16th IFAC World Congress.

[13] Farinaz Behrooz, Norman Mariun, Mohammad Hamiruce Marhaban, Mohd Amran Mohd Radzi, and Abdul Rahman Ramli. Review of control techniques for hvac systems — nonlinearity approaches based on fuzzy cognitive maps. *Energies*, 11(3), 2018.

[14] Abdul Afram and Farrokh Janabi-Sharifi. Theory and applications of hvac control systems – a review of model predictive control (mpc). *Building and Environment*, 72:343–355, 2014.

[15] D. Subbaram Naidu and Craig G. Rieger. Advanced control strategies for heating, ventilation, air-conditioning, and refrigeration systems—an overview: Part i: Hard control. *HVAC&R Research*, 17(1):2–21, 2011.

[16] D. Subbaram Naidu and Craig G. Rieger. Advanced control strategies for hvac&r systems—an overview: Part ii: Soft and fusion control. *HVAC&R Research*, 17(2):144–158, 2011.

[17] Gianluca Serale, Massimo Fiorentini, Alfonso Capozzoli, Daniele Bernardini, and Alberto Bemporad. Model predictive control (mpc) for enhancing building and hvac system energy efficiency: Problem formulation, applications and opportunities. *Energies*, 11(3), 2018.

[18] Samuel Prívara, Jiří Cigler, Zdeněk Váňa, Frauke Oldewurtel, Carina Sagerschnig, and Eva Žáčeková. Building modeling as a crucial part for building predictive control. *Energy and Buildings*, 56:8–22, 2013.

[19] Georgios Lymperopoulos and Petros Ioannou. Building temperature regulation in a multi-zone hvac system using distributed adaptive control. *Energy and Buildings*, 215:109825, 2020.

[20] Rasmus Halvgaard, Niels Kjølstad Poulsen, Henrik Madsen, and John Bagterp Jørgensen. Economic model predictive control for building climate control in a smart grid. In *2012 IEEE PES Innovative Smart Grid Technologies (ISGT)*, pages 1–6, 2012.

[21] Georgios D. Kontes, Georgios I. Giannakis, Víctor Sánchez, Pablo De Agustin-Camacho, Ander Romero-Amorrortu, Natalia Panagiotidou, Dimitrios V. Rovas, Simone Steiger, Christopher Mutschler, and Gunnar Gruen. Simulation-based evaluation and optimization of control strategies in buildings. *Energies*, 11(12):3376, 2018.

[22] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. A Bradford Book, Cambridge, MA, USA, 2018.

[23] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518(7540):pp. 529–533, February 2015.

[24] 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, 2020.

[25] Davide Coraci, Silvio Brandi, Marco Savino Piscitelli, and Alfonso Capozzoli. Online implementation of a soft actor-critic agent to enhance indoor temperature control and energy efficiency in buildings. *Energies*, 14(4), 2021.

[26] Silvio Brandi, Antonio Gallo, and Alfonso Capozzoli. A predictive and adaptive control strategy to optimize the management of integrated energy systems in buildings. *Energy Reports*, 8:1550–1567, 2022.

[27] Silvio Brandi, Massimo Fiorentini, and Alfonso Capozzoli. Comparison of online and offline deep reinforcement learning with model predictive control for thermal energy management. *Automation in Construction*, 135:104128, 2022.

[28] Michael L. Littman, Thomas L. Dean, and Leslie Pack Kaelbling. On the complexity of solving markov decision problems. *CoRR*, abs/1302.4971, 2013.

[29] Ofir Nachum, Mohammad Norouzi, Kelvin Xu, and Dale Schuurmans. Bridging the gap between value and policy based reinforcement learning. *CoRR*, abs/1702.08892, 2017.

[30] Christopher J.C.H. Watkins and Peter Dayan. Technical note: Q-learning. *Machine Learning*, 8(3):279–292, May 1992.

[31] Hai Nguyen and Hung La. Review of deep reinforcement learning for robot manipulation. In *2019 Third IEEE International Conference on Robotic Computing (IRC)*, pages 590–595, 2019.

[32] John Langford. *Efficient Exploration in Reinforcement Learning*, pages 309–311. Springer US, Boston, MA, 2010.

[33] Hado van Hasselt, Arthur Guez, and David Silver. Deep reinforcement learning with double q-learning. *CoRR*, abs/1509.06461, 2015.

[34] José R. Vázquez-Canteli and Zoltán Nagy. Reinforcement learning for demand response: A review of algorithms and modeling techniques. *Applied Energy*, 235:1072–1089, 2019.

[35] Satinder P. Singh, Tommi Jaakkola, and Michael I. Jordan. Learning without state-estimation in partially observable markovian decision processes. In William W. Cohen and Haym Hirsh, editors, *Machine Learning Proceedings 1994*, pages 284–292. Morgan Kaufmann, San Francisco (CA), 1994.

[36] Tuomas Haarnoja, Aurick Zhou, Kristian Hartikainen, George Tucker, Sehoon Ha, Jie Tan, Vikash Kumar, Henry Zhu, Abhishek Gupta, Pieter Abbeel, and Sergey Levine. Soft actor-critic algorithms and applications, 2019.

[37] Petros Christodoulou. Soft actor-critic for discrete action settings. *CoRR*, abs/1910.07207, 2019.

[38] Zhiang Zhang, Adrian Chong, Yuqi Pan, Chenlu Zhang, and Khee Poh Lam. Whole building energy model for hvac optimal control: A practical framework based on deep reinforcement learning. *Energy and Buildings*, 199:pp. 472–490, 2019.

[39] Zhiang Zhang, Adrian Chong, Yuqi Pan, Chenlu Zhang, Siliang Lu, and Khee Poh Lam. A deep reinforcement learning approach to usingwhole building energymodel for hvac optimal control. 2018.

[40] Samuel L. Smith, Pieter-Jan Kindermans, Chris Ying, and Quoc V. Le. Don't decay the learning rate, increase the batch size, 2018. (accessed October 14, 2021).

[41] Abien Fred Agarap. Deep learning using rectified linear units (relu), 2019.

[42] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017.

[43] T.; Hinton G. Tieleman. Lecture 6.5-rmsprop: divide the gradient by a running average of its recent magnitude. COURSERA 4 (2012) 26-31, 2012.

[44] MATLAB. *version 7.10.0 (R2019b)*. The MathWorks Inc., Natick, Massachusetts, 2019.

[45] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh

Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[46] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[47] Matthias Plappert. keras-rl. https://github.com/keras-rl/keras-rl, 2016.

[48] Ashley Hill, Antonin Raffin, Maximilian Ernestus, Adam Gleave, Anssi Kanervisto, Rene Traore, Prafulla Dhariwal, Christopher Hesse, Oleg Klimov, Alex Nichol, Matthias Plappert, Alec Radford, John Schulman, Szymon Sidor, and Yuhuai Wu. Stable baselines. https://github.com/hill-a/stable-baselines, 2018.

[49] Ján Drgoňa, Javier Arroyo, Iago Cupeiro, David Blum, Krzysztof Arendt, Donghun Kim, Enric Olle, Juraj Oravec, Michael Wetter, Draguna Vrabie, and L. Helsen. All you need to know about model predictive control for buildings. *Annual Reviews in Control*, 50, 10 2020.

[50] Hao Gao, Christian Koch, and Yupeng Wu. Building information modelling based building energy modelling: A review. *Applied Energy*, 238:320–343, 2019.

[51] Abdul Afram and Farrokh Janabi-Sharifi. Review of modeling methods for hvac systems. *Applied Thermal Engineering*, 67(1):507–519, 2014.

[52] Zakia Afroz, GM Shafiullah, Tania Urmee, and Gary Higgins. Modeling techniques used in building hvac control systems: A review. *Renewable and Sustainable Energy Reviews*, 83:64–84, 2018.

[53] Drury Crawley, Linda Lawrie, Frederick Winkelmann, W.F. Buhl, Y.Joe Huang, Curtis Pedersen, Richard Strand, Richard Liesen, Daniel Fisher, Michael Witte, and Jason Glazer. Energyplus: Creating a new-generation building energy simulation program. *Energy and Buildings*, 33:pp. 319–331, 04 2001.

[54] University of Wisconsin-Madison. Solar Energy Laboratory. *TRNSYS, a transient simulation program*. Madison, Wis. : The Laboratory, 1975., 1975.

Loose-leaf for updating.;March 31, 1975.;"This manual, and the TRNSYS program it describes, were developed under grants from the RANN program of the National Science Foundation (Grant GI 34029), and from the Energy Research and Development Administration (Contract E(11-1)-2588).

[55] Modelica Association. Modelica® - a unified object-oriented language for physical systems modeling. Tutorial, December 2000.

[56] Michael Wetter, Wangda Zuo, Thierry Nouidui, and Xiufeng Pang. Modelica buildings library. *Journal of Building Performance Simulation*, 7, 07 2014.

[57] Michael Wetter. A modular building controls virtual test bed for the integrations of heterogeneous systems. 2008.

[58] David Broman, Christopher Brooks, Lev Greenberg, Edward A. Lee, Michael Masin, Stavros Tripakis, and Michael Wetter. Determinate composition of fmus for co-simulation. In *Proceedings of the Eleventh ACM International Conference on Embedded Software*, EMSOFT '13. IEEE Press, 2013.

[59] Silvio Brandi, Davide Coraci, Davide Borello, and Alfonso Capozzoli. Energy management of a residential heating system through deep reinforcement learning. In John R. Littlewood, Robert J. Howlett, and Lakhmi C. Jain, editors, *Sustainability in Energy and Buildings 2021*, pages 329–339, Singapore, 2022. Springer Singapore.

[60] Zhiang Zhang, Adrian Chong, Yuqi Pan, Chenlu Zhang, and Khee Poh Lam. Whole building energy model for hvac optimal control: A practical framework based on deep reinforcement learning. *Energy and Buildings*, 199:472–490, 2019.

[61] Georgios D. Kontes, Georgios I. Giannakis, Víctor Sánchez, Pablo De Agustin-Camacho, Ander Romero-Amorrortu, Natalia Panagiotidou, Dimitrios V. Rovas, Simone Steiger, Christopher Mutschler, and Gunnar Gruen. Simulation-based evaluation and optimization of control strategies in buildings. *Energies*, 11(12), 2018.

[62] Srinarayana Nagarathinam, Vishnu Menon, Arunchandar Vasan, and Anand Sivasubramaniam. Marco - multi-agent reinforcement learning based control of building hvac systems. In *Proceedings of the Eleventh ACM International Conference on Future Energy Systems*, e-Energy '20, page 57–67, New York, NY, USA, 2020. Association for Computing Machinery.

[63] Ki Uhn Ahn and Cheol Soo Park. Application of deep q-networks for model-free optimal control balancing between different hvac systems. *Science and Technology for the Built Environment*, 26(1):61–74, 2020.

[64] Ruoxi Jia, Ming Jin, Kaiyu Sun, Tianzhen Hong, and Costas Spanos. Advanced building control via deep reinforcement learning. *Energy Procedia*, 158:6158–6163, 2019. Innovative Solutions for Energy Transitions.

[65] Shunian Qiu, Zhenhai Li, Zhengwei Li, Jiajie Li, Shengping Long, and Xiaoping Li. Model-free control method based on reinforcement learning for building cooling water systems: Validation by measured data-based simulation. *Energy and Buildings*, 218:110055, 2020.

[66] Thomas Schreiber, Sören Eschweiler, Marc Baranski, and Dirk Müller. Application of two promising reinforcement learning algorithms for load shifting in a cooling supply system. *Energy and Buildings*, 229:110490, 2020.

[67] Christian Blad, Simon Bøgh, and Carsten Kallesøe. A multi-agent reinforcement learning approach to price and comfort optimization in hvac-systems. *Energies*, 14(22), 2021.

[68] Samir Touzani, Anand Krishnan Prakash, Zhe Wang, Shreya Agarwal, Marco Pritoni, Mariam Kiran, Richard Brown, and Jessica Granderson. Controlling distributed energy resources via deep reinforcement learning for load flexibility and energy efficiency. *Applied Energy*, 304:117733, 2021.

[69] Joon-Yong Lee, Sen Huang, Aowabin Rahman, Amanda D. Smith, and Srinivas Katipamula. Flexible reinforcement learning framework for building control using energyplus-modelica energy models. In *Proceedings of the 1st International Workshop on Reinforcement Learning for Energy Management in Buildings & Cities*, RLEM'20, page 34–38, New York, NY, USA, 2020. Association for Computing Machinery.

[70] Yujiao Chen, Leslie K. Norford, Holly W. Samuelson, and Ali Malkawi. Optimal control of hvac and window systems for natural ventilation through reinforcement learning. *Energy and Buildings*, 169:195–205, 2018.

[71] Zhanhong Jiang, Michael J. Risbeck, Vish Ramamurti, Sugumar Murugesan, Jaume Amores, Chenlu Zhang, Young M. Lee, and Kirk H. Drees. Building hvac control with reinforcement learning for reduction of energy cost and demand charge. *Energy and Buildings*, 239:110833, 2021.

[72] Zhengbo Zou, Xinran Yu, and Semiha Ergan. Towards optimal control of air handling units using deep reinforcement learning and recurrent neural network. *Building and Environment*, 168:106535, 2020.

[73] G.T. Costanzo, S. Iacovella, F. Ruelens, T. Leurs, and B.J. Claessens. Experimental analysis of data-driven control for a building heating system. *Sustainable Energy, Grids and Networks*, 6:81–90, 2016.

[74] Yan Du, Helia Zandi, Olivera Kotevska, Kuldeep Kurte, Jeffery Munk, Kadir Amasyali, Evan Mckee, and Fangxing Li. Intelligent multi-zone residential hvac control strategy based on deep reinforcement learning. *Applied Energy*, 281:116117, 2021.

[75] William Valladares, Marco Galindo, Jorge Gutiérrez, Wu-Chieh Wu, Kuo-Kai Liao, Jen-Chung Liao, Kuang-Chin Lu, and Chi-Chuan Wang. Energy optimization associated with thermal comfort and indoor air control via a deep

reinforcement learning algorithm. *Building and Environment*, 155:105–117, 2019.

[76] Bingqing Chen, Zicheng Cai, and Mario Bergés. Gnu-rl: A precocial reinforcement learning solution for building hvac control using a differentiable mpc policy. In *Proceedings of the 6th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation*, BuildSys '19, page 316–325, New York, NY, USA, 2019. Association for Computing Machinery.

[77] Xiongfeng Zhang, Renzhi Lu, Junhui Jiang, Seung Ho Hong, and Won Seok Song. Testbed implementation of reinforcement learning-based demand response energy management system. *Applied Energy*, 297:117131, 2021.

[78] Oscar De Somer, Ana Soares, Koen Vanthournout, Fred Spiessens, Tristan Kuijpers, and Koen Vossen. Using reinforcement learning for demand response of domestic hot water buffers: A real-life demonstration. In *2017 IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT-Europe)*, pages 1–7, 2017.

[79] B. Dupont, P. Vingerhoets, P. Tant, K. Vanthournout, W. Cardinaels, T. De Rybel, E. Peeters, and R. Belmans. Linear breakthrough project: Large-scale implementation of smart grid technologies in distribution grids. In *2012 3rd IEEE PES Innovative Smart Grid Technologies Europe (ISGT Europe)*, pages 1–8, 2012.

[80] Ioannis Antonopoulos, Valentin Robu, Benoit Couraud, Desen Kirli, Sonam Norbu, Aristides Kiprakis, David Flynn, Sergio Elizondo-Gonzalez, and Steve Wattam. Artificial intelligence and machine learning approaches to energy demand-side response: A systematic review. *Renewable and Sustainable Energy Reviews*, 130:109899, 2020.

[81] Yong Liang, Long He, Xinyu Cao, and Zuo-Jun Shen. Stochastic control for smart grid users with flexible demand. *IEEE Transactions on Smart Grid*, 4(4):2296–2308, 2013.

[82] Paulo Lissa, Michael Schukat, Marcus Keane, and Enda Barrett. Transfer learning applied to drl-based heat pump control to leverage microgrid energy efficiency. *Smart Energy*, 3:100044, 2021.

[83] Steven T Taylor. *Fundamentals of Design and Control of Central Chilled-Water Plants*. American Society of Heating, Refrigerating and Air-Conditioning Engineers . . . , 2017.

[84] Derk J Swider. A comparison of empirically based steady-state models for vapor-compression liquid chillers. *Applied thermal engineering*, 23(5):539–556, 2003.

[85] Young Ran Yoon and Hyeun Jun Moon. Performance based thermal comfort control (ptcc) using deep reinforcement learning for space cooling. *Energy and Buildings*, 203:109420, 2019.

[86] Avisek Naug, Marcos Quiñones Grueiro, and Gautam Biswas. Continual adaptation in deep reinforcement learning-based control applied to non-stationary building environments. In *Proceedings of the 1st International Workshop on Reinforcement Learning for Energy Management in Buildings & Cities*, RLEM'20, page 24–28, New York, NY, USA, 2020. Association for Computing Machinery.

[87] S. Murugesan, Z. Jiang, M. J. Risbeck, J. Amores, C. Zhang, V. Ramamurti, K. H. Drees, and Y. M. Lee. Less is more: Simplified state-action space for deep reinforcement learning based hvac control. In *Proceedings of the 1st International Workshop on Reinforcement Learning for Energy Management in Buildings & Cities*, RLEM'20, page 20–23, New York, NY, USA, 2020. Association for Computing Machinery.

[88] Choton K. Das, Octavian Bass, Ganesh Kothapalli, Thair S. Mahmoud, and Daryoush Habibi. Overview of energy storage systems in distribution networks: Placement, sizing, operation, and power quality. *Renewable and Sustainable Energy Reviews*, 91:1205–1230, 2018.

[89] Tom Terlouw, Tarek AlSkaif, Christian Bauer, and Wilfried van Sark. Optimal energy management in all-electric residential energy systems with heat and electricity storage. *Applied Energy*, 254:113580, 2019.

[90] F. Ruelens, B. J. Claessens, S. Quaiyum, B. De Schutter, R. Babuška, and R. Belmans. Reinforcement learning applied to an electric water heater: From theory to practice. *IEEE Transactions on Smart Grid*, 9(4):3792–3800, 2018.

[91] Simeng Liu and Gregor P. Henze. Experimental analysis of simulated reinforcement learning control for active and passive building thermal storage inventory: Part 2: Results and analysis. *Energy and Buildings*, 38(2):148–161, 2006.

[92] Giuseppe Pinto, Davide Deltetto, and Alfonso Capozzoli. Data-driven district energy management with surrogate models and deep reinforcement learning. *Applied Energy*, 304:117642, 2021.

[93] Jose R. Vazquez-Canteli, Gregor Henze, and Zoltan Nagy. Marlisa: Multi-agent reinforcement learning with iterative sequential action selection for load shaping of grid-interactive connected buildings. In *Proceedings of the 7th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation*, BuildSys '20, page 170–179, New York, NY, USA, 2020. Association for Computing Machinery.

[94] Anjukan Kathirgamanathan, Eleni Mangina, and Donal P. Finn. Development of a soft actor critic deep reinforcement learning approach for harnessing energy flexibility in a large office building. *Energy and AI*, 5:100101, 2021.

[95] Georgios Martinopoulos, Konstantinos T. Papakostas, and Agis M. Papadopoulos. A comparative review of heating systems in eu countries, based on efficiency and fuel cost. *Renewable and Sustainable Energy Reviews*, 90:687 – 699, 2018.

[96] Ali Baniasadi, Daryoush Habibi, Waleed Al-Saedi, Mohammad A.S. Masoum, Choton K. Das, and Navid Mousavi. Optimal sizing design and operation of electrical and thermal energy storage systems in smart buildings. *Journal of Energy Storage*, 28:101186, 2020.

[97] Masoume Shabani and Javad Mahmoudimehr. Techno-economic role of pv tracking technology in a hybrid pv-hydroelectric standalone power system. *Applied Energy*, 212:84–108, 2018.

[98] Juha Koskela, Antti Rautiainen, and Pertti Järventausta. Using electrical energy storage in residential buildings – sizing of battery and photovoltaic panels based on electricity cost optimization. *Applied Energy*, 239:1175–1189, 2019.

[99] Sepehr Sanaye and Ahmadreza Sarrafi. A novel energy management method based on deep q network algorithm for low operating cost of an integrated hybrid system. *Energy Reports*, 7:2647–2663, 2021.

[100] Chenxiao Guan, Yanzhi Wang, Xue Lin, Shahin Nazarian, and Massoud Pedram. Reinforcement learning-based control of residential energy storage systems for electric bill minimization. In *2015 12th Annual IEEE Consumer Communications and Networking Conference (CCNC)*, pages 637–642, 2015.

[101] Amjad Anvari-Moghaddam, Ashkan Rahimi-Kian, Maryam S. Mirian, and Josep M. Guerrero. A multi-agent based energy management solution for integrated buildings and microgrid system. *Applied Energy*, 203:41–56, 2017.

[102] Gabriel Zsembinszki, Cèsar Fernández, David Vérez, and Luisa F. Cabeza. Deep learning optimal control for a complex hybrid energy storage system. *Buildings*, 11(5), 2021.

[103] Naren Srivaths Raman, Ninad Gaikwad, Prabir Barooah, and Sean P. Meyn. Reinforcement learning-based home energy management system for resiliency. In *2021 American Control Conference (ACC)*, pages 1358–1364, 2021.

[104] Yuekuan Zhou, Sunliang Cao, Jan L.M. Hensen, and Peter D. Lund. Energy integration and interaction between buildings and vehicles: A state-of-the-art review. *Renewable and Sustainable Energy Reviews*, 114:109337, 2019.

[105] Chunhua Liu, K. T. Chau, Diyun Wu, and Shuang Gao. Opportunities and challenges of vehicle-to-home, vehicle-to-vehicle, and vehicle-to-grid technologies. *Proceedings of the IEEE*, 101(11):2409–2427, 2013.

[106] B. Svetozarevic, C. Baumann, S. Muntwiler, L. Di Natale, M.N. Zeilinger, and P. Heer. Data-driven control of room temperature and bidirectional ev charging using deep reinforcement learning: Simulations and experiments. *Applied Energy*, 307:118127, Feb 2022.

[107] Elena Mocanu, Decebal Constantin Mocanu, Phuong H. Nguyen, Antonio Liotta, Michael E. Webber, Madeleine Gibescu, and J. G. Slootweg. Online building energy optimization using deep reinforcement learning. *IEEE Transactions on Smart Grid*, 10(4):3698–3708, 2019.

[108] William Curran, Tim Brys, Matthew Taylor, and William Smart. Using pca to efficiently represent state spaces, 2015.

[109] Siliang Lu, Weilong Wang, Chaochao Lin, and Erica Cochran Hameen. Data-driven simulation of a thermal comfort-based temperature set-point control with ashrae rp884. *Building and Environment*, 156:137–146, 2019.

[110] 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, 2020.

[111] Guanyu Gao, Jie Li, and Yonggang Wen. Deepcomfort: Energy-efficient thermal comfort control in buildings via reinforcement learning. *IEEE Internet of Things Journal*, 7(9):8472–8484, 2020.

[112] Xiaolei Yuan, Yiqun Pan, Jianrong Yang, Weitong Wang, and Zhizhong Huang. Study on the application of reinforcement learning in the operation optimization of hvac system. *Building Simulation*, 14(1):75–87, Feb 2021.

[113] Tianshu Wei, Yanzhi Wang, and Qi Zhu. Deep reinforcement learning for building hvac control. In *Proceedings of the 54th Annual Design Automation Conference 2017*, DAC '17, New York, NY, USA, 2017. Association for Computing Machinery.

[114] José R. Vázquez-Canteli, Stepan Ulyanin, Jérôme Kämpf, and Zoltán Nagy. Fusing tensorflow with building energy simulation for intelligent energy management in smart cities. *Sustainable Cities and Society*, 45:243–257, 2019.

[115] José Vázquez-Canteli, Jérôme Kämpf, and Zoltán Nagy. Balancing comfort and energy consumption of a heat pump using batch reinforcement learning with fitted q-iteration. *Energy Procedia*, 122:415–420, 2017. CISBAT 2017 International ConferenceFuture Buildings & Districts – Energy Efficiency from Nano to Urban Scale.

[116] Kento TOMODA and Yasuyuki SHIRAISHI. Operational control for earth-to-air heat exchanger by reinforcement learning (part 1): Applicability verification of algorithm considering counterfactual reward. *Journal of Environmental Engineering (Transactions of AIJ)*, 86(785):708–718, 2021.

[117] David Silver, Guy Lever, Nicolas Heess, Thomas Degris, Daan Wierstra, and Martin Riedmiller. Deterministic policy gradient algorithms. In *International conference on machine learning*, pages 387–395. PMLR, 2014.

[118] Hepeng Li, Zhiqiang Wan, and Haibo He. Real-time residential demand response. *IEEE Transactions on Smart Grid*, 11(5):4144–4154, 2020.

[119] Dorota A. Chwieduk. Towards modern options of energy conservation in buildings. *Renewable Energy*, 101:1194–1202, 2017.

[120] Hans Auer and Reinhard Haas. On integrating large shares of variable renewables into the electricity system. *Energy*, 115:1592–1601, 2016. Sustainable Development of Energy, Water and Environment Systems.

[121] Haider Tarish Haider, Ong Hang See, and Wilfried Elmenreich. A review of residential demand response of smart grid. *Renewable and Sustainable Energy Reviews*, 59:166–178, 2016.

[122] Davide Deltetto, Davide Coraci, Giuseppe Pinto, Marco Savino Piscitelli, and Alfonso Capozzoli. Exploring the potentialities of deep reinforcement learning for incentive-based demand response in a cluster of small commercial buildings. *Energies*, 14(10), 2021.

[123] Hongxun Hui, Yi Ding, Weidong Liu, You Lin, and Yonghua Song. Operating reserve evaluation of aggregated air conditioners. *Applied Energy*, 196:218–228, 2017.

[124] Luis Pérez-Lombard, José Ortiz, and Christine Pout. A review on buildings energy consumption information. *Energy and Buildings*, 40(3):394–398, 2008.

[125] Abhinandana Boodi, Karim Beddiar, Malek Benamour, Yassine Amirat, and Mohamed Benbouzid. Intelligent systems for building energy and occupant comfort optimization: A state of the art review and recommendations. *Energies*, 11(10), 2018.

[126] Poul O Fanger et al. Thermal comfort. analysis and applications in environmental engineering. *Thermal comfort. Analysis and applications in environmental engineering.*, 1970.

[127] Toby Cheung, Stefano Schiavon, Thomas Parkinson, Peixian Li, and Gail Brager. Analysis of the accuracy on pmv – ppd model using the ashrae global thermal comfort database ii. *Building and Environment*, 153:205–217, 2019.

[128] Mahmoud M. Abdelrahman, Adrian Chong, and Clayton Miller. Personal thermal comfort models using digital twins: Preference prediction with bim-extracted spatial–temporal proximity data from build2vec. *Building and Environment*, 207:108532, 2022.

[129] Sajjad Baghaee and Ilkay Ulusoy. User comfort and energy efficiency in hvac systems by q-learning. In *2018 26th Signal Processing and Communications Applications Conference (SIU)*, pages 1–4, 2018.

[130] Brida V. Mbuwir, Frederik Ruelens, Fred Spiessens, and Geert Deconinck. Battery energy management in a microgrid using batch reinforcement learning. *Energies*, 10(11), 2017.

[131] Bert J. Claessens, D. Vanhoudt, J. Desmedt, and F. Ruelens. Model-free control of thermostatically controlled loads connected to a district heating network. *Energy and Buildings*, 159:1–10, 2018.

[132] Hussain Kazmi, Fahad Mehmood, Stefan Lodeweyckx, and Johan Driesen. Gigawatt-hour scale savings on a budget of zero: Deep reinforcement learning based optimal control of hot water systems. *Energy*, 144:159–168, 2018.

[133] Danilo Fuselli, Francesco De Angelis, Matteo Boaro, Stefano Squartini, Qinglai Wei, Derong Liu, and Francesco Piazza. Action dependent heuristic dynamic programming for home energy resource scheduling. *International Journal of Electrical Power & Energy Systems*, 48:148–160, 2013.

[134] Yuan Wang, Kirubakaran Velswamy, and Biao Huang. A long-short term memory recurrent neural network based reinforcement learning controller for office heating ventilation and air conditioning systems. *Processes*, 5(3), 2017.

[135] Bert J. Claessens, Peter Vrancx, and Frederik Ruelens. Convolutional neural networks for automatic state-time feature extraction in reinforcement learning applied to residential load control. *IEEE Transactions on Smart Grid*, 9(4):3259–3269, 2018.

[136] Jingran Ma, Joe Qin, Timothy Salsbury, and Peng Xu. Demand reduction in building energy systems based on economic model predictive control. *Chemical Engineering Science*, 67(1):pp. 92–100, 2012.

[137] Jianli Chen, Godfried Augenbroe, and Xinyi Song. Lighted-weighted model predictive control for hybrid ventilation operation based on clusters of neural network models. *Automation in Construction*, 89:pp. 250–265, 2018.

[138] Gregor P. Henze, Robert H. Dodier, and Moncef Krarti. Development of a Predictive Optimal Controller for Thermal Energy Storage Systems. *HVAC&R Research*, 3(3):pp. 233–264, 1997.

[139] S.H Cho and M Zaheer-uddin. Predictive control of intermittently operated radiant floor heating systems. *Energy Conversion and Management*, 44(8):pp. 1333–1342, 2003.

[140] Gianluca Serale, Massimo Fiorentini, Alfonso Capozzoli, Paul Cooper, and Marco Perino. Formulation of a model predictive control algorithm to enhance the performance of a latent heat solar thermal system. *Energy Conversion and Management*, 173:pp. 438–449, oct 2018.

[141] Massimo Fiorentini, Josh Wall, Zhenjun Ma, Julio H. Braslavsky, and Paul Cooper. Hybrid model predictive control of a residential HVAC system with on-site thermal energy generation and storage. *Applied Energy*, 187:pp. 465–479, feb 2017.

[142] Sayani Seal, Benoit Boulet, and Vahid R. Dehkordi. Centralized model predictive control strategy for thermal comfort and residential energy management. *Energy*, 212:118456, 2020.

[143] David J. Biagioni, Xiangyu Zhang, Peter Graf, Devon Sigler, and Wesley Jones. A Comparison of Model-Free and Model Predictive Control for Price Responsive Water Heaters. *Proceedings of the 1st International Workshop on Reinforcement Learning for Energy Management in Buildings & Cities (RLEM'20)*, page pp. 29–33, 2020.

[144] Glenn Ceusters, Román Cantú Rodríguez, Alberte Bouso García, Rüdiger Franke, Geert Deconinck, Lieve Helsen, Ann Nowé, Maarten Messagie, and Luis Ramirez Camargo. Model-predictive control and reinforcement learning in multi-energy system case studies. *Applied Energy*, 303:117634, 2021.

[145] Rob Guglielmetti, D. Macumber, and N. Long. Openstudio: An open source integrated analysis platform; preprint. 01 2011.

[146] Lin Zhang. Simulation analysis of built environment based on design builder software. *Applied Mechanics and Materials*, 580-583:3134–3137, 07 2014.

[147] pyep. https://pypi.org/project/pyEp/.

[148] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym, 2016.

[149] David Blum, Javier Arroyo, Sen Huang, Ján Drgoňa, Filip Jorissen, Harald Taxt Walnum, Yan Chen, Kyle Benne, Draguna Vrabie, Michael Wetter, and Lieve Helsen. Building optimization testing framework (boptest) for simulation-based benchmarking of control strategies in buildings. *Journal of Building Performance Simulation*, 14(5):586–610, 2021.

[150] Paul Scharnhorst, Baptiste Schubnel, Carlos Fernández Bandera, Jaume Salom, Paolo Taddeo, Max Boegli, Tomasz Gorecki, Yves Stauffer, Antonis Peppas, and Chrysa Politi. Energym: A building model library for controller benchmarking. *Applied Sciences*, 11(8), 2021.

[151] Alphabuilding. https://github.com/WalterZWang/AlphaBuilding-MedOffice.

[152] Jose R Vazquez-Canteli, Sourav Dey, Gregor Henze, and Zoltan Nagy. Citylearn: Standardizing research in multi-agent reinforcement learning for demand response and urban energy management, 2020.

[153] Advanced controls test bed (actb). https://github.com/WalterZWang/AlphaBuilding-MedOffice.

[154] Gabriel Dulac-Arnold, Richard Evans, Hado van Hasselt, Peter Sunehag, Timothy Lillicrap, Jonathan Hunt, Timothy Mann, Theophane Weber, Thomas Degris, and Ben Coppin. Deep reinforcement learning in large discrete action spaces, 2016.

[155] A. Lonza. Reinforcement learning algorithms with python. Packt, 2019.

[156] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework, 2019.

[157] Gabriele Comodi, Francesco Carducci, Balamurugan Nagarajan, and Alessandro Romagnoli. Application of cold thermal energy storage (ctes) for building demand management in hot climates. *Applied Thermal Engineering*, 103:1186–1195, 2016.

[158] Alessia Arteconi, Jing Xu, Eleonora Ciarrocchi, Luca Paciello, Gabriele Comodi, Fabio Polonara, and Ruzhu Wang. Demand side management of a building summer cooling load by means of a thermal energy storage. *Energy Procedia*, 75:3277–3283, 2015. Clean, Efficient and Affordable Energy for a Sustainable Future: The 7th International Conference on Applied Energy (ICAE2015).

[159] Daniele Ioli, Alessandro Falsone, and Maria Prandini. Optimal energy management of a building cooling system with thermal storage: A convex formulation. *IFAC-PapersOnLine*, 48(8):1150–1155, 2015. 9th IFAC Symposium on Advanced Control of Chemical Processes ADCHEM 2015.

[160] Haoshan Ren, Yongjun Sun, Ahmed K. Albdoor, V.V. Tyagi, A.K. Pandey, and Zhenjun Ma. Improving energy flexibility of a net-zero energy house using a solar-assisted air conditioning system with thermal energy storage and demand-side management. *Applied Energy*, 285:116433, 2021.

[161] Gabriele Comodi, Andrea Giantomassi, Marco Severini, Stefano Squartini, Francesco Ferracuti, Alessandro Fonti, Davide Nardi Cesarini, Matteo Morodo, and Fabio Polonara. Multi-apartment residential microgrid with electrical and thermal storage devices: Experimental analysis and simulation of energy management strategies. *Applied Energy*, 137:854–866, 2015.

[162] Pantelis Dimitroulis and Miltiadis Alamaniotis. A fuzzy logic energy management system of on-grid electrical system for residential prosumers. *Electric Power Systems Research*, 202:107621, 2022.

[163] Emrah Biyik and Aysegul Kahraman. A predictive control strategy for optimal management of peak load, thermal comfort, energy storage and renewables in multi-zone buildings. *Journal of Building Engineering*, 25:100826, 2019.

[164] Tetsuya Wakui, Kento Sawada, Ryohei Yokoyama, and Hirohisa Aki. Predictive management for energy supply networks using photovoltaics, heat

pumps, and battery by two-stage stochastic programming and rule-based control. *Energy*, 179:1302–1319, 2019.

[165] Jia Liu, Xi Chen, Hongxing Yang, and Yutong Li. Energy storage and management system design optimization for a photovoltaic integrated low-energy building. *Energy*, 190:116424, 2020.

[166] Vanika Sharma, Mohammed H. Haque, and Syed Mahfuzul Aziz. Energy cost minimization for net zero energy homes through optimal sizing of battery storage system. *Renewable Energy*, 141:278–286, 2019.

[167] Sašo Medved, Suzana Domjan, and Ciril Arkar. Contribution of energy storage to the transition from net zero to zero energy buildings. *Energy and Buildings*, 236:110751, 2021.

[168] Tiansong Cui, Shuang Chen, Yanzhi Wang, Qi Zhu, Shahin Nazarian, and Massoud Pedram. An optimal energy co-scheduling framework for smart buildings. *Integration*, 58:528–537, 2017.

[169] Angela Amato, Matteo Bilardo, Enrico Fabrizio, Valentina Serra, and F. Spertino. Energy evaluation of a pv-based test facility for assessing future self-sufficient buildings. *Energies*, 14:329, 01 2021.

[170] Reino Ruusu, Sunliang Cao, Benjamin Manrique Delgado, and Ala Hasan. Direct quantification of multiple-source energy flexibility in a residential building using a new model predictive high-level controller. *Energy Conversion and Management*, 180:1109–1128, 2019.

[171] William F. Holmgren, Clifford W. Hansen, and Mark A. Mikofski. pvlib python: a python package for modeling solar energy systems. *Journal of Open Source Software*, 3(29):884, 2018.

[172] Wilhelm Durisch, Bernd Bitnar, Jean-C. Mayor, Helmut Kiess, King hang Lam, and Josie Close. Efficiency model for photovoltaic modules and demonstration of its application to energy yield estimation. *Solar Energy Materials and Solar Cells*, 91(1):79–84, 2007.

[173] Mark Z. Jacobson and Vijaysinh Jadhav. World estimates of pv optimal tilt angles and ratios of sunlight incident upon tilted and tracked pv panels relative to horizontal panels. *Solar Energy*, 169:55–66, 2018.

[174] Daniel Görges. Relations between model predictive control and reinforcement learning. *IFAC-PapersOnLine*, 50(1):4920–4928, 2017. 20th IFAC World Congress.

[175] N. S. Raman, A. M. Devraj, P. Barooah, and S. P. Meyn. Reinforcement learning for control of building hvac systems. *2020 American Control Conference (ACC)*, pages pp. 2326–2332, 2020.

[176] Martin Herceg, Michal Kvasnica, Colin N. Jones, and Manfred Morari. Multi-parametric toolbox 3.0. *2013 European Control Conference (ECC)*, pages pp. 502–510, 2013.

[177] Gurobi. Gurobi Solver. 2020.

[178] Alberto Bemporad. Model predictive control design: New trends and tools. *Proceedings of the 45th IEEE Conference on Decision and Control, Vols 1-14*, (1):pp. 6678–6683, 2006.

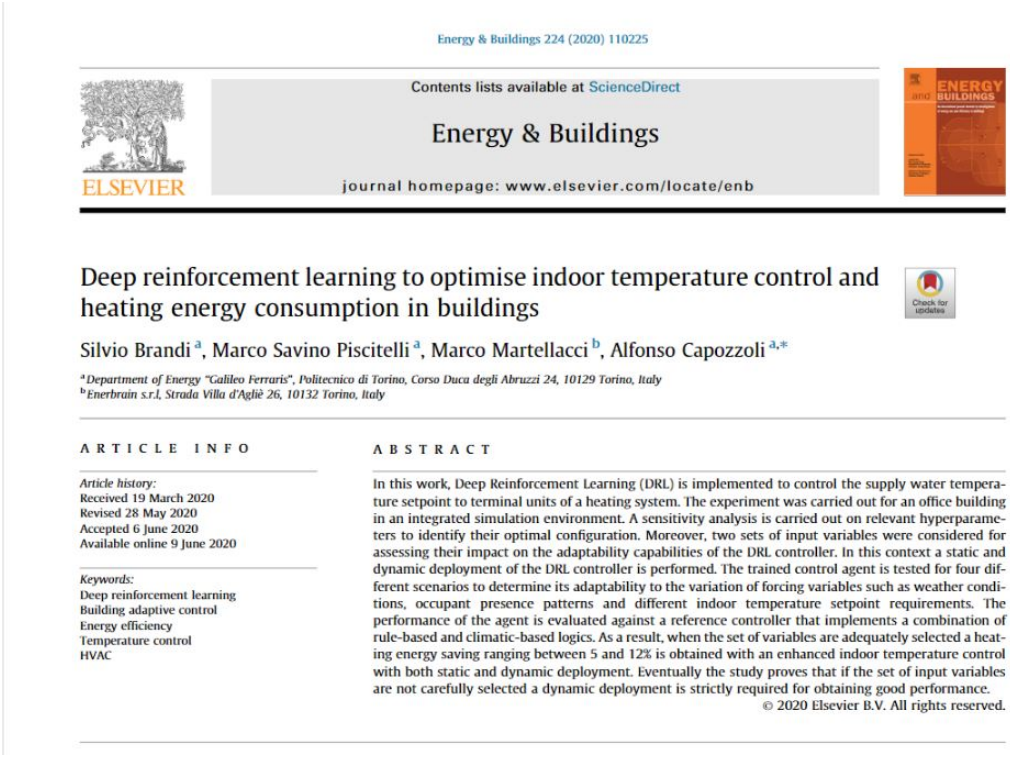
[179] Javier Arroyo, Carlo Manna, Fred Spiessens, and Lieve Helsen. Reinforced model predictive control (rl-mpc) for building energy management. *Applied Energy*, 309:118346, 2022.

[180] Giuseppe Pinto, Zhe Wang, Abhishek Roy, Tianzhen Hong, and Alfonso Capozzoli. Transfer learning for smart buildings: A critical review of algorithms, applications, and future perspectives. *Advances in Applied Energy*, 5:100084, 2022.

Appendix A

Journal papers included in this dissertation

This Appendix lists the papers published by the author that have been included/partially included in this dissertation.



Energy & Buildings 224 (2020) 110225

Contents lists available at ScienceDirect

Energy & Buildings

journal homepage: www.elsevier.com/locate/enb

Deep reinforcement learning to optimise indoor temperature control and heating energy consumption in buildings

Silvio Brandi [a], Marco Savino Piscitelli [a], Marco Martellacci [b], Alfonso Capozzoli [a,*]

[a] Department of Energy "Galileo Ferraris", Politecnico di Torino, Corso Duca degli Abruzzi 24, 10129 Torino, Italy
[b] Enerbrain s.r.l, Strada Villa d'Agliè 26, 10132 Torino, Italy

ARTICLE INFO

Article history:
Received 19 March 2020
Revised 28 May 2020
Accepted 6 June 2020
Available online 9 June 2020

Keywords:
Deep reinforcement learning
Building adaptive control
Energy efficiency
Temperature control
HVAC

ABSTRACT

In this work, Deep Reinforcement Learning (DRL) is implemented to control the supply water temperature setpoint to terminal units of a heating system. The experiment was carried out for an office building in an integrated simulation environment. A sensitivity analysis is carried out on relevant hyperparameters to identify their optimal configuration. Moreover, two sets of input variables were considered for assessing their impact on the adaptability capabilities of the DRL controller. In this context a static and dynamic deployment of the DRL controller is performed. The trained control agent is tested for four different scenarios to determine its adaptability to the variation of forcing variables such as weather conditions, occupant presence patterns and different indoor temperature setpoint requirements. The performance of the agent is evaluated against a reference controller that implements a combination of rule-based and climatic-based logics. As a result, when the set of variables are adequately selected a heating energy saving ranging between 5 and 12% is obtained with an enhanced indoor temperature control with both static and dynamic deployment. Eventually the study proves that if the set of input variables are not carefully selected a dynamic deployment is strictly required for obtaining good performance.

Fig. A.1 Brandi S., Piscitelli M.S., Martellacci M., Capozzoli A. 2020. *Deep reinforcement learning to optimise indoor temperature control and heating energy consumption in buildings*. Energy and Buildings 224, 110225.

energies

MDPI

Article

Online Implementation of a Soft Actor-Critic Agent to Enhance Indoor Temperature Control and Energy Efficiency in Buildings

Davide Coraci, Silvio Brandi, Marco Savino Piscitelli and Alfonso Capozzoli *

TEBE Research Group, BAEDA Lab, Department of Energy "Galileo Ferraris", Politecnico di Torino, Corso Duca degli Abruzzi 24, 10129 Turin, Italy; davide.coraci@polito.it (D.C.); silvio.brandi@polito.it (S.B.); marco.piscitelli@polito.it (M.S.P.)

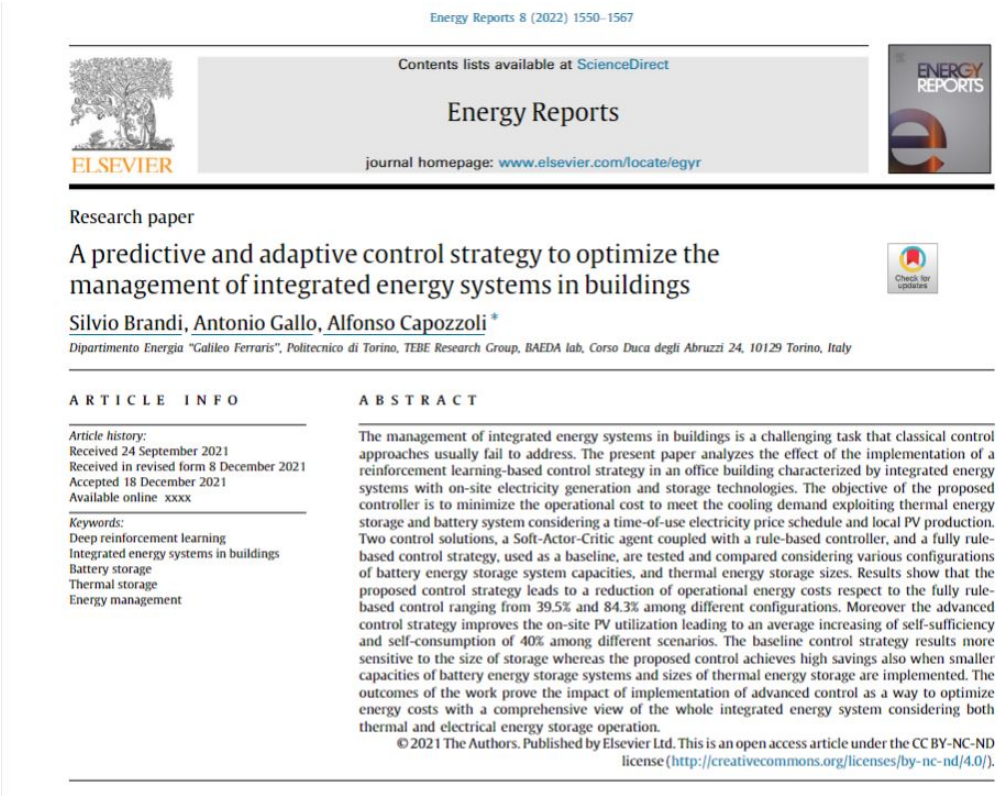
* Correspondence: alfonso.capozzoli@polito.it

Abstract: Recently, a growing interest has been observed in HVAC control systems based on Artificial Intelligence, to improve comfort conditions while avoiding unnecessary energy consumption. In this work, a model-free algorithm belonging to the Deep Reinforcement Learning (DRL) class, Soft Actor-Critic, was implemented to control the supply water temperature to radiant terminal units of a heating system serving an office building. The controller was trained online, and a preliminary sensitivity analysis on hyperparameters was performed to assess their influence on the agent performance. The DRL agent with the best performance was compared to a rule-based controller assumed as a baseline during a three-month heating season. The DRL controller outperformed the baseline after two weeks of deployment, with an overall performance improvement related to control of indoor temperature conditions. Moreover, the adaptability of the DRL agent was tested for various control scenarios, simulating changes of external weather conditions, indoor temperature setpoint, building envelope features and occupancy patterns. The agent dynamically deployed, despite a slight increase in energy consumption, led to an improvement of indoor temperature control, reducing the cumulative sum of temperature violations on average for all scenarios by 75% and 48% compared to the baseline and statically deployed agent respectively.

Keywords: automated system optimisation; building adaptive control; deep reinforcement learning; soft actor-critic; heating system

Citation: Coraci, D.; Brandi, S.; Piscitelli, M.S.; Capozzoli, A. Online Implementation of a Soft Actor-Critic Agent to Enhance Indoor Temperature Control and Energy Efficiency in Buildings. *Energies* **2021**, *14*, 997. https://doi.org/10.3390/en14040997

Fig. A.2 Coraci D., Brandi S., Piscitelli M.S., Capozzoli A. 2021. *Online Implementation of a Soft Actor-Critic Agent to Enhance Indoor Temperature Control and Energy Efficiency in Buildings*. Energies 14, 997.



Energy Reports 8 (2022) 1550–1567

Contents lists available at ScienceDirect

Energy Reports

journal homepage: www.elsevier.com/locate/egyr

Research paper

A predictive and adaptive control strategy to optimize the management of integrated energy systems in buildings

Silvio Brandi, Antonio Gallo, Alfonso Capozzoli *

Dipartimento Energia "Galileo Ferraris", Politecnico di Torino, TEBE Research Group, BAEDA lab, Corso Duca degli Abruzzi 24, 10129 Torino, Italy

ARTICLE INFO

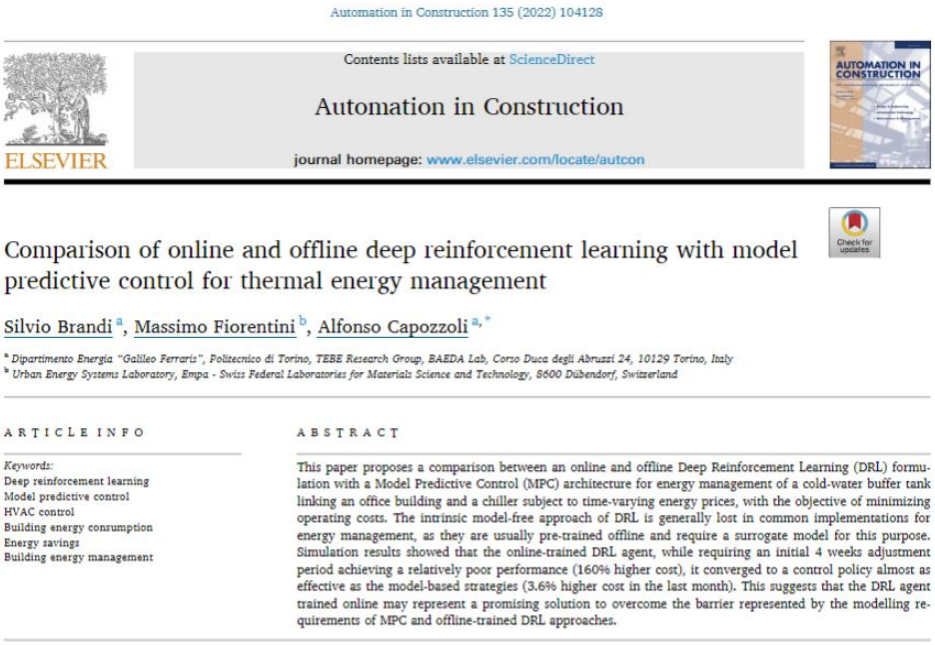
Article history:
Received 24 September 2021
Received in revised form 8 December 2021
Accepted 18 December 2021
Available online xxxx

Keywords:
Deep reinforcement learning
Integrated energy systems in buildings
Battery storage
Thermal storage
Energy management

ABSTRACT

The management of integrated energy systems in buildings is a challenging task that classical control approaches usually fail to address. The present paper analyzes the effect of the implementation of a reinforcement learning-based control strategy in an office building characterized by integrated energy systems with on-site electricity generation and storage technologies. The objective of the proposed controller is to minimize the operational cost to meet the cooling demand exploiting thermal energy storage and battery system considering a time-of-use electricity price schedule and local PV production. Two control solutions, a Soft-Actor-Critic agent coupled with a rule-based controller, and a fully rule-based control strategy, used as a baseline, are tested and compared considering various configurations of battery energy storage system capacities, and thermal energy storage sizes. Results show that the proposed control strategy leads to a reduction of operational energy costs respect to the fully rule-based control ranging from 39.5% and 84.3% among different configurations. Moreover the advanced control strategy improves the on-site PV utilization leading to an average increasing of self-sufficiency and self-consumption of 40% among different scenarios. The baseline control strategy results more sensitive to the size of storage whereas the proposed control achieves high savings also when smaller capacities of battery energy storage systems and sizes of thermal energy storage are implemented. The outcomes of the work prove the impact of implementation of advanced control as a way to optimize energy costs with a comprehensive view of the whole integrated energy system considering both thermal and electrical energy storage operation.

Fig. A.3 Brandi S., Gallo A., Capozzoli A. 2022. *A predictive and adaptive control strategy to optimize the management of integrated energy systems in buildings*. Energy Reports 8, pp: 1550-1567.



Automation in Construction 135 (2022) 104128

Contents lists available at ScienceDirect

Automation in Construction

journal homepage: www.elsevier.com/locate/autcon

Comparison of online and offline deep reinforcement learning with model predictive control for thermal energy management

Silvio Brandi [a], Massimo Fiorentini [b], Alfonso Capozzoli [a,*]

[a] Dipartimento Energia "Galileo Ferraris", Politecnico di Torino, TEBE Research Group, BAEDA Lab, Corso Duca degli Abruzzi 24, 10129 Torino, Italy
[b] Urban Energy Systems Laboratory, Empa - Swiss Federal Laboratories for Materials Science and Technology, 8600 Dübendorf, Switzerland

ARTICLE INFO

Keywords:
Deep reinforcement learning
Model predictive control
HVAC control
Building energy consumption
Energy savings
Building energy management

ABSTRACT

This paper proposes a comparison between an online and offline Deep Reinforcement Learning (DRL) formulation with a Model Predictive Control (MPC) architecture for energy management of a cold-water buffer tank linking an office building and a chiller subject to time-varying energy prices, with the objective of minimizing operating costs. The intrinsic model-free approach of DRL is generally lost in common implementations for energy management, as they are usually pre-trained offline and require a surrogate model for this purpose. Simulation results showed that the online-trained DRL agent, while requiring an initial 4 weeks adjustment period achieving a relatively poor performance (160% higher cost), it converged to a control policy almost as effective as the model-based strategies (3.6% higher cost in the last month). This suggests that the DRL agent trained online may represent a promising solution to overcome the barrier represented by the modelling requirements of MPC and offline-trained DRL approaches.

Fig. A.4 Brandi S., Fiorentini M., Capozzoli A. 2022. *Comparison of online and offline deep reinforcement learning with model predictive control for thermal energy management*. Automation in Construction 135, 104128.