Deep Reinforcement Learning for Optimal Energy Management in Smart Multi-Energy Systems

Dhekra Bousnina

PSL UNIVERSITÉ PARIS

THÈSE DE DOCTORAT
DE L'UNIVERSITÉ PSL

Préparée à Mines Paris - PSL

Deep Reinforcement Learning for Optimal Energy Management in Smart Multi-Energy Systems

Apprentissage par Renforcement Profond pour la Gestion Energétique Optimale dans les Systèmes Intelligents Multi-Energies

Soutenue par

Dhekra BOUSNINA

Le 15 décembre 2023

École doctorale n°84

Sciences et Technologies de l'Information et de la Communication

Spécialité

Contrôle, Optimisation, Prospective

Composition du jury :

Robert BELLINI Adjoint au directeur Adaptation au changement climatique, Aménagement et Trajectoires bas-carbone, ADEME	*Président*
Erwin FRANQUET Professeur, Université Côte d'Azur	*Rapporteur*
Safa BHAR LAYEB Professeure associée, HDR, Université Tunis El Manar	*Rapporteure*
Eric PEIRANO Directeur Général Adjoint en charge des quartiers bas carbone, HDR, Efficacity	*Examinateur*
Welington de OLIVEIRA Professeur Associé, HDR, Mines Paris-PSL	*Examinateur*
Gilles GUERASSIMOFF Professeur, Mines Paris-PSL	*Directeur de thèse*

MINES PARIS | PSL

Abstract

This PhD research work introduces an energy management approach for Smart Multi-Energy Systems (SMES) that leverages the power of Deep Reinforcement Learning (DRL) algorithms. We propose a Smart Energy Management System (SEMS) that is designed to optimize the management of flexible energy systems within SMES, including heating, cooling and electricity storage systems as well as District Heating and Cooling Systems (DHCS) such as district-level Thermo-Refrigerating Heat Pumps. The study focuses on applying the proposed approach on the Meridia Smart energy (MSE) case-study, a real-world demonstration project for SMES that is currently under construction within the Nice Meridia eco-district in southern France. MSE consists of an eco-district that encompasses 50 buildings, many of which are equipped with photovoltaic (PV) panels. The occupants will be supplied with heat and cold produced locally in the eco-district thanks to a geothermal Fourth Generation DHCS. In addition to local electricity, heating and cooling production, the SMES also integrates multi-energy storage systems, namely an innovative heat storage system by phase-changing materials, a cold storage by ice storage tanks and a battery storage offering additional flexibility.

The decision making problem is tackled using a DRL approach. The developed DRL framework is based on an actor-critic architecture and is benchmarked against Model Predictive Control (MPC), which is one of the most widely used methods for advanced process control in both industrial and academic level. Simulation results on a first case-study, a simplified simulation model drawn from MSE, demonstrate that the proposed DRL approach succeeds in learning a strategy that closely approximates the theoretical MPC optimum (within 98%) in terms of overall energy cost reduction. Notably, it even outperformed some MPC variants with realistic forecasts. This study represents one of the

initial attempts in the literature to simultaneously benchmark DRL and MPC approaches for multi-energy management in SMES case-studies and suggests that the DRL approach holds promise for energy and cost-efficient and sustainable management of SMES.

The proposed DRL framework is further applied on a second more complex case-study where a digital twin has been developed for the MSE eco-district under Dymola (Modelica language). This digital twin, exported as a Functional Mock-up Unit (FMU) and wrapped as an Open AI Gym environment serves as training, validation and testing environment for the DRL agent. Simulation results of applying the DRL approach on the MSE digital twin re-affirm the findings from the first case-study and showcased that DRL is a promising approach to address the problem of optimal energy management in SMES.

Future works will involve transitioning the DRL framework from simulation to the real-world systems of the MSE project and exploring additional use-cases and optimization objectives such as collective self-consumption and participation in frequency regulation markets and demand response mechanisms, and expanding the scope of the considered SMES to involve the management of other multi-energy systems including electric vehicles, charging stations, public lighting as well as controllable building devices.

Résumé

Cette thèse introduit une approche de gestion de l'énergie pour les Systèmes Multi-Energies Intelligents (SMEI) basée sur des algorithmes d'Apprentissage par Renforcement Profond (Deep Reinforcement Learning, DRL). Nous proposons un Système de Gestion Multi-Énergies Intelligente (SGMEI) conçu pour optimiser la gestion des systèmes d'énergie flexibles au sein des SMEI, notamment les systèmes de stockage de chaleur, de froid et d'électricité, ainsi que les systèmes de production dans les réseaux de chaleur et de froid tels que les Thermo-Frigo Pompes (TFPs).

Cette étude vise l'application de l'approche proposée sur l'étude de cas Meridia Smart Energie (MSE), un projet démonstrateur pour les SMEI, actuellement en construction dans l'écoquartier de Nice Meridia, dans le sud de la France. Le projet MSE englobe un écoquartier composé d'environ 50 bâtiments, dont plusieurs sont équipés de panneaux photovoltaïques. Les occupants de cet écoquartier seront alimentés par de la chaleur et du froid produits localement grâce à un ensemble de TFPs géothermiques et acheminés aux bâtiments grâce à un réseau de chaleur et de froid de quatrième génération. En plus de la production locale d'électricité, de chaud et de froid, le SMEI de MSE intègre également des systèmes de stockage multi-énergies, notamment un système innovant de stockage de chaleur par matériaux à changement de phase, un stockage de froid par des bacs de glace et un stockage électrique par batterie. Le problème de gestion optimisée est abordé en utilisant une approche basée sur des algorithmes de DRL. Les algorithmes développés reposent sur une architecture dite acteur-critique et sont benchmarkés avec une approche basée sur du Contrôle Prédictif (Model Predictive Control, MPC), l'une des techniques les plus largement utilisées pour le contrôle avancé des processus, tant dans l'industrie que dans le milieu académique. Les résultats de simulation sur le premier cas d'étude,

un modèle de simulation pour un SMEI simplifié inspiré de MSE, démontrent que l'agent DRL développé réussit à apprendre une stratégie qui s'approche étroitement de l'optimum théorique obtenu par le contrôleur MPC (à hauteur de 98%) en termes de réduction des coûts énergétiques globaux dans le SMEI. À noter que la performance de l'agent DRL a même surpassé certaines variantes du MPC qui ont été alimentées par des prévisions réalistes. Cette étude représente l'une des premières tentatives dans la littérature qui visent à comparer simultanément les approches DRL et MPC pour la gestion multi-énergie optimisée sur des cas d'étude de SMEI, suggérant que l'approche DRL est prometteuse pour la gestion énergétique et économique durable des SMEI.

L'approche DRL proposée est ensuite appliquée à un second cas d'étude, pour lequel un jumeau numérique a été développé sous Dymola (langage Modelica) pour l'écoquartier MSE. Ce jumeau numérique est exporté en tant qu'Unité de Modélisation Fonctionnelle (Functional Mock-up Unit, FMU) et encapsulé en tant qu'environnement Open AI Gym, servant ainsi d'environnement d'entraînement, de validation et de test pour l'agent DRL. Les résultats de simulation de l'application de l'approche DRL sur le jumeau numérique de MSE valident les conclusions de l'étude de cas 1, confirmant que le DRL est une approche prometteuse pour la résolution du problème de la gestion optimisée multi-énergies dans les SMEI.

Les travaux futurs consisteront à assurer la transition des algorithmes DRL développés des modèles de simulation vers les systèmes réels de MSE et à explorer des cas d'usage et des objectifs d'optimisation supplémentaires, tels que l'autoconsommation collective au sein de l'écoquartier et la participation aux mécanismes d'effacement et aux marchés de réglage de fréquence, ainsi que l'extension du SMEI pour inclure la gestion de systèmes multi-énergies supplémentaires, tels que les véhicules électriques, les stations de recharge, l'éclairage public ainsi que les charges pilotables dans les bâtiments.

To my parents, my husband, my sister, my brother and all my loved ones,

To the cherished memory of our dear colleague at Idex Timothy Leab, whose contributions to this project will forever be remembered,

Acknowledgments

I would like to begin by expressing my deepest gratitude to the members of the jury for agreeing to examine and evaluate this work: I first sincerely thank Dr. Robert BELLINI for agreeing to be the president of this jury. I am also deeply grateful to Pr. Erwin FRANQUET and Dr. Safa BHAR LAYEB for agreeing to review this manuscript and for dedicating their time to provide their comments and feedback on my work. I would also like to sincerely thank Dr. Eric PEIRANO for kindly agreeing to be a member of this jury and I greatly appreciate his thorough examination of my work. I would like furthermore to express my deepest gratitude to Dr. Welington de OLIVEIRA for agreeing to be a member of this jury and for his continuous guidance and support over the years that dates back to my Energy Systems Optimization (OSE) master's internship in 2018, when all this incredible journey in energy systems optimization began.

This journey in energy systems optimization started six years ago when I first met Pr. Nadia MAÏZI and Pr. Gilles GUERASSIMOFF while applying for the specialized master's degree OSE. I cannot be thankful enough to both of them for their belief in me at that time and for the great opportunity they provided me. Their constant trust, support and guidance not only pushed me forward through this exciting path of energy systems optimization but also helped me uncover the captivating world within it.
Following my OSE master's adventure, Pr. Gilles GUERASSIMOFF granted me the opportunity to pursue my doctoral journey and I wish to extend my profound gratitude to him for making this journey all the most enriching and fulfilling. His constant support and guidance, and his invaluable advice have been the cornerstone of my success. Our meetings and discussions have always been not only scientifically enriching but also per-

sonally inspiring to me. His trust in me was the driving force that continually pushed me to meet his expectations. I just could not have hoped for a better supervisor, and I honestly feel exceptionally fortunate to have had his mentorship throughout this journey.

This work would not have been possible without the financial support of the ADEME, which funded the Meridia Smart Energy Project (MSE) via the Program Investissements d'avenir that they operate. I would like to express my profound gratitude to all the people who contributed to the approval process of this project and gave me the opportunity to pursue this work, in particular to Patricia SIDAT who has been overseeing this project from its inception.

This work would also not have been possible without the collaboration with the Idex team who warmly welcomed me during the first two years of my doctoral journey. I am profoundly thankful to Driss SALMI for initiating this collaboration and for the numerous enriching discussions we've had throughout this project. I also would like to thank Jacques Brunet for providing me the smoothest integration and the warmest welcome into his team at Idex. It is with a heavy heart that I also dedicate this work to the memory of Timothy LEAB with whom I shared an office at Idex during the first two years of this research. His unwavering positivity and energy made the journey all the more pleasant.

My deepest appreciation also extends to all the members the MSE project who contributed their expertise and effort to ensure the success of this project. I would like to mention in particular Somchay NORINDR, Eduard MALDONADO, Bertrand DESPRES, Chloé ZHANG, Vincent DE GRIEVE, Laurent PARODI and Jonathan GAGNOR from Idex, Olivier SORIANO, Fabrice BENTIVOGLIO and Jean-François FOURMIGUE from the CEA and Ludivine MUNTZER and Régis MARTIN from the Metrople Nice Cote d'Azur.

Throughout this doctoral journey, my affiliation with the CMA research center of Mines Paris-PSL has been an absolute privilege. It is a place where every individual's kindness and brilliance harmoniously blend to create a unique and inspiring working atmosphere:

Within the CMA team, I first would like to express my heartfelt gratitude to Jean-Paul MARMORAT for his timely support, assistance and feed-back.
I would also like to deeply thank Damien CORRAL, our data science expert and Python guru, for his vital help during this work and for all the enriching and passionate discussions that we had on reinforcement learning, and in advance for all those that we will have in the future.

I am also particularly thankful to Valérie ROY and Valentina SESSA for their guidance, support and very warm encouragements especially during all the thesis monitoring committees. I'm also thankful to the researchers at the CMA Sophie DEMASSEY, Sandrine SELOSSE and Edi ASSOUMOU for their kindness and for being a continuous source of inspiration to me.

During my time at the CMA, I also had the great pleasure of co-supervising four bright and dedicated OSE interns alongside Gilles. They played a pivotal role in the development of the Dymola digital twin within this project. I would like to acknowledge their significant contributions and the valuable insights I've gained while working with each of them: thank you Younes BAGHDAD, Kim PERRIGUEY, Imaane KHOYRATEE and Axel RICHET.

I'm also deeply appreciative of the administrative team at the CMA who played a vital role in creating a fantastic working atmosphere: special thanks to Alice, Amel, Claire, Cédric, Sébastien and Sabrina, whose kind help and support has been invaluable at every phase of this journey.

I am very grateful for the friendships I've cultivated with my fellow doctoral candidates and postdoctoral researchers at the CMA: Naima, you've been such an incredible friend and neighbor to me, and your support, love and positivity have been a constant source of encouragement for me over the past years. To all the PhD candidates and my friends with whom I shared my office and incredible moments: Rabab, Gregorio, Sophie, Victor, Amir, Lucas, Thibaut, Zixuan, Xsenia, Charlène, Marie and Cindy, thank you all for your kindness and support. Many special thanks also go to Gildas and Louis.

As a wise person once said that Parents are the bones on which children cut their teeth, it goes without saying that I am eternally grateful to my Parents to whom I owe everything and who I am. It is with immense gratitude and a profound sense of indebtedness that I acknowledge the invaluable contribution of my parents, my sister and my brother without whom I would not have been able to come this far.

Last but not least, I would like to deeply thank my husband for his constant support and encouragement throughout this doctoral journey. His boundless love, not only as a life partner but also as a scientific mentor, has been an invaluable source of strength. Having successfully completed his own PhD, he knew how to give me insightful advice and guidance and helped me navigate the challenges of my research. I am truly blessed to have such a wonderful husband who believes in my dreams and shares in my passion for knowledge. Thank you for being my confidant, my source of inspiration and above all my greatest supporter.

Contents

List of Figures

List of Tables

List of Algorithms

List of acronyms

1GDCHS	First Generation District Heating and Cooling System
2GDHCS	Second Generation District Heating and Cooling System
3GDHCS	Third Generation District Heating and Cooling System
4GDHCS	Fourth Generation District Heating and Cooling System
5GDHCS	Fifth Generation District Heating and Cooling System
BESS	Battery Energy Storage System
CAES	Compressed Air Energy Storage
DCS	District Cooling System
DDPG	Deep Deterministic Policy Gradient
DER	Distributed Energy Resources
DHS	District Heating System
DHCS	District Heating and Cooling System
DHW	Domestic Hot Water
DL	Deep Learning
DP	Dynamic Programming
DPG	Deep Policy Gradient
DQN	Deep Q-Networks
DQL	Deep Q-Learning
DR	Demand Response
DRL	Deep Reinforcement Learning
DSM	Demand Side Management
EMS	Energy Management System
ESS	Energy Storage System
EV	Electric Vehicle

FMI	Functional Mock-up Interface
FMU	Functional Mock-up Unit
GA	Genetic Algorithm
GHG	Greenhouse Gases
ICES	Integrated Community Energy Systems
IES	Integrated Energy Systems
LP	Linear Programming
MDP	Markov Decision Process
MG	Micro-Grid
MILP	Mixed Integer Linear Programming
ML	Machine Learning
MPC	Model Predictive Control
MSE	Meridia Smart Energy
NLP	Natural Language Processing
PCM	Phase Change Materials
PPO	Proximal Policy Optimization
PSO	Particle Swarm Optimization
PV	Photo-voltaic
RES	Renewable Energy Sources
RL	Reinforcement Learning
SAC	Soft Actor Critic
SDCS	Smart District Cooling System
SDHS	Smart District Heating System
SES	Smart Energy Systems
SG	Smart Grid
SMEMS	Smart Multi Energy Management System
SMES	Smart Multi Energy Systems
SoC	State of Charge
STG	Smart Thermal Grid
TD	Temporal Difference
TD3	Twin-Delayed DDPG
TESS	Thermal Energy Storage System
TRHP	Thermo-Refrigerating Heat Pump
WSHP	Water Source Heat Pump

General Introduction

Context

Within the radical changes that the energy landscape is currently undergoing, Smart Grids (SG) are playing a major role in the modernization of the electrical grid [1]. These smart electricity networks have the great advantage of integrating in a cost-effective way the behavior and actions of all the users connected to them, including consumers, producers and prosumers, to ensure a cost-efficient and sustainable operation of the power system while guaranteeing quality and security of supply [2]. Besides electrical networks, District Heating and Cooling Systems (DHCS) also play a paramount role in the implementation of the new Smart Energy Systems (SES) [3]. In fact, the recently emerging concept of Smart Thermal Grids (STG) also comes up with numerous advantages including flexibility potentials and ability to adapt to the changes that affect the thermal demand and supply in short, medium and long terms. Thus, smart thermal grids, as well, are expected to be an integrated part of the future energy system [4], [5].

However, research works on the optimal control and energy management within the smart grid context traditionally focus solely on the electrical usages. Though, jointly optimizing the electrical networks together with other energy vectors interacting with them like heating and cooling networks has a great potential to increase the overall economic and environmental efficiency and flexibility of the energy systems. This idea leads to a generalization of the smart grid concept towards Smart Multi Energy Grids (SMEG) [6] and Smart (Multi-Energy) Systems [7] that lie on the interaction between electricity and other energy vectors including heat, cold, gas and hydrogen as well as other sectors that electricity might interact with like the transport sector and water networks. Such multi-energy

systems can enable the flexible utilization of various energy vectors while incorporating storage systems, manageable loads and other flexibility potentials, based on a multitude of criteria such as energy efficiency, reliability, costs and emissions. Considering these interactions in the optimal management of energy systems allows to unlock considerable efficiency and flexibility potentials and represents one of the main advantages of smart (multi) energy systems.

Optimal control of smart (multi) energy systems is essential to guarantee a reliable operation for their flexible components and ensure an optimal management of controllable loads, production units and storage systems while minimizing energy and operational costs [8]. One of the most popular and widely used optimal control techniques is Model Predictive Control (MPC), also referred to as Receding Horizon Control [9], [10]. MPC is a feedback control method where the optimal control problem is solved at each time step to determine a sequence of control actions over a fixed time horizon. Only the first control actions of this sequence are then applied on the system and the resulting system state is measured. At the next time step, the time horizon is moved one step forward and a new optimization problem is then solved, taking into account the new system state and updated forecasts of future quantities. This receding time horizon and periodic adjustment of the control actions make the MPC robust against the uncertainties inherent to the model and forecasts [11]. MPC has been used in many successful applications in the field of the energy management in smart grids and smart districts including [12]–[15].

Nevertheless, MPC and model-based approaches in general, rely on the development of accurate models and predictors and on the usage of appropriate solvers. This does not only require domain expertise but also involves a need to re-design these components each time that a change occurs on the architecture or scale of the smart energy system [16]. Furthermore, classical optimization approaches based on Mixed Integer Linear Programming (MILP), Dynamic Programming (DP) or heuristic methods like Particle Swarm Optimization (PSO) generally suffer from time-consuming procedures. In fact, they have to compute all or part of possible solutions in order to choose the optimal one, and have to re-run a generally time-consuming optimization procedure each time that an optimal decision needs be taken. Therefore, such methods, despite their ability to provide quite accurate results, generally fail to consider on-line solutions for large-scale real databases [17].

Learning-based techniques, on the other hand, do not require accurate system models and uncertainty predictors and can, thus, be an alternative to model-based approaches. Reinforcement Learning (RL) [18] is a learning-based method that has been gaining popularity over the past few years when it comes to dealing with challenging sequential decision making tasks [19]. It consists of a learning paradigm in which an agent learns to control a system by interacting with its environment and receiving feedback in the form of a numerical reward. The learning agent takes a sequence of actions over time to maximize its cumulative reward and it learns an optimal strategy over time through trial and error by observing the consequences of its actions on the environment. The RL paradigm has initially been developed in the context of control theory and operations research in the 1950s and 1960s with early research works like those of Richard Bellman [20] who laid the theoretical foundations for RL by developing dynamic programming and the Bellman equation, and Arthur Samuel [21] who developed one of the earliest self-learning programs. More recently, Richard Sutton and Andrew Barto made significant contributions to the theoretical and practical advancements of RL, and their influential book [18] became a foundational resource for researchers and students interested in this field.

RL has actually been around for several decades but its practical applications on real life and complex decision making problems remained limited. Indeed, RL-based approaches fail to handle large state and action spaces owing to the curse of dimensionality [22]: as the number of states and actions to be handled by the RL agent increases, its computational and memory requirements grow exponentially, making it impractical for many real-world problems. This major historical limitation of RL largely restricted its broader adoption and effectiveness. Around 2013, a transformative breakthrough emerged and led to a revival of interest in the RL field thanks to the work of Mnih et al. [23] who successfully combined RL with Deep Learning (DL), giving birth to Deep Reinforcement Learning (DRL). DRL combines the strong nonlinear perceptual capability of Deep Neural Networks (DNNs) with the robust decision making ability of RL [24]. This way, DNN enabled RL agents to efficiently represent and approximate complex value functions and thus allowed them to effectively learn from high dimensional state spaces. Unlike RL, DRL exhibits strong generalization capabilities in problems with complex state spaces. In 2015, the authors of Mnih et al. [23], in collaboration with researchers from DeepMind published a paper in the journal Nature [25] where they demonstrated that DRL agents

could achieve a superhuman performance on a range of Atari games. This work showcased the potential of DRL for complex decision-making tasks and has, as a consequence, revitalized the field of RL and opened the doors to numerous successful applications of DRL on challenging decision making problems. One of the main advantages of DRL compared to other classical optimization approaches is that, once it learned an optimal strategy, it can take optimal decisions in a few milliseconds without having to re-compute any costly optimization procedure. This makes DRL algorithms less time-consuming than classical optimization approaches and makes them, as a consequence, more suitable for real-time optimization problems. What's more, DRL algorithms can succeed in learning optimal strategies directly from sensory inputs like images or sensor data without requiring accurate models and predictors, handcrafted features or appropriate solvers. This end-to-end learning approach makes DRL applicable for a wide range of control tasks without requiring domain expertise.

DRL has, this way, shown successful applications in various real-life problems with large state spaces like Atari and Go games [26], robotics [27], [28], autonomous driving [29], [30] and other complex control tasks [25]. More recently, [31] proposed a novel assembling methodology of Q-learning agents -a type of RL agents- trained several times with the same training data for stock market forecasting. The use of DRL aimed at avoiding problems that may occur when using supervised learning-based classifiers like overfitting. Other recent successful applications of DRL include intrusion detection systems as presented in [32]. Furthermore, [33] proposed a new ensemble DRL model for predicting wind speed and the comparison of the proposed model with nineteen alternative mainstream forecasting models showed that the DRL-based approach provided the best accuracy. Moreover, Google has announced in 2018 that it gave control over the cooling of several of its data centers to a DRL algorithm [34].

DRL has also recently gained a widespread recognition in Natural Language Processing (NLP) and conversational AI thanks to increasingly popular applications like Open AI's ChatGPT [35] that uses DRL techniques to fine-tune its language generation capabilities and improve the quality and relevance of its text responses in real-time conversations [36].

Thesis objectives and contributions

This PhD research work focuses on the application of DRL for the optimal energy management problem in smart (multi) energy systems. This work is part of the Merdia Smart energy (MSE) eco-district, which is a demonstrator project for smart energy systems under construction in the city of Nice, south of France, since 2019. MSE is a smart energy system that involves a Fourth Generation District Heating and Cooling system (4GDHCS) that will provide the 50 buildings ($\simeq$ 3500 households) of the eco-district with heat and cold. These buildings will also be equipped with photovoltaic (PV) panels on their rooftops. The MSE smart energy system also includes multi-energy storage systems namely an electricity storage through a Battery Energy storage system (BESS), an innovative heat storage through Phase-Change Materials (PCM) and a cold storage by an ice on coil technology. A consortium has been created around the MSE project. It is composed of four entities: the *Métropole Nice Côte d'Azur*, public service delegate, the *Idex* company, energy service provider of the eco-district, as well as two research laboratories, namely the *Center for Atomic Energy and Alternative Energies (CEA)* who developed, designed and constructed the innovative phase-change material heat storage system, and the *Center for Applied Mathematics (CMA) of Mines Paris* who is in charge of developing the optimal energy management algorithms of the multi-energy storage systems of the MSE smart energy system, through the present thesis. This consortium received funding from the *ADEME, the french Agency for Ecological Transition* [1] through the Program *Investissements d'Avenir* [2] that they operate.

The aim of this thesis is therefore to develop an energy management system for smart multi-energy systems starting from the MSE case-study. This energy management system ensures the optimized management of the flexible energy systems, particularly the three storage systems, with the aim of minimizing the overall energy consumption costs within the smart energy system. The developed solution has to be adaptable to achieve other potential optimization objectives such as maximizing the self-consumption and energy self-sufficiency of the eco-district or minimizing its Greenhouse Gas (GHG) emissions. The solution should also be reproducible in order to be deployable in any smart energy

[1] ADEME, the French Agency for Ecological Transition, https://www.ademe.fr/en/frontpage/

[2] French government, Le Programme d'Investissements d'Avenir, https://www.gouvernement.fr/le-programme-d-investissements-d-avenir

system similar to MSE.

To address the optimal energy management problem, we propose a DRL-based approach. More specifically, we formulated the optimal control problem as a Markov Decision Process (MDP) and developed a DRL agent to perform real-time scheduling of the multi-energy storage systems within the considered smart energy system. The DRL agent that we developed is based on an actor-critic algorithm called Deep Deterministic policy Gradient (DDPG). One of the reasons behind this choice is the ability of these algorithms to deal with continuous action spaces. We tested this approach on two simulated smart energy system case studies, both drawn from the MSE real-world project. The first case-study, denoted *case study 1* is a simplified simulation model developed under Python, where the DRL-based framework was tested and benchmarked against an MPC-based framework that we developed for this case-study. In the second case study, *case study 2*, we developed a digital twin -a detailed simulation model- for the MSE smart energy system under the Modelica language to account for the dynamics of the different energy systems that it involves. We then tested the developed DRL-based framework for the cost-optimal energy management of the multi-energy storage systems of this digital twin. Thus, the main contributions of this research work are the following:

- **DRL for the optimal operation of smart multi-energy systems:** We developed a Deep Reinforcement Learning-based framework for the optimal energy management in smart energy systems. Unlike most of the previous works where mono-fluid (electrical or thermal) Smart Grids are considered, we focus in this work on simultaneously managing multi-energy (electrical, heating, cooling) smart grids that interact with the main utility grid. A variable electricity price signal is considered and a DRL-based energy management system is developed to take price-responsive control actions. This extends the applicability of DRL-based approaches to more complex and interconnected energy systems.

- **A DDPG algorithm for multiple continuous actions:** We propose the use of an actor-critic (DDPG) algorithm instead of the most commonly used value-based algorithms (mainly Deep Q-Learning (DQL) algorithms). At each time step of the control horizon, multiple continuous actions are simultaneously taken by the DDPG agent to optimally schedule the various storage systems as well as the thermal production units. The DDPG algorithm is actually capable to deal with the continuous

action and state spaces that are inherent to the smart energy system model. Discretizing the action space, as would be required by a DQL approach, would result in a loss of precision. Avoiding this precision loss would require a fine discretization, which would lead the agent to explore more actions and hence increase the complexity of the problem.

- **Benchmark against an MPC controller:** The proposed DRL-based approach is tested on two different case studies, both drawn from the MSE smart energy system real-world use case. In *case study 1*, we develop a simulation model under Python where the dynamics of the different energy systems involved in the smart energy system are considered in a simplified way. The cost-optimal energy management problem of the three storage systems is formulated as a Markov Decision Process (MDP) and is solved using DRL. The same problem is then converted into a linear Programming (LP) problem and is embedded into a Model Predictive Control (MPC) framework. The DRL based approach is then benchmarked with the MPC based approach through this case study. Simulation results showed a close performance of the DRL agent to a perfect-forecast MPC controller and therefore suggest that DRL is a promising approach for dealing with optimal energy management problems in smart multi-energy systems. To the best of our knowledge, this work represents one of the first studies that simultaneously benchmark DRL and MPC on a smart multi-energy system case study. Indeed, the only work in literature considering this subject is the paper of Ceusters et al. [37] which was conducted simultaneously to our research work. In consistency with the conclusions of this paper, our work also showcased that DRL performs close to perfect foresight MPC and can even outperform MPC with realistic forecasts. It accordingly suggests that DRL is a promising approach for the optimal energy management in smart energy systems featuring multi-energy storage systems.

- **Reward shaping to address sparse reward issues:** the authors in [37] stated that their DRL agent showed difficulties in managing the electrical and thermal storage systems as this involves a short-term penalty (when charging) in order to achieve mid-term reward. They suggested investigating, in future works, whether these difficulties are due to the hyper parameters used, the implemented objective, the

choice of PPO (Proximal Policy Optimization) as a DRL algorithm, or the use of the RL-based approach itself. We think that the difficulties faced by the PPO agent may stem either from the use of PPO or from *the sparse reward* problem, a challenging issue faced by RL agents in situations where the agent does not receive enough reward feedback from the environment, typically when the actions that it selects do not yield any reward. In our work, we propose shaping the reward signal in a way that ensures a more effective learning and guarantees avoiding the sparse reward problem. Our simulation results show that the DDPG agent succeeds, after training, in efficiently operating the three storage systems.

- **Application on a Modelica digital twin developed for the real-world MSE smart energy system:** the proposed DRL-based framework is also applied on *case study* 2 where a digital twin of the MSE smart energy system is developed under Dymola (Modelica language) to better account for the dynamics of its energy systems. The Dymola digital twin is then exported as a Functional Mock-up Unit (FMU) using the Functional Mock-up Interface (FMI) standard that allows models from different modeling tools to be integrated and co-simulated. The FMU is then integrated into the developed Python framework and interacts with the DRL agent after being wrapped into an OpenAI Gym [38] environment. Simulation results showed that the DRL agent also succeeds in efficiently operating the three energy storage systems in the more complex environment of this case study. Nonetheless, most of the hyper-parameters we used for this case study are the same as for the simpler case study 1. This suggests, in consistence with the work of Ceusters et al. [37] that the hyper-parameters are not totally environment-specific but are more likely to be task-specific. Future work includes transitioning the developed DRL-based framework from simulation to real-world application on the MSE smart energy system project. It will be deployed, as a first instance, as a decision-support tool for the optimal operation of the three storage systems and the heat and cold power plant (mainly through load shedding). In the future, the developed DDPG agent will be extended to operating real-time energy management of the various energy systems within the eco-district including the storage systems, district heating and cooling production units, controllable loads of the buildings, heated water storage tanks, electric vehicle charging stations and the public lighting of the district.

Thesis outline

The remainder of this manuscript is divided into two parts that are organized as follows, as illustrated in figure 1:

- **Part I** presents an overview on smart multi-energy systems as well as the optimal control methods applied for their optimal operation. It then focuses on the two optimal control approaches on which we dwell in this work, namely Deep Reinforcement Learning and Model Predictive Control. These two techniques are reviewed and then simultaneously benchmarked on the same simulated smart energy system case study 1.

  - **Chapter 1** introduces and defines key concepts around smart multi energy systems, including smart (electrical) grids, District Heating and Cooling Systems (DHCS), smart thermal grids and their integration into the larger cross-sectoral concepts of multi-energy smart grids, integrated energy systems and smart multi-energy systems.
  - **Chapter 2** focuses on the optimal energy management in smart grids, DHCS and smart multi-energy systems and reviews the optimization methods used within these energy management systems.
  - **Chapter 3** proposes an in depth focus on the reinforcement learning paradigm and its combination with deep learning by reviewing its theory and reporting previous works proposing its application for energy systems.
  - **Chapter 4** sheds light on the model predictive control strategy and reviews its previous applications in energy systems. Differences and similarities between MPC and DRL are then highlighted.
  - **Chapter 5** proposes benchmarking MPC and DRL based approaches through their simultaneous application for the optimal energy management in smart multi-energy systems through simulated case study 1.

- **Part II** is devoted to the application of the proposed DRL-based approach on the Meridia Smart Energy case study project that represents the context in which this research work was conducted.

– **Chapter 6** describes the MSE eco-district, the multi-energy systems that it involves as well as the strategic objectives that mostly drive the energy management systems that we developed for these systems.

– **Chapter 7** presents the Dymola simulation model that we developed for the MSE eco-district and details the overall modeling approach that we adopted.

– **Chapter 8** presents the application of the DRL-based approach proposed in this work, on the developed MSE digital twin.

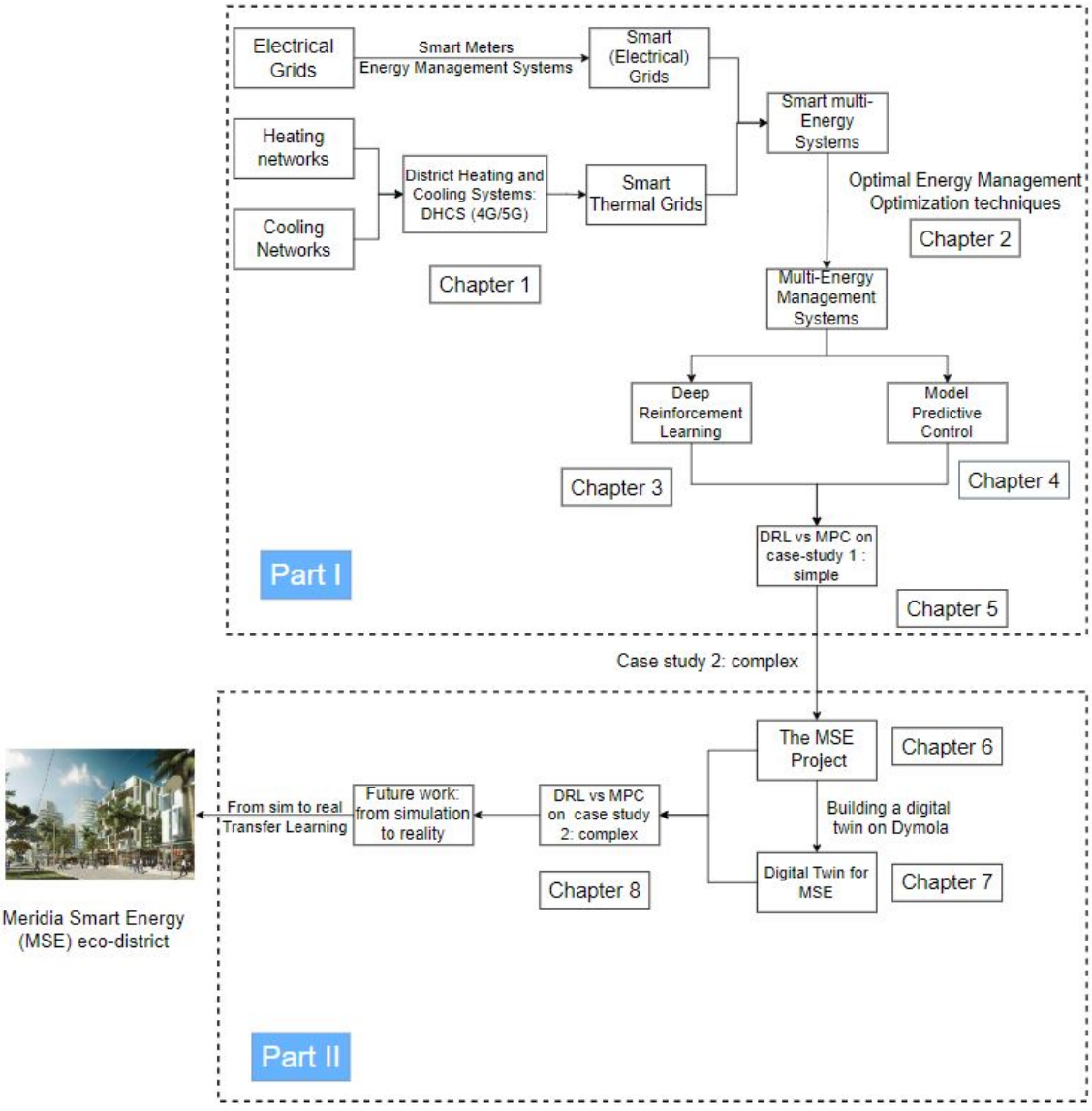


Figure 1: Plan of the manuscript.

Optimal energy management in Smart Multi-Energy Systems: benchmarking DRL and MPC based approaches

Introduction of Part I

This first part of the dissertation provides an insightful overview of smart multi-energy systems by defining their key concepts and reviewing the array of optimal control methods that can be applied for their optimal operation. A focus is then made on two particular approaches, namely deep reinforcement learning and model predictive control by reviewing their principles and methodologies and reporting previous works applying them for the optimal energy management in energy systems. These two techniques are then simultaneously benchmarked through their application on case study 1: a simulated smart multi-energy system that is drawn from the MSE project.

Chapter 1

Smart Energy Systems: an overview

Résumé

Ce premier chapitre introduit et définit les concepts clés qui sous-tendent ce travail de recherche. On commence par définir les concepts de réseaux électriques intelligents, communément appelés smart grids, ainsi que leurs composantes clés telles que les ressources énergétiques distribuées, les compteurs intelligents et les systèmes de gestion de l'énergie. Ensuite, on explore les analogues thermiques de ces réseaux, à savoir les réseaux thermiques intelligents ou smart grids thermiques qui émergent avec les réseaux de chaleur et de froid de quatrième génération. Enfin, l'intégration de ces deux notions au sein de cadres plus vastes et intersectoriels favorisant les synergies entre les différents vecteurs énergétiques est passée en revue, en introduisant les concepts de réseaux intelligents multi-énergies, de systèmes énergétiques intégrés et en aboutissant aux systèmes multi-énergies intelligents.

1.1 Introduction

In this first chapter, we introduce and define key energy systems' concepts around which the present research work is structured. We first define the concept of Smart Electrical Grids together with their key components like Distributed Energy Resources, smart meters, information and communication technologies, and Energy Management Systems. Then, a focus is made on the thermal energy systems by defining District Heating and Cooling Systems (DHCS) and their classification into five different generations as well as the new concept of Smart Thermal Grids that emerges particularly within the fourth

and fifth generations. Finally, we review the integration of these two concepts, namely Smart Electrical Grids and Smart Thermal Grids, into the larger cross-sectoral concepts of Multi-Energy Smart Grids, Smart Energy Hubs, Integrated Energy Systems and Smart Multi-Energy Systems.

1.2 Smart Electrical Grids

1.2.1 The concept

The radical changes that the energy landscape is currently undergoing led to an increasing share of renewable energy generation and a growing use of Distributed Energy Resources and energy storage systems. These changes have also brought about the adoption of new technologies mainly for the power and energy management of these resources in order to maintain the main objectives of the energy system: economical and environmental efficiencies and security of supply. All these aspects contributed to the emergence of Smart Grids [39], also known as smart electrical grids or smart power grids and sometimes also referred to as intelligent grids, intelligrids, futuregirds, intergrids or intragrids [1]. The concept of Smart Grids does not have one unique universally accepted definition [2]. For instance, the Trans-European Networks – Energy (TEN-E) Regulation defines a Smart Grid as "an electricity network that can integrate in a cost efficient manner the behaviour and actions of all users connected to it, including generators, consumers and those that both generate and consume, in order to ensure an economically efficient and sustainable power system with low losses and high levels of quality, security of supply and safety" [40]. Similarly to this definition, the European Technology Platform [41] defines it as "an electricity network that can intelligently integrate the actions of all users connected to it – generators, consumers and those that do both in order to efficiently deliver sustainable, economic and secure electricity supplies". Different definitions of Smart Grids can also be found in [42]. Overall, the main components of a Smart Grid that differentiate it from a traditional grid are information and communication technologies, smart meters, Energy Management Systems (EMS), Distributed Energy Resources (DER), Demand Side Management (DSM) and Smart users [43] that will be discussed in next sections. For further details on the definitions of a Smart Grid, its various manifestations, its potential benefits and challenges as well as a brief review of the developments and research directions in the context of Smart Grids, we refer the interested reader to the work of El-Hawary [44].

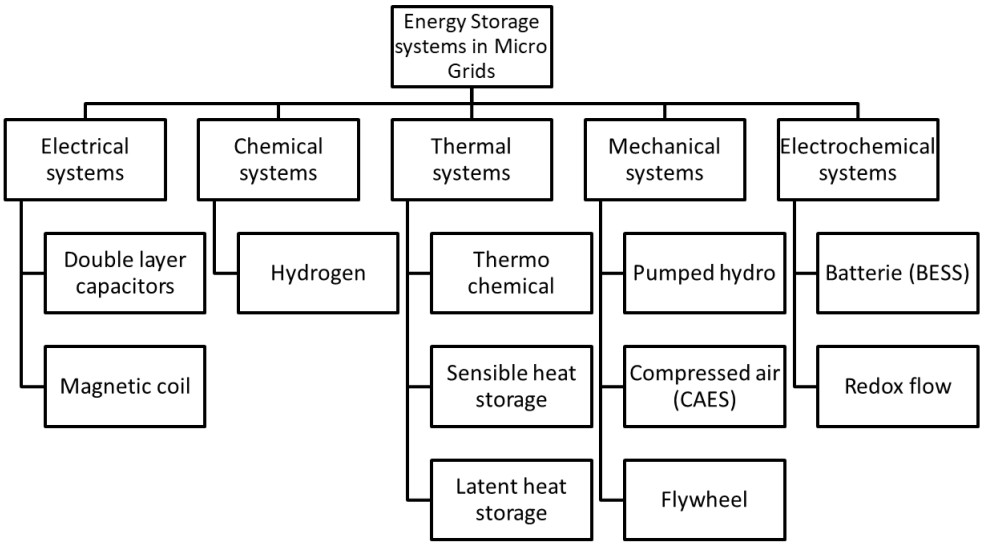


Figure 1.1: Types of energy storage systems in microgrids and Smart Grids (adapted from [47]).

1.2.2 Distributed Energy Resources

Distributed Energy Resources (DER) represent one of the main components of a Smart Grid. They can be defined as "small power sources that can help to meet regular power demand", according to ([2]). They include renewable energy generation, mainly wind generators and photo-voltaic generators, as well as fuel cells, gas turbines, micro turbines and internal combustion engines [45], [46] and energy storage technologies like Battery Energy Storage Systems (BESS). Distributed generation in Smart Grids can also include thermal generation (on which we focus in section 1.3) and Electric Vehicles (see paragraph 1.2.5).

Among Distributed Energy Resources, energy storage systems are regarded as a promising solution for many of the challenges that come out with the Smart Gird concept, in particular the intermittency of the renewable energy generation, and thus are playing a key role in the development of Smart Grids. Energy Storage Systems in smart Grids include electrical, chemical, thermal, mechanical and electrochemical storage systems, as summarized in figure 1.1 [47].

Electrical and electrochemical Energy Storage systems (ESS) consist basically in Battery Energy Storage Systems (BESS) which are deemed to be efficient for many Smart Grid applications such as self-consumption, ancillary services (like frequency control), arbitrage, peak shaving, load shifting and maintaining voltage on the distribution grid [48]–

[50].

When it comes to mechanical ESS, they consist principally in Pumped Hydro Storage Systems [51]–[55], flywheels [56], [57] , and Compressed Air Energy Storage (CAES) [56], [58], [59] which are considered as an attractive storage technology for Smart Grids thanks to their high ramp rate and quick start-up time [47].

Concerning Thermal Energy Storage Systems (TESS), they can be classified into three types according to the material being used for the heat or cold storage into: sensible heat, latent heat and thermo-chemical heat storage systems. Further details on each of these TESS classes, their materials and the pros and cons of each type can be found in [60]. A complete review on thermal energy storage facilities in District heating and Cooling Systems can be also found in [61].

Besides DER, a Smart Grid also contains a number of smart meters that produce a large amount of online operational data, which represents one of the major both opportunities and challenges of the Smart Grids.

1.2.3 Smart meters and information and communication technologies

Smart meters are one of the key devices in a Smart Grid. Their main role is to collect energy consumption information from load devices and communicate them to the utility company and/or the Smart Grid operator. These advanced energy meters are made up of different sensors and control devices, together with a specific communication infrastructure [62]. According to [47], a Smart Grid can contain up to millions of smart meters. Thus, the deployment of smart meters, together with other applications in Smart Grids result in the generation of huge amounts of online operational data. That is why, communication systems are also key elements in the Smart Grid infrastructure. They ensure the transmission of data between sensors and power appliances and smart meters, and between smart meters and utilities' data centers using both wired and wireless communication media [63].

1.2.4 Demand Side Management

The equilibrium between supply and demand is of paramount importance for the reliable operation of a power grid. In fact, if consumption outweighs or falls behind production, the frequency of the grid diverges from its nominal value. This nominal value is of $50Hz$

for the European Synchronous Zone that involves most of Europe and other neighboring countries, as well as for other countries like China, India, Indonesia and some African countries. Other countries including the American continent, as well as Japan, south Korea and some African countries have a nominal frequency of $60Hz$.

When the grid frequency deviates from its nominal value, regulation is needed to bring the frequency back to its pre-defined value. Even though the supply side can be controlled according to the load, it is still not easy to ensure the task of maintaining this equilibrium without controlling the demand side. Moreover, the increasing share of intermittent renewable energy generation adds another layer of complexity to this problem. Renewable energy generation not only depends on weather conditions, but it also has generation peaks that often do not always occur at the same time as peak demands. Thus, an interesting alternative to maintaining the balance between supply and demand is to use new strategies that rely more on demand control and end-users' engagement. Demand side Management (DSM) or Demand Response (DR) [64] is one of these promising strategies and can be defined as " the planning, implementation and monitoring of utility activities that are designed to influence the customer's use of electricity, in a way that changes the time pattern and magnitude of utility's load" [65]. In other words, this consists in controlling the energy consumption of the end-users by encouraging them to consume less power during peak hours, or to shift their energy consumption to off-peak hours in order to flatten the demand curve. DSM can therefore offer a wide range of potential benefits for the power systems as explained in [66]–[68].

1.2.5 Electric Vehicles

With the recent advances in battery storage technologies and the emergence of electric mobility, Plug-in Electric vehicles came into play and offer an additional distributed storage capacity in the Smart Grid context via vehicle-to-grid integration [69]. In fact, Grid-to-Vehicle, Vehicle-to-Grid and Smart charging technologies are seen as key steps towards achieving economical and environmental benefits, as stated by [70]. This paper proposes a multi-objective-techno-economic-environmental optimization approach for the scheduling of charge-discharge of Electric Vehicles in Smart Grids by connecting the stakeholders involved in the Smart Grids with the control of electric vehicles. One of the principal challenges that arise with the integration of electric vehicles in the power system is their highly uncertain behaviour and thus the difficulty to predict EV arrival and departure times for

Electric Vehicle Charging Stations (EVCS). The paper of Pflaum et al. [71] provides a brief review of related works that propose EVCS control strategies taking this uncertainty explicitly into account, and advances a robust EVCS management strategy which provides a day-ahead upper limit profile of the EVCS's power consumption.
Dang et al. [72] studied the case of an eco-district with electric vehicles, which are considered as flexibility providing units due to their Vehicle-to-Grid capabilities. The Energy Management System (EMS) of the eco-district developed in this work has an electric vehicle power management aiming at maximizing the electric vehicle charging power during off peak periods, and minimizing the discharge from electric vehicles when the demand reaches the peak load or when the energy generation is too low to meet the total demand. The results showed that the proposed EMS allowed a reduction of 70% in overloading duration and 17% in total electricity costs. Similarly, [73] designed an optimal management and day-ahead scheduling strategy for a microgrid including a Vehicle-to-grid system. Experimental tests showed that the proposed strategy results in daily cost savings of nearly 10%.

1.2.6 Energy Management Systems

Energy Management Systems (EMS) are designed to ensure monitoring, analysis and control of energy systems and equipments in Smart Grids by means of meters, sensors and control algorithms [74]. They dynamically adapt to distributed energy systems to make them more effective and reliable by controlling distributed devices [75]. They can therefore be defined as control devices which are responsible for defining the optimum scheduling of dispatchable units by using information such as load, renewable energy generation, weather and energy grid prices forecasts, energy storage units' state of charge, etc. [76], [77]. These information are used as inputs to perform optimal scheduling by determining optimal set points for the dispatchable units of the Smart Grid system. Developing an EMS mainly relies on formulating an optimization problem where the objective function can be single-objective or multi-objective. Single-objective functions basically consider the minimum-cost operation of the Smart Grid. When it comes to multi-objective functions, they involve a combination of two or more objectives that belong to the following four types [78] summarized in figure 1.3: capital and operational objectives (that include production and fuel costs, maintenance costs as well as start-up and shut-down costs), energy storage objectives (mainly related to costs, charging and discharging efficiencies

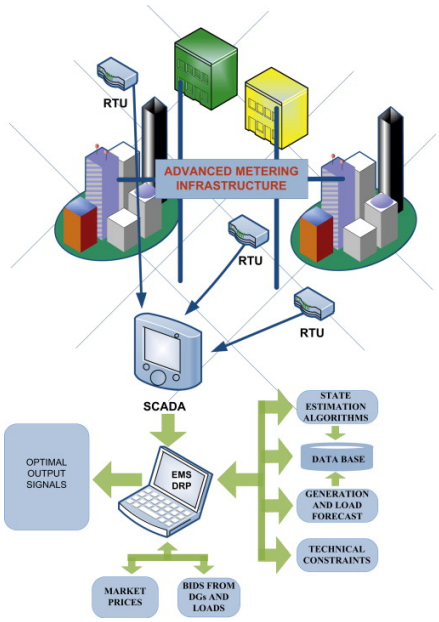


Figure 1.2: Energy Management Systems in Smart Grids [74].

and lifetime of the storage systems), environmental objectives (that involve maximizing Renewable Energy Sources (RES) and minimizing Greenhouse Gases (GHG) emissions and penalty costs related to them) and finally miscellaneous objectives (including penalty and dissatisfaction costs, power losses, etc). A compendium of these optimization objectives, together with constraints, tools and algorithms for Energy Management Systems is provided in [78]. Furthermore, an extensive literature review and analysis of the main trends in the field of centralized Energy Management Systems in Microgrids can be found in [79]. A deeper focus on Energy Management Systems literature is made in the next chapter of this manuscript.

1.3 Smart Thermal Grids

1.3.1 District Heating and Cooling Systems

The heating and cooling demands are currently responsible for a share of about 50% of the overall final energy consumption in Europe [80]. Households account for nearly 79% of this energy consumption which is used basically for space and water heating together with space cooling [81]. While almost 75% of this energy is still generated using fossil fuels, the European Commission issued a set of guidelines aiming at reaching the long-term carbon neutrality as well as the target GHG emissions, among which the transition of the heating and cooling sector is claimed to be of paramount importance [82], [83].

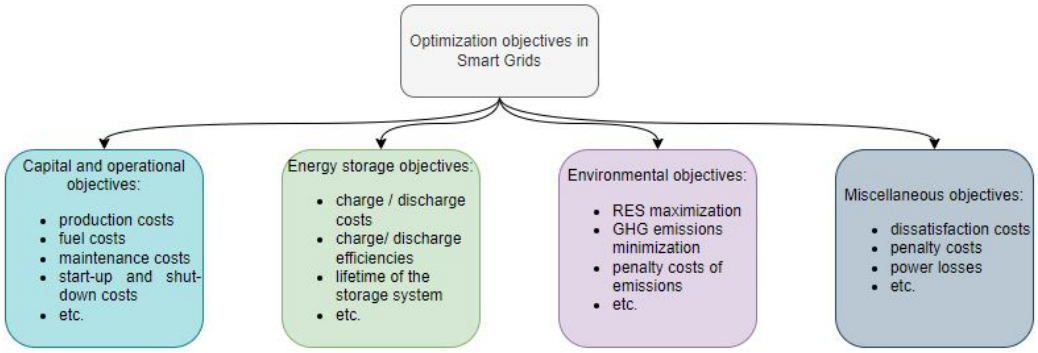


Figure 1.3: Types of optimization objectives in Smart Grids Energy Management systems.

District Heating and Cooling Systems, hereafter denoted DHCS, consist in centralised power plants that distribute heat and cold in urban areas by feeding hot water and cold water into a network of pipes. They are acknowledged to provide an efficient energy supply to cover the heating and cooling demand of buildings while reducing primary energy consumption and GHG emissions and operating in a more cost-effective way relatively to individual solutions [84]. In France in 2021, there were 833 District Heating Networks that deliver $25.4\ TWh$ of net heat to about 43945 buildings and 32 District Cooling Networks that deliver $0.81\ TWh$ of net cold to 1401 buildings [85].

Since their origination in the 14th century [84], district energy systems have used a variety of energy sources such as geothermal or ground source [86]–[89], solar thermal [90]–[93], fossil fuels [94], [95], biomass [96]–[98], waste incineration [99]–[102] and waste heat [103]–[107]. In their review of district heating and cooling systems, Lake et al. [108] described the technology and examined the advantages and disadvantages of each of the aforementioned energy sources.

In the first generation of District Heating Systems (1GDHS), which inception dates back to the 1880s in the USA, steam was used as the heat carrier. This technology was utilized in most of the DHS in the USA and Europe between 1880 and 1930 and is nowadays considered as obsolete basically because of the significant heat losses generated by high temperature steam not to mention the safety issues due to steam explosion.

In the second generation (2GDHS), the heat carrier used is pressurised hot water for which the supply temperature goes up to more than 100 ˚C. One of the main reasons behind im-

plementing this technology was to economize on fuel and achieve better comfort by using combined heat and power (CHP). It was used notably in the large DHS of the former URSS. Its remains can still be found today in some countries as old parts of the present water-based DHS. The third generation (3GDHS), sometimes also denoted "Scandinavian district heating technology", has also pressurised hot water as heat carrier but with supply temperatures generally below 100 ˚C. This technology was introduced in the 1970s and was used in most of DHS extensions and new systems in Europe, USA, Canada, China and Korea as well as replacements in Europe and the former URSS. The primary reason behind implementing 3GDHS was to ensure security of supply, following the two major oil crises of 1973 and 1979 [109], by enhancing energy efficiency and replacing oil with local or cheaper fuels like coal, biomass, waste and even geothermal and solar heat in some areas.

When it comes to District Cooling systems (DCS), three similar generations can be identified based on the advances in the technologies used for generating and distributing cold to the end-users as well as the strategies of operating the DCS. The first generation, 1GDCS, consists in pipeline refrigeration systems that appeared mainly in Europe and in North America in the late 1880s [110], [111]. The distribution fluids that were used in this technology are pressurized ammonia and brine solutions. Thereafter appeared the second generation (2GDCS) in the 1960s that had cold water as distribution fluid and was installed in many cities like Hambourg (Germany), Hartford (USA) and La Defense area in Paris (France). The third generation (3GDCS) has cold water as distribution fluid as well. Its technology relies on the diversification of cold supply : absorption chillers, mechanical chillers, excess cold streams, natural cooling from lakes and cold storage as reported by Lund et al. [112]. This technology was implemented in many locations in the 1990s following the 1987 Montreal Protocol on Substances that Deplete the Ozone Layer [113] that resulted in banning Chlorofluorocarbon (CFC) refrigerants [114].

1.3.2 Fourth Generation District Heating and Cooling Systems

The traditional high temperature DHS suffer from several shortcomings that can compromise their financial return, namely the network thermal losses that can go up to 30% of the supplied energy, high investment costs of installations, in addition to a heating demand decline due to the renovation of the existing buildings [115]. All these reasons motivate the current research focus on Fourth Generation and Fifth Generation District Heating and

Cooling Systems (4GDHCS and 5GDHCS) that are acknowledged as a promising solution to achieve high efficiencies by operating at low temperature. The concept of Fourth Generation District Heating System (4GDHS) was first introduced by Lund et al. [112] in relation to the challenges of reaching a future renewable non-fossil heat and cold supply as part of global sustainable energy systems. It is defined as "a coherent technological and institutional concept for which the DHS provides heat supply for low-energy buildings, with low grid losses and using low-temperature heat sources." The recent paper of Jodeiri et al. [116] gives a review of the challenges brought about by the integration of high shares of RES and waste heat permitted by the 4GDHS. In fact, unlike the previous three generations of DHS, the development of this fourth generation includes meeting major challenges like ensuring energy efficient buildings and integrating DHCS as part of the operation of Smart Energy Systems. The term Smart Energy Systems refers to the concept of integrated Smart Electricity, Gas and Thermal Grids that will be dealt with in the next section of this chapter.

Similarly to the concept of 4GDHS, fourth Generation District Cooling Systems (4GDCS) can be defined as "new smart district cooling systems that are more interactive with the electricity, heating and gas grids" [117]. In other words, the underlying goal behind 4GDCS is to achieve a cross-sectoral integration of the DCS as part of the smart energy systems by exploiting the Combined Heating and Cooling (CHC) synergy by using synchronously both ends of a heat pump or a combination of a heat pump and chiller working in parallel. This can be done by using the surplus cold obtained from heat generation for covering cooling demand and using the surplus heat obtained from cooling generation for covering heating demand. Seasonal storages can also play a preponderant role in exploiting this synergy by allowing the storage of cold from the heat pumps during winter for later use during summer and storing low-temperature heat from the chiller during summer for a later use in winter. This way, thermal storages help 4GDCS take part in smart energy systems where flexibility potentials allow for a better integration of RES.

The concept of Fourth Generation District Heating and Cooling systems (4GDHCS) evolves through the synergy between 4GDHS and 4GDCS and their integration in the multi-energy and cross-sectoral Smart Energy Systems [118]. A recent review on 4GDHCS as well as their integration in the state-of-the-art Smart Energy Systems was proposed by Fabozzi et al. [119], and a methodology for the optimal design and control of these

networks with thermal Energy Storage was proposed by Van der Heijde et al. [120].

1.3.3 Fifth Generation District Heating and Cooling Systems

After the emergence of the concept of 4GDHCS, the term Fifth Generation District Heating and Cooling Systems (5GDHCS) appeared for the first time in 2015 within the H2020 project known with the name FLEXYNETS which is an acronym for Fifth generation, Low temperature, high EXergy district heating and cooling NETworks [121]. A definition of 5GDHCS, which is in accordance with the classification of DHCS proposed by [112] and adopted in our work, is given by Buffa et al. [122] as a thermal energy supply system that has hybrid substations with Water Source Heat Pumps (WSHP) and water or brine as a carrier medium. It operates at low (close to ambient) temperature levels, which allows to use renewable heat sources at low thermal exergy content and to directly exploit industrial and urban excess heat. The reversible operation of its hybrid substations enables covering the heating and cooling demands of different buildings simultaneously and with the same pipelines. Thanks to these hybrid substations, the 5GDHCS technology permits sector coupling of thermal, electrical and gas networks as integrated parts of decentralised smart energy systems. [122] also provides a review of 5GDHCS projects in Europe and carries out an analysis of the pros and cons of these new technologies. It is worth mentioning that among the 40 5DHCS projects reported in this paper, 15 are located in Germany, 15 in Switzerland and 5 in Italy. Germany and Switzerland are thus recognised as pioneers of this technology, while none of the projects mentioned in this article are implemented in France.

The nomenclature of "fifth generation" was adopted in several research works [123]–[125]. Meanwhile, other recent research papers argue that this nomenclature is not in line with the labels established to DHS from 1GDHS to 4GDHS. For instance, Lund et al. [126] identify the similarities and differences between 4GDHCS and 5GDHCS. They explain that 5GDHCS might be a promising solution with several advantages, but is rather regarded as a complementary technology to 4GDHCS and can thus coexist with it. They conclude that this complementarity induces an absence of chronological succession between 4GDHCS and 5GDHCS that would justify the use of the term "generation" to qualify 5GDHCS.

Anyhow, Whether they are considered as two distinct generations or as complementry technologies, 4GDHCS and 5GDHCS both belong intrinsically to the new concept of

Table 1.1: Different generations of District Heating Systems.

Generation	1GDHS	2GDHS	3GDHS	4GDHS	5GDHS
Period of peak technology	1880-1930	1930-1980	1980-2020	2020-2050	2020-2050
Heat carrier	Steam	Pressurised hot water (>100˚C)	Pressurised hot water (<100˚C)	Pressurised hot water (around 50˚C)	Pressurised hot water (between 0˚C and 30˚C)

Smart Thermal Grids and are inherently related to the concept of Smart Energy Systems. We discuss both of these concepts respectively in the next sections of this chapter.

1.3.4 The concept of Smart Thermal Grids

DHCS play a paramount role in the implementation of the new Smart Energy Systems [3]. That is why the concept of Smart Thermal Grids (STG), similarly to the Smart (electrical) Grids, has recently emerged to allow the efficient integration of DER and Renewable Energy Generation within DHCS [5]. Smart Thermal Grids, also referred to for instance by the European Commission [127] as "smart heating and cooling grids", can be defined as " a network of pipes connecting the buildings in a neighbourhood, town centre or whole city, so that they can be served from centralised plants as well as from a number of distributed heating or cooling production units including individual contributions from the connected buildings" [128]. They are in fact able to ensure the same functions performed by classical thermal grids, but are developed in order to make an efficient utilization of the intermittent thermal DER and to provide the required energy when needed through optimal and intelligent management, as stated by [3] who investigated the impact of the integration of distributed and centralized thermal energy storage and solar energy systems within a community-level Smart Thermal Grid.

Like Smart (electrical) Grids, Smart Thermal Grids focus on the efficient and optimal operation of a grid structure allowing for distributed and renewable energy generation and possibly involving interaction with consumers as highlighted by [112] who also explained that the two concepts differ slightly. Indeed, the major challenges faced by Smart Thermal Grids come from the use of low-temperature heat sources and the interaction with low-

energy buildings, whereas the main challenges faced by Smart (electricity) Grids result from the intermittency of renewable electricity generation.
All in all, these two concepts not only complement each other but they are also both essential for the implementation of the future sustainable multi-energy systems, i.e., electrical, heating, cooling and gas systems, that we detail in the following section. Hence, Smart Thermal Grids are expected to be an integrated part of the future Smart Energy Systems [4] since they come up with a bunch of advantages like flexibility and ability to adapt to the changes that affect the thermal demand and supply in short, medium and long term [5].

1.4 Towards integrated solutions: Smart Multi-Energy Systems

1.4.1 Smart Multi-Energy Grids

Research works within the Smart Grid context traditionally focus solely on the electrical usages. However, jointly optimizing the electrical networks together with the other energy vectors interacting with them like heating and cooling networks, as well as gas and hydrogen networks, has a great potential to increase the overall economic and environmental efficiency and flexibility of the energy systems. This idea brings about an extension of the Smart Grid concept towards the concept of Smart electrical-thermal-gas grids or Smart Multi Energy Grids that lies on the interaction between electricity and other energy sectors (gas, hydrogen, heat and cold) as well as other sectors that electricity might interact with like the transport sector. These interactions allow to unlock considerable efficiency and flexibility potentials which represents one of the main benefits of Smart Multi Energy Grids [6].

1.4.2 Smart Multi-Energy Systems

Smart Multi-Energy Grids do indeed belong to the larger concept of Multi-Energy Systems (MES) which refers to energy systems where multiple energy vectors (electricity, heating, cooling, transport, fuel, etc.) optimally interact with each others at different levels (for instance a district, city, region..) as stated by Mancarella [129] who explained

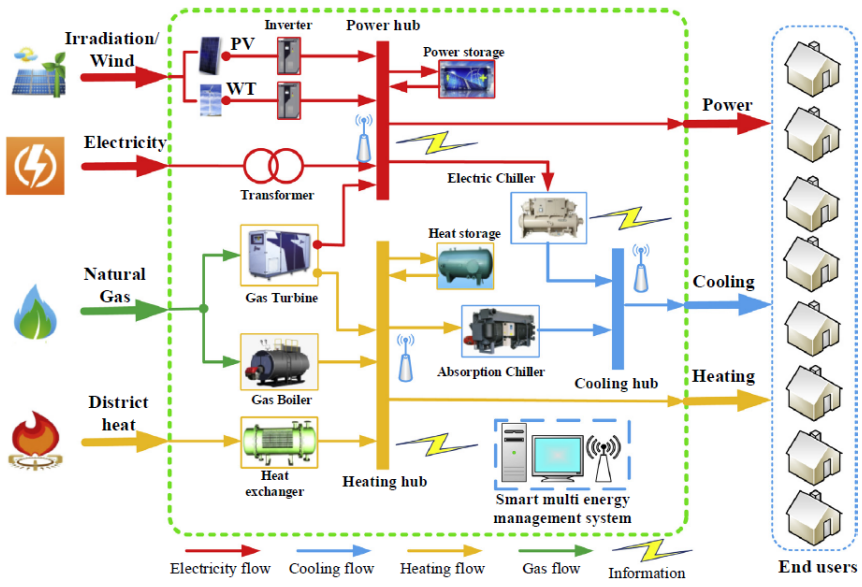


Figure 1.4: A typical community-level Smart Multi-energy System [8].

that Multi-Energy Systems can feature better technical, economic and environmental performance compared to classical independent or separate energy systems and at both the planning and operational stage, which is currently being recognized by a wealth of research being performed on related topics [130].

If a Multi-Energy System is acting in a smart environment and equipped with smart devices (e.g. smart meters, sensors, actuators), communication infrastructures, and embedded smart energy management systems, then it is referred to as a Smart Multi Energy System (SMES) as defined by [8] who also proposed the architecture of a typical community level Smart Multi-Energy System (SMES) (figure 1.4): it can be connected to the main utility grid as well as gas and district heating networks. As output, it can provide power, heating and cooling to the end-users. Storage systems can involve power storage, mainly Battery Energy Storage Systems (BESS), heat storage (e.g., in heated water storage tanks) or cooling storage (e.g., ice storage tanks).

The increasing operational complexity of such multi-energy systems leads to a need for advanced monitoring, forecasting and optimization algorithms [131]. Thus, a Smart Multi-Energy System is also equipped with a Smart Multi-Energy Management System (SMEMS). In fact, similarly to Energy Management Systems (EMS) previously defined in relation with the Smart Electrical Grids, Smart Multi-Energy Management Systems are developed to operate optimal energy management of Smart Multi-Energy Systems [8]. These SMEMS constitute the subject of the next chapter of this dissertation.

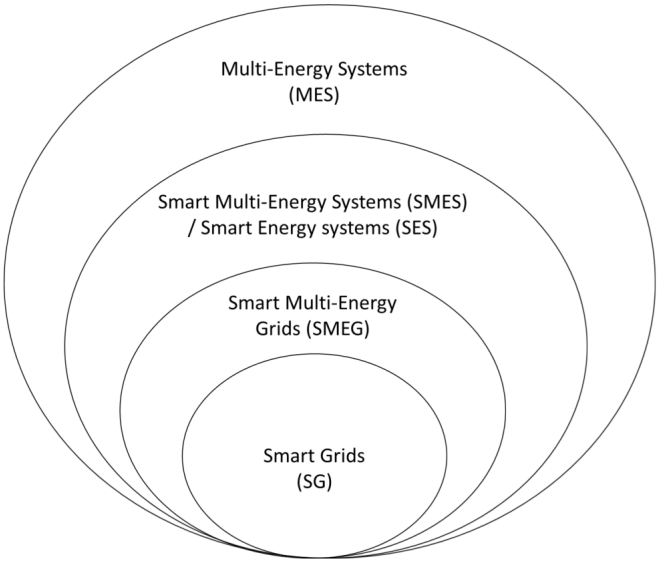


Figure 1.5: From Smart Grids to Multi-Energy Systems.

1.4.3 Smart Energy Hubs

Community-level and district-level Multi-Energy Systems are installed close to end-users. Energy transmission limits can thus be neglected since they involve short-distance transmissions. Therefore, energy flows in Multi-Energy Systems can be modeled with the concept of energy hub [8]. This concept was defined for the first time in [132] as an interface between consumers, producers, storage and transmission devices in different ways. This interface is made directly or via conversion equipment and by handling one or different carriers. However, its main idea relies on considering the external interactions of the energy systems through an input-output equivalent model which dates back to the work performed by Leontief in economics in 1986 [133].

In 2007, Geidl et al. [134], [135] introduced the use of the concept of Energy Hubs in a Multi-Energy Systems context for the analysis of multi-energy conversion through an input-output perspective. A complete review on this concept as well as an overview of the works carried out on energy hub models can be found in [136].

In 2015, Sheikhi et al. [137] proposed a modification of the classic Energy Hub model to present an upgraded model in the smart environment and introduced for the first time the concept of "Smart Energy Hubs". Actually, the recent emergence of Smart Grids in the power grid, coupled with the development of the Smart Grid concepts for the other energy carriers and infrastructure such as gas and district heating networks (via smart Thermal

Grids), led to the understanding of the fact that operating optimal management of smart energy system requires consideration of various energy carriers that lead to the creation of an integrated smart energy management system in the form of a Smart Energy Hub (SEH) as detailed by Mohammadi et al. [43]. Rayati et al. [138] defined the term SEH as "a unit in a smart energy infrastructure, where multiple energy carriers, e.g. natural gas and electricity, can be converted, conditioned and stored". Thus, a SEH is an energy hub, which entails appliances, loads and energy production systems that use smart meters and that include two-way communication links for an optimal and smarter operation.

1.4.4 Smart Energy Systems

The use of the term Smart energy systems was proposed for the first time in 2012 by Lund et al. [7] who later gave it a specific definition in [128], in [139] and in [140] as an approach in which smart electrical thermal and gas grids, together with storage technologies, are combined and jointly coordinated in a way that allows to identify and exploit synergies between them and hence achieve an optimal solution for each individual sector and for the overall energy system. In fact, some of the main synergies that can be identified and exploited thanks to the combination of these different energy carriers into integrated energy systems are, to name a few:

- Using electricity for heating and cooling purposes, which allows for the use of heat and cold storage instead of or together with electricity storage. Thermal storage may often be more efficient and cost-effective than electricity storage.

- Using electricity for gas (power-to-gas [141]) allows for the use of gas storage instead of, or together with electricity storage. Gas storage can also be more efficient and cost-effective than electricity storage.

- Using electricity for heating and cooling can also be used for power balance and ancillary services such as power markets regulation.

- Heat pumps for heating can also be used to meet cooling demands by providing district cooling systems with cold and vice versa.

- Using electricity for mobility (electric vehicles) not only allows for fuel replacement but also for providing power balancing services by exploiting the additional

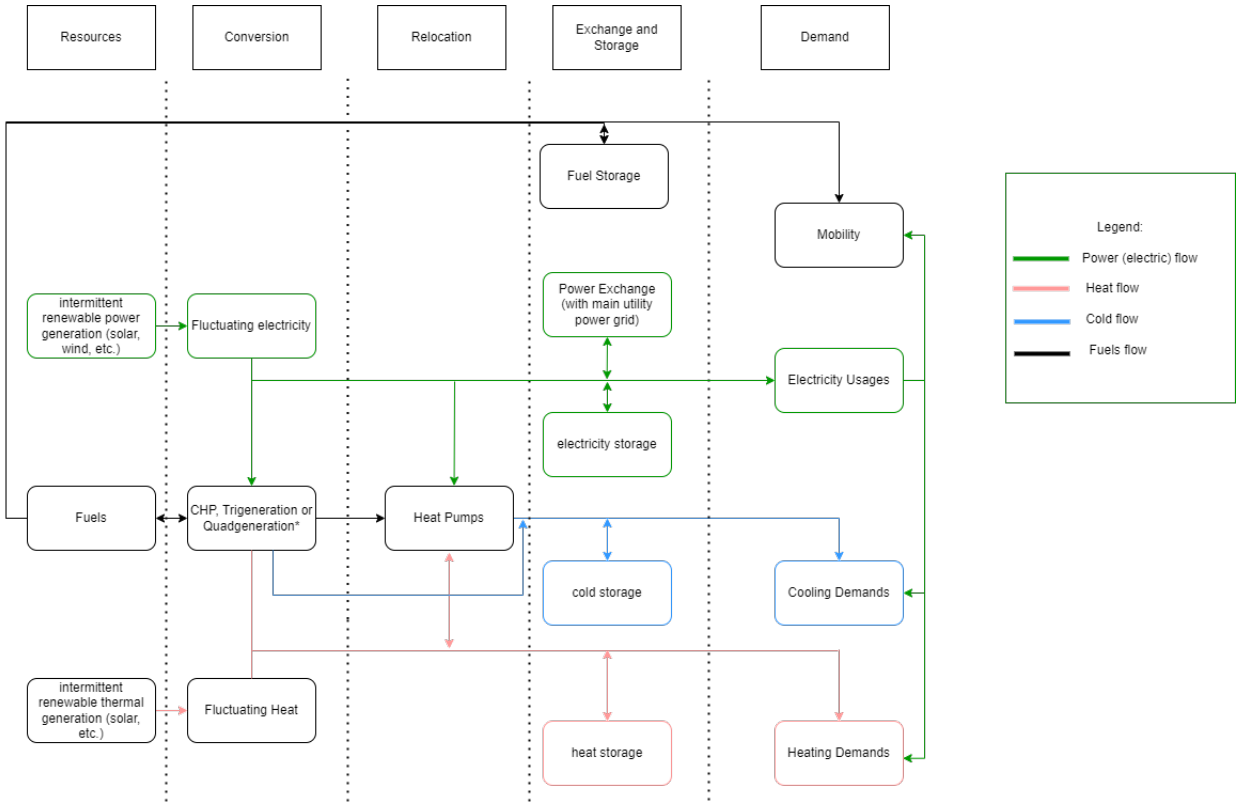


Figure 1.6: Overview of the components of Smart Energy Systems.

storage capacity that can be offered by the electric vehicles fleet (vehicle-to-grid systems[142]).

- Using heat produced by DHS for biogas production. In fact, biogas production only requires low-temperature heat. Supplying this heat by DHS is more effective than producing biogas at the plant.

Many of these synergies are present and exploited in the Smart Energy systems studied in this research work, in particular in the Meridia Smart Energy project case study presented in details in chapter 6.

An overview of the different energy components of a Smart Energy System, inspired from the definition and diagrams provided by the work of Connolly et al. [128], is presented in figure 1.6.

1.4.5 Integrated Energy Systems

In the paper where they propose a definition of Smart Energy Systems, Lund et al. [140] mentioned that the terms Integrated Energy Systems (IES) and Integrated Community En-

ergy Systems (ICES) are sometimes used in literature to refer to Smart Energy Systems [143]–[146]. Integrated Energy Systems are seen as a promising solution that can provide multiple energy supplements and collaborations through the coupling of independent energy systems such as power, heat, and gas, which can lead to a reduction of operating costs and an improvement of integrated energy efficiency [144].

The definitions of all the terms presented above are summarised in table 1.2. To the best of our knowledge, the literature lacks studies or reviews where all these terms are grouped, defined and discussed. Most of the aforementioned terms, in particular Smart Energy Systems, Smart Multi-energy Grids or Multi-energy Smart Grids and Integrated Energy Systems can be suitable to refer to the energy systems studied in our research work. For the sake of simplicity, we will almost solely use the terms Smart Energy Systems and Smart Multi-Energy Systems to refer to these systems in the remainder of this manuscript.

Table 1.2: Summary of concepts and definitions.

Concept	Definition
Smart (Electrical) Grids (SEG /SG)	Electricity networks that can intelligently integrate the actions of all users connected to them, including generators, consumers and those that do both, in order to efficiently deliver sustainable, economic and secure electricity supplies [41], [42].
Smart Thermal Grids (STG)	A network of pipes connecting the buildings in a neighbourhood, town centre or whole city, so that they can be served from centralised plants as well as from a number of distributed heating or cooling production units including individual contributions from the connected buildings [128].
Smart Multi-Energy Grids (SMEG)	Extension of the concepts of Smart Electrical Grids and Smart Thermal Grids where focus is not made solely on electricity grids or thermal networks but on "all-energy systems" by adopting a holistic approach for the deployment of all-energy systems at both operational and planning point of views [6].
Multi-Energy systems (MES)	Energy systems where multiple energy vectors (electricity, heating,cooling, transport, fuel...) optimally interact whith each others at different levels (for instance a district, city, region, etc.) [129]

Continuation of Table 1.2	
Concept	Definition
Smart Multi-Energy Systems (SMES)	Multi-energy systems that are acting in a smart environment and equipped with smart devices (e.g. smart meters, sensors, actuators), communication infrastructures, and embedded smart Energy Management Systems. They can be connected to the main power utility grid as well as gas and district heating networks [8].
Energy Hubs (EH)	A concept used to model energy flows in Multi-Energy Systems. It is an interface between consumers, producers, storage and transmission devices in different ways. This interface is made directly or via conversion equipment and by handling one or different carriers [132].
Smart Energy Hubs (SEH)	Units in a smart energy infrastructure where multiple energy carriers, e.g. natural gas and electricity, can be converted, conditioned and stored [138] .
Integrated Energy Systems (IES)	A solution that can provide multiple energy supplements and collaborations through the coupling of independent energy systems such as power, heat, and gas, which can lead to a reduction of operating costs and an improvement of integrated energy efficiency [144].
Smart Energy Systems (SES)	An approach in which smart electrical thermal and gas grids, together with storage technologies are combined and jointly coordinated in a way that allows to identify and exploit synergies between them and hence achieve an optimal solution for each individual sector and for the overall energy system [140].

1.5 Conclusion

This first chapter introduced the key concepts of energy systems that drive our research work, starting from Smart Electrical Grids, District Heating and Cooling Systems and Smart Thermal grids and arriving to integrated energy systems that bring electricity grids together with heating and cooling networks as well as other energy vectors (e.g gas) and sectors (e.g mobility) they may interact with. The definitions of the terms Smart Multi-energy Grids, Smart Energy Hubs, Smart Energy Systems, and Smart Multi-Energy Systems were reviewed and the latter was adopted as nomenclature for the remainder of this manuscript.

One of the major advantages brought about by Smart Energy Systems is that they allow for a joint operation and optimization of the different energy vectors which offers a great potential for enhancing the overall flexibility and economical and environmental efficiency of these combined energy systems. The next chapter of this thesis deals with the optimal energy management of these Smart Energy Systems by explaining its objectives, discussing previous work and reviewing the optimization techniques used for this purpose.

Chapter 2

Optimal energy management in Smart Energy Systems

Résumé

Ce second chapitre aborde la question de la gestion optimale de l'énergie dans les systèmes multi-énergie intelligents. En effet, même avec une conception et un dimensionnement optimaux, la performance technique, environnementale de tels systèmes est déterminée par leur fonctionnement et leur gestion [147]. Par conséquent, l'optimisation du pilotage des systèmes multi-énergies intelligents joue un rôle crucial dans l'exploitation optimale des synergies entre les différents vecteurs énergétiques tout en garantissant un fonctionnement optimal et des performances élevées pour chacun des systèmes mono-énergétiques individuels les constituant. Ce chapitre propose une revue des méthodes d'optimisation utilisées dans le développement de systèmes de gestion de l'énergie dans les réseaux électriques intelligent, les réseaux de chaleur et de froid et les systèmes multi-énergies intelligents.

2.1 Introduction

A Smart Energy System, even when optimally designed, still has its actual technical, environmental and economical performance defined by its operation and management [147]. Thus, the optimization of the Smart Energy Systems' operation plays a great role in optimally exploiting the synergies between the different energy vectors while ensuring an optimal operation and high performance for each of the individual mono-energy systems.

In this chapter, we shed light on the optimal energy management of Smart Electrical Grids, DHCS and Smart Multi-Energy Systems and we review the optimization techniques used within these energy management systems.

2.2 Smart Multi-Energy Management Systems

Similarly to Energy Management Systems in electrical Smart Grids contexts, Multi-Energy Management Systems, also known as Smart Multi-Energy Management Systems, play the role of the brain or control center in a Smart Energy System by ensuring optimal control and energy management of their energy generation, consumption, distribution and storage systems. Meanwhile, given the complexity of these multi-energy systems, the development of Smart Multi-Energy Management Systems is by no means an easy task since it involves multiple vectors coupling, several operational objectives and different time scales (short, medium and long-term) [148]. Extensive recent research works in the (Smart) Multi-Energy Systems context focused on modeling, optimal configuration planning, optimal energy management and optimal energy flow [8], [149] . The focus of the present work is mainly on optimal operational planning and control strategies in Smart Multi-Energy Systems. The optimal operational planning of the Smart Energy Systems is generally performed based on economic, environmental and supply security considerations. It decides the optimal operation of local resources as well as the interactions with the main utility grids. It aims at ensuring the load demand satisfaction for the end-users while satisfying technical, operational and economical constraints. Meanwhile, the intra-day optimal control of the Smart Energy System aims at adapting the energy management strategies to the real time realization of the unknown quantities (load demand, renewable energy generation, electricity prices, etc.) within a given control horizon and time step.

2.2.1 Optimal energy management in District Heating and Cooling Systems

The intelligent control for optimal operation is a major future challenge for the improvement of DHCS [112]. Nevertheless, due to their complexity and high parameters combinatorics, the determination of optimal production and distribution plans in DHCS is difficult, if not impossible, when solely based on empirical laws and expert judgement. That is why there is an emphasis in the literature on the need for operational optimization

techniques to ensure an intelligent control of the DHS. These operational optimization techniques as well as their taxonomy are detailed further in section 2.3. A review of modeling techniques as well as the main exact and heuristic optimization techniques applied to DHS can be found in [150]. Talebi et al. [151] also presented a review on the modeling and optimization of DHS and [152] reviewed the optimization approaches and tools for DHCN.

The paper of Benonysson et al. [153] was amongst the first studies that considered operational cost optimization in DHS by formulating a problem for supply temperature selection that embeds consumers, DHN and production plants. The control problem of supply temperature in DHS was also addressed by Grosswindhager et al. [154] who proposed a predictive control strategy based on Fuzzy Direct Matrics Control. A predictive control strategy was also proposed by Sandou et al. [155] for the short-term control and optimization of complex DHN. Idowu et al. [156] investigated the use of Machine Learning techniques for the optimization of energy usage in DHS with a CHP plant. In 2015, Vesterlund and Dahl [157] presented a method for the modeling and optimization of complex DHS with meshed networks, i.e., networks containing loops. Guelpa et al. [158] proposed an optimization method to minimize mainly pumping costs in large DHN, Giraud et al. [159] dwelled on the optimal control of pressurized hot water in DHS and Vesterlund et al. [160] dealt with the problem of minimizing total operational costs of complex DHN. Morvaj et al. [161] focused on the simultaneous design and operational optimization of urban distributed energy systems including DHN. Later on, Li et al. [162] studied the optimization problem in DHCS from three different perspectives, namely energy distribution, heat substations and end-users perspectives. They also paved the way for future research in the new concept of Smart Thermal Grids by suggesting research directions towards further development, testing, design and optimization. Within this framework, Zhang et al. [163] proposed a method to solve the problem of flow rate control optimization in smart DHS. When it comes to 4GDHCS, Schweiger et al. [164] presented a framework for their dynamic thermo-hydraulic simulation and optimization and van der Heijde [120] proposed a methodology for optimal design and control of 4GHDCS with thermal storage. Lesko et al. [165] also proposed a solution for the operational optimization of DHS with thermal storage. In fact, several studies focused on operational optimization of thermal storage within DHCS. For instance, [166] solved the problem of dynamic optimization for op-

timal charging and discharging periods of a thermal energy storage to achieve electrical and cooling load shifting and [167] introduced an approach for performance prediction and real time control for cooling storage in DCS.
After reviewing these studies on the operational optimization of district heating and cooling systems, the next section will focus on electrical Smart Grids and Micro-grids.

2.2.2 Optimal energy management in electrical Smart Grids and Microgrids

Similarly to DHCS and Smart Thermal Grids, intelligent control is also essential for electrical Smart Grids to, inter alia, ensure the optimal management of loads, production and storage units, minimize costs, fossil fuel consumption and Green House Gases emissions, and to warranty a reliable operation for the Smart Grid components. A review of the optimal control techniques applied to the energy management and optimal control of microgrids -which are vital components of Smart Grid architectures- is presented in [168]. These optimal control techniques are classified into classic methods such as Mixed integer Linear Programming and predictive optimization, and non classic methods like game theory, non linear programming and Particle Swarm Optimization, on which we focus further in section 2.3. More recently, Twaha and Ramli [169] presented a review of the most common optimization approaches for distributed energy generation systems including both stand-alone and grid-connected systems, and Pourbehzadi et al. [170] presented a comprehensive review of solution methodologies for the optimal operation of hybrid microgrids under uncertainties of RES. In [169], the optimization methods are classified into mathematical optimization methods and computer programming optimization methods. The first class of methods includes combinatorial optimization, dynamic optimization, linear and non linear programming, integer programming and network flow theory; and the second class involves metaheuristic methods, linear programming as well as Dynamic Programming. The paper shows that Artificial Intelligence techniques are the most used optimization methods for Distributed Energy Generation Systems and highlights Particle Swarm Optimization as the most dominating Artificial Intelligence method used for this purpose. Nevertheless, Reinforcement Learning and Deep Reinforcement Learning are not mentioned among these Artificial Intelligence techniques used for optimization within this framework. In the present work, we focus more on these two techniques that

we present in section 2.3.

2.2.3 Optimal energy management in Smart Multi-Energy Systems

Electrical, thermal and gas grids are traditionally operated separately. Meanwhile, in the future Smart Multi-Energy Systems, these different vectors need to be combined and controlled in a coordinated way. This coordinated optimal planning and control allows to identify synergies between them and achieve an optimal operation for each of these individual energy vectors, as well as for the whole Multi-Energy System [139]. Thus, a number of countries around the world like The US, France, Germany and China are developing plans around building Multi-Energy Systems, many of them are mentioned in [8]. Such energy systems are embedded with several energy converters and storage devices in a way that allows various energy vectors to be converted, distributed and stored locally in these community or district-level energy systems. Their inputs are connected to energy networks like DHCN, electricity grids and natural gas networks, and their outputs supply end-users with heat, cold, power and natural gas simultaneously [171].
As well, extensive research works are focusing on several aspects related to Multi-Energy Systems like modeling (based on the Energy Hub concept), optimal structure and configuration planning, optimal energy flow, and optimal energy management strategies [149]. for instance, [8] explained that the Energy Management System of a Smart Multi-Energy System, also referred to as Smart Multi-Energy Management System (SMEMS) is responsible for collecting data about the status of devices, energy prices, weather data as well as forecasts on weather, PV and wind generation and power, heating and cooling loads. Based on these information, it establishes optimal energy management strategies and dispatch schemes and sends control signals to each of the considered controllable elements in order to optimally manage the operation of the whole Multi-Energy System.

When it comes to optimization objectives in the Multi-Energy Systems and the smart Grids context, the optimal operation problems can have either one or a combination of two or more of the following objectives: Operational optimization and overall energy costs minimization, Levelized Energy Costs minimization, Optimal Power Flow (OPF) [8], [149], load peak reduction, self-consumption maximization, Green House Gases emission minimization, fossil fuel consumption minimization [172], etc. For instance, Weber et al. [173] used an evolutionary algorithm optimizer to compute the trade-off between multiple objectives, namely the trade-off between the CO_2 emissions and investment and

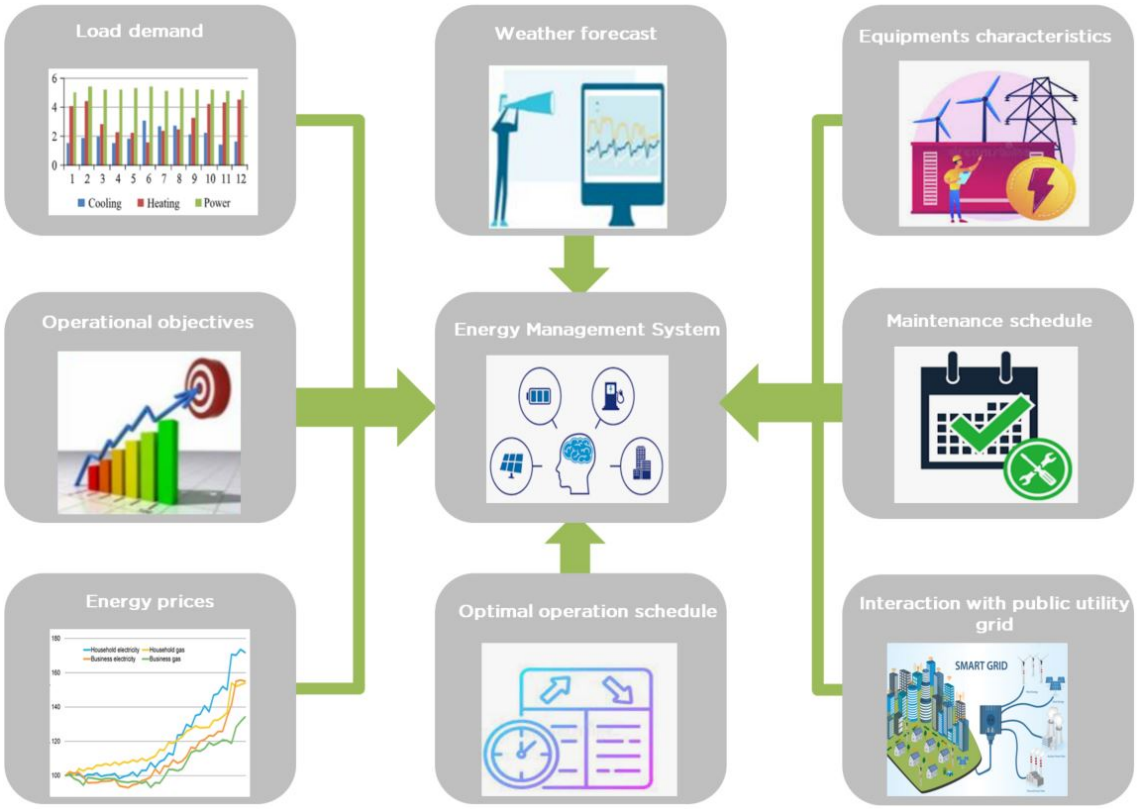


Figure 2.1: Energy management in Smart Energy systems (adapted from [148]).

operational costs. Moreover, a multi-objective model for the day-ahead operational planning of a microgrid was introduced by Hosseinnezhad et al. in [174]. Ma et al. [175] also proposed a multi-objective optimization approach for a Multi-Energy System integrating PV, CCHP and ground source heat pump.

Similarly to the previous works, the case studies under consideration in the present work have multiple optimization objectives and hence, multi-objective optimization can be the way to go. More details about these case studies as well as the choice of the methods for their optimal operation will be discussed further in the next chapters.

2.2.4 Markov Decision Process formulation

The goal of optimal operation in Smart Multi-Energy Systems is to develop an optimal planning and control strategy that maximizes some defined performance criteria for a given Smart Multi-Energy System. Since this problem belongs to the sequential decision making type of problems, it can be modeled as a Markov Decision Process (MDP).

The concept of Markov Decision Process was first introduced by Bellman in 1957 [176]. It is used to describe a stochastic process controlled by a sequence of control actions,

under uncertainty [177]. MDP is in fact a fundamental formalism for Decision Theoretic Planning, Reinforcement Learning (cf. Chapter 3) and other learning problems in stochastic domains. It has become the *de facto* standard formalism for learning sequential decision making problems as pointed out by [178]. Thus, many stochastic optimization problems in different domains like finance, telecommunication, routing problems, inventory problems and stochastic scheduling can be formulated as MDPs as detailed by [179]. If the state and action spaces are finite, then the MDP is called Finite Markov Decision Process (Finite MDP).
An MDP consists of states, actions, rewards and transition functions between states:

- State: for a given sequential decision making problem, a state is a unique characterization of all what is important in the state of this problem [178]. As a matter of example, if we consider the problem of playing the game of chess, a state would be a complete configuration of all the board pieces.

- Action: an action a refers to an action that is taken to control the system states. It can be either a unique action or a set of multiple actions, and can be composed either of discrete or continuous actions.

- Reward: it specifies explicitly the goal of the optimization by specifying the reward from being in a given state or taking a certain action for a state.

- Transition function: by applying an action a to a state s, the system moves from the state s to a new state s'. A transition function T defines the probability of making a transition to state s' after taking action a in state s. If the transition function T is known, the MDP decision making problem can be solved using Dynamic Programming, and if T is unknown, Reinforcement Learning algorithms (introduced in the next section and detailed in Chapter 3) such as Q-Learning can be used [180]. These ideas are discussed further in chapter 3.

For an extensive review on Markov Decision Processes, we refer the interested reader to [181] and [182]. As well, the book of Sigaud and Buffet [183] discusses MDPs and problems of Artificial Intelligence that can be formalized as MDPs, mainly sequential decision making under uncertainty and Reinforcement Learning problems. As a matter of example, [184] showed how the problem of home energy management with Electric Vehicle

charging can be formulated as an MDP. More details about the MPD formulation and the way of solving it mainly through Reinforcement Learning techniques will be discussed further in Chapter 3. In the next section, we present an overview of the optimization techniques used in the literature for solving the optimal energy management problem in Smart Energy Systems.

2.3 Optimization techniques for energy management in Smart Energy Systems

Optimization methods used in Energy Management Systems of Smart Energy Systems can be classified into three categories, as presented by [185]–[187] and as shown in figure 2.2 namely, rule-based techniques, optimization based techniques and hybrid techniques that we detail in this section.

2.3.1 Rule-based techniques

Rule-based techniques are methods that rely on allocating reference points based on existing situations, defined scenarios and decision trees, as explained by [187]. These methods can be nearly optimal for relatively simple use-cases such as energy systems with only one energy storage device as in [188] who developed a rule-based control strategy for a Battery Energy Storage System for dispatching solar and wind energy or also in [189] and in [190]. However, there is a common belief within the research community that rule-based control and expert judgement are not sufficient to perform the optimal control for more complex or hybrid energy systems in Smart (electrical) Grids, Smart Thermal Grids and Smart Multi-Energy Systems. In fact, due to their complexity and high parameters combinatorics, the development of optimal management plans and control strategies, for example in District Heating Systems, is difficult, if not impossible, when based excusively on empirical laws and/or expert judgements [159].

2.3.2 Optimization-based techniques

2.3.2.1 Exact mathematical methods

Optimization-based techniques can be classified according to whether they are exact mathematical methods, approximate methods or hybrid methods. Another possible and well-known classification would be to divide them into classic methods, which are the

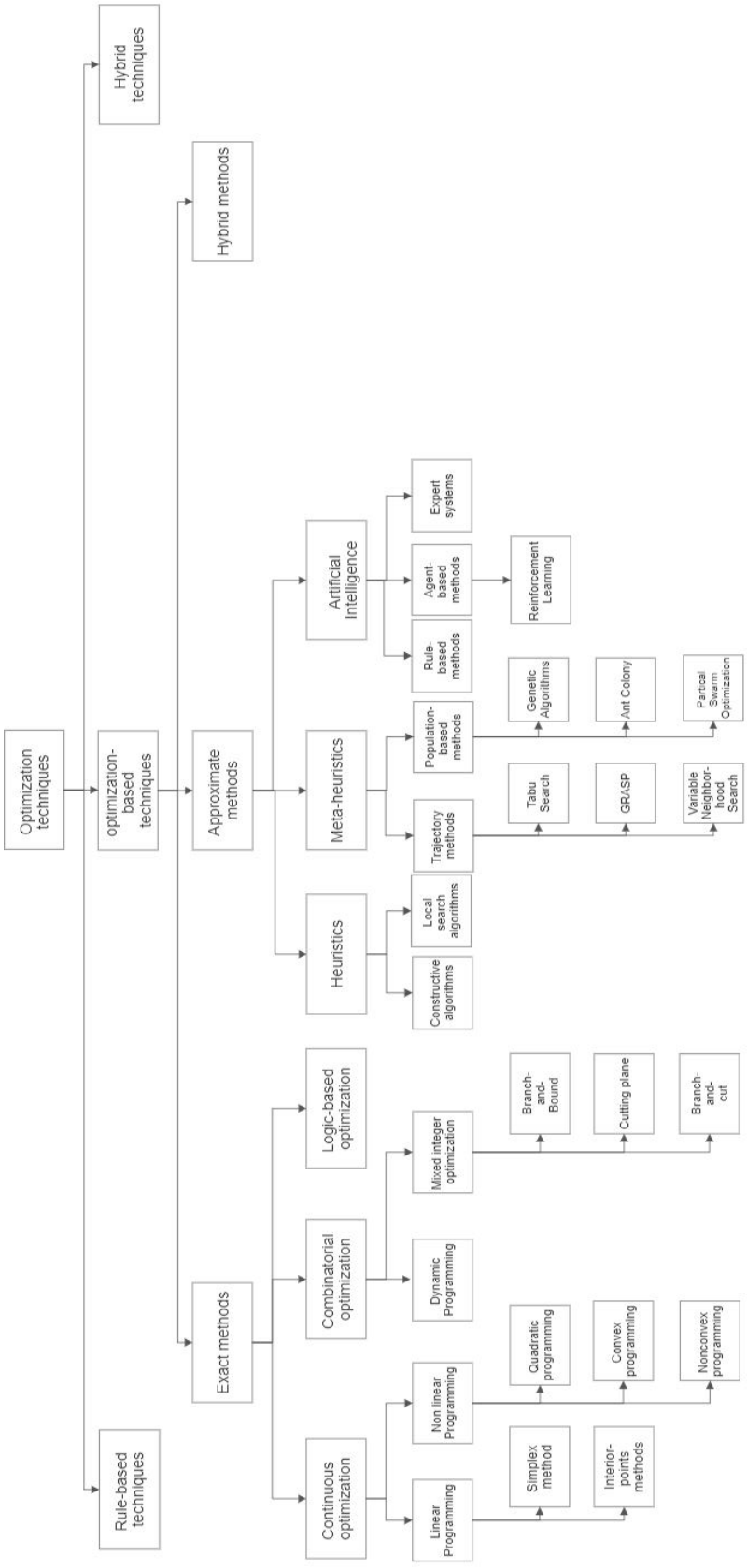


Figure 2.2: Classes of optimization methods used in the Energy Management Systems' context.

most widely used ones like Mixed Integer Linear Programming (MILP) and Model Predictive Control (MPC), and non classic methods such as game theroy, Non Linear Programming, Partcile Swarm Optimization and Genetic Algorithms. In this work we opt for the first type of classification. Main exact mathematical methods are:

- Linear Programming(LP): an LP formulation of the optimization problem is used if both the objective function and all the equality and inequality constraints are linear. LP problems can be mathematically expressed as:

$$\min_{x} \qquad f(x) = c^T x \tag{2.1a}$$

$$\text{s.t. } Ax \leq b \tag{2.1b}$$

$$x \geq 0 \tag{2.1c}$$

  Where x represents the vector of decision variables, $f(x)$ represents the objective function to be optimized, A is a matrix of known coefficients, c and b are vectors of known coefficients, and the inequality 2.1b includes the constraints of the problem. The most used algorithms for solving LPs include the simplex method introduced by Dantzig [191], and interior-point algorithms [192]. LPs belong to the linear models family which also includes Integer Programming (if variables are of type integer, most commonly binary) and Mixed Integer Linear Programming (if variables are of both real and integer types).

- Non-Linear Programming (NLP): corresponds to mathematical problem formulations where all the variables are continuous and that contain any kind of non-linearity in the constraints and/or in the objective function [185]. This kind of problems are solved using unconstrained and constrained optimization algorithms. The most common algorithms designed for NLP optimization include Newton-Raphon methods, conjugate gradient methods, quasi-Newton methods and successive quadratic programming methods (for quadratic optimization problems) [193].

- Mixed Integer Programming: Mixed Integer Programming (MIP) methods are used to solve mixed-integer linear or non-linear programming problems. A general MIP

problem can be expressed as follow:

$$\min_{x,y} \quad f(x,y) = c^T x + hy \tag{2.2a}$$

$$\text{s.t. } Ax + Gy \leq b \tag{2.2b}$$

$$x \geq 0 \tag{2.2c}$$

$$x = (x_1, x_2, ..., x_n)^T \in \mathbb{R}^n \tag{2.2d}$$

$$y \in 0, 1 \tag{2.2e}$$

Such problems deal with integer control variables which makes them require specific solving methods due to discontinuity of these variables. The main algorithms developed to solve this type of problems include Branch-and-bound [194], Cutting-plane [195] and the Branch-and-cut method which is a hybrid of the previous two methods [196].

Mixed Integer Linear programming (MILP) is one of the most extensively explored optimization-based techniques in the Smart Grid and district heating context [197], [198]. For instance, [165] proposed an optimization solution based on MILP problems for the operational optimization of DHS with thermal storage. Similarly, a MILP model is proposed for the optimal control of a pressurized water DHS in [159]. The proposed algorithm optimizes the use of production means, supply temperature and differential pressure. The optimal control is based on an MPC framework associated with a dynamic non-linear model of the network. This model is built using the simulation platform Dymola and the model library DistrictHeating [199].

Mixed Integer Programming models also include MINLP (Mixed Integer Non-Linear Programming) which are also used in power management, for instance when dealing with probabilistic constraints. [200] illustrates a solution of a mixed-integer stochastic nonlinear optimization problem with joint probabilistic constraints for the power management of a hydro plant coupled with a wind farm.

- Dynamic programming: Dynamic programming refers to a class of algorithms that can be used to compute optimal policies given a model of the environment as a

Markov Decision Process [18]. It deals with optimal control problems where decisions are made in stages and where the outcome of each decision can be relatively anticipated but not totally predictable. Therefore, a decision can not be viewed in isolation since a trade-off should be made between the desirability of low present cost and the undesirability of high future costs as explained in the book of [201] on Dynamic Programming and optimal control. Dynamic Programming captures this trade-off by ranking decisions based on the sum of the present cost and the estimated future costs at each stage. It aims to transform a complex decision making problem into a sequence of embedded simpler problems by decomposing a multi-stage problem into a sequence of single-stage decision problems, which are easier to solve sequentially.

If the considered multi-stage optimization problem is of stochastic nature, i.e., if one or several parameters of the problem are modeled as stochastic variables or processes, then probabilistic or stochastic Dynamic Programming is used to solve it [202], [203].

Even though Dynamic Programming is widely considered as the only feasible way of solving general stochastic optimal control problems, it suffers from what Bellman refers to as *the curse of dimensionality*, meaning that its computational requirements grow exponentially with the number of state variables. Nevertheless, Dynamic Programming is still far more efficient and more widely applicable than any other general method as explained by [18]. In other words, Dynamic programming is sometimes thought to be of limited applicability because of the curse of dimensionality, but the difficulties created by large state spaces and resulting in the curse of dimentionality are indeed inherent difficulties of the problem, not of Dynamic Programming as a solution method. Thus, Dynamic Programming is still comparatively better suited to handling large state spaces than other competing methods such as direct search and linear programming. The Dynamic Programming approach is discussed further in Chapter 3.

2.3.2.2 Approximate methods

Meta-heuristics: meta-heuristics are "solution methods that orchestrate an interaction between local improvement procedures and higher level strategies to create a process capable of escaping from local optima and performing a robust search of a solution space" as

defined in the handbook of meta-heuristics [204]. The most popular meta-heuristic methods inlude Genetic Algorithms (GA), Particle Swarm Optimization (PSO), Aunt Colony and Simulated annealing. Among these methods, Genetic Algorithms and Particle Swarm Optimization are the most widely used meta-heuristic methods in the energy management context.

Genetic Algorithms are based on search techniques that mimic the process of natural evolution, the Darwin thinking of natural selection and natural genetics. Genetic algorithms start from a population, a set of feasible random solutions (chromosomes), and use search operations with probabilistic rules of selection of new generation in the evolution process, ensuring their improvement. The major advantage of evolutionary algorithms is the search techniques that allows to achieve a global optimum, whereas other methods ensure it only if some convex properties of the problem are satisfied. The no starting point dependency and the ability to specialize the problem formulation makes Genetic Algorithms very interesting for engineering application, ensuring a high probability to find a global optimum.

As a matter of example, Genetic Algorithms are used in [205] for the operational cost optimization of electricity and heating networks in buildings with distributed energy generation, electric storage with batteries and thermal storage, and in [120] for the optimal design and control of Fourth Generation District Heating Networks with thermal storage. Genetic Algorithms are indeed part of a more general class of methods called evolutionary algorithms. Evolutionary algorithms are used for instance in [173] for the design and optimization of a district energy system. Similarly, the optimal integration of renewable energy sources for autonomous tri-generation combined cooling, heating and power system is achieved based on evolutionary Particle Swarm Optimization algorithm in [206]. Genetic algorithms and Particle Swarm Optimization are also combined in the work of Mazafar et al. [207] who proposed an approach for the optimal allocation of renewable energy sources and electric vehicle charging stations based on improved Genetic Algorithm - Particle Swarm Optimization algorithm.

Artificial Intelligence: most of the aforementioned optimization methods proceed by computing all or part of possible solutions before choosing the optimal one. This strategy appears to be time consuming, and thus can hardly be suitable for (near) real-time deci-

sion making process. This limitation, together with the availability of large amounts of historical, on-line and operational data in the context of Smart Energy Systems and in the era of big data, contributed to a growing research focus on the use of Machine Learning methods for optimization in Smart Grids [208] and Smart Energy Systems.

- Reinforcement Learning (RL): Reinforcement Learning is a learning paradigm that deals with learning to control a system in order to maximize a numerical performance measure that expresses a long-term objective, referred to as the numerical reward signal, as defined by [209]. The reward signal maximization implies learning what to do, by mapping situations to actions. The learner and decision-maker is called *the agent*, and the system it interacts with (to take actions) is called the environment (figure 2.3). Thus the terms agent, environment and action are used in lieu of the terms controller, controlled systems and control signal. Other key components of Reinforcement Learning include *the reward signal*, *the value function*, *the policy* and *the model of the environment*. The reward signal is the value sent by the environment to the agent at each time step. It defines the goal of the Reinforcement Learning since the objective of the agent is to maximize the total reward it receives from the environment. Thus, the reward signal defines the immediate "desirability" of the environment states [18]. Conversely, the value function of a state indicates the total expected reward to be accumulated over the future, starting from that state, which means that it indicates the long-term "desirability" of the state. Therefore, a given state can have a low immediate reward and a high value, and the opposite also can be true. The policy defines the way in which the agent behaves (takes actions) at a given time. And finally, the model of the environment is built to imitate the behavior of the environment. Thus it can be used for instance to predict the system state and reward that result from a given state-action pair. However, this model is optional and is used only in Model-based Reinforcement Learning approaches where a model is first learned and then used for planning [210]. In the opposite, model-free Reinforcement Learning is a trial-and-error learning approach that does not require a model of the environment. It is based on the agent's learning to associate optimal actions to states, without establishing transition probabilities between states. One of the most widely used model-free Reinforcement Learning techniques is $Q_{Learning}$ [210], [211]. Further details about the Reinforcement Learning theory

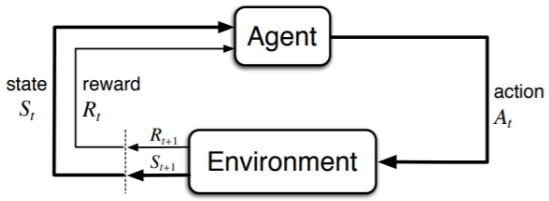


Figure 2.3: The agent-environment interaction in Reinforcement Learning [18].

as well as its aforementioned components will be given in Chapter 3.

The main characteristics that distinguish Reinforcement Learning from other Machine Learning methods are, as explained by [18], summarized in the three following points:

- Reinforcement Learning problems are closed-loop problems in the sense that actions taken by the learning systems do influence its later input.
- The learner in a Reinforcement Learning problem does not have direct instructions of what actions to take, as it is the case in many other Machine Learning methods, but rather acts by trying actions to discover which of them yield the most reward.
- Actions may have a consequence, not only on the immediate reward but also on the next situation.

The Reinforcement Learning paradigm has its roots in Dynamic Programming algorithms. However, for classical Dynamic Programming, there is a need to generate the transition probability and reward matrices from the given random variables in order to obtain an optimal solution, while for Reinforcement Learning, given the same distributions of governing random variables, a near-optimal solution can be solved by the use of a simulator and RL algorithm without the overhead of transition probability and reward matrices [156].

In traditional RL algorithms, action spaces are generally assumed to be discrete, whereas more complex and actual tasks often have a large state space and a continuous action space. In those cases, traditional Reinforcement Learning will encounter the curse of dimensionality when dealing with high-dimensional input data, which greatly limits the practical application of RL algorithms [22]. As a matter of fact, in complex and stochastic environments with high-dimensional state spaces, like

the typical Smart Grid or Smart Multi-Energy System environment, RL faces the problem of curse of dimensionality, ie the fact that the number of parameters to be learned increases exponentially with the number of variables, which results in a low learning efficiency [212]. This major limitation of RL can be overcome by Deep Reinforcement Learning (DRL) techniques.

- Deep Reinforcement Learning (DRL) is a state-of-the-art Machine Learning method that combines Deep Learning with Reinforcement Learning [25]. In other words, it combines the strong nonlinear perceptual capability of Deep Neural Networks (DNNs) with the robust decision making ability of Reinforcement Learning [22], [24].

  Unlike Reinforcement Learning, Deep Reinforcement Learning algorithms exhibit strong generalization capabilities in problems with complex state spaces. They have for example shown successful applications in problems with a very large number of states such as playing Atari, Go games [26], robotics, clinical trials [213], autonomous driving and other complex control tasks [25], [27]–[30]. In fact, with the development of distributed monitors and controllers, Smart Energy Systems control tasks can have very complex state spaces. Deep Reinforcement Learning algorithms, with their generalization abilities and strong representation power, could, therefore, be promising candidates for decision making problems where a large amount of features can be used. This idea was exemplified by Di Wu et al. [184] who showed that problems like home energy management can be dealt with using both batch RL (off-line RL algorithm) and DQN (on-line DRL algorithm).

  A deeper focus on the theory of (Deep) Reinforcement Learning and related work on its applications in power systems is made in chapter 3.

2.3.3 Uncertainty approaches

Despite the important advances in operational research and optimization techniques, one of the remaining current challenges consists in considering uncertainty in the optimization process. In fact, most of real life problems do contain some kind of uncertainty. For example, in economy, uncertainties might come for example from uncertain economic growth rates. In agriculture, as well as production, we face uncertainties on prices, demand and weather conditions. As far as the energy sector is concerned, sources of

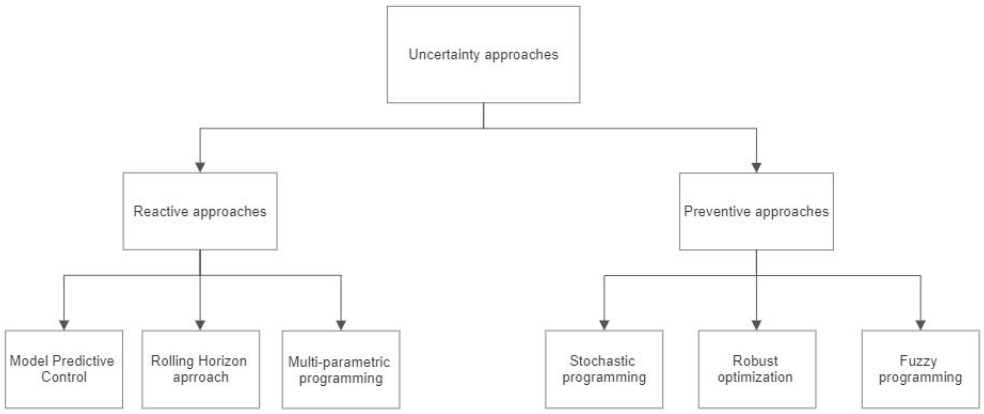


Figure 2.4: Uncertainty approaches.

uncertainties are also numerous [214]. For instance in the Smart Multi-Energy Systems context, market prices, demand for electricity, heating, cooling and gas consumption and weather conditions especially in renewable energy production are all associated to sources of uncertainty that may have a serious impact on the problem and, in that case, should be taken into account.

The approaches used to address problems under uncertainty can be classified into two categories as presented in Figure 2.4: reactive approaches and preventive approaches. In reactive approaches, a nominal plan is obtained using a deterministic formulation (i.e., that does not take uncertainties into account) and is then adjusted according to updated forecasts and system data. On the other hand, in preventive approaches, all possible scenarios are taken into consideration in order to find a good solution that is feasible for all the considered cases. Stochastic programming, robust optimization and fuzzy programming are among the most widely used preventive approaches [185].

2.3.3.1 Model Predictive Control

Model Predictive Control (MPC) [9], [10] is a feedback control method that appeared in the late 1970s and became widely used for advanced process control in both academic and industrial applications [215]. It has then experienced remarkable progress from both theoretical aspects and industrial issues perspectives [216] and gave therefore rise to thousands of successful industrial applications [217]. In the MPC mechanism, the optimal control

problem is solved at each time step to determine a sequence of control actions over a fixed time horizon. Only the first control action of this sequence is then applied to the controlled system and the new resulting system state is measured. At the next time step, the time horizon is moved one step forward and a new optimization problem is then solved, taking into account the new system state and updated forecasts of future quantities. The receding time horizon and the periodic adjustment of the control actions make the MPC robust against the uncertainties inherent to the model and forecasts [11]. A more in depth focus on the MPC theroy is made in Chapter 4.

2.3.3.2 Rolling Horizon approach

Model Predictive Control (MPC) belongs to predictive control approaches that also include other well known methods like Rolling Horizon Control -also referred to as Receding Horizon Control (RHC)- and Generalized Predictive Control (GPC). Even though these three control strategies have originally been investigated separately, MPC and RHC rely on similar principles and are often considered as identical when the MPC is based on the state-space model [218]. For instance, Kopanos and Pistikopoulos [219] who introduced a Rolling Horizon approach for the reactive scheduling of a network of Combined Heat and Power units, followed a state-space representation of the scheduling problem and gave the example of MPC as a typical Rolling Horizon approach.

2.3.3.3 Stochastic Programming

Stochastic programming, also referred to as optimization under uncertainty [220], [221], is the branch of optimization that addresses stochastic programs, i.e., optimization problems of the form:

$$\min_{x} f(x, \omega) \qquad s.t. \qquad x \in X(\omega), \tag{2.3}$$

where the objective function f and/or the feasible set X depend on an uncertain parameter ω. In stochastic programming, the decision variables are generally divided into two sets: the *first-stage variables* which are *here-and-now* decisions that have to be taken before the actual realization of the random events is revealed, and the *second-stage variables*, also called *recourse variables* which are *wait and see* decision variables that are selected, at a certain cost, after the random events take place. One of the most widely used stochastic programming approaches is the two-stage stochastic formulation [222] partly because of its simplicity relatively to other stochastic formulations.

To solve a stochastic program, we need to make assumptions on the probability distributions of its random variable. Nevertheless, making such assumptions is not always possible, especially when it comes to variables that depend for example on political decisions, such as oil prices.

2.3.3.4 Robust optimization

Another way to handle uncertainties in optimization problems -by making weaker assumptions on the random data- is to consider the worst case situation between different scenarios. This approach is referred to as robust optimization [223], which is commonly used worldwide but does not belong to the field of stochastic programming. It is acknowledged that worst case formulations generally result in expensive and conservative decisions and may lead to heavy investment costs which are intended to cover some extreme situations that may meanwhile have very low chance to occur. Nevertheless, using such robust methods can be very interesting or even essential for several kinds of problems where the most pessimistic scenario must be given a due importance and particularly considered. This can be the case, for example, for the sizing of dams where the most pessimistic cases should be considered because of the severe damage that can result from wrongly planned dams.

2.3.3.5 Fuzzy programming

The fuzzy mathematical programming was introduced by Bellman and Zadeh in 1970 [224], and then gained into popularity thanks to the work of Zimmermann in 1991 [225]. Similarly to stochastic programming, it addresses the problem of optimization under uncertainty, but is based on modeling uncertainty in a different way. In fact, while stochastic programming considers uncertainty by modeling it through discrete or continuous probability functions, fuzzy programming relies on considering random parameters as fuzzy numbers and constraints as fuzzy sets. Some constraint violation is allowed and we define the degree of satisfaction of a constraint as a membership function of the constraint in question [226]. Several types of membership functions have been proposed in the literature, but the most used one is the linear membership function $u(x)$ defined as follows: for a linear constraint $a^t x \leq \beta$, where x represents the decision vector and β is a random parameter that can take values ranging from b to $b + \Delta b$ where $\Delta b \geq 0$, the linear

membership function $u(x)$ is written as:

$$u(x) = \begin{cases} 1, & \text{if } a^t x \leq b, \\ 1 - \frac{a^t x - b}{\Delta b}, & \text{if } b < a^t x \leq b + \Delta b, \\ 0, & \text{if } a^t x > b + \Delta b. \end{cases} \tag{2.4a}$$

2.3.4 Hybrid-techniques

Hybrid techniques are methods composed of two or more of the previously mentioned optimization techniques. They can for instance put together MPC with genetic algorithms, Neural Networks or Reinforcement Learning. For example, [167] proposed a novel approach based on a Neural Network based Model Predictive controller coupled with a Genetic algorithm for the real time optimal control of a district cooling systems with thermal energy storage (ice storage). Furthermore, Model Predictive Control is combined with Reinforcement Learning in [227] for the control of systems described by Markov Decision Processes. Reinforcement Learning is used to make the MPC learn from experience which, inter alia, speeds up the decision making process. Furthermore, Smarra et al. [228] used a Neural Network based data-driven state model as a plant simulator in the MPC closed-loop optimization. The proposed method deals with building energy optimization and climate control.

2.3.5 Conclusion of the state-of-the-art

In light of the previous literature revue of related works on optimal energy management, planning and control of Electrical, thermal and Smart Multi-Energy Systems, and on the optimization techniques used for this purpose, we conclude that:

- Exact optimization-based methods like Mixed Integer Linear Programming, as well as approximate optimization-based methods like Genetic Algorithms and Particle Swarm Optimization can be suitable to solve optimal energy management problems and thus are widely used in the Smart Grid context. Nevertheless, their procedure can be time consuming and thus non suitable for an on-line optimization where real-time or near real-time decision making is needed. In fact one common key feature between all these methods is that they have to compute all or part of the possible solutions before choosing the optimal one, which makes their procedures generally

time consuming.

- This reason, added to the additional complexity related to the dynamic properties of the Smart Multi-Energy System case studies considered in the present research work, together with the large amount of on-line and operational data brought about by the Smart Energy Systems' concept, turn the choice of the operational optimization methods towards Machine Learning techniques. Among ML techniques, Reinforcement Learning is the area that deals with sequential decision making under uncertainty, and is thus suitable for cost optimization and optimal control problems. However, Reinforcement Learning fails to deal with large amounts of states and/or actions, due to the curse of dimensionality. Thus, the state-of-the art Deep Reinforcement Learning approach which is evolving through the combination of Reinforcement Learning with Deep Learning can be the solution to overcome this limitation.

- Despite the growing interest in the use of the Deep Reinforcement Learning approach and its success with many real-life problems, this method has hitherto been applied exclusively to mono-fluid Smart Grid contexts. To the best of our knowledge, there is a lack of studies considering the use of Deep Reinforcement Learning for the optimal control of Smart Multi-Energy systems. In this work, we apply Deep Reinforcement Learning in a Smart Multi-Energy System context. The main objective is to develop optimal planning and optimal control strategies for a real-life case study: the Meridia Smart Energy (MSE) Smart Multi-Energy System presented in further details in Chapter 6.

- Model Predictive Control (MPC) is one of the most widely used methods for advanced process control in both industrial and academic applications. In this work, we apply MPC, besides the Deep Reinforcement Learning approach, on a Smart Multi-Energy System case-study. A comparative study is carried out to evaluate the trade-off between performance and computational time of these approaches. To the best of our knowledge, this work represents one of the initial attempts in literature to simultaneously benchmark Deep Reinforcement Learning and Model Predictive control in a Smart Multi-Energy Systems context.

2.4 Conclusion

This chapter introduced the problem of optimal energy management in Smart Energy Systems and its formulation as a Markov Decision Process (MDP). The optimization-based exact as well as approximate methods used in literature to solve this category of problems were presented and the uncertainty approaches used to address these problems taking into account uncertainty were reviewed. The MDP formulation will be discussed in further details in Chapter 3. Besides, a more in depth focus will be made in the next chapters on two particular methods to address the problem: Deep Reinforcement Learning and Model Predictive Control. Applying and benchmarking these two methods for the problem of intelligent control for the optimal operation of a Smart Multi-Energy System case study represents one of the main contributions of this work.

Chapter 3

Deep Reinforcement Learning: theory and applications in Smart Energy Systems

"Artificial Intelligence = Reinforcement Learning + Deep Learning."

David Silver [229]

Résumé

Ce chapitre commence par une brève introduction aux paradigmes de l'Apprentissage Machine suivie d'une présentation des principales applications de ces paradigmes dans les systèmes énergétiques intelligents. Ensuite, on propose un focus sur le paradigme de l'apprentissage par renforcement en explorant son histoire, sa théorie et ses particularités et en s'appuyant principalement sur la méthodologie et les notations du livre référence de Sutton et Barto [18] sur l'apprentissage par renforcement. Enfin, on conclut ce chapitre par une revue de littérature concernant les travaux ayant appliqué de cette approche pour la gestion des systèmes énergétiques intelligents, mettant en évidence les contributions principales de ce travail par rapports aux travaux antérieurs dans ce domaine.

3.1 Introduction

The Reinforcement Learning (RL) paradigm was first briefly introduced in this manuscript in Chapter 2. In the present chapter, we propose a more in depth focus on the theory of

RL and Deep RL (DRL) based mainly on the approach and notations presented in Sutton and Barto's reference book of RL [18]. This chapter starts with a brief introduction of the different Machine Learning paradigms followed by an introduction of key applications of these Machine Learning paradigms in the field of smart energy systems. We then focus on RL and DRL theory, review their applications in power systems, report previous works using these approaches for optimal energy management in Smart Energy Systems and finally specify the contributions of this PhD research work in this domain.

3.2 Machine Learning paradigms

Machine Learning (ML) is the field of scientific study of algorithms and statistical models that give a computer the ability to learn without being explicitly programmed [230]. ML paradigms are organized into a taxonomy based on their desired outcome [231]. The most commonly used types include Supervised Learning, Unsupervised Learning and Reinforcement Learning.

- Supervised Learning (SL): SL is the most commonly used field of ML. It is concerned about learning a classification or a regression task from a set of labeled training data, by generating a function that maps the inputs to desired outputs. SL is the learning paradigm studied in most current research in the field of ML according to [18].

- Unsupervised Learning (UL): it refers to the task of learning models from data-sets of inputs without labeled examples. In other terms, it is mainly about finding hidden structures and discovering patterns and relationships in sets of unlabeled data.

- Reinforcement Learning (RL): RL is a learning paradigm that deals with how a software agent learns to control a system by taking actions in an environment in order to maximize a numerical performance measure that expresses a long-term objective, referred to as the reward signal. While Supervised and Unsupervised Learning models are generally myopic and only consider instant reward, Reinforcement Learning is rather sequential and considers long-term cumulative rewards and thus is far-sighted [232]. Reinforcement Learning is different from Supervised Learning in that Supervised Learning is not adequate for learning from interaction. In fact, Supervised Learning requires, for each given situation of the training set, a label that

specifies the correct action to be taken by the system in that situation. Meanwhile, in interactive problems, it is usually impractical to get the correct behaviour for all the situations of a training set that is representative of all the situations in which the learning agent is required to make decisions. Reinforcement Learning is also different from Unsupervised Learning even though the fact that both paradigms do not require examples of desired behaviour is misleading and might let one think of Reinforcement Learning as a kind of Unsupervised Learning. Unlike Unsupervised Learning, Reinforcement Learning is not about trying to find a hidden structure in a set of unlabeled data but rather about trying to maximize a reward signal. More fundamental distinctive features of Reinforcement Learning that make it different from Supervised, Unsupervised and other learning paradigms are detailed in section 3.4.2.2.

3.3 Selective key Machine Learning applications in Smart Energy Systems

In the following, we present some of the main applications of Machine Learning methods in the context of Smart Energy Systems as well as the classes of Machine Learning techniques mostly used for each application.

3.3.1 Electrical load forecast

Forecasting electrical load in order to accurately predict the amount of power that is likely to be consumed by the end-users is a crucial task in Smart Energy Systems. Methods used for load forecast can be classified into three categories as explained in [233]: physical methods, also called engineering methods [167], statistical methods and Machine Learning methods. Statistical methods used for electrical load forecasts include popular techniques such as ARMA (AutoRegressive Moving Average) and SARIMA (Seasonal AutoRegressive Integrated Moving average) [234]. Machine Learning methods include popular and widely used methods like Support Vector Machines (SVM) [235]–[237], Regression Trees (RT) [238], Multi Linear Regression (MLR), Artificial Neural Networks (ANN) [239], and other state-of-the-art neural network-based methods like Long-Short Term Memory (LSTM) [240].

3.3.2 Thermal load forecast

Methods used for thermal load forecast are generally similar to those used for electrical load forecast [156]. For instance, methods like regression models and ANN are commonly used for both electric and thermal load forecasting. Some methods, mainly physics-based models, are more used for thermal-load forecasting since they take into consideration critical factors that influence the thermal load demand like the physical properties of the buildings and the heat transfer dynamics. These methods are less commonly used for electrical load forecasting since the factors that influence electrical load demand such as the usage patterns and the time of the day are better captured using other ML metods.

Other state-of-the art methods like Deep Neural Network based prediction methods can also be used for the thermal load forecasting task. For instance, a Deep Learning approach is applied in [241] for the day-ahead forecast of thermal load in District Heating Systems. The proposed approach is compared with a linear model and the paper concludes that the Deep Learning model provides higher accuracy, even though simple linear models can perform very well in predicting heat loads for DHS especially when non-linearities can be accounted for.

3.3.3 Renewable energy generation forecast

The intermittent nature of renewable energy generation, mainly solar and wind power generation, lead to a an important research focus on renewable power generation forecast in order to efficiently manage their integration into Smart Energy Systems. In fact, accurately forecasting renewable power generation can help the Smart Energy Systems' operators to optimize grid stability, ensure an optimal energy dispatch as well as optimally manage the energy storage and other flexible storage systems. Machine Learning methods that are used for this purpose are broken down into three classes, namely Supervised methods like Artificial Neural Networks and Support Vector Machines [242], unsupervised methods like Deep Belief Networks [208], as well as hybrid models [242].

3.3.4 Flexibility quantification

Flexibility in smart thermal grids can be defined as " the ability to speed up or delay the injection or extraction of energy into or from a system" [243]. It only requires that the sys-

tem has thermal inertia in a way that guarantees that the energy balance can be respected all the time. The sources of this inertia come from thermal capacities that can be found in the heat or cold carrier, the heat or cold storage systems as well as the thermal inertia of the buildings provided with heat or cold. Vandermeulen et al. [243] explained that the question of flexibility quantification in the field of thermal networks is very relevant especially when the problem of their control is considered.

When it comes to Smart Electrical Grids, flexibility refers to the capacity to increase or decrease the electrical load at a certain time frame [244]. The work of Mocanu [208] addressed the problem of flexibility identification, prediction and estimation of optimal flexibility in electrical Smart Grids using mainly Machine Learning approaches. Classification is among data-driven methods used for flexibility detection and deep learning methods are used for prediction. The work proposed a five-order restricted Boltzmann machine approach and explained that this method outperformed most of these state-of-the art methods in accomplishing flexibility identification, prediction and estimation of optimal flexibility all at once.

3.3.5 Frequency control

Frequency control, also referred to as frequency regulation aims at maintaining the power system frequency close to its nominal value, e.g., 50 Hz in the in the synchronous grid of Continental Europe and 60 Hz in the U.S. In fact, the equilibrium between supply and demand is of paramount importance for a power grid. If consumption outweighs or falls behind production, the frequency of the grid diverges from its nominal value. In this case, regulation is needed to bring the frequency back to its pre-defined value. In practice, this can be done by absorbing the surplus power from the grid, or by injecting the missing power when supply and demand are not balanced. To ensure this operation of frequency regulation, Frequency Containment Reserves (FCR) are procured by the Transmission System Operators. Deep Learning and Reinforcement Learning are among the most used Machine Learning methods for this frequency control task [245].

3.3.6 Voltage control

Voltage Control is another task in the power systems context for which Machine Learning approaches such as Deep Learning and Reinforcement Learning can be used. It refers to the operation of keeping the voltage magnitude across the power network close to

its nominal value. Unlike frequency control which is a fast-timescale control problem, voltage control deals with both fast-timescale and slow-timescale controllable devices. Moreover, the increasing penetration of renewable energy generation such as PV and wind generation in distribution systems comes along with new challenges to the voltage control operation due to the rapid fluctuations and significant uncertainties of intermittent renewable generation. Several studies proposed the use of reinforcement learning-based approaches to deal with the problem of voltage control including the work of Yang et al. [246], Duan et al. [247] and Cao et al. [248].

3.3.7 Energy management

As introduced in Chapter 2 Energy Management Systems (EMS) are control software that use information flow to monitor, control and optimize the operation of the system by managing the power flow and maintaining the power balance in a reliable and efficient way. Thus, they play a paramount role in Smart Energy Systems by enabling effective control and optimization of energy generation, consumption and storage systems. Machine Learning methods can be used for this purpose by building models that can for example learn from historical data and predict future energy demand or be used to identify patterns and recognize anomalies in energy consumption data. These models can be incorporated within Energy Management Systems in order to significantly improve the efficiency and sustainability of the energy management in Smart Energy Systems.
The present work focuses on this specific application of ML in smart energy systems, namely energy management, and proposes to explore the potential of the DRL approach in fulfilling this task. The remainder of this chapter dwells on the theory of reinforcement learning and deep reinforcement learning and reviews their previous applications in the energy systems' context.

3.4 Deep Reinforcement Learning: theory

3.4.1 Deep Learning

Artificial Neural Networks (ANN) are functions that mimic the neural processing in the brains of biological organisms through a set of algorithms [249]. They consist in functions $f : X \to Y$ parameterized with weights $\theta \in \mathbb{R}^{n_\theta}$ that take $x \in X$ as input and give as

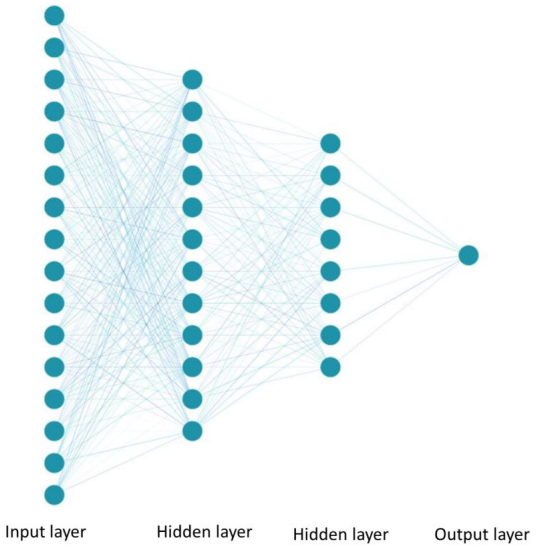


Figure 3.1: Example of a Deep Neural Network (DNN) with fully connected hidden layers (illustrated using NN-SVG tool [250]).

output $y \in Y$ such that:

$$y = f(x; \theta). \tag{3.1}$$

Deep Neural Networks (DNN) are Artificial Neural Networks with more than one hidden processing layer based on non-linear transformations as illustrated in Figure 3.1. These layers are trained with the objective to minimize a given cost function such as Root Mean Square error (RMS) for regression and cross-entropy for classification.
The last few years have witnessed a resurgence of interest in Deep Learning that can be traced back basically to:

- The increase in computational power mainly due to the use of GPUs.
- The availability of a continuously developing ecosystem of software and datasets.
- Several advances in the deep learning methodology, some of which are mentioned in [19] like the work of Srivastava et al. [251] and Szegedy et al. [252].

Therefore, Deep Learning or DNN is a particular scheme of ML. It has been applied in all ML paradigms, usually for Supervised or Unsupervised Learning, but can also be integrated with Reinforcement Learning mainly for function approximation.

- Deep Learning in Supervised Learning: Deep Learning is used for several successful applications in the field of Supervised Learning, mainly in image recogni-

tion applications. When provided with data in sufficient qualities and quantities, it achieves low error rates and generally exceeds human performance in related challenges [19].

- Deep Learning in Unsupervised Learning: Deep Learning has also numerous applications in the field of Unsupervised Learning like image generation applications where one of the most promising architectures is the Generative Adversarial Networks (GAN) introduced by Goodfellow et al. [253].

- Deep Learning in Reinforcement Learning: Deep Learning has been recently associated with RL to give birth to Deep Reinforcement Learning (DRL, DeepRL). This combination has shown a great success in learning complex tasks from high-dimensional inputs that were previously thought to be too complex to be completed by a software agent. In fact, the ability of an RL agent to solve problems, especially those with large state spaces, is strongly linked to its ability to appropriately generalize from its past experience. This is where comes the need for Supervised Learning methods within RL since they can provide the RL agent with a strong generalization capability. ANN and DNN are not the only or always the best way to do this as stated by Sutton and Barto [18], but in problems with complex and high dimensional state spaces, DNN exhibit a strong non linear perceptual capability that can provide RL with the ability to generalize from its past experience.

3.4.2 Reinforcement Learning

3.4.2.1 A brief history

Two main threads have constituted the early history of Reinforcement Learning [18]: trial and error learning and optimal control. These two threads evolved independently during decades before meeting around a third thread, namely temporal-difference methods (see paragraph 3.4.2.7), in the late 1980s to give birth to the modern Reinforcement Learning. One of the most powerful aspects of modern Reinforcement Learning is its interaction and integration with several disciplines including statistics, optimization, operational research, control theory and other mathematical subjects, as well as psychology and neuroscience. Indeed, among all Machine Learning schemes, Reinforcement Learning remains the closest form to natural human and animals learning process and many of

its algorithms were inspired by the learning process of biological entities.
In their Introduction to Reinforcement Learning book [18], Sutton and Barto mentioned that Harry Klopf, to whom they dedicate the book, was the individual whose ideas led to the distinction between Reinforcement Learning and Supervised Learning by reviving the trial-and-error thread within artificial intelligence [254], [255]. Klopf stated that a fundamental aspect of adaptive behaviour was being missed as Machine Learning researchers were dwelling almost solely on Supervised Learning. This essential aspect consists in the hedonic aspect of learning i.e., the fact that experiences range from pleasure to pain. These ideas of trial-and-error learning and "pleasure-pain systems" were among the earliest to be thought of implementing when it came to the possibility of implementing artificial intelligence on a computer. In fact in 1948, Alan Turing talked about the design of "a pleasure-pain system" that works similarly to the Law of Effect: "When a pain stimulus occurs, all tentative entries are cancelled, and when a pleasure stimulus occurs, they are all made permanent" [256].

3.4.2.2 Distinctive features of Reinforcement Learning

Besides the use of a numerical reward signal to formalize the idea of goals and purposes which remains one of the most distinctive features of the Reinforcement Learning paradigm, we can identify other important features that distinguish Reinforcement Learning from other Machine Learning paradigms:

Trial and error search

The Reinforcement Learning paradigm relies on the fact that the agent learns to take actions in order to maximize a cumulative total reward signal without being told which actions are likely to lead to achieving this goal. Instead, it has to learn which actions to take by trying them. This way, it learns to select actions based on evaluative feedback that does not require knowing what the desired actions should be.

Evaluative feedback

One of the most important distinctive features of Reinforcement Learning with respect to Supervised Learning is that it uses evaluative feedback - i.e., feedback that evaluates the actions taken - rather than instructive feedback -i.e., feedback that instructs by explicitly giving the correct actions-. One feature that distinguishes evaluative feedback from instructive feedback is that the first depends completely on the action taken whereas the

latter specifies the correct action to be taken regardless of the action actually taken by the agent.

Delayed reward

In the most interesting and complex sequential decision making problems considered in Reinforcement Learning, actions taken in a particular situation might affect not only the instant immediate reward of the agent but also all the next situations of the system and, as a consequence, the whole batch of subsequent rewards. Thus, a Reinforcement Learning agent is concerned about immediate rewards as well as delayed rewards.

The exploration-exploitation conflict

The exploration-exploitation conflict, dilemma or trade-off is a well-known issue that arises in Reinforcement Learning and not in other kinds of Learning. It refers to the balance that the agent has to find between *exploiting* what it has already experienced in order to obtain high reward values and *exploring* possibly better action selections. In other words, the trade-off that the agent has to make - as it accumulates knowledge of its environment- between following what seems hitherto to be the most promising strategy with respect to what it has experienced so far, and trying new experiences to know more about the environment. Indeed, in order to obtain high reward values, the agent has to prefer actions that it has already tried and found out to yield high rewards, whereas to discover these actions, it has to try actions that it has not selected before. The dilemma consists in that following solely exploitation or exploration leads to failure in the task and that is why the agent has to balance both by trying various actions and gradually preferring those that yield the most reward.

3.4.2.3 Core Reinforcement Learning components

Reward signal

At each time step of the learning, the RL agent receives a numerical signal (in the form of a single number) from the environment. This number is called the reward. The reward signal then defines the goal of an RL agent since the unique objective of this agent is to maximize its cumulative reward over the long run. If we see the agent as a biological system, the reward signal would be reflected for example by the experience of pain for negative values or pleasure for positive values.

Value function

The difference between the value function and the reward signal is that the reward signal specifies the immediate return and thus defines "what is good immediately" while the value function rather defines "what is good in the long term". In other words, for a given state of the environment, the reward indicates the immediate desirability of that state, whereas the value function determines the long-term desirability of that state, i.e., the total amount of accumulated rewards that the agent expects to get over the future, by starting from that state and considering the whole set of states that are likely to follow, together with their respective rewards. It is worth mentioning that most of Reinforcement Learning algorithms involve estimating the value-function.

Policy

The policy is the core element of an RL agent in that it defines, by itself, the way of behaving of the agent. It consists in a mapping of environment states and actions whose form can range from simple functions and lookup tables to more computationally extensive functions like search processes or stochastic policies defining probabilities associated to actions.

Model

The model of the environment is an optional component in the sense that it is part only of model-based RL. The model mimics the behaviour of the environment in a way that allows it to predict, for each state and action set, the resulting next state and reward. It is then used for a task referred to as $planning$ which consists in deciding on a sequence of actions, by considering possible future situations before they actually occur. A large part of the methods used to solve RL problems do not rely on models and $plannig$ but rather on trial-and-error learning. These methods are referred to as model-free methods.

3.4.2.4 Problem setup

The Markov Decision Process formalism

Markov Decisions Processes (MDPs) are a classical standard formalisation of sequential decision making problems and constitute a mathematically idealized form of the Reinforcement Learning problems and more generally of the problems of learning goal-directed behavior from interaction. The MDP framework suggests that an abstraction of such problems of goal-directed learning from interaction can be made by reducing them

to three main signals iteratively transiting between the agent and the environment:

- The first signal consist in the *actions* and represents the decisions taken by the agent. In general, they can be any kind of decision we might want to learn how to make.
- The second signal consists in the *states* and represents the basis on which the agent's decisions are taken. In general, they can be any information one can know about the environment that can be useful in making decisions.
- The third signal consists in the *rewards* and is used to define the goal of the agent: it communicates to the agent what one wants to achieve but not how to achieve it.

In an MDP, the actions taken in a given time step do not only influence immediate rewards but also subsequent states of the environment and, as a consequence, future rewards. The sequence of time steps is considered to be discrete, even though many ideas of this theory could be extended to continuous-time cases like in [257]. At each time step, the learning agent receives an observation, which is a representation of the *state* of the environment $S_t \in \mathcal{S}$ and takes an *action* $A_t \in \mathcal{A}(s)$. At the next time step, it receives an observation of the new state S_{t+1} as well as a numerical reward $r_{t+1} \in \mathcal{R} \subset \mathbb{R}$. We denote by *finite* MDP an MDP where the state set $\mathcal{S}$, the action set $\mathcal{A}$ and the reward set $\mathcal{R}$ all have a finite number of elements and the random variables R_t and S_t have discrete probability distributions that depend solely on the previous state and action. We then define the four-argument ordinary deterministic *dynamics function* of the MDP $p : \mathcal{S} \times \mathcal{R} \times \mathcal{S} \times \mathcal{A} \to [0, 1]$ for particular values $s' \in \mathcal{S}$ and $r' \in \mathcal{R}$ of these random variables as the probability of occurrence of these values at time t given particular values s and r of the previous state and action as:

$$p(s', r|s, a) \doteq Pr\{S_t = s', R_t = r|S_{t-1} = s, A_{t-1} = a\}, \qquad \forall s', s \in \mathcal{S}, r \in \mathcal{R}, a \in \mathcal{A}(s). \tag{3.2}$$

This dynamics function defines a probability distribution of each state s and action a, which means that

$$\sum_{s' \in \mathcal{S}} \sum_{r \in \mathcal{R}} p(s', r|s, a) = 1, \qquad \forall s \in \mathcal{S}, a \in \mathcal{A}(s). \tag{3.3}$$

In the Reinforcement Learning theory, the states are generally assumed to have what is called the *Markov property*, which means that the state includes all necessary information

about the past agent-environment interactions so that that the probability of each value of state S_t and reward R_t only depend on the state and action of the previous time step S_{t-1} and A_{t-1} and, given them, it does by no means depend on earlier states and actions. Once the dynamics function p defined, we can then define the *state-transition probabilities function* $p : \mathcal{S} \times \mathcal{S} \times \mathcal{A} \to [0, 1]$ as:

$$p(s'|s, a) \doteq Pr(S_t = s'|S_{t-1} = s, A_{t-1} = a = \sum_{r \in \mathcal{R}} p(s', r|s, a)), \tag{3.4}$$

As well as the expected rewards for a state-action pair $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$:

$$r(s, a) \doteq \mathbb{E}[R_t|S_{t-1} = s, A_{t-1} = a] = \sum_{r \in \mathcal{R}} r \sum_{s' \in \mathcal{S}} p(s', r|s, a), \tag{3.5}$$

and the expected reward for a state-action-next state triples $r : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$:

$$r(s, a, s') \doteq \mathbb{E}[R_t|S_{t-1} = s, A_{t-1} = a, S_t = s'] = \sum_{r \in \mathcal{R}} r \frac{p(s', r|s, a)}{p(s'|s, a)}. \tag{3.6}$$

We also define the *expected return* that we want the agent to maximize where the return G_t for a time step t is defined as a specific function of the sequence of rewards received after that time step $R_{t+1}, R_{t+2}, R_{t+3}, ...,$. The simplest form of this function is the sum:

$$G_t \doteq R_{t+1} + R_{t+2} + R_{t+3} + ... + R_T, \tag{3.7}$$

Where T refers to the time of termination i.e., the time step corresponding to the terminal state of the environment. Such random variable is only defined for episodic tasks that naturally break into identifiable subsequences such as game plays. However, for tasks that go on naturally without a limit, such as control problems considered in the present work, the termination time step would be $T = \infty$. Such tasks are referred to as *continuing tasks*. In such cases, the return function G_t that we seek to maximize can easily be infinite. That is why we define the concept of *discounted return*:

$$G_t \doteq R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + ... = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}, \tag{3.8}$$

Where $0 \leq \gamma \leq 1$ is a key parameter referred to as the *discount rate* or *discount factor*. If $\gamma < 1$, then the infinite sum in the return function would have a finite value if the reward sequence R_k is bounded. If $\gamma = 0$ then the approach is said to be *myopic* since the agent is only looking to choose A_t in order to maximize its immediate reward R_{t+1}. Thus, the parameter γ is used to determine the present value of future rewards: the more the value of γ is closer to 1, the more importance the agent gives to future rewards and is said to be *farsighted*.

This allows us to write the discounted return function defined above in a conventional form that is available for both episodic and continuing tasks:

$$G_t \doteq \sum_{k=t+1}^{T} \gamma^{k-t-1} R_k, \tag{3.9}$$

With the possibility of having $T = \infty$ or $\gamma = 1$, but not both at the same time.

Value functions and policies

Value functions and *policies* are key concepts in Reinforcement Learning. A value function of a given state is an estimate of *how good* it is - in terms of expected future return- for the agent to be in that state, and a value function of a given state-action pair is an estimate of *how good* it is for the agent to take that action in that specific state. Most of Reinforcement Learning algorithms involve estimating value functions. A value function is defined with respect to a *a policy* π which is a representation of a given strategy, behavior, or way of acting, by mapping states to probabilities of taking each possible action. If the agent is following a policy $\pi(a|s)$ at a time step t, this means that $\pi(a|s)$ defines a probability distribution of $a \in \mathcal{A}(s)$ for each $s \in \mathcal{S}$. The *state-value function* of a policy π denoted $v_\pi(s)$ is the expected return of starting from state s and following the policy π and is defined as:

$$v_\pi(s) \doteq \mathbb{E}_\pi[G_t|S_t = s] = \mathbb{E}_\pi[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1}|S_t = s], \qquad \forall s \in \mathcal{S}. \tag{3.10}$$

We also define the *action-value function* of a policy π as:

$$q_\pi(s, a) \doteq \mathbb{E}_\pi[G_t|S_t = s, A_t = a] = \mathbb{E}_\pi[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1}|S_t = s, A_t = a]. \tag{3.11}$$

A key property of value functions that is used in Reinforcement Learning as well as in Dynamic Programming consists in the recursive relationship between the value $v_\pi(s)$ of any state s and the value of its possible successors:

$$\begin{aligned} v_\pi(s) &\doteq \mathbb{E}_\pi[G_t|S_t = s] \\ &= \mathbb{E}_\pi[R_{t+1} + \gamma G_{t+1}|S_t = s] \\ &= \sum_a \pi(a|s) \sum_{s'} \sum_r p(s', r|s, a)[r + \gamma \mathbb{E}_\pi[G_{t+1}|S_{t+1} = s']] \\ &= \sum_a \pi(a|s) \sum_{s',r} p(s', r|s, a)[r + \gamma v_\pi(s')], \qquad \forall s \in \mathcal{S}. \end{aligned} \tag{3.12}$$

This equation is known as the *Bellman equation* for v_π. It is a fundamental equation in Reinforcement Learning and Dynamic Programming theory. It states that the value of a given state is equal to the discounted value of the expected successor state added to the reward expected on the way.

Optimal value functions and policies

We can define an *optimal policy* π_* for a finite MDP as a policy that is better than or equal to all other policies, in terms of expected return. The rule that defines whether a policy π is better than a policy π' is the following:

$$\pi \geq \pi' \text{ if and only if } v_\pi(s) \geq v_{\pi'}(s), \qquad \forall s \in \mathcal{S}. \tag{3.13}$$

There is always at least one optimal policy, and it might not be unique. In such cases, all optimal policies are denoted π_* and their common state-value function (resp. action-value function) is referred to as *optimal state-value function* v_* (resp. *optimal action-value function* q_*). They are defined as follows:

$$v_*(s) \doteq \max_\pi v_\pi(s), \qquad \forall s \in \mathcal{S}. \tag{3.14}$$

$$q_*(s, a) \doteq \max_\pi q_\pi(s, a), \qquad \forall s \in \mathcal{S}, a \in \mathcal{A}. \tag{3.15}$$

The relation between these two functions can be expressed as:

$$q_*(s, a) = \mathbb{E}[R_{t+1} + \gamma v_*(S_{t+1})|S_t = s, A_t = a]. \tag{3.16}$$

The optimal action-value function of a pair (s, a) gives the expected return for taking the action a in the state s and following the optimal policy π_*. Similarly to the Bellman equation 3.12, we define the Bellman equation for v_* which is known as the *Bellman optimality equation* for v_*. It is a special consistency condition that optimal value functions have to meet and it states that the value of a given state under an optimal policy π_* is equal to the expected return that corresponds to the best action from that state and it is written as:

$$\begin{aligned} v_*(s) &= \max_{a \in \mathcal{A}(s)} q_{\pi_*}(s, a) \\ &= \max_a \mathbb{E}_{\pi_*}[G_t | S_t = s, A_t = a] \\ &= \max_a \mathbb{E}_{\pi_*}[R_{t+1} + \gamma G_{t+1} | S_t = s, A_t = a] \\ &= \max_a \mathbb{E}_{\pi_*}[R_{t+1} + \gamma v_*(S_{t+1}) | S_t = s, A_t = a] \\ &= \max_a \sum_{s', r} p(s', r | s, a)[r + \gamma v_*(s')], \qquad \forall s \in \mathcal{S}. \end{aligned} \tag{3.17}$$

This equation was first popularized by Bellman who referred to it as the *basic functional equation* [20]. For a finite MDP, this equation has a unique solution, and for continuous time and state problems, the equivalent of this equation is called *Hamilton-Jacobi equation*.

We also define the *Bellman optimality equation* for q_* as:

$$\begin{aligned} q_*(s, a) &= \mathbb{E}[R_{t+1} + \gamma \max_{a'} q_*(S_{t+1,a'}) | S_t = s, A_t = a] \\ &= \sum_{s', r} p(s', r | s, a)[r + \gamma \max_{a'} q_*(s', a')]. \end{aligned} \tag{3.18}$$

The Bellman optimality equation given by 3.17 consists in system of n equations with n unknowns, where n is the number of states. If the dynamics function p of the environment is known, then one can use any method for solving systems of nonlinear equations to solve the system of equations for v_* and the related system of equations for q_*. Once v_* is obtained, computing the optimal policy π_* becomes relatively straightforward: any policy that is *greedy* with respect to v_* is an optimal policy. In other words, if we have the optimal value function v_*, then the actions that appear to be best after a one-step ahead search are optimal actions, and any policy that allocates non-zero probabilities solely to

these actions is an optimal policy. Thus, a one-step search using v_* and evaluating only short-term consequences does yield long-term optimal policies since the optimal value function v_* already considers the consequence, in terms of reward, of all possible future behaviors.

Nevertheless, computing optimal policies using this method, i.e., by explicitly solving the Bellman optimality equation, is not always feasible neither practical. In fact, three conditions need to be satisfied in order for this to be true:

- The states of the MDP have the Markov property.
- The dynamics of the environment are known with precision.
- Sufficient computational resources are available.

These assumptions are rarely true in interesting decision making tasks in practice. For instance, even if the first two conditions are satisfied, computing an optimal policy can generally not be done without extreme computational expenses. That is why many decision making techniques like heuristic search and dynamic programming methods focus on approximately solving the Bellman optimality equation. The same assumption is true in Reinforcement Learning where we also have to settle for approximate solutions. One key feature of Reinforcement Learning when approximately solving an MDP, in comparison with other approaches, is in putting more effort into learning a good behavior for frequently encountered situations (states) to the detriment of less frequently encountered ones.

3.4.2.5 Dynamic Programming

Dynamic Programming involves the set of algorithms used to compute optimal policies for MDPs where a perfect model of the environment is known. Despite its limited utility for Reinforcement Learning, due to its great computational cost and the fact that it requires a perfect model of the environment, Dynamic Programming is still a cornerstone for the Reinforcement Learning theory [258]. Indeed, most of the Reinforcement Learning approaches can be seen as attempts to fulfill the same goal as Dynamic Programming, i.e., computing value functions and using them to structure the search for good policies, with less computational efforts and without requiring the knowledge of a perfect model of the environment.

The *prediction problem*, also referred to as *policy evaluation* in the Dynamic Programming literature deals with computing a state-value function v_π for an arbitrary policy π. Let us consider the Bellman equation for v_π presented in 3.12. If the dynamics of the environment are accurately known then the $v_\pi(s), s \in \mathcal{S}$ are the solution of a system of n linear equations with n unknowns that can be solved using iterative solution methods. The first approximation v_0 is chosen arbitrarily, with assigning 0 value to the terminal state if it exists. Then, the Bellman equation is iteratively used as an update rule for the successive approximations:

$$\begin{aligned} v_{k+1}(s) &\doteq \mathbb{E}_\pi[R_{t+1} + \gamma v_k(S_{t+1})|S_t = s] \\ &= \sum_a \pi(a|s) \sum_{s',r} p(s', r|s, a)[r + \gamma v_k(s')], \qquad \forall s \in \mathcal{S}, \end{aligned} \tag{3.19}$$

where $\pi(a|s)$ is the probability of taking action a in state s under policy π.

The *iterative policy evaluation* algorithm consists in computing iteratively a sequence $\{v_k\}$. This sequence converges to v_π as $k \to \infty$. The pseudo-code for the iterative policy evaluation algorithm is presented in Algorithm 1.

The main reason behind evaluating a policy π by computing its value function, is to find

Algorithm 1: Pseudo-code for iterative policy evaluation algorithm to estimate $V \approx v_\pi$

```
Input the policy π to be evaluated
Fix the parameter θ > 0, a small threshold that determines accuracy of estimation
Initialize V(s) arbitrarily for s ∈ S with V(s_T) ← 0, where s_T is the terminal
  state, if any
while Δ ≥ θ do
    Δ ← 0
    for each s ∈ S do
        v ← V(s)
        V(s) ← Σ_a π(a|s) Σ_{s',r} p(s', r|s, a)[r + γV(s')]
        Δ ← max(Δ, |v − V(s)|)
    end
end
```

better policies. The *policy improvement theorem* states that for a pair of deterministic policies π and π', if for every state $s \in \mathcal{S}$, the value of selecting the action $a = \pi'(s)$ and then following π is greater than the value of just following π all the time, then the policy

π' is at least as good as (or better than) the policy π, that is if:

$$q_\pi(s, \pi'(s)) \geq v_\pi(s) \qquad \forall s \in \mathcal{S}, \tag{3.20}$$

then,

$$v_{\pi'}(s) \geq v_\pi(s) \qquad \forall s \in \mathcal{S} \tag{3.21}$$

Which means that

$$\pi' \geq \pi \tag{3.22}$$

This theorem holds if the pair of policies π and π' are identical, except for a single state s where $\pi'(s) = a \neq \pi(s)$. In this case, if $q_\pi(s, a) > v_\pi(s)$ this actually means that the changed policy π' is better than π.

If we perform a policy change, not only for a single state but for all states $s \in \mathcal{S}$, by selecting for each state s the action a that maximizes the quantity $q_\pi(s, a)$ as follows:

$$\begin{aligned} \pi'(s) &\doteq \arg\max_a q_\pi(s, a) \\ &= \arg\max_a \mathbb{E}[R_{t+1} + \gamma v_\pi(S_{t+1}|S_t = s, A_t = a] \\ &= \arg\max_a \sum_{s',r} p(s', r|s, a)[r + \gamma v_\pi(s')], \end{aligned} \tag{3.23}$$

then this new policy is at least as good as (or better than) the original policy, according the policy improvement theorem. This changed policy is called *greedy* policy, since it takes the action that appears to be best in the short term according to the value function v_π of the original policy π. The process that leads to this greedy policy is named *policy improvement*. This process always leads us to an improved policy which is strictly better than the original one, except when the original policy is already optimal. In fact, if we suppose that the greedy policy π' is just as good as, but not better than the original policy π i.e., $v_{\pi'} = v_\pi$, then (3.23) implies that

$$\begin{aligned} v_{\pi'}(s) &= \max_a \mathbb{E}[R_{t+1} + \gamma v_{\pi'}(S_{t+1}|S_t = s, A_t = a] \\ &= \max_a \sum_{s',r} p(s', r|s, a)[r + \gamma v_{\pi'}(s')]. \end{aligned} \tag{3.24}$$

This equation is none other than the Bellman optimality equation 3.17, which means that $v_\pi = v_{\pi'} = v_*$ and that π and π' are optimal policies. If we carry on iteratively alternating policy evaluations and policy improvements as defined above, this process will converge to the optimal policy and the optimal value function of the considered MDP. Indeed, a finite MDP has a finite number of deterministic policies and this process will therefore converge in a finite number of steps. The pseudo-code for this process, referred to as *policy iteration*, is presented in Algorithm 2. Even though policy iteration algorithm

Algorithm 2: Pseudo-code for policy iteration algorithm to estimate $\pi \approx \pi_*$ and $V \approx v_*$

Initialization
Initialize $V(s) \in \mathbb{R}$ and $\pi(s) \in \mathcal{A}(s)$ arbitrarily for $s \in \mathcal{S}$ with $V(s_T) \leftarrow 0$, where s_T is the terminal state, if any
Policy evaluation
Fix the parameter $\theta > 0$, a small threshold that determines accuracy of estimation
while $\Delta \geq \theta$ **do**
    $\Delta \leftarrow 0$
    for *each* $s \in \mathcal{S}$ **do**
        $v \leftarrow V(s)$
        $V(s) \leftarrow \sum_{s',r} p(s', r|s, \pi(s))[r + \gamma V(s')]$
        $\Delta \leftarrow \max(\Delta, |v - V(s)|)$
    end
end
Policy improvement
policy-is-stable $\leftarrow true$
for *each* $s \in \mathcal{S}$ **do**
    old-action $\leftarrow \pi(s)$
    $\pi(s) \leftarrow \arg\max_a \sum_{s',r} p(s', r|s, a)[r + \gamma V(s')]$
    if old-action $\neq \pi(s)$, then policy-is-stable $\leftarrow$ false
end
If policy-is-stable then stop and return $V \approx v_*$ and $\pi \approx \pi_*$,
else go to **Policy evaluation**.

guarantees convergence to the optimal policy and value function, one of its limitations is that each of its iterations includes policy evaluation which may involve many sweeps, i.e., updates for each state of the state set.

One way of ensuring faster convergence of the policy iteration algorithm consists in truncating the policy evaluation step mainly by breaking it after one sweep. This algorithm is referred to as *value iteration* and is detailed in Algorithm 3. This algorithm is obtained

by converting the Bellman optimality equation 3.17 to an update operation as follows:

$$\begin{aligned} v_{k+1}(s) &\doteq \max_a \mathbb{E}[R_{t+1} + \gamma v_k(S_{t+1}|S_t = s, A_t = a] \\ &= \max_a \sum_{s',r} p(s', r|s, a)[r + \gamma v_k(s')] \qquad \forall s \in \mathcal{S}. \end{aligned} \tag{3.25}$$

This algorithm involves in each sweep one sweep of policy evaluation and one sweep of policy improvement and achieves faster convergence when multiple sweeps of policy evaluation are inserted between each sweeps of policy improvement.

This idea of putting together policy evaluation and policy improvement processes and alternating them is referred to as *generalized policy iteration* (GPI). These processes are interleaved until they both stabilize, i.e., both do not yield any more change. This only happens when the policy found is greedy with respect to its evaluation function which means that the Bellman optimality equation 3.17 has been met and that the policy and value function found are optimal.

Not only Dynamic Programming methods but also most of Reinforcement Learning approaches can be considered as generalized policy iterations (GPI). One other common thread between Dynamic Programming and Reinforcement Learning is *bootstrapping*. In fact, all Dynamic Programming methods update estimates of the value of a state on the basis of estimates of successor states i.e., update estimates based on other estimates. This is called bootstrapping, and many Reinforcement Learning methods, even those who, unlike Dynamic Programming methods, do not require a model of the environment, do bootstrapping. Mentions of connections between Reinforcement Learning and Dynamic Programming appeared in the literature since the early 1960s with Minsky [259] and later in the work of Werbos [260] in 1977.

Nevertheless, Dynamic Programming suffers from several limitations, some of which are inherent to the approach itself like the fact that it involves sweeps over the entire state set. Other limitations, like the *curse of dimensionality*, limit its applicability on very large problems. The term curse of dimensionality was first introduced by Bellman [20] and refers to the exponential growth of the number of states with the number of state variables. However, this limitation is often thought of as a difficulty that is inherent to the problem to be solved and not to Dynamic Programming itself as a solution method.

If we denote by n the number of states and k the number of actions of an MDP, the number

of deterministic policies would be k^n. Dynamic Programming still guarantees to find an optimal policy in a number of computational operations that is less than some polynomial function of k and n. On the other hand, any method of direct search through the policy space would have to examine each of the k^n policies in order to offer the same guarantee. This means that Dynamic Programming would be exponentially faster. Even though Linear Programming methods can also be used to solve MDPs and have better worst-case convergence guarantees than those provided by Dynamic Programming, these methods become impractical at approximately 100 times smaller state spaces than Dynamic Programming methods. Thus, only Dynamic Programming approaches are still feasible for the largest MDP problems .

Algorithm 3: Pseudo-code for value iteration algorithm to estimate $\pi \approx \pi_*$

Initialize $V(s) \in \mathbb{R}$ arbitrarily for $s \in \mathcal{S}$ with $V(s_T) \leftarrow 0$, where s_T is the terminal state, if any
Fix the parameter $\theta > 0$, a small threshold that determines accuracy of estimation
while $\Delta \geq \theta$ **do**
    $\Delta \leftarrow 0$
    for *each* $s \in \mathcal{S}$ **do**
        $v \leftarrow V(s)$
        $V(s) \leftarrow \max_a \sum_{s',r} p(s', r|s, a)[r + \gamma V(s')]$
        $\Delta \leftarrow \max(\Delta, |v - V(s)|)$
    end
end
Return a deterministic policy $\pi \approx \pi_*$ such that
$\pi(s) = \arg\max_a \sum_{s',r} p(s', r|s, a)[r + \gamma V(s')]$

3.4.2.6 Monte Carlo methods

Monte Carlo methods are methods for estimating value functions and discovering optimal policies. They represent solution methods for Reinforcement Learning problems based on averaging sample returns i.e., on sampling and averaging returns for each state-action pair. Unlike Dynamic Programming, they do not require a model of the environment and can learn optimal policies and value functions from experience, that is from sample sequences of states, actions and rewards from real or simulated agent-environment interactions. They have therefore several advantages over Dynamic Programming methods, namely:

- A model of the environment is only needed to generate sample transitions and not the probability distributions of all possible transitions as it is the case for Dynamic Programming.

Table 3.1: Comparison of key properties of RL solution methods.

Method	Does bootstrapping	Requires a model
Dynamic Programming	✓	✓
Monte Carlo methods	✗	✗
Temporal Difference Learning	✓	✗

- Monte Carlo methods are able to learn optimal policies and value functions from actual experience as well as from simulated experience.

- The computational costs for estimating the value of a given state does not depend on the total number of states.

- Monte Carlo methods do not do bootstrapping which makes them less influenced by violations of the Markov Property.

A particularly interesting class of solution methods for Reinforcement Learning problems, called Temporal-Difference Learning (TD Learning), combines key aspects of Monte Carlo methods and Dynamic Programming: they do bootstrapping like Dynamic Programming, and learn from experience like Monte Carlo methods. These TD Learning methods are the object of the next section.

3.4.2.7 Temporal Difference Learning

Temporal-Difference Learning is a both crucial and original idea in Reinforcement Learning. It combines interesting features from Monte Carlo and Dynamic Programming methods in that it is able to learn directly from raw experience without requiring a model of the environment and that it updates estimates based on other available learned estimates. In effect, similarly to MC methods, TD Learning methods use experience to solve the *prediction problem*. We denote by *prediction problem* the problem of evaluating a policy π by estimating its value function v_π, in contrast to the problem of finding an optimal policy π_* which is referred to as the *control problem*.
Indeed, MC methods update their estimate V of the value function v_π for a given experience following a policy π using the following update rule:

$$V(S_t) \leftarrow V(S_t) + \alpha[G_t - V(S_t)], \tag{3.26}$$

Where α is a step-size parameter. In other terms, for a *visit* to a given state, MC methods need to wait until the end of the episode so that the return G_t for that visit is known and then use it for the update of $V(S_t)$. Unlike MC methods, TD methods do not have to wait until the end of the episode and can update the estimate of $V(S_t)$ just at the next time step using the following update increment:

$$V(S_t) \leftarrow V(S_t) + \alpha[R_{t+1} + \gamma V(S_{t+1}) - V(S_t)], \tag{3.27}$$

The quantity between brackets in the update rule is called the Temporal Difference error and denoted δ_t. It is a crucial quantity that appears in different forms in Reinforcement Learning. It is an error that measures the difference between the estimated value $V(S_t)$ and the better estimated value $R_{t+1} + \gamma V(S_{t+1})$:

$$\delta_t \doteq R_{t+1} + \gamma V(S_{t+1}) - V(S_t). \tag{3.28}$$

For a given time step t, the TD error δ_t depends on the reward R_{t+1} and next state S_{t+1}. As a result, the TD error for the time step t is not available until the next time step $t+1$. The TD method using this update rule is referred to as *one-step TD* or *TD(0)*. The pseudo-code for TD(0) is given in algorithm 4. TD learning comprises other general methods called *n-step TD* or *TD(λ)*.

Algorithm 4: Pseudo-code for Temporal Difference method TD(0) to estimate v_π

```
Input the policy π to be evaluated
Fix the step size parameterα : 0 < α ≤ 1
Initialize V(s) arbitrarily for s ∈ S⁺ with V(s_T) ← 0, where s_T is the terminal
  state, if any
for each episode do
    Initialize S
    for each step of the episode, if state S is not terminal do
        A ← action given by the policy π for S
        Execute the action A and observe the resulting new state S′ and reward R
        V(S) ← V(S) + α[R + γV(S′) − V(S)]
        S ← S′
    end
end
```

TD methods have advantages over both Dynamic Programming and Monte Carlo

Methods, namely:

- Unlike DP methods, they do not require a model of the environment's dynamics, its rewards and next state probability distributions.

- Unlike MC methods, they are implemented in an online and totally incremental way. They just have to wait one time step and not an entire episode to learn, and they learn from each transition, regardless of what subsequent actions are taken, and still can guarantee convergence, regardless of the step size parameter α, provided that it is chosen to be sufficiently small. In spite of these known advantages, some questions are still open like which method among TD learning and Monte Carlo methods converges faster, and which one makes a more efficient use of limited data.

As a matter of fact, Temporal Difference methods are not restricted to Reinforcement Learning. They can more generally be used to make long-term predictions about dynamical systems. They can therefore be useful for many applications among which prediction of power station demands, weather patterns, customer behaviour, financial data and election outcomes were mentioned by Sutton and Barto [18]. Nonetheless, these potential applications have not yet been sufficiently extensively explored, as stated by the authors. In the remainder of this section, we present two of the most widely used TD methods, SARSA and Q-Learning.

SARSA

SARSA (State Action Reward State Action) is an on-policy TD method used for the control problem. Instead of considering transitions from a state to another and learning the state values, here we consider transitions from a state-action pair to another and learning state-action values $q_\pi(s, a)$ using the quintuple $S_t, A_t, R_{t+1}, S_{t+1}, A_{t+1}$ that gives its name to this algorithm. The update rule for the action-value function $q_\pi(s, a)$ is as follows:

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha[R_{t+1} + \gamma Q(S_{t+1}, A_{t+1}) - Q(S_t, A_t)], \tag{3.29}$$

and the pseudo-code for SARSA is given in algorithm 5. This algorithm is on-policy because it continually estimates the action-value function q_π for the current behaviour policy π, and at the same time changes π greedily with respect to q_π.

Algorithm 5: Pseudo-code for the SARSA algorithm to estimate $Q \approx q_*$

Fix the step size parameter$\alpha : 0 < \alpha \leq 1$ and a small $\epsilon > 0$
Initialize $Q(s, a)$ arbitrarily for $s \in \mathcal{S}^+$,$a \in \mathcal{A}(s)$ with $Q(s_T, .) \leftarrow 0$, where s_T is the terminal state, if any
for *each episode* **do**
    Initialize S
    Choose A from $\mathcal{A}(s)$ given by a policy derived from Q (an ϵ-greedy policy for example)
    for *each step of the episode, if state S is not terminal* **do**
        Execute the action A and observe the resulting new state S' and reward R
        Choose A from $\mathcal{A}(s)$ given by a policy derived from Q (an ϵ-greedy policy for example)
        $Q(S, A) \leftarrow Q(S, A) + \alpha[R + \gamma Q(S', A') - Q(S, A)]$
        $S \leftarrow S', A \leftarrow A'$
    end
end

Q-Learning

Q-Learning [261] is one of the most well-known and most widely used Reinforcement Learning algorithms. Its development constituted one of the earliest breakthroughs in Reinforcement Leaning. It consists in an off-policy Temporal-Difference algorithm for the control problem where the learned action-value function Q approximates the optimal action-value function q_* independently of the policy followed. It is defined by the following update rule:

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha[R_{t+1} + \gamma \max_a Q(S_{t+1}, a) - Q(S_t, A_t)]. \tag{3.30}$$

The pseudo-code for Q-Learning is given by algorithm 6.

3.4.2.8 Function approximation

The methods discussed above are referred to as tabular solution methods because the value functions and the policies are stored in tabular forms. A way of generalizing these methods to large and continuous state and action spaces is by using function approximation. This consists in taking samples from a desired function and attempting to build on them and generalize in order to construct an approximation of the whole function (e.g the value function). In fact, for numerous decision making problems where one wants to apply reinforcement learning, the state space is large and combinatorial. For instance, let us consider using reinforcement learning to train an agent to play the game of Ms.

Algorithm 6: Pseudo-code for the Q-Learning algorithm to estimate $\pi \approx \pi_*$

```
Fix the step size parameterα : 0 < α ≤ 1 and a small ε > 0
Initialize Q(s, a) arbitrarily for s ∈ S+,a ∈ A(s) with Q(s_T, .) ← 0, where s_T is
  the terminal state, if any
for each episode do
    Initialize S
    for each step of the episode, state S is not terminal do
        Choose A from A(s) given by a policy derived from Q (an ε-greedy
          policy for example)
        Execute the action A and observe the resulting new state S′ and reward R
        Q(S, A) ← Q(S, A) + α[R + γ max_a Q(S′, a) − Q(S, A)]
        S ← S′
    end
end
```

Pac-Man. In this game, there are over 250 pellets that Ms. Pac-Man can eat. Each pellet has two possible states (present or already eaten). This means that the total number of states is about 10^{75}, which is greater than the number of atoms in our galaxy [262]. It is thus unrealistic to try to keep track of an estimate of all the Q-values or to expect to find the optimal value function or an optimal policy for such tasks. The purpose is then to use function approximation in order to find good approximate solutions that require limited computational resources. Linear function approximation was one of the most used methods mainly due to its well established theoretical properties. Special non linear function approximators, namely Artificial Neural Networks (ANN), became then a popular choice mainly after the work of Mnih et al. [25], even though Sutton and Barto [18] explained that the application of ANNs in reinforcement learning as function approximation dates back to the work of Farley and Clark in 1954 [263]. The interested reader may refer to the work of Schmidhuber [264] for a review on applications of ANNs in reinforcement learning.

In algorithm 7, we present, as a matter of example the pseudo-code for TD(0) with function approximation called Semi-gradient TD(0).

3.4.3 Deep Reinforcement Learning

We talk about Deep Reinforcement Learning, denoted DRL or Deep RL, when Deep Neural Networks are used to approximate the value function, the policy or the model (i.e., the state transition function and the reward function) in Reinforcement Learning [232]. In fact, real-world complex problems generally have high dimensional, often even

Algorithm 7: Pseudo-code for TD(0) with function approximation to estimate $\hat{v} \approx v_\pi$

Input the policy π to be evaluated
Input a differentiable function $\hat{v} : \mathcal{S}^+ \times \mathbb{R}^d \leftarrow \mathbb{R}$ such that $\hat{v}(s_T, .) \leftarrow 0$, where s_T is the terminal state, if any
Fix the step size parameter $\alpha > 0$
Initialize the weights $\mathbf{w} \in \mathbb{R}^d$ of the value function arbitrarily (for example $\mathbf{w} = 0$)
for *each episode* **do**
    Initialize S
    for *each step of the episode, state S is not terminal* **do**
        $A \leftarrow \pi(.|S)$
        Execute the action A and observe the resulting new state S' and the reward R
        $\mathbf{w} \leftarrow \mathbf{w} + \alpha[R + \gamma\hat{v}(S', \mathbf{w}) - \hat{v}(S, \mathbf{w})]\nabla\hat{v}(S, \mathbf{w})$
        $S \leftarrow S'$
    end
end

continuous, action spaces. Therefore, it can be of interest for Reinforcement Learning algorithms to rely on Deep Neural Networks for learning the value function, the policy or the model, for two main reasons [19]: on one hand, they are suitable for dealing with high-dimensional sensor inputs without requiring an exponential data extension. On the other hand, they are also suitable for online learning. In other terms, they are able to exploit additional samples of data collected during learning in order to gradually improve the function approximators.

3.4.3.1 Value-based DRL methods

Value-based methods are among the most popular classes of algorithms in Deep Reinforcement Learning. They rely on attempting to learn the optimal action-value function that maps states to the expected cumulative reward obtained by taking a particular action in that state. Deep Q-Learning (DQL) is one of the most well-known value-based methods. It consists in a neural network-based algorithm that uses experience replay and target networks to stabilize training and improve learning efficiency. It attempts to minimize the mean squared error between the predicted Q-value and the target Q-value for each state-action pair. A pseudo-code for DQN is given in Algorithm 8, where the action-value function approximated by a DNN is denoted Q, the target action-value function used to compute the targets for the update step is denoted Q', D refers to a replay memory uti-

lized to store transitions to be used for the training of the Q network, and C refers to the number of steps after which the target network Q' is to be updated with weights of the current network Q.

Nonetheless, one of the major limitations of value-based methods such as DQN is that

Algorithm 8: Pseudo-code for Deep Q-Learning algorithm with experience replay, adapted from [25]

Initialize replay memory D to capacity N
Initialize action-value function Q with random weights θ
Initialize target action-value function Q' with weights θ' (same as Q)
Initialize state s
for *each episode in range(num-episodes)* **do**
    Initialize cumulative reward signal R to to 0
    Initialize sequence $s_1 = \{\mathbf{x}_1\}$ and pre-processed sequence $\phi_1 = \phi(s_1)$
    for *each step t in range(num-steps-per-episode)* **do**
        Take action a_t ϵ-greedily with respect to Q, i.e select random action a_t with probability ϵ, otherwise select $a_t = argmax_a Q(\phi(s_t, a; \theta)$
        Execute action a_t and observe reward r_t and image x_{t+1} of next state s_t
        Set $s_{t+1} = s_t, a_t, x_{t+1}$ and pre-process $\phi_{t+1} = \phi(s_{t+1})$
        Store transition $(\phi_t, a_t, r_t, \phi_{t+1})$ in replay memory D
        Sample mini-batch of transitions $(\phi_j, a_j, r_j, \phi_{j+1})$ from D
        Compute target Q-values y_j for mini-batch as follows:
        if *episode terminates at step $j + 1$* **then**
            $y_j = r_j$
        else
            $y_j = r_j + \gamma . max_{a'} Q'(\phi_{j+1}, a'; \theta')$
        end
        Update parameters of the Q-network by minimizing loss between $Q(s_t, a_t)$ and $Q'(s_t, a_t)$ (perform a gradient descent step on $(y_j - Q(\phi_j, a_j; \theta))^2$ with respect to the parameters θ of the network)
        Every C steps, update Q' with the weights of Q
    end
end

they fail to handle environments with continuous action spaces. This is mainly due to the fact that the output of the network represents the Q-values for a discrete set of actions, and selecting the best action from a continuous range would require intensive additional processing effort. Another limitation of value-based methods is that they may suffer from instability in environments with sparse rewards or long-term dependencies. In these cases, the Q-values may be inaccurate or biased, leading to poor performance.

To address these limitations, policy-based methods have been developed as an alternative to value-based methods. Instead of learning the optimal action-value function, policy-

based methods learn the optimal policy directly, which maps states to actions. This can be more effective in continuous action spaces and environments with sparse rewards.

3.4.3.2 Policy-based DRL methods

Policy-based methods in deep reinforcement learning aim to learn a policy directly, without estimating a value function. In other terms, instead of computing the value of each state-action pair, they directly learn a policy that maps states to actions. One popular policy-based method is the Reinforce algorithm that is based on the policy gradient theorem. It estimates the gradient of the expected return with respect to the policy parameters and updates the policy in the direction of the gradient. Specifically, the update rule scales the gradient by the advantage function, which is a measure of how much better the action taken was compared to the average action taken from that state. The Reinforce algorithm has been shown to work well in high-dimensional and continuous action spaces.
Deep Policy Gradient (DPG) [265] are methods that extend the Reinforce algorithm to Deep Neural Networks by using stochastic gradient descent to update the policy parameters. These methods have been shown to work well in high-dimensional and continuous action spaces, and have the potential to improve sample efficiency and convergence speed compared to value-based methods.

3.4.3.3 Actor-Critic methods

Actor-critic methods are a class of reinforcement learning algorithms that combine the advantages of both value-based and policy-based methods. In actor-critic architectures, there are two eponymous networks: an actor network that generates actions based on the current state of the environment, and a critic network that estimates the value of the current state-action pair. The actor network is trained to maximize the expected return of the policy, while the critic network is trained to estimate the state-value function, which is the expected return starting from the current state. By combining the advantages of both value-based and policy-based methods, actor-critic methods can achieve better performance than either methods alone.
One of the most successful actor-critic algorithms is Deep Deterministic Policy Gradient (DDPG). It is an off-policy algorithm that learns a deterministic policy function in continuous action spaces in order to approximate the optimal policy. It combines the actor-critic approach with the insights from Deep Q Networks (DQN) by using two Deep Neural

Networks, namely an actor network and a critic network. Thus, in addition to being able to handle continuous action spaces, it is also well-suited for high-dimensional state and action spaces.

In DDPG algorithms, the actor network maps the state to a deterministic action, while the critic network estimates the value function by learning the optimal Q-value for the current state-action pair. Unlike DQN, the critic network in DDPG uses the predicted action of the actor network to estimate the Q-value. This is referred to as the *target Q-value*, as it is the value that the critic network is trying to approximate.

DDPG uses experience replay and target networks to stabilize the learning process. The experience replay buffer stores tuples in the form of (state, action, reward, next state), which are randomly sampled to update the networks. The target networks are copies of the actor and critic networks, with their weights updated slowly using a soft update rule to provide more stable target Q-values.

Among the main advantages of DDPG over other Deep Reinforcement Learning algo-

Algorithm 9: Pseudo-code for the DDPG algorithm

Initialize the actor network μ and the critic network Q with random weights θ^{μ} and θ^{Q}
Initialize target networks μ' and Q' with the weights $\theta^{\mu'} \leftarrow \theta^{\mu}$ and $\theta^{Q'} \leftarrow \theta^{Q}$
Initialize the experience replay Buffer D
for $episode \leftarrow 0$ **to** $N_{episodes}$ **do**
    Initialize a random process R for action exploration
    Get initial observation of state S_1 at time step $t = 1$
    for $t \leftarrow 1$ **to** N_{steps} **do**
        Select action $a_t = \mu(s_t|\theta^{\mu}) + R_t$ according to the current policy and exploration noise
        Execute action a_t in the environment and observe the resulting reward r_t and the new state s_{t+1}
        Store the transition (s_t, a_t, r_t, s_{t+1}) in experience replay buffer D
        Sample a random mini-batch of N transitions (s_i, a_i, r_i, s_{i+1}) from D
        Set $y_i(r_i, s_{i+1}) = r_i + \gamma.Q'(s_{i+1}, \mu'(s_{i+1}|\theta^{\mu'})|\theta^{Q'})$
        Update the critic by minimising the loss $L = 1/N \sum_i Q(s_i, a_i|\theta^{Q}) - y_i)^2$
        Update the actor policy using the policy gradient $\nabla_{\theta^{\mu}} 1/N \sum_{s \in B} Q(s, \mu(s|\theta^{\mu})|\theta^{Q})$
        Update the target networks: $\theta^{Q'} \leftarrow (1 - \rho).\theta^{Q} + \rho.\theta^{Q'}$ and $\theta^{\mu'} \leftarrow (1 - \rho).\theta^{\mu} + \rho.\theta^{\mu'}$
    end
end

rithms, it is worth mentioning its ability to learn policies in continuous action spaces,

which is difficult for traditional value-based methods like DQN. Additionally, DDPG is known to be relatively sample efficient and can learn from a small number of samples. This makes it well-suited for real-world applications with limited sample availability.
However, like other actor-critic methods, DDPG can suffer from instability during the learning process. The use of target networks and soft updates help mitigate this issue, but tuning the hyperparameters properly can still be a challenging task.
Overall, DDPG is a powerful actor-critic algorithm that has shown great success in learning policies for continuous action spaces. Its ability to learn from limited samples makes it a promising approach for real-world applications. However, careful tuning of hyperparameters remains necessary to ensure successful training.
Another example of actor-critic methods is the Trust Region Policy Optimization (TRPO) algorithm. TRPO is a policy optimization method that iteratively improves the policy by taking small steps in the policy parameter space while ensuring that the change in the policy does not significantly affect the performance of the agent. It has been shown to be robust to hyperparameter choices and can achieve high sample efficiency.
The well-known Proximal Policy Optimization (PPO) algorithm is a variant of TRPO. It simplifies the optimization procedure by using a clipped surrogate objective function that limits the change in the policy. PPO was shown to be more stable and sample-efficient than TRPO, making it a popular choice for many Deep Reinforcement Learning applications.
Overall, actor-critic methods are a powerful class of reinforcement learning algorithms that can achieve state-of-the-art performance in a wide range of applications. By combining the benefits of value-based and policy-based methods, actor-critic methods provide a versatile and effective approach for solving complex sequential decision-making problems. This has been among the strong motivations behind our choice of using the actor-critic based Deep Reinforcement Learning algorithms in the present research work.

3.5 Literature review of previous work on RL and DRL applications in energy systems

3.5.1 Previous work on Reinforcement Learning

Reinforcement Learning has been used for numerous energy optimization applications including cost optimization in the Smart Grids and District Heating Systems' context. Indeed, among Machine Learning techniques, Reinforcement Learning methods are the most suitable for cost optimization problems since they can learn an optimal strategy, as explained by Mocanu [208].

For instance, Idowu et al. [156] proposed a Reinforcement Learning based approach to optimize energy usage and thus minimize the production costs of CHPs in a District Heating System. The optimization is done in the consumers homes, and believed to lead to an efficiency through the whole District Heating Network.

Similarly, Di Wu et al. [184] proposed a Reinforcement Learning application in the electrical Smart Grid context. The paper deals with the problem of optimal energy management of a residential home with Electric Vehicle charging. The problem is formalized as a Markov Decision Process, with the objective of minimizing the long-term operating costs. Model-free Reinforcement Learning control algorithms were developed to address this problem.

Moreover, Kuznetsova et al. [266] used a Reinforcement Learning algorithm for the optimal energy management of an electrical microgrid composed of wind turbine and Battery Energy Storage System and a local consumer. The Reinforcement Learning algorithm aims at finding an optimal schedule for the battery so as to increase the use rate of the battery during high electricity demand periods, decrease the electricity withdraw from the external grid, and increase the use of renewable electricity produced locally by the wind turbine. When it comes to multi-energy systems, Reinforcement Learning was used for instance for the optimization of energy costs in a multi-carrier system modeled as a Smart Energy Hub (SEH) in [138]. An Energy Management System was developed to find a near optimal solution based on Reinforcement Learning algorithms together with Monte Carlo estimation. The simulation results in this paper show that operating costs can be reduced by up to 40% , peak load reduced by 17% and CO_2 emission social cost reduced

by 50% while keeping the comfort level.

Several studies also focus on the comparison of Reinforcement learning with other optimization techniques like Model Predictive Control. For instance Ernst et al. [267] compared these two approaches on a power system problem, namely electrical power oscillations damping problem, and showed that Reinforcement Learning can definitely be competitive with Model Predictive Control even in contexts where an accurate deterministic model of the system is available.

3.5.2 Previous work on Deep Reinforcement Learning

The application of Reinforcement Learning in complex and real-time control tasks within power systems remained limited due to curse of dimensionality, since such problems generally involve large state and action sets. Deep Reinforcement Learning approaches were recently proposed as a promising solution to overcome this limitation. That is why an increasing focus is currently made on application of DRL for the optimal energy management of energy systems. For instance, there have recently been several successful applications of DRL within the energy sector in the context of microgrids, smart homes, Smart Grids and District Heating Systems. For instance, in the context of Smart Electrical Grids, two DRL algorithms, namely Deep Q-Networks (DQN) and Deep Policy Gradient (DPG) have been used in [208] for building energy optimization in a smart grid. Similarly, François-Lavet et al. [19], [268] proposed a Deep Reinforcement Learning solution for the energy management of an electricity microgrid featuring PV panels and both a long term and short term storage systems (respectively a hydrogen storage and battery). The problem of optimally operating these storage systems is formulated as a partially observable Markov Decision Process (MDP) where the decision is taken under uncertainty. The considered uncertainty comes mainly from the electricity consumption and the PV production. The developed Deep Q-Network (DQN) agent was tested on the case of a residential customer microgrid located in Belgium and showed to successfully extract knowledge from the past PV production and electricity consumption time series. Even though this study empirically demonstrates that the proposed DRL model generalizes sufficiently well to unseen situations of electricity demand and PV production, it has some inherent limitations. First, it does not account for the microgrid's interaction with the main utility grid. Second, the model focuses exclusively on electrical load demand and does not consider other energy vectors and usages. Besides, it does not consider

multiple actions: the DRL agent's actions are restricted, with direct control limited the hydrogen storage, while actions on the battery storage are dynamically adapted based on the balance equation of the microgrid. Lastly, the DRL algorithm used in this work (DQN) only permits discrete actions (i.e charging at maximum rate, discharging at maximum rate or remaining idle), limiting the granularity of the control. These four limitations are addressed in the current research work through the case studies that we introduce in chapters 5 and 6 of this dissertation.

In 2017, Mocanu [208] introduced a Deep Reinforcement Learning approach for building energy optimization. To our knowledge, this was the first time that the Deep Policy Gradient algorithm is used in a Smart Grid context. The developed solution aimed at the online optimization of the planning of electrical devices for both residential buildings and an aggregation of buildings. Besides the Deep Policy Gradient method, Deep Q-Networks ware also proposed for solving the same decision Making problem in this work. Similarly to [268], the sequential decision making problem was formulated as a Markov Decision process.

DRL has also been used for handling continuous action spaces in the Smart Grid context. For instance, the work of Mocanu [208] extended DRL approaches for unsupervised energy prediction using SARSA and Deep Q-Learning together with Deep Belief Networks for continuous action spaces. Hirata et al. [269] compared the results of a Deep Reinforcement Learning approach with those of a MILP for Smart Grid optimization and showed that the DRL agent successfully learnt to adjust its action during the training phase to maximize the reward.

On the other hand, some recent research works also focused on the use of Deep Reinforcement Learning Strategies in District Heating Systems. For instance, Zhang et al. [163] presented a Deep Reinforcement Learning method for the flow rate control of District Heating Systems during the heating process. Simulated experiments on a real life case study showed savings of about 552 MWh of heat quantity and 42276.45 tons of water per hour compared to a manual control.

When it comes to Smart Multi-Energy Systems, there is still a lack of research works that consider Deep Reinforcement Learning for the intelligent control in this context. In fact, the only research work where Deep Reinforcement Learning is applied to the optimization in an integrated electrical and heating system is, to our knowledge, the work of zhang

et al. [22] where DRL is applied for the control of energy conversion of wind power to minimize operating costs in an integrated energy system.

The paper of Glavic et al. [270] presents a detailed survey of past applications of the RL paradigm for solving electric power system decision and control problems. More recently, Chen et al. [245] presented recent advances and future challenges within using RL for selective key applications in power systems. Among major applications of RL in power systems, they highlighted frequency regulation, voltage control and energy management. Figure 3.2 presents the schemes for RL-based decision-making in the power systems context.

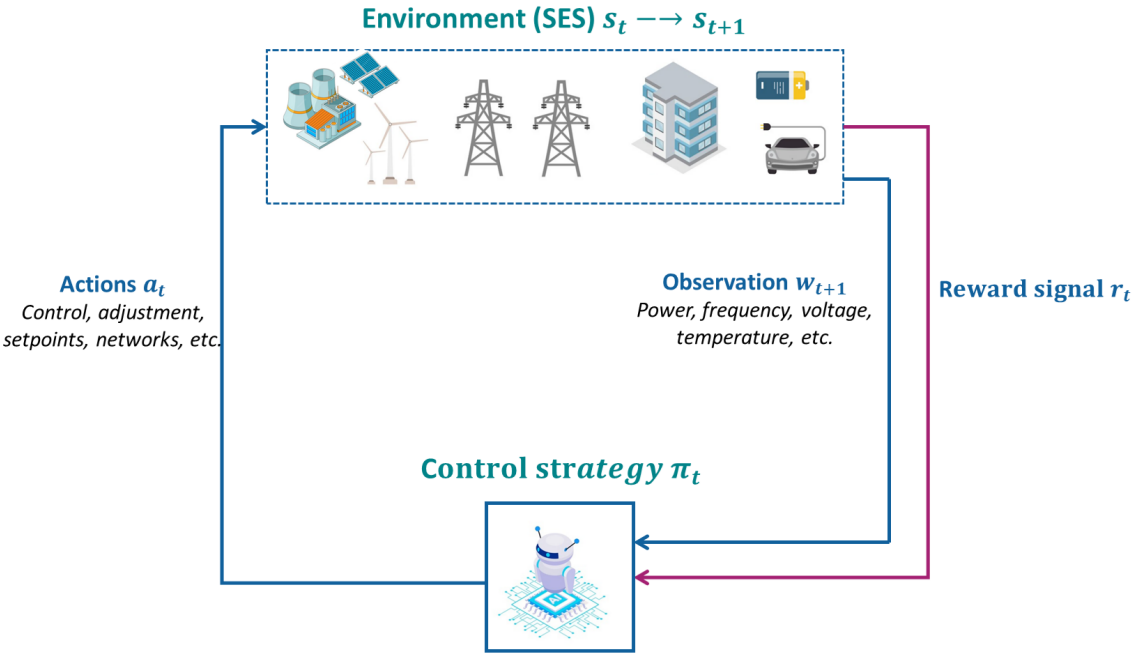


Figure 3.2: RL shemes for power systems control applications, adapted from [245].

The present work deals with the energy management applications. The interested reader can refer to [245] for a review of RL-based applications in frequency and voltage regulation, and to [271] for a more general review of the research and practice of Reinforcement Learning and Deep Learning in Smart Grids. For instance, besides the previously mentioned work of François-Lavet et al. [268], Ji et al. [16] also proposed a DQN approach to develop real-time generation schedules for a microgrid while optimizing its daily operational costs. DQL algorithms have also been applied in the work of Qin et al. [272] for the coordinated operation of wind farms and energy storage and in

the paper of Lin et al. [273] for the on-line optimization of a microgrid featuring PV and wind generation, diesel generators, fuel cells, electric load and a BESS. Among the various DRL algorithms, the conventional DQL remains the most widely used approach and algorithms such as Deep Policy Gradient (DPG) and Actor-Critic (AC) are rarely investigated. This is primarily due to the simplicity of the DQL and to the fact that it handles well discrete action spaces. Meanwhile, DQL can not be directly applied to problems with continuous action spaces since they need to discretize the action space which leads to an explosion of the number of actions and, as a consequence, to a decreased performance [274], [275]. Indeed, considering only discrete actions for the planning and control of the Smart Grid components significantly restrains their flexibility potentials and prevents from obtaining the best optimal scheduling and control strategies. Unlike DQL, Deep Policy Gradient (DPG) algorithms are capable of dealing with environments with continuous actions spaces. In this respect, Mocanu et al. [276] proposed the use of DQL and DPG for online building energy optimization through the scheduling of electricity consuming devices. The results showed that DPG algorithms are more suitable than DQN to perform online energy resources scheduling. Even though this work pioneered the use of DRL for online building energy optimization, the actions it considers are restricted to the on/off status of flexible load devices in a smart building. Besides, the DPG algorithms are also often criticized for the fact that their gradient estimator may have a large variance, which is likely to lead to slow convergence [277]. In order to overcome this limitation, Actor-Critic (AC) algorithms were proposed to combine the strong points of DPG and DQL approaches by estimating both the policy and the Q-value function during the training. In this respect, two DRL algorithms were designed for Smart Grid optimization in [17]: on the one hand, DQL was applied for the discrete action control tasks like charging/discharging the BESS or switching the buy/sell modes of the grid. On the other hand, an AC algorithm named H-DDPG (Hybrid-Deep Deterministic Policy Gradient) was developed to deal with continuous state and action spaces. Yet, only the results of the DQN approach were presented in the paper and benchmarked with the results of a Mixed Integer Linear Programming (MILP) optimization Matlab tool. Even though DDPG algorithms were proposed for some applications in the energy systems context, namely for dealing with cost optimization problems in Smart Home energy systems in [278], for flow rate control in Smart District Heating Systems (SDHS) in [163], and for solving the Nash

Equilibrium in energy sharing mechanisms in [279], most of these applications consider mono-action and/or mono-fluid use-cases. In other words, they consider solely electrical or thermal Smart Grids and do not consider jointly optimizing the uses of several energy vectors within Smart Multi-Energy Systems. Besides, most of the previous works consider applications on the Smart Home or building level and do not consider testing these approaches on a larger smart district level. Finally, thourough comparisons of the performance of DPG or Actor-Critic based approaches with other widely used techniques like MPC for dealing with energy management systems in Smart Grids have rarely been reported in the literature.

3.5.3 Contribution of the present work

In the present work, we propose an Actor-Critic-based approach to deal with the real-time energy management of smart multi-energy systems. More specifically, we formulated the optimal control problem as an MDP and developed a DDPG agent to perform real-time scheduling of the multi-energy systems within the Smart energy system. This approach was applied on two smart multi-energy system cases-studies detailed respectively in chapters 5 and 8. The main contributions of the present work are the following:

- Unlike most of previous works where mono-fluid (electrical or thermal) Smart Grids are considered, we focus on multi-energy (electrical, heating, cooling, hydrogen) smart grids that interact with the main utility grid. A variable electricity price signal is considered and a DRL-based energy management system is developed to take price-responsive control actions.

- The DDPG algorithm is proposed instead of the mostly used DQN to deal with the continuous action and state spaces inherent to the smart multi-energy system model. At each time step of the control horizon, multiple continuous actions are simultaneously taken by the DDPG agent to optimally schedule the different storage systems as well as the thermal production units.

- The proposed approach is first tested on a simplified residential multi-energy system model then on a more complex case-study represented by a detailed digital twin. Ultimately, this approach will be applied on a real-life district-level smart multi-energy system that is currently under construction in France. More specifically, the

developed DDPG agent is aimed at operating real-time energy management of the various energy systems within an eco-district: heating and cooling storage systems, a battey energy storage system and a geothermal District Heating and Cooling system. At a later stage of this project, the scope of the energy systems controlled by the agent will be extended to the Electric Vehicle Charging Stations and the public lighting of the district as well as controllable loads of the buildings and heated water storage tanks. Simulation results of these two case studies are discussed in chapters 5 and 8 of this manuscript.

- The proposed approach is benchmarked with an MPC-based approach. A comparative study through the two Smart Energy System case studies is carried out to evaluate the trade-off between performance and computational time of these approaches. To the best of our knowledge, this work represents one of the initial studies that simultaneously benchmark Deep Reinforcement Learning and Model Predictive Control in a Smart Multi-Energy Systems' context, besides the work of Ceusters et al. [37] that was conducted simultaneously with our research work [280].

3.6 Conclusion

This chapter introduced the theory of Reinforcement Learning as well as its combination with Deep Neural Networks as function approximators, that gives raise to Deep Reinforcement Learning. These methods can be used to solve sequential decision making problem modeled as MDPs. Despite the numerous successful applications of DRL on various domains, most of these applications are still much more focused on academic than on real world applications. Unlike DRL, Model Predictive Control (MPC) techniques have shown numerous successful real life applications and were largely adopted in practice [267], [281]. That is why, we use MPC as a benchmark method to evaluate the DRL approaches developed in the present work. The next chapter reviews the theory as well as previous applications of MPC for the optimal energy management in Smart Energy Systems.

Chapter 4

Model Predictive Control: theory and applications in Smart Energy Systems

Résumé

Ce chapitre commence par une présentation de la théorie du Contrôle Prédictif basé sur les Modèles (MPC), une stratégie de contrôle en boucle fermée avancée qui remonte à la fin des années 1970 et qui compte des milliers d'applications réussies en matière de contrôle de processus, tant dans le milieu académique qu'industriel. On explore ensuite comment le MPC peut être appliqué pour résoudre les problèmes décrits dans les deux premiers chapitres de ce manuscrit. Enfin, on effectue une brève revue des applications des approches basées sur le MPC pour la gestion des systèmes énergétiques.

4.1 Introduction

Model Predictive Control (MPC) is an advanced feedback control strategy that dates back to the late 1970s and has thousands of successful process control applications in both academic and industrial fields. This chapter introduces MPC theory and how it can be applied to handle the problems described in the first two chapters of this manuscript. Applications of MPC-based approaches for energy systems management are then briefly reviewed.

4.2 Predictive optimization approaches

Model Predictive Control (MPC), also known as Receding Horizon Control (RHC) [9], [10] belongs to the family of predictive optimization techniques. Predictive optimization techniques are methods that explicitly consider forecasts in the decision making process. For instance, in the case of energy and power systems control, one considers forecasts on future uncertain quantities like loads, renewable power generation, prices, as well as weather. Three types of predictive optimization approaches can be distinguished:

- deterministic approaches: that suppose to have perfect forecasts and thus do not consider uncertainty on these forecasts.
- stochastic approaches: which consider uncertainty on forecasts, using probability distributions (chance-constrained stochastic programming) or recourse functions (stochastic optimization with recourse) [282].
- robust approaches: which are more conservative paradigms that consider the optimal control for the worst case [223], [283], [284].

4.3 Model Predictive Control

MPC is one of the most popular and widely used predictive approaches for the optimal control. As presented in chapter 2, it consists in a feedback control method where the optimal control problem is solved at each time step to determine a sequence of control actions over a fixed time horizon. The sequence of control actions is computed based on a model of the dynamical system and its predicted future evolution [285], [286]. The control objective and the mathematical model are formulated as a real-time optimization problem that repeatedly computes the sequence of control actions. Only the control action of this sequence that is associated with the current time step is then applied to the controlled system and the new resulting system state is measured. At the next time step, the time horizon is moved one step forward and a new optimization problem is then solved, taking into account the new system state and updated forecasts of future quantities. The operation scheme of MPC is illustrated in figure 4.1.

The receding time horizon and the periodic adjustment of the control actions make the MPC robust against the uncertainties that are inherent to the model and forecasts [11]. In

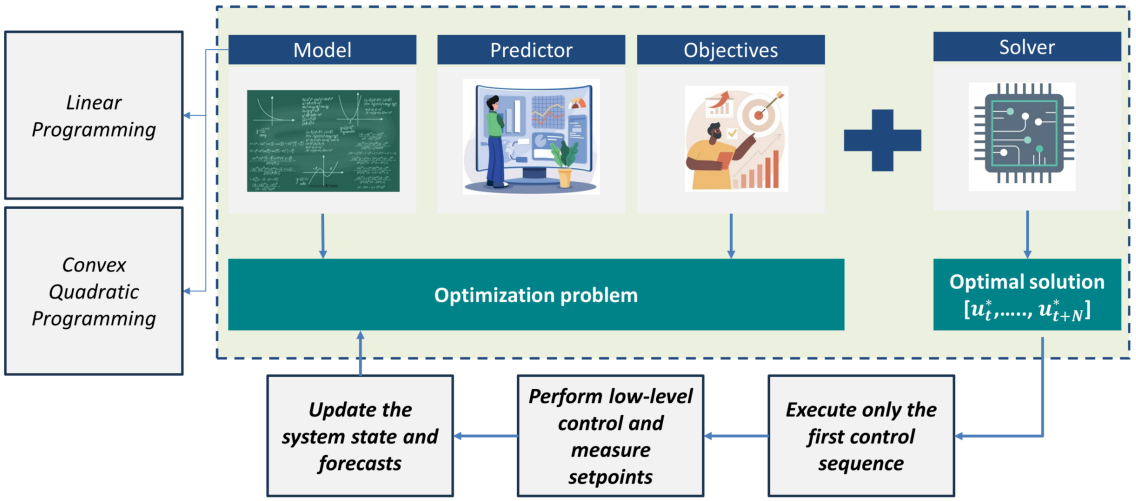


Figure 4.1: Model Predictive Control scheme.

fact, the regularly repeated optimization procedure provides a closed-loop feedback that allows the MPC to override model uncertainties and external disturbances. Nevertheless, this procedure is computational intensive since large-scale optimal control problems have to be solved numerically in real time.

In the remainder of this section, we briefly introduce four of the most widely used variants of MPC, namely linear MPC, non-linear MPC, data-driven MPC and learning-based MPC.

4.3.1 Linear MPC

Linear models are among the most widely used prediction models within MPC. This is referred to as linear MPC and results in a Linear Program (LP) or a convex Quadratic Program (QP) optimization model for which fast optimization algorithms have been developed in the past decades [287]. Such advances in solution methods for linear MPC allowed its application for larger scaled systems as well as systems with fast dynamics. In fact, MPC is able to approximate and solve most optimal control problems numerically with much lower computational effort than more classical approaches like dynamic programming, and without facing the curse of dimensionality.

One additional great advantage of MPC over other approaches is that system limitations such as the maximum capacity or the maximum charge and discharge power of an energy storage system can be directly handled by adding them as constraints in the optimization problem of the MPC.

4.3.2 Nonlinear MPC

Nonlinear MPC, denoted NMPC, is the extension of the well-established linear MPC to the nonlinear world for applications where process nonlinearities and nonlinear constraints need to be explicitly considered. It therefore holds much promise for more complex practical cases than the traditional MPC for linear-constrained systems. For instance, Schirrer et al. [288] proposed an NMPC for the optimal control of heating and cooling activities for a low-energy office building. Nontheless, the incorporation of nonlinear models that NMPC involves also brings about more challenging problems related mainly to computational and theoretical control difficulties [289]. Overall, linear MPC remains the prevailing choice over nonlinear MPC in current practice.

4.3.3 Data-driven MPC

The data-driven modeling approach is based on learning from historical measurable data and without modeling physical details of the system. In fact, despite its effectiveness, MPC faces major challenges coming mainly from the necessity of accurately modeling the dynamics of the considered physical system. Several authors pointed out that capturing such an accurate dynamical model requires time, cost and effort [290], [291]. To overcome this limitation that challenges the cost-effectiveness of the MPC as an energy management method, data-driven MPC can be used to reduce the modeling costs and exploit real data from existing systems. It can be particularly useful when accurate models of the system are not readily available or when the system dynamics are too complex or difficult to be modeled using traditional methods. It also has the advantage of being able to adapt to changes in system behaviour and to provide robust control performance based on real-time data. For instance, Smarra et al. [228] used a data-driven MPC using random forests for the building energy optimization and climate control. Similarly, Ferreira et al. [292] proposed a Neural Network-based MPC approach for the optimization of thermal comfort and energy savings in public buildings. The Artificial Neural Networks are used as predictive models and the discrete optimization problem considered is solved using the Branch and Bound method.

4.3.4 Learning-based MPC

In Learning-based Model Predictive Control (LB-MPC), a learning technique such as Reinforcement Learning, Neural Networks or Gaussian processes, is used to enable the MPC to learn from incorporating experience built up through the interaction with the system during the decision making process. This approach allows combining the advantages of MPC and Machine Learning techniques: the MPC part is used to provide the robustness and decision making over the relatively short term, while the learning part is used to provide robustness, decision making and adaptation over the long term.
When the learning technique used is Reinforcement Learning, the approach is then denoted RLB-MPC [293]. In this case, the Reinforcement Learning agent learns a value function which is progressively taken into account as experience increases. This allows to speed up the decision making process, to take control actions over an infinite time horizon instead of the inherently finite time horizon of the MPC, and to adapt the actions to slowly changing system and desired performances as explained by [227]. This paper proposes a learning-based MPC approach to control systems described by Markov Decision processes. The approach initially used a pure MPC, and experience is then more and more taken into account as experience increases over time. Once the built up experience is sufficient, the agent starts fully using it and thus requires less computational burden than with non-learning approaches.

4.4 Model Predictive Control applications in energy systems

MPC is an advanced control mechanism that has thousands of successful applications in numerous fields like industrial and chemical processes, supply chains, economic and finance stochastic control, control of hybrid electric vehicles, automotive applications and aerospace applications [294].
When it comes to the field of energy systems and their optimal management, MPC has well-reported advantages over more traditional approaches, as mentioned for instance by O'Dwyer et al. [295] and Killian and Kozek [296]. It thus had many successful applications in this domain, ranging from smart building control [297], [298] to microgrid

control [299] and district-level energy management [300], [301]. One of the simplest and most used versions of MPC for these applications is the certainty-equivalent MPC that consists in replacing the unknown quantities by their available forecasts to solve the optimization problem.

4.4.1 Applications in microgrids and Smart Grids

MPC-based control approaches for energy management systems of microgrids are proposed for example in Parisio et al. [13] for the efficient optimization of a microgrid where the overall optimization problem is formulated as a Mixed Integer Linear Program (MILP). The method was applied to an experimental microgrid located in Athens, Greece. An experimental case study was also conducted in [302] where a two-stage stochastic approach was adopted for the energy management problem of the microgrid to take into account the uncertainties due to the fluctuation of demand and renewable energy generation. The stochastic optimization problem was stated as a MILP and incorporated in an MPC framework in order to further compensate the uncertainty. In [15], a stochastic MPC approach was developed to take into account uncertainties on frequency deviations in the optimal control of a PV-battery microgrid used to participate in the frequency control market. Similarly to [302], the two-stage stochastic optimization problem is stated as an LP and incorporated in an MPC framework to further compensate uncertainties.

MPC was also used for the operational optimization of combined heat and power microgrids. For example, Gambino et al. [303] used an MPC for the optimal control of a combined electric and heat power microgrid. The problem was formulated as a MILP model and MPC is used to take the system uncertainties into account. The developed algorithms have been applied on a microgrid located in Finland and the proposed approach is compared to a heuristic one. Tang et al. [304] developed an MPC-based strategy for the optimal control of thermal storage of commercial buildings in a smart grid during fast Demand Response (DR) events.

A robust MPC has been used in [305] for the planning and control at various scales in the electrical grid like smart homes, shared electric vehicle charging stations and wind farms integrated in the transmission network. The approach is based on providing scenarios as inputs to the optimization problem and optimizing the worst-case performance over those scenarios. At the scale of a smart home, Gelleschus et al. [11] considered the control of home energy systems composed of a PV-battery-heat pump system with energy storage

via an MPC controller. Different optimization solvers are examined for the MILP optimization problem of the MPC such as Branch-and-cut-based solvers, a hybrid Genetic Algorithm and dynamic programming. The simulation results showed that the branch-and-cut algorithms performed best with respect to reliability and computational time criterion. The authors explained that genetic algorithms fail to find feasible solution because of the many constraints in the problem's formulation and the dynamic programming fails to improve the calculation time because of the many states of the system. Jorissen et al. [306] also applied MPC at home energy systems' level. The paper considers the problem of building heating ventilation and air conditioning systems control. The proposed approach is based on a white-box NMPC.

4.4.2 Applications in District Heating Systems

MPC-based strategies were also applied in the Distict Heating Systems (DHS) level. For instance, Gambino et al. [307] considered the problem of operational optimization of a district heating power plant with thermal energy storage. The optimal control strategy aimed at reducing the operational costs by optimally managing the boilers, thermal storage and load curtailment. The optimization problem is formulated as a MILP and incorporated in an MPC framework. In addition, Verilli et al. [308] proposed a MILP-based MPC controller for the optimal operation of a district heating power plant with flexible loads and thermal energy storage. In [309], the authors developed a stochastic MPC approach for the optimal energy management of a district heating power plant. A two-stage stochastic formulation for the optimization problem of the MPC framework is proposed to deal with the uncertainty on the power demand, the renewable energy generation, as well as the weather conditions. Lie-Jensen et al. [310] also considered the problem of unit commitment and heat production unit control through the case study of a DHS in Norway. The authors suggested the use of MPC for the optimal control of flow rate and heat production units, the MILP optimization approach for the unit commitment problem and multi-linear regression for the heat load forecast. Cupeiro Figueroa et al. [311] proposed a methodology that introduces a shadow-cost in the objective function of an MPC while dealing with the optimal control of hybrid geothermal systems. In fact, the finite-horizon nature of the MPC makes it hard to properly consider long term objectives and constraints. In this paper, a prediction horizon of the weather forecasts in the order of days is used within the MPC framework in order to enable the system to react in

advance to changes that occur on the heating and cooling loads of buildings. However, when geothermal heat pump systems are considered, the ground dynamics are in the order of months and even years. Thus, the prediction horizon in the order of days may be sub-optimal. That is why, current research works in the field of MPC focus on tackling this challenge for instance by adding shadow-costs to the objective function as in [311].

4.4.3 Applications in multi-energy systems

A few works considering the application of MPC in multi-energy systems have also been reported. For instance, Huang et al. [312] proposed an MPC-based strategy for the daily operational optimization of a multi-energy-system composed of an alkaline electrolyzer that converts renewable power into heat and green hydrogen. The system is modeled as a dynamic power-to-hydrogen-and-heat model, and the MPC-based approach is compared with a traditional rule-based approach and showed operational cost savings of about 59%. Blaud et al. [313] also considered the comparison of MPC and rule-based approaches in multi-energy system case studies. The paper proposes the modeling of multi-energy systems' dynamics based on Multi-Prosumer Node (MPN) formulation. The MPC controller aims at minimizing economic costs and takes into account forecasts on loads, weather, renewable power generation and cost of grid power purchase. The simulation results also showed cost savings of 84.24% in summer and 8.21% in winter with respect to a rule-based control benchmark.

Arnold et al. [314] proposed an MPC approach for the optimal control of multi-carrier energy systems modeled using the energy hub concept. They considered mainly the coupling of electricity and gas energy systems equipped with storage systems to boost their efficiency and reliability. The same authors proposed in [315] the optimal control of coupled electricity and gas networks modeled as energy hubs using a distributed MPC scheme. An Artificial Neural Network-based MPC was introduced in [316] for the optimal management in District Cooling Systems (DCS) belonging to multi-energy systems. The multi-energy systems considered in this work include PV panels, connection to the power grid and variable-load air-to-water heat pumps used as backup systems, along with the DCS. The designed MPC controller aims at reducing the electrical energy withdrawn from the power grid for the backup cooling systems in cases of temporal mismatch between energy demand and supply. Last but not least, Aliu [317] focused on the incorporation of electric vehicles with high penetration in the optimal operation of multi-energy

systems using a stochastic MPC scheme. This approach allows to address uncertainties related to the characteristics, availabilities and owner charging preferences of electric vehicles.

4.5 On the relationship between MPC and DRL

While DRL and MPC are both approaches for system control, their main differences lay in their way of approaching the control problem. Indeed, DRL is a model-free method where the agent learns an optimal strategy by trial and error through interaction with the environment, whereas MPC is a model-based method where the controller relies on a model of the system dynamics as well as a prediction of its future behavior in order to generate a control signal. Another key difference is also the level of problem complexity they can be handled by each approach. While DRL has been successfully applied to solve complex control tasks, MPC would be better suited for problems with known system dynamics since it can handle constraints on the system variables more efficiently. A comparison of MPC and DRL properties in summarised in Table 4.1.

Overall, the performance of reinforcement learning and model predictive control algorithms depends on the specific application and problem considered. While there is no absolute winner between these two approaches, researchers continue to explore the strengths and weaknesses of each of them and look for ways to combine them for a boosted performance. That is why, several studies have recently been focusing on the comparison of Model Predictive Control and Reinforcement Learning. For instance, Gorges [258] reviewed the principles of both methods and studied their relations for discrete-time linear time-invariant systems. Alamir [318] also considered the comparison of three approaches, namely Reinforcement Learning, stochastic Model Predictive Control and certification via randomized optimization for learning against uncertainty in control engineering problems. Many studies considered comparing these two approaches through applications in several fields like building energy management [319], [320] and electrical power oscillations damping [267]. To the best of our knowledge, no studies prior to our work considered bench-marking DRL and MPC for the optimal energy management in Smart Multi-Energy Systems. Indeed, the only work in literature considering this subject is the paper of Ceusters et al. [37] which was conducted simultaneously to our research work.
In this work, we propose the use of Deep Reinforcement Learning for the optimal en-

Table 4.1: Comparison of MPC and DRL properties, adapted from [258].

MPC		DRL	
Disadvantages	A model is required Convexity is usually required Adaptivity is immature (usually based on robustness) Online complexity is high (except for explicit and neural MPC)	Advantages	No model required Convexity is not required Adaptivity is inherently mature Online complexity is low
Advantages	Offline complexity is low stability theory is mature (e.g. based on terminal cost) Feasibility theory is mature (e.g. based on terminal constraints) Robustness theory is mature Constraint handling is inherently mature	Disadvantages	Offline complexity is high Stability theory is immature Feasibility theory is immature Robustness theory is immature Constraint handling is immature (except for input constraints)

ergy management in smart multi-energy systems through two multi-energy system case studies. These case studies are both drawn from the Meridia smart Energy (MSE) multi-energy system project. The first case study is referred to as "case study 1" or "toy model case study", since it serves as a fundamental starting point by presenting a simplified representation of the multi-energy system. It is implemented in Python and it served as a Proof of Concept (PoC) by capturing essential elements of the system while maintaining a manageable level of complexity. This case study is presented in the next chapter of this manuscript, together with implementation details and simulation results of applying the deep reinforcement learning approach and the MPC-based approach on this first simulated case study. The second case study, denoted "case study 2" involves a comprehensive model and an advanced and more detailed representation of the dynamics of the multi-energy system under the Modelica language. This case study as well as its simulation results are the subject of part II of this dissertation.

4.6 Conclusion

This chapter presented a brief theory and practice overview of the Model Predictive Control strategy. For a more in depth review of the MPC theory, the interested reader can refer for instance to the work of Garcia et al. [9]. In the present work, MPC is used as a benchmark approach to evaluate the performance of the Reinforcement Learning-based approaches developed for the optimal energy management in smart multi-energy systems. Even though NMPC may exhibit a better performance, we opt for the use of LMPC as a benchmark approach basically to align with the prevailing common practices and current industry standards. In the next chapter of this dissertation, we compare simultaneously Deep Reinforcement Learning and Model Predictive Control through their application on a first Smart Multi-Energy System simulated case study.

Chapter 5

Deep Reinforcement Learning and Model Predictive Control in Multi-Energy System case study 1

Résumé

Plusieurs études ont signalé les similitudes entre le Contrôle Prédictif basé sur les Modèles (MPC) et l'Apprentissage par Renforcement Profond (DRL), tant de manière formelle [321] que par des simulations [267]. D'autres les ont présentés comme deux approches complémentaires [322]. Dans ce chapitre, nous comparons ces deux approches à travers une étude de cas simulée d'un système multi-énergie. Le cas d'usage proposé s'inspire du système multi-énergie intelligent Meridia Smart Energie (MSE). Il est simulé en Python, et le problème de contrôle correspondant est formulé à la fois sous la forme d'un processus de décision markovien et sous la forme d'un problème d'optimisation linéaire à horizon glissant. La première formulation permet de résoudre le problème en utilisant une approche d'Apprentissage par Renforcement Profond (DRL), tandis que la seconde permet d'utiliser une approche basée sur le MPC. Ce chapitre présente l'étude de cas, la formulation du problème, les détails de mise en œuvre et les résultats de simulation et de comparaison des solutions basées à la fois sur le DRL et le MPC pour cette première étude de cas.

5.1 Introduction

Several studies have reported the similarities that exist between Model Predictive Control (MPC) and Deep Reinforcement Learning (DRL) formally as in [321] and by simulations [267], and others presented them as two complementary frameworks [322]. In this chapter, we compare these two approaches through a simulated multi-energy system case-study. The use-case under consideration is inspired from the Meridia Smart Energy (MSE) multi-energy system. It is simulated in Python and the corresponding sequential decision making problem is formulated both as a Markov Decision process and a receding horizon linear optimization problem. The first formulation allows for solving the problem using a Reinforcement Learning (RL) approach and the second allows for using an MPC-based approach. The case study, problem formulation, implementation details and simulation results of both the DRL- and MPC-based solutions for this first case-study are presented in this chapter.

5.2 Case study description

5.2.1 The use case

The multi-energy system use case considered in case study 1 is drawn from the Meridia Smart Energy System (MSE) project and considers the following energy systems in a simplified representation: residential electric heating and cooling loads, distributed renewable energy generation (by PV panels), a Thermo-Refrigerating Heat Pump (TRHP) that consumes electricity and produces simultaneously heat and cold for the buildings thermal needs, a Battery Energy Storage System (BESS), a heat storage system and a cold storage system. All these components are connected to the public utility grid. The grid connection is assumed to be sufficiently large that the electrical demand of the overall system can always be fulfilled. The structure of the multi-energy system is shown in figure 5.1 and the properties of each of the energy systems are given in table 5.1. At each time step, the electric loads of the buildings are met by the local PV generation, by discharging the BESS or by withdrawing electricity from the public utility grid. The heating demand, on the other hand, is met either by directly producing heat via the heat and cold production unit (TRHP) or by discharging the heat storage system. Similarly, cooling loads are ensured by directly producing cold by the heat and cold production units or by discharging

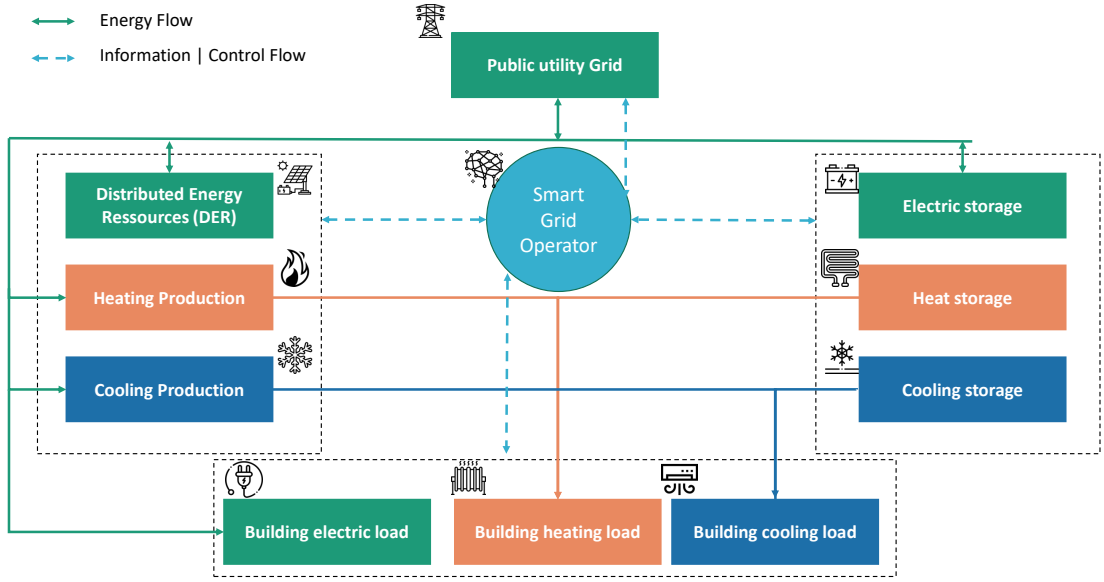


Figure 5.1: Architecture of the multi-energy system considered in case-study 1 [280].

the cooling storage system.

The objective of the optimal control problem considered here is to ensure the optimal operation of the different storage systems in a way that minimizes the overall energy consumption costs within the multi-energy system. This sequential decision making problem was formulated as a Markov Decision Process (MDP) for the RL solution, and as a Linear Programming (LP) optimization problem for the MPC solution. These two formulations are detailed below.

5.2.2 MDP problem formulation

This optimal energy management problem aims at operating the controllable units of the multi-energy system, specifically the three energy storage systems in this case study, while minimizing the daily operational costs. To solve this sequential decision making problem, we formulate it as a Markov Decision Process (MDP). In fact, the energy level of each of the energy storage systems, at each time step, depends only on the energy level and the charge/discharge power of the previous time step. Hence, the scheduling of the different energy storage systems and production units satisfies the Markov property and can be formulated as an MDP denoted $M = (S, A, T, R, \gamma)$ where its key components, the state space S, the action space A, the reward signal R and the transition function T are designed as follows:

Table 5.1: Properties of energy systems of case study 1.

Parameter	Value
Size of the battery ξ_{bat}	1500 kWh
Battery charge efficiency η_{bat}	90%
Battery discharge efficiency η_{bat}	90%
Size of the heat storage ξ_{HS}	1200 kWh
Heat storage charge efficiency η_{HS}	75%
Heat storage discharge efficiency η_{HS}	75%
Size of the cold storage ξ_{CS}	800 kWh
Cold storage charge efficiency η_{CS}	75%
Cold storage discharge efficiency η_{CS}	75%
Peak power generation of PV	600 kWp
Maximal heat/cold generated by TRHP	1500 kW

- State space: the environment state at each time step $t \in H$ is denoted by s_t and is composed of six types of information:
  $s_t = (s_t^{Storage}, s_t^{Load}, s_t^{Gen}, s_t^{Grid}, s_t^{Prod}, s_t^{Temp})$ where $s_t^{Storage} \in S^{Storage}$ denotes the storage operation of the Smart Energy System and describes the amount of energy stored in each of the electric, heating and cooling storage systems: $s_t^{Storage} = (s_t^{Bat}, s_t^{HS}, s_t^{CS})$. $s_t^{Load} \in S^{Load}$ contains the electric, heating and cooling loads denoted $s_t^{Load,e}, s_t^{Load,h}$ and $s_t^{Load,c}$ respectively. Similarly, $s_t^{Gen} \in S^{Gen}$ contains the current amount of distributed energy generation. In our case-study, it consists of PV generation. $s_t^{Grid} \in S^{Grid}$ contains the electricity prices λ_t as well as the amount of power P_{Grid} withdrawn or injected into the main utility grid at time step t. The convention used for the grid power P_{Grid} is such that its value is positive when power is drawn from the grid and negative if power is supplied to the grid. Finally, $s_t^{Prod} \in S^{Prod}$ contains the quantities of heat and cold produced by the TRHPs at time step t and $s_t^{Temp} \in S^{Temp}$ contains the outdoor temperature.

- Action space: the aim of the energy management system is to decide the charging/discharging power of each energy storage system $P_t^{SS} = (P_t^{Bat}, P_t^{HS}, P_t^{CS})$, the amount of energy to be purchased from the public utility grid P^{Grid} and the thermal energy (heat or cold) produced by the TRHPs Q^{TRHP}. The actions on the energy storage systems are composed by charge / discharge powers of the battery

P^{Bat}, the heat storage system P^{HS} and the cold storage system P^{CS} such that:

$$P^{Bat,min} \leq P_t^{Bat} \leq P^{Bat,max} \qquad \forall t, \tag{5.1}$$

$$P^{HS,min} \leq P_t^{HS} \leq P^{HS,max} \qquad \forall t, \tag{5.2}$$

$$P^{CS,min} \leq P_t^{CS} \leq P^{CS,max} \qquad \forall t, \tag{5.3}$$

where $P^{Bat,min}$ is the minimum battery power, i.e maximum battery discharging power, $P^{Bat,max}$ is the maximum battery charging power, $P^{HS,min}$ the maximum heat storage discharging power, $P^{HS,max}$ its maximum charging power, $P^{CS,min}$ the maximum cold storage discharging power and $P^{CS,max}$ its maximum charging power. It is worth noting that P_t^{Bat}, P_t^{HS} and P_t^{CS} can have either positive or negative values (positive values meaning charging the storage system and negative values meaning discharging it), and that these are direct actions to be taken by the agent, while P^{Grid} and Q^{TRHP} are dynamically adapted according to the balance equations of the multi-energy system.

- Reward signal: when an action $a_t \in A_t$ is applied on the system, this triggers the environment to move from state s_{t-1} to state s_t and hence a reward r_t is obtained. Since the aim of the agent is to minimize the total energy costs within the Smart Energy System, the reward signal r_t corresponds to the negative of rescaled instantaneous operational revenues at time step t:

$$r_t = -\alpha.[C_t^{Gen}.P_t^{Gen} + C_t^{Grid}.P_t^{Grid}] \tag{5.4}$$

Where C_t^{Gen} is the cost of distributed power generation and C_t^{Grid} is the cost of power purchase from the public utility grid i.e. the variable energy price, and α is a factor by which we re-scale the cost function, such that

$$0 < \alpha \leq 1 \tag{5.5}$$

5.2.3 LP problem formulation

The sequential decision making problem defined by the MDP above can be formulated as an LP optimization problem. In fact, the inter-conversion between MDP and LP formulations consists in translating the dynamics of a sequential decision-making problem into two distinct mathematical frameworks. While in the MDP formulation, the problem's temporal evolution is captured through states, actions, transition probabilities, and immediate rewards, translating this into an LP formulation consists in converting actions into decision variables that can take on either real or integer values, converting states into variables and replacing transition probabilities with deterministic constraints that enforce the state transitions. The heart of the transformation lies in the reward signal, which evolves from driving decision-making in the MDP to shaping the objective function in the LP. This translation necessitates capturing the problem's dynamics through a series of constraints that maintain energy balances, adhere to storage system capacities, and satisfy demand requirements. The LP formulation that we propose for the sequential decision making problem of the case study considered in this chapter is as follows:

$$\min \quad \sum_{t=1}^{H} C_t^{Gen}.P_t^{Gen} + C_t^{Grid}.P_t^{Grid} \tag{5.6a}$$

$$\text{s.t. } P_t^{\text{Grid}} = P_t^{\text{Load,e}} + P_t^{\text{Bat}} + P_t^{\text{Gen}} + P_t^{\text{TRHP,e}} \qquad \forall t \tag{5.6b}$$

$$P_t^{\text{TRHP,h}} = P_t^{load,h} + P_t^{\text{HS}} \qquad \forall t \tag{5.6c}$$

$$P_t^{\text{TRHP},c} = P_t^{load,c} + P_t^{\text{CS}} \qquad \forall t \tag{5.6d}$$

$$P_t^{\text{TRHP,h}} + P_t^{\text{TRHP,c}} = COP^{TRHP}.P_t^{\text{TRHP,e}} \qquad \forall t \tag{5.6e}$$

$$P_t^{Bat} = P_t^{\text{Bat,ch}} + P_t^{\text{Bat, disch}} \qquad \forall t \tag{5.6f}$$

$$E_1^{Bat} = E_{\text{init}}^{Bat}.(1 - k_{\text{sd}}^{Bat}) + \Delta_t \left(P_0^{\text{Bat, ch}} \eta_{\text{Bat, ch}} - \frac{1}{\eta_{\text{Bat, disch}}}.P_0^{\text{Bat, disch}} \right) \tag{5.6g}$$

$$E_{\text{t+1}}^{Bat} = E_{\text{t}}^{Bat}.(1 - k_{\text{sd}}^{Bat}) + \Delta_t \left(P_t^{\text{Bat, ch}}.\eta_{\text{Bat, ch}} - \frac{1}{\eta_{\text{Bat, disch}}}.P_t^{\text{Bat, disch}} \right), \forall t \tag{5.6h}$$

$$E^{\text{Bat, min}} \leq E_t^{Bat} \leq E^{\text{Bat, max}} \qquad \forall t \tag{5.6i}$$

$$P^{\text{Bat,min}} \leq P_t^{Bat} \leq P^{\text{Bat,max}} \qquad \forall t \tag{5.6j}$$

$$P_t^{HS} = P_t^{\text{HS,ch}} + P_t^{\text{HS, disch}} \qquad \forall t \tag{5.6k}$$

$$E_1^{HS} = E_{\text{init}}^{HS}.(1 - k_{\text{sd}}^{HS}) + \Delta_t \left(P_0^{\text{HS, ch}} \eta_{\text{HS, ch}} - \frac{1}{\eta_{\text{HS, disch}}}.P_0^{\text{HS, disch}} \right) \tag{5.6l}$$

$$E_{\text{t+1}}^{HS} = E_{\text{t}}^{HS}.(1 - k_{\text{sd}}^{HS}) + \Delta_t \left(P_t^{\text{HS, ch}}.\eta_{\text{HS, ch}} - \frac{1}{\eta_{\text{HS, disch}}}.P_t^{\text{HS, disch}} \right) \quad \forall t \quad (5.6m)$$

$$E^{\text{HS, min}} \leq E_t^{HS} \leq E^{\text{HS, max}} \quad \forall t \quad (5.6n)$$

$$P^{\text{HS,min}} \leq P_t^{HS} \leq P^{\text{HS,max}} \quad \forall t \quad (5.6o)$$

$$P_t^{CS} = P_t^{\text{CS,ch}} + P_t^{\text{CS, disch}} \quad \forall t \quad (5.6p)$$

$$E_1^{CS} = E_{\text{init}}^{CS}.(1 - k_{\text{sd}}^{CS}) + \Delta_t \left(P_0^{\text{CS, ch}} \eta_{\text{CS, ch}} - \frac{1}{\eta_{\text{CS, disch}}}.P_0^{\text{CS, disch}} \right) \quad (5.6q)$$

$$E_{\text{t+1}}^{CS} = E_{\text{t}}^{CS}.(1 - k_{\text{sd}}^{CS}) + \Delta_t \left(P_t^{\text{CS, ch}}.\eta_{\text{CS, ch}} - \frac{1}{\eta_{\text{CS, disch}}}.P_t^{\text{CS, disch}} \right) \quad \forall t \quad (5.6r)$$

$$E^{\text{CS, min}} \leq E_t^{CS} \leq E^{\text{CS, max}} \quad \forall t \quad (5.6s)$$

$$P^{\text{CS,min}} \leq P_t^{CS} \leq P^{\text{CS,max}} \quad \forall t \quad (5.6t)$$

Where the label Bat in the notation refers to the battery, HS refers to the heat storage, CS refers to the cold storage, ch refers to charging of storage systems, $disch$ refers to discharging, $TRHP$ refers to the thermo-refrigerating heat pumps and t refers to the time step. Equation (5.6a) represents the objective function that aims at minimizing the power generation and grid power consumption costs over the multi-energy system. Constraints (5.6b) to (5.6d) represent the power balance equations of the system, equation (5.6e) links between the heat and cold power generated by the thermo-refrigerating heat pump and the electricity that it consumes via the coefficient of performance COP^{TRHP}. Constraints (5.6f) to (5.6j) define the dynamics of the battery energy storage system, (5.6k) to (5.6o) define the dynamics of the heat storage system and 5.6p to (5.6t) define those of the cold storage.

We propose to address this optimal control problem using both a DRL and an MPC based approach relying respectively on the MDP and LP formalisms defined above. Inter-conversion between two formalisms is done by converting the reward signal (5.4) of the MDP into the objective function (5.6a) of the LP and the action space of the MDP into the decision variables of the LP. The state space of the MDP as well as the transition between states following actions define the environment of the RL agent. They are governed by a Python simulator that defines the energy balance equations and the dynamics of the energy storage systems based on the same equations that define the constraints (5.6b) to (5.6t) of the LP problem.

5.3 The Deep Reinforcement Learning approach

5.3.1 Algorithm architecture

The optimal control problem considered in this work involves continuous action spaces that reflect the charge / discharge power of each of the storage systems. In future versions, this control problem will be extended to integrate optimized decisions on the heat and cold power to be produced by the thermo-refrigerating heat pumps and the chillers at a given time, the electric vehicles' smart charging, domestic hot water storage tanks and public lighting of the eco-district, as well as decisions regarding flexibility services such as load shedding and frequency regulation. For most of these actions, adopting continuous action spaces would ensure a precise and fine-grained control, while discrete action spaces would inevitably result in a loss of granularity and ultimately lead to a sub-optimal performance.

In order to harness the benefits of both value-based and policy-based RL algorithms, the *actor-critic* architectures emerged. As illustrated in figure 5.2, actor-critic paradigms involve two eponymous components: the *actor* which is responsible for representing the policy and making decisions on actions based on state observation, and the *critic* who simultaneously approximates the Q-value function and evaluates the quality of the actions chosen by the actor. In this work, we opt for the actor-critic paradigm to solve the optimal multi-energy management problem. Among prominent actor-critic algorithms such as Deep Deterministic Policy Gradient (DDPG) [274], Twin-Delayed Deep Deterministic Policy Gradient (TD3) [323] and Soft Actor Critic (SAC) [324], we opt for the DDPG algorithm, for which we provide the pseudo-code in Algorithm 10. This choice of the DDPG algorithm is grounded in its strong sample efficiency and generalization capabilities. This choice also finds support in several studies that conducted benchmarks comparing DDPG with other DRL algorithms for optimal control in energy systems. For instance, Wang et al. [320] considered a comparison of different DRL agents including DDPG and SAC against MPC and other rule-based models for the optimal control of a heat pump system in a residential house. Their findings on a standardized virtual building simulator revealed that, among DRL agents tested, only the DDPG algorithm consistently outperformed the baseline controller across all the considered heating scenarios.

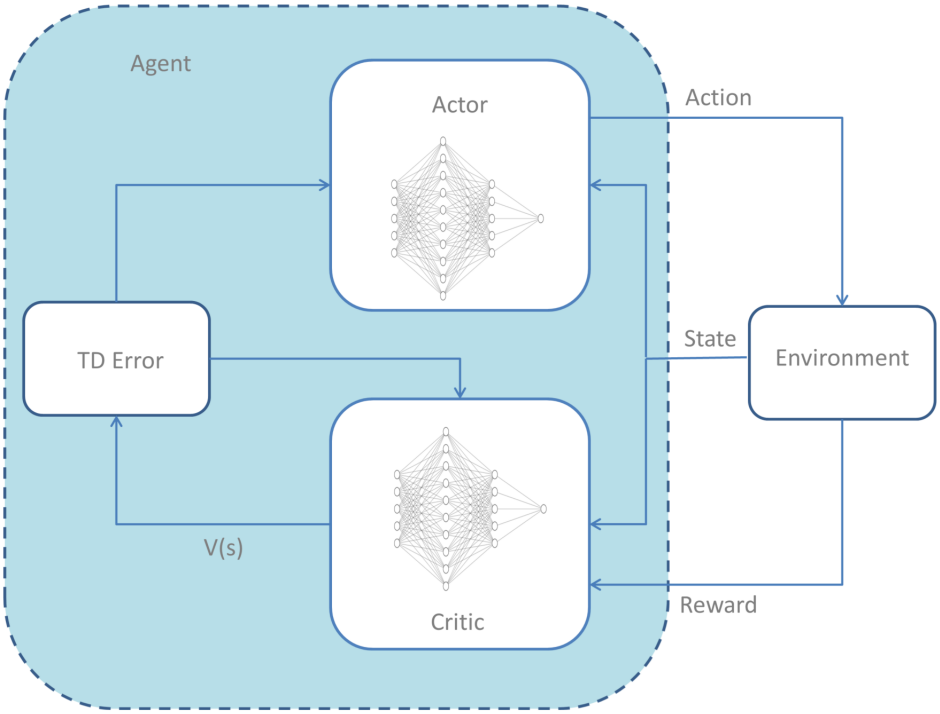


Figure 5.2: An illustration of the actor-critic architecture (adapted from [325]).

Algorithm 10: DDPG algorithm

Initialize the actor network μ and the critic network Q with random weights θ^{μ} and θ^{Q} ;
Initialize target networks μ' and Q' with the weights $\theta^{\mu'} \leftarrow \theta^{\mu}$ and $\theta^{Q'} \leftarrow \theta^{Q}$;
Initialize the experience replay Buffer B ;
for $episode \leftarrow 0$ **to** $N_{episodes}$ **do**
    Initialize a random process R for action exploration;
    Get initial observation of state S_1 at time step $t = 1$;
    for $t \leftarrow 1$ **to** N_{steps} **do**
        Select action $a_t = \mu(s_t|\theta^{\mu}) + R_t$ according to the current policy and exploration noise ;
        Execute action a_t in the environment and observe the resulting reward r_t and the new state s_{t+1} ;
        Store the transition (s_t, a_t, r_t, s_{t+1}) in experience replay buffer B;
        Sample a random mini-batch of N transitions (s_i, a_i, r_i, s_{i+1}) from B ;
        Set $y_i(r_i, s_{i+1}) = r_i + \gamma.Q'(s_{i+1}, \mu'(s_{i+1}|\theta^{\mu'})|\theta^{Q'})$;
        Update the critic by minimising the loss $L = 1/N \sum_i Q(s_i, a_i|\theta^{Q}) - y_i)^2$;
        Update the actor policy using the policy gradient $\nabla_{\theta^{\mu}} 1/N \sum_{s \in B} Q(s, \mu(s|\theta^{\mu})|\theta^{Q})$;
        Update the target networks: $\theta^{Q'} \leftarrow (1 - \rho).\theta^{Q} + \rho.\theta^{Q'}$ and $\theta^{\mu'} \leftarrow (1 - \rho).\theta^{\mu} + \rho.\theta^{\mu'}$
    end
end

5.3.2 Reward signal engineering

A reinforcement learning agent aims to acquire a policy that maximizes its cumulative sum of rewards over time. Consequently, the reward signal holds paramount importance in the learning task of a reinforcement learning agent, making reward engineering a critical and sometimes challenging task. This process involves hand-crafting the structure of the reward signal to effectively encapsulate the underlying objectives of the task while supplying adequate feedback to allow a meaningful learning for the RL agent. In many real-world scenarios, including the one explored in this study, rewards can be scarce, sporadic, irregular, infrequent, or unpredictable. This unpredictability can pose challenges for the agent in deciphering the correct actions to take, a phenomenon commonly known as the "sparse reward" problem. Addressing this issue is a complex task and is still not fully comprehended [326], [327]. Various authors have proposed approaches to tackle this problem, such as Trott et al. [328], Colas et al. [329], and Amin et al. [330]. Noteworthy solutions to mitigate sparse reward problems include structuring the reward signal by introducing penalty components or indicative functions, guiding the RL agent in navigating the learning process.

One of the distinctive aspects of the problem addressed in this study, that further contributes to the parse reward issue, is the agent's need to embrace diminutive or negative instantaneous rewards in order to optimize its future returns. This entails learning a strategy that deviates from conventional scenarios where the agent typically avoids immediate negative rewards to secure higher future rewards. For example, in the context of optimal energy management of storage systems, the agent must learn to charge the storage systems in specific time steps, often enduring negative instantaneous rewards, to discharge them later during opportune moments and maximize overall rewards. Another complication arises when handling storage systems that reach full charge or discharge. Traditionally, these boundary constraints are managed using low-level controllers in addition to optimal controllers. For instance, if the optimal controller suggests further discharging when the storage system is already fully discharged, the low-level controller intervenes to ensure compliance with boundary conditions and safe operational limits. Integrating such low-level control with a reinforcement learning agent may introduce a potential sparse reward problem. For instance, when the storage system is fully charged, and the RL agent's suggested action is to further charge it, the state of charge remains unchanged, resulting in no

variation of the system state. On the other hand, the reward signal may vary for example because of the variation of the energy prices and/ or the grid power consumption. This scenario may mislead the agent into mistakenly believing that taking no action on the storage system is the optimal strategy, potentially trapping it in a local optimum.

If the challenge of sparse rewards is not effectively addressed, for instance through appropriate reward shaping, it can slow down the learning process or even lead to its failure. In practical terms, the agent might expend numerous learning episodes exploring actions that do not yield sufficient reward feedback. Consequently, the agent may converge to strategies that represent a local optimum. An illustrative example of such strategies is opting for minimal or nearly no utilization of the storage systems, which is certainly misaligned with the desired optimal strategy. For instance, Ceusters et al. [37] encountered a similar challenge while addressing the optimal energy management problem in multi-energy systems using a deep reinforcement learning algorithm called PPO (Proximal Policy Optimization). Specifically, they highlighted challenges wherein the RL agent struggled to effectively operate both electrical and thermal storage systems. The authors attributed this issue to the inherent nature of operating storage systems, which involves enduring a momentary penalty (increased consumption during charging) to secure a mid-term reward (achieved by discharging during opportune moments, such as when energy prices are higher). Additionally, they suggested potential reasons for these difficulties, including the hyper-parameters chosen for the PPO algorithm, the use of the PPO algorithm itself, or even the application of the RL approach to this specific problem.

We posit that a likely explanation for this challenge lies in the sparse reward problem. Indeed, we encountered comparable issues when implementing the DDPG algorithm in our case study and successfully addressed them by enhancing the reward signal through the incorporation of a penalty function. We propose a function that takes the following

structure:

$$Penalty_t^{Bat} = \begin{cases} -|P_t^{Bat}| & \text{if } \Delta SoC^{Bat} = 0, \\ 0 & \text{otherwise.} \end{cases} \tag{5.7a}$$

$$Penalty_t^{HS} = \begin{cases} -|P_t^{HS}| & \text{if } \Delta SoC^{HS} = 0, \\ 0 & \text{otherwise.} \end{cases} \tag{5.7b}$$

$$Penalty_t^{CS} = \begin{cases} -|P_t^{CS}| & \text{if } \Delta SoC^{CS} = 0, \\ 0 & \text{otherwise.} \end{cases} \tag{5.7c}$$

$$\tag{5.7d}$$

Where P_t^{Bat}, P_t^{HS} and P_t^{CS} are the actions taken by the RL agent respectively for the battery, heat storage and cold storage systems operation. ΔSoC^{Bat}, ΔSoC^{HS} and ΔSoC^{CS} are the state of charge variations of these respective storage systems yielded by the actions of the agent. This way, the agent incurs a penalty each time it executes an action on a storage system that fails to induce any changes in its state of charge. This penalty is particularly relevant when the agent tries to charge a storage system already at full capacity or discharge a system already at its minimum state of charge. Traditionally, these constraints on storage systems are managed by a low-level controller that is external to the agent. This controller receives the actions dictated by the high-level control agent (RL agent, MPC, etc.) and adjusts them to adhere to the system's constraints. While this conventional constraint handling guarantees compliance with the system's constraints, it falls short in providing the RL agent with sufficient feedback to avoid sparse reward issues during training.

By adding the penalty component to the reward signal, it becomes as follows:

$$r_t = -\alpha.[C_t^{gen}.P_t^{gen} + C_t^{grid}.P_t^{grid}] + Penalty_t^{Bat} + Penalty_t^{hS} + Penalty_t^{CS}. \tag{5.8}$$

This approach effectively addressed the sparse reward problem by furnishing the RL agent with more insightful feedback. Following this modification, the DDPG agent successfully acquired a strategy for the efficient and simultaneous operation of the electrical and thermal storage systems, as demonstrated in the simulation results section.

5.3.3 The exploration-exploitation dilemma

As explained in chapter 3, striking the balance between exploration and exploitation in reinforcement learning is essential to ensure an effective learning. This allows the agent to find a trade-off between exploiting known strategies that already yielded high rewards, while simultaneously exploring new actions that may lead to obtain potentially better strategies. In this work, we addressed this issue by investigating the use of action noise and parameter noise.

5.3.3.1 Action noise

Action noise consists in injecting randomness into the actions taken by the RL agent during training, so as to promote exploration. In this work, we propose the use of two types of action noise: a correlated noise, namely Ornstein-Uhlenbeck (OU) noise and an uncorrelated noise, namely normal distribution noise.

OU noise is a correlated noise, which means that it introduces a correlation between its successive noise samples. It is a stochastic process that is often used to model noises with temporal correlations. It is commonly used as action noise in RL and is defined by the following differential equation:

$$dx_t = \theta(\mu - x_t)dt + \sigma dW_t \tag{5.9}$$

Where x_t is the value of the process at time step t, μ is the mean that the process tends to revert to, θ is a parameter that monitors the speed of mean reversion towards μ, dW_t is referred to a Wiener process, it represents the randomness of the process, and σ is a volatility parameter that scales the intensity of the noise. The discrete-time equation of the OU process that we use in this work has the following form:

$$x_{t+1} = x_t + \theta(\mu - x_t)\Delta t + \sigma\sqrt{\Delta t}.Z_t \tag{5.10}$$

Where $Z_t \sim \mathcal{N}(0, 1)$ is a sample from a standard normal distribution.

In this work, we tested both OU action noise and normal distribution noise to promote better exploration of the RL agent. Normal distribution noise $x \sim \mathcal{N}(\mu, \sigma)$, where μ is its mean and σ is its standard deviation, is an uncorrelated action noise that adds random perturbations to the agent's actions with no inherent correlations between successive noise

samples.

5.3.3.2 Parameter noise

Plappert et. al. [331] proposed an alternative to the approach commonly used to promote exploration in DRL by injecting noise in the action space. Their alternative approach relies on adding noise directly to the agent's parameters. Their work showed that parameter noise is a promising alternative to the conventional action space noise in that it leads to a more efficient exploration even in problems with sparse rewards for which action noise is not likely to be effective. The difference between action noise and parameter noise is illustrated in figure 5.3.

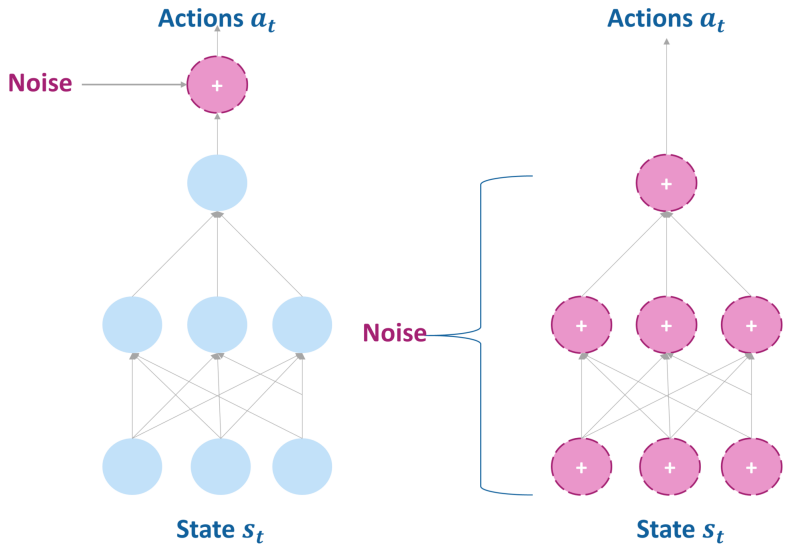


Figure 5.3: An illustration of the difference between action noise (left) and parameter noise (right) (adapted from from Open AI [332]).

In our work, we implemented and compared parameter noise in DDPG, alongside the two previously mentioned variants of action noise. Our simulation results, presented further in section 5.6 support that parameter noise may outperform both correlated and uncorrelated action noise in exhibiting effective exploration in some specific environments. The results of comparison between the effectiveness of these three noise exploration types in case-study 1 will be presented in forthcoming sections of this chapter.

5.3.4 Hyper-parameters

Besides the exploration noises, the DDPG algorithm also relies on various parameters that govern its behavior and can significantly influence its stability, its convergence speed as well as its efficiency in the learning task. That is why, a careful consideration and an effective tuning of these hyper-parameters are essential. These parameters include the architectures of the actor and the critic, their respective learning rates, the discount factor γ, the soft update parameter τ and the configuration parameters of the replay buffer, namely the size of the replay buffer and the batch size:

- **Architecture of the actor model:** the actor's deep neural network defines the policy that governs the actions selected by the actor and involves a mapping from states to actions. Thus, its depth and width can influence the DRL agent's capacity to represent complex policies.

- **Architecture of the critic model:** the role of the critic's deep neural network is to evaluate the actions taken by the actor by approximating the value function and thus aiding the agent in understanding the expected cumulative rewards yielded by its current policy. It encapsulates a mapping between pairs of states and actions chosen by the actor as input, with their corresponding estimated Q-values as output. By being able to predict and compare values of different actions, the agent can make informed decisions about which actions are expected to yield the most favorable rewards. Thus, similarly to the actor network, the architecture of the critic network can affect the ability of the DDPG agent to deal with complex and high-dimensional state spaces.

- **Learning rates of the actor and the critic:** they define the step size at which the actor and the critic models update their respective parameters in response to observed experiences. Even though high learning rates may result in a faster convergence, it may also lead to instability and overshooting in the learning process. Hence, balancing these rates is essential to achieve a compromise between speed and stability. It is also important to consider the ratio between the learning rate of the actor (LRA) and the learning rate of the critic (LRC). In fact, even though this ratio may simply be taken equal to 1 ($LRA = LRC$) leading to the actor and the critic updating their parameters at the same pace, several research works like

[333] suggested that a better performance of the DDPG can be obtained if the critic is trained at a slightly higher learning rate than the actor (around 2 to 10 times higher). This strategy allows the value estimate to converge more quickly and accurately while allowing the policy of the actor to adapt more slowly and profit from more exploratory and adaptive learning (since the actor's policy updates will anyways be effectively guided by the critic).

- **Discount factor** $0 \leq \gamma < 1$**:** also referred to as discount rate, it is a key parameter in reinforcement learning. It defines the importance that the agent gives to future rewards (against immediate rewards) in the decision-making process. Its value is commonly chosen in the range between 0.9 and 0.99. A low value of γ (close to 0) leads the agent to prioritize immediate rewards while a higher value (close to 1) leads it to attribute more weights to future rewards and consequently encourages long-term planning.

- **Soft update parameter** $0 < \tau < 1$**:** it dictates the rate at which the target networks are updated from the online networks. A higher value of τ leads to more frequent updates. This may boost the convergence speed, but can also result in noise and instability. Conversely, smaller values of τ may lead to smoother updates and an enhanced stability while slowing down the learning process. As a consequence, tuning this parameter is also essential to reach a trade-off between stability and convergence speed.

- **Size of the replay buffer:** the replay buffer stores past experiences, i.e tuples in the form (state, action, reward, next state), in order for the agent to learn from them through sample replay. The size of the replay buffer determines the number of past experiences that will be retained for learning. Its value often ranges between 10^4 and 10^6. While increasing the buffer size can allow capturing a wider range of experiences and as a result an improved exploration and stability, it will however increase the memory requirements. Inversely, a smaller buffer size allows conserving memory but can impact the learning and convergence by limiting the diversity of experiences.

- **Batch size:** for each update of the neural networks, a set of experiences are sampled from the replay buffer. The batch size determines the number of sampled experi-

ences. Its value often ranges between 32 and 256 experiences. Similarly to other hyper-parameters, the selection of the batch size is important since it influences the accuracy of the gradients estimates as well as the computational resources requirements.

5.4 Implementation details

We developed a custom DRL framework using Python, wherein the Deep Deterministic Policy Gradient (DDPG) agent was crafted and designed to interact with the simulation model, which serves as the experimental platform for our experiments and has been encapsulated as an OpenAI Gym environment. Instead of relying on pre-existing libraries or frameworks such as OpenAI's Baselines [334] or Stable Baselines [335], we intentionally built the DDPG agent from scratch. This deliberate choice was intended to foster a profound understanding of the algorithm's functioning and provide greater flexibility in tailoring and refining the algorithm to suit the specific problem addressed in this study.
For the testing and parameter-tuning of the agent, we first reduced the case-study to a mono-action environment where the agent learns to manage only one storage system (the battery in this case). All the other components defined for this case study remained the same, except for the heat and cold storage systems that were not used in this first mono-action use-case. Once this initial phase of testing and fine-tuning the DRL agent on this first use-case was successfully completed, this use-case was extended to the multiple-action environment where the agent learns a strategy to simultaneously manage the three multi-energy storage systems. Starting with a mono-action use-case aimed at gaining insights into the algorithm's behaviour and understanding its intricacies on a simpler setup before progressively introducing complexity in the setup and tackling the broader problem of simultaneously orchestrating multiple multi-energy storage systems. Besides, fine-tuning the agent for the mono-action use-case and then for the extended multiple-action use-case also allowed us to understand the scalability of the DRL agent and the adaptability of its hyper-parameters and to ensure that it can transition from managing a single component to managing the synergic operation of multi-energy systems. The training data comprising two consecutive years are split into two sets: a one year training set and a one year validation set. This division represents a common practice in the ML field that is adopted to avoid over-fitting as explained by François-Lavet et al. [268]. The training

set basically offers a substantial amount of historical data to the agent to learn from and the validation set serves as a checkpoint for evaluating the agent's performance on unseen data and fine-tuning the hyper-parameters. A visualization of the exogenous data used in the simulations is provided in Figure 5.4 for a typical winter day and in Figure 5.5 for a typical summer day. It should be noted that the input state data, actions and reward signals are normalized, as this normalization is a common practice in Deep Reinforcement Learning since it helps the Deep Neural Networks to train more effectively by preventing issues related to exploding or vanishing gradients. Besides, the DRL agent can learn optimal policies more easily when using normalized reward signals, especially when using activation functions such as *tanh* or *sigmoid* which naturally work within the range between -1 and 1.

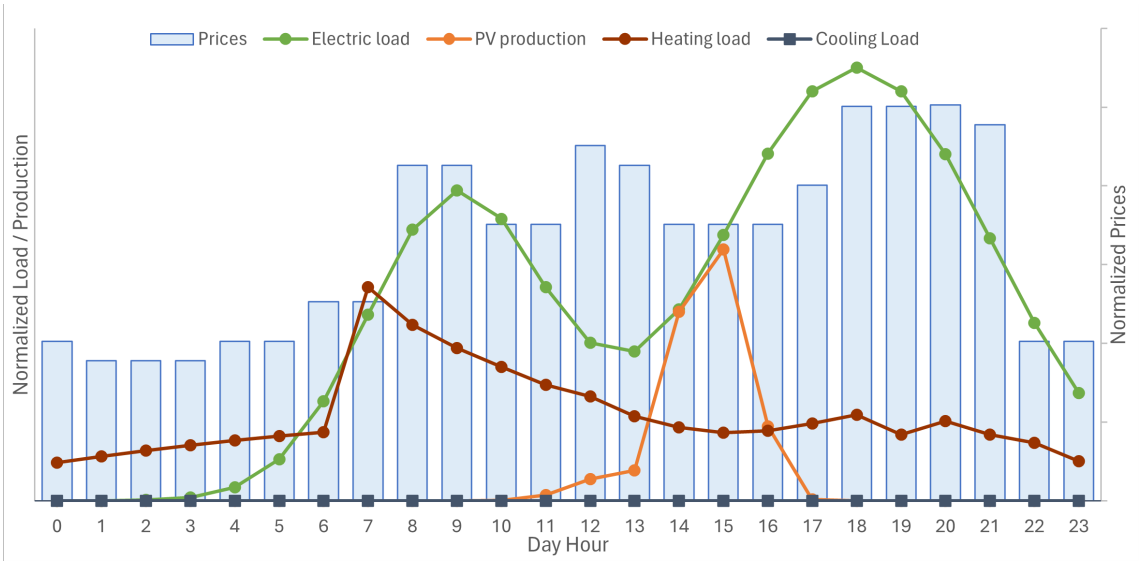


Figure 5.4: Visualization of the data used for simulations for a typical winter day: electric, heating and cooling loads, PV generation and electricity prices.

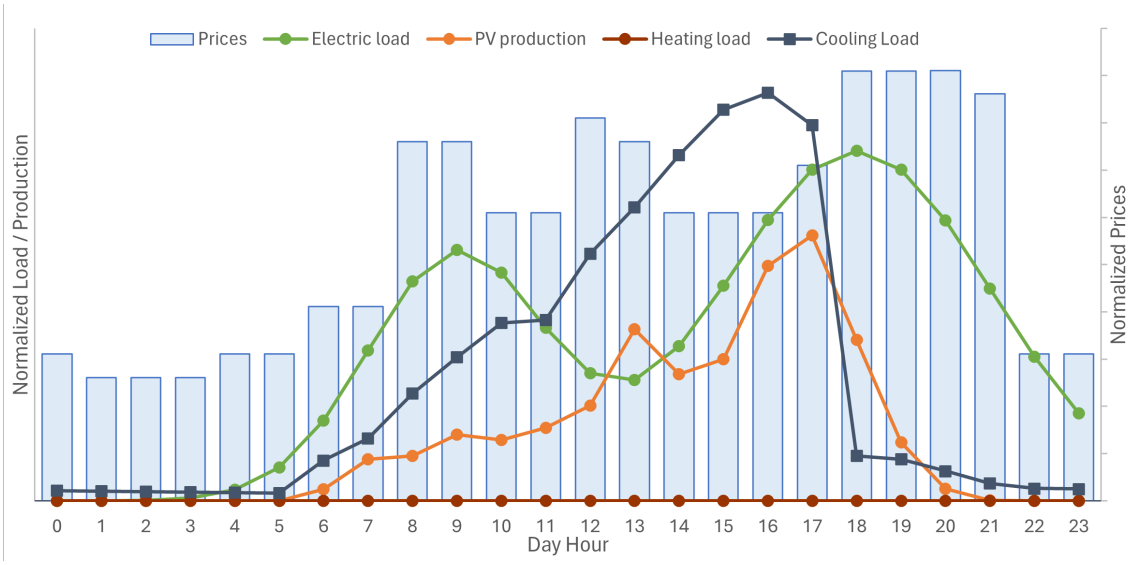


Figure 5.5: Visualization of the data used for simulations for a typical summer day: electric, heating and cooling loads, PV generation and electricity prices.

5.5 The MPC-based benchmark approach

In order to provide a comprehensive evaluation of the DRL approach, we incorporated an MPC-based algorithm as a benchmark to the DRL approach. The MPC framework was developed under Python. As explained in Chapter 4, this predictive control approach operates in a closed-loop fashion and optimizes control actions over a given time horizon and given provided forecasts on future quantities for a specified forecast horizon. In our implementation, we conducted simulation cycles that cover periods of one year with hourly time-steps and involve a control horizon and a forecast horizon that were both set to 24 hours.

At each time-step of the simulation, an energy manager class is called to solve the previously defined LP optimization problem for the upcoming 24 hours. The control actions that correspond to the immediate time-step are then executed in the simulation model. As the closed-loop simulation advances step by step, the system state is updated and the forecast horizon is adjusted accordingly. We provide the controller with perfect forecasts over future electricity prices, electric, heat and cold loads and PV generation, in order to obtain a theoretical MPC optimum and thus establish a robust baseline for assessing the effectiveness of the proposed DRL-based energy management approach.

5.6 Simulation results

5.6.1 Single-action-environment results

5.6.1.1 Training and validation results

The DDPG agent was trained in the single-action environment through a series of learning episodes of one-year simulation, with hourly time-steps. The training aimed at acquiring a strategy for the efficient operation of the battery energy storage system while minimizing the overall operational costs of the smart energy system. The DRL agent succeeded in learning a strategy to optimally manage the storage system that achieved a total reward of approximately 98% of the theoretical MPC optimum. Figure 5.6 provides a representation of cumulative costs over one random week of the year, obtained by the trained DDPG as well as those obtained by the theoretical MPC controller. Remarkably, the two cumulative costs closely track each other, which underscores similarity in performance between the two agents. We note that the costs and rewards depicted in the figures are presented in a normalized format with reference to the total costs obtained by the MPC controller. This normalization aims at facilitating the direct comparison between the performance of the DRL agent and the MPC controller to which we associate a cost value of 1 (or 100%). In Figure 5.7, we illustrate an example of the battery management strategies devised by the DRL and the MPC agents for a week period. Even though these strategies are not identical, both agents share a common approach in that they charge the battery during low electricity price periods and profit from higher price periods to discharge the battery. As a result, the two agents achieve comparable energy consumption costs, which indicates the effectiveness of the DDPG agent in achieving performance levels akin to those obtained with the MPC controller.

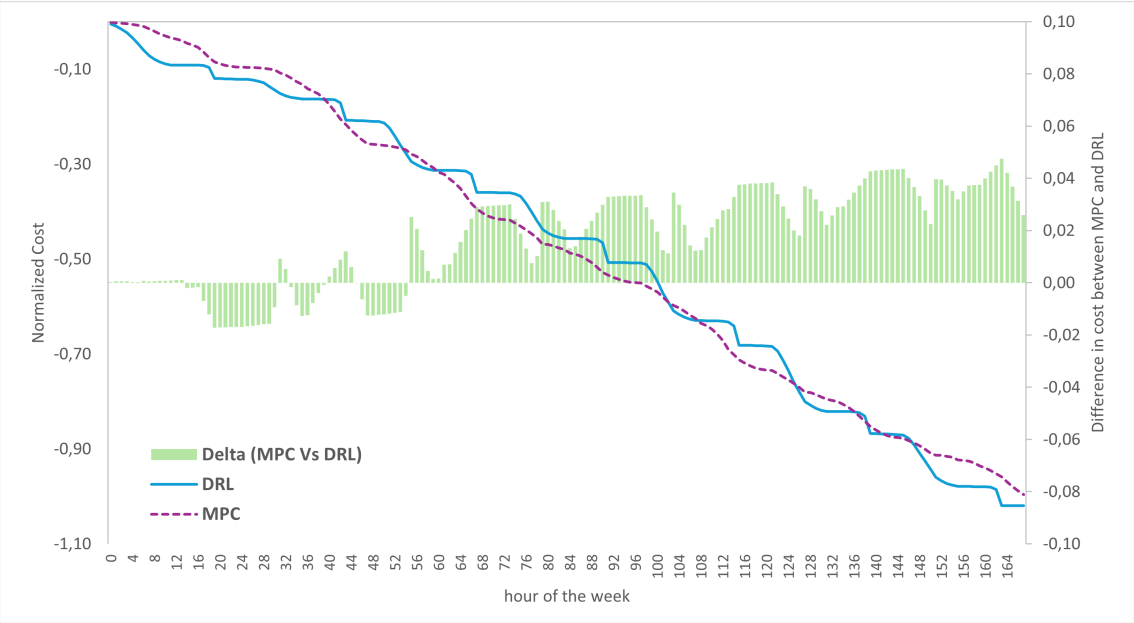


Figure 5.6: Difference between normalized cumulative costs over one random week of the year obtained by the DRL agent and the MPC controller.

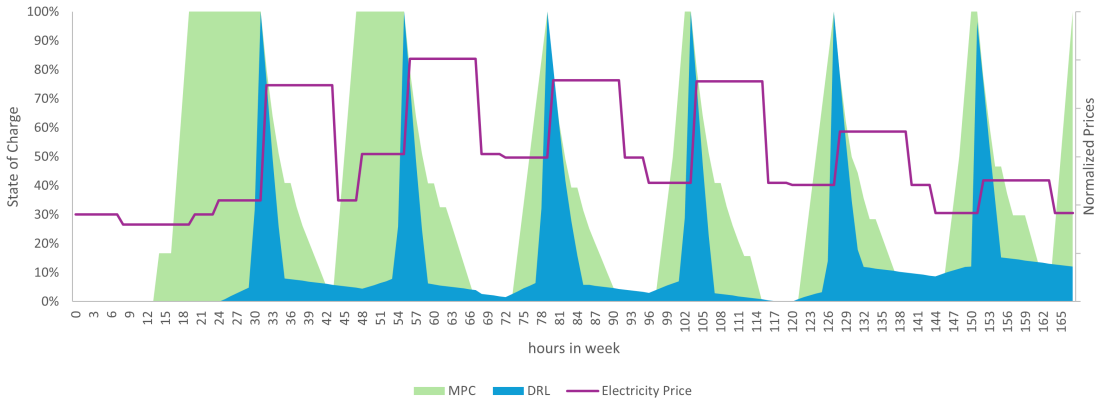


Figure 5.7: Illustration of the battery energy management policies obtained by the DRL agent and the MPC controller over one random week of the year.

5.6.1.2 Hyper-parameter tuning

Similarly to the approach adopted by Wang et al. [320] and regarding the excessive training duration of the DRL agent, we opted for a hyper-parameter tuning that relies on manually varying one hyper-parameter at a time. The outcomes of these experiments are presented below.

Tuning of the actor and critic DNNs The actor network aims at finding an optimal policy through a mapping from states to actions. We designed a network that is composed

of four densely connected layers: an input layer that receives the state information and thus involves as many nodes as the size of the state space, two subsequent hidden layers that attempt to extract further meaningful features from the state space information, and an output layer that provides the actions' values and use hyperbolic tangent ($\tanh$) as an activation function to make sure that the actions fall within the desired bounds (-1 and 1). On the other hand, the critic network aims at evaluating the quality of the actions selected by the actor by taking both the states and the actions as inputs and estimating the Q-values. The state inputs are transformed through two hidden layers, and the actions are processed through one other hidden layers. Both layers are then concatenated and processed through two additional hidden layers, and the final layer outputs a single value that provides an estimated Q-value for the given state-action pair.
For the tuning of the actor and critic neural networks' parameters, we tested five different activation function types as well as five values for the size of the hidden layers, but did not change the number of hidden layers of each of the neural networks. Actually, the accuracy of a neural network depends on the number of hidden layers used, but is more importantly defined by the type of activation function, as stated by Sharma et al. [336].

- **Type of the activation functions:** for the activation functions, we tested the five following functions: ReLU (rectified Linear Unit), ELU (Exponential Linear Unit), SELU (Scaled Exponential Linear Unit), Softplus, Softmax and Sigmoid. The results of these tests, as illustrated in Figure 5.8 and Table 5.2 show that RELU is the activation function that allowed achieving the most accurate results for our case study.

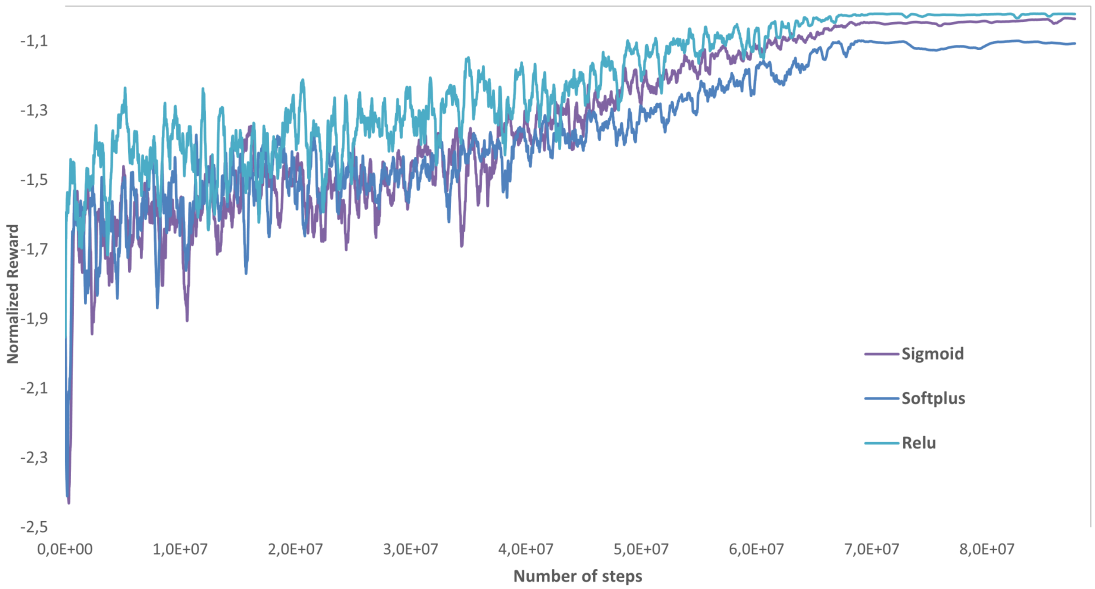


Figure 5.8: Learning curve of the DDPG agent for three different types of activation functions for the actor and the critic

Activation function	Sigmoid	Softmax	Softplus	Selu	Elu	**Relu**
Normalized Reward	-1.04	-2.00	-1.11	-1.02	-1.04	**-1.02**

Table 5.2: Normalized final episodic rewards obtained for six different types of the activation functions used in the actor and the critic neural networks

- **Size of the hidden layers:** for the number of nodes that compose each hidden layer of the actor and the critic networks, we conducted experiments with five different sizes: 32, 64, 128, 256 and 512. The results of these experiments, as shown in Figure 5.9 and Table 5.3 revealed that hidden layer sizes that are smaller than 128 are not sufficient to achieve sufficiently accurate and stable solutions. Meanwhile, we also noticed that increasing the number of neurons beyond 128 (for example 256 and 512) did not have a significant impact on the quality of the solution.

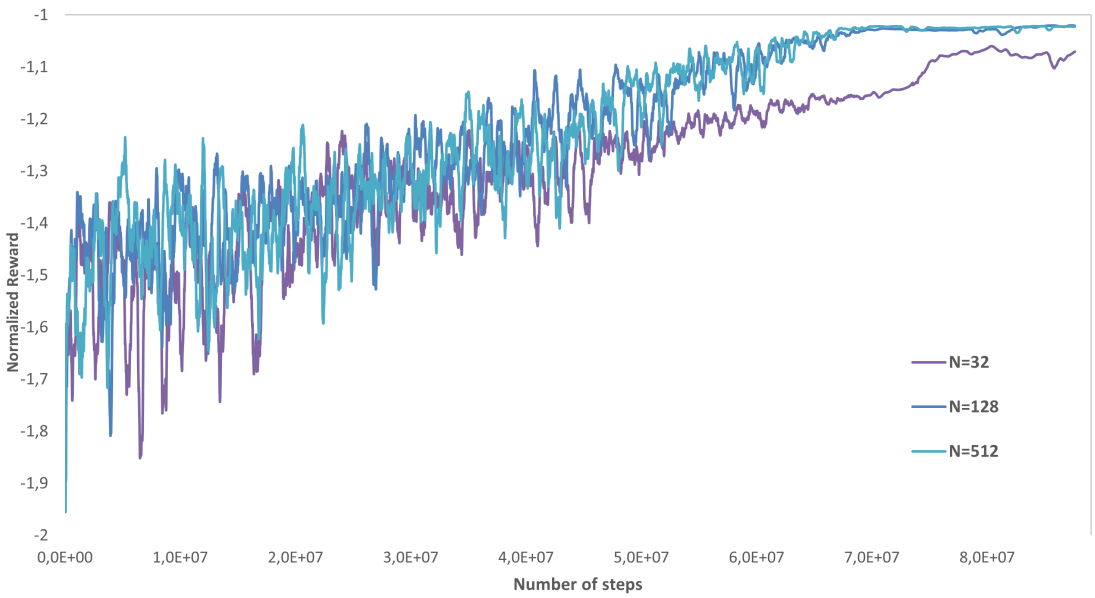


Figure 5.9: Learning curve of the DDPG agent for three different sizes of the hidden layers for the actor and the critic neural networks

Hidden Layers Size	N=32	N=64	**N=128**	N=256	N=512
Normalized Reward	-1.07	-1.03	**-1.02**	-1.02	-1.02

Table 5.3: Normalized final episodic rewards obtained for five different values of the size of the hidden layers of the actor and the critic neural networks

Tuning of the learning rates The DDPG algorithm features two distinct learning rates, one for the actor (LRA) and one for the critic (LRC). As explained above and in consistence with recent literature in the field such as [337] and [333], we opted for a training of the critic at a slightly higher (twice higher) learning rate than the actor. To tune the learning rates, we tested six different values for the LRA ranging from 10^{-1} to 10^{-6} while maintaining the ratio $LRC = 2 * LRA$). Figure 5.10 presents the learning curves of the DDPG agent for different values of LRA. We presented the learning curves for only three out of these six values to improve the readability of the figure. The final episodic normalized reward after a training cycle of each of the six tested learning rates are presented in Table 5.4 and show that the learning rate that yields the better results is $LRA = 10^{-4}$.

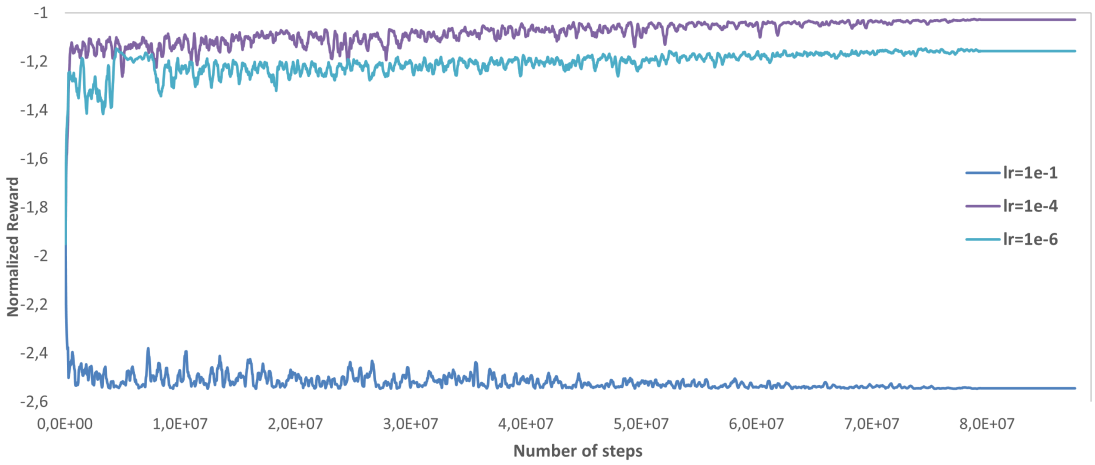


Figure 5.10: Learning curve of the DDPG agent for different values of the learning rate of the actor

Actor Learning Rate	10^{-1}	10^{-2}	10^{-3}	$\mathbf{10^{-4}}$	10^{-5}	10^{-6}
Normalized Reward	-2.55	-2.55	-1.13	**-1.03**	-1.07	-1.15

Table 5.4: Normalized final reward obtained by the DDPG agent for different values of the learning rate of the actor

Tuning of the discount factor The discount factor γ is one of the most crucial hyper-parameters for the DDPG agent. We tested five values of γ ranging between 0.9 and 0.99. The results, presented in Figures 5.11 and Table 5.5 show that the value of the discount factor that leads to the best final reward after training cycle is $\gamma = 0.975$.

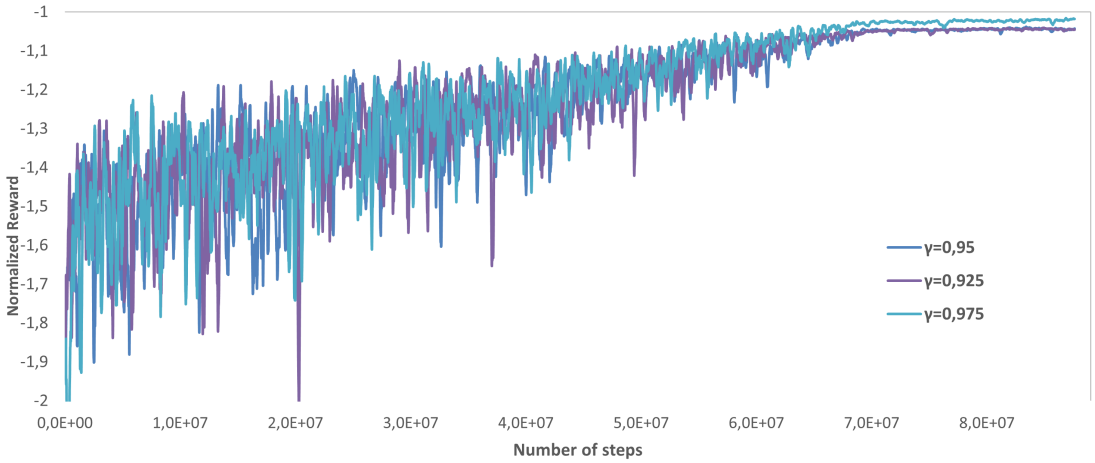


Figure 5.11: Learning curve of the DDPG agent for different values of the discount factor γ

Discount Factor (γ)	0.9	0.925	0.95	**0.975**	0.99
Normalized Reward	-1.04	-1.06	-1.04	**-1.02**	-1.06

Table 5.5: Normalized final reward obtained by the DDPG agent for different values of the discount factor γ

Tuning of the soft update parameter For the soft update parameter τ, we tested four different values ranging between 5.10^{-1} and 5.10^{-4}. To compare these values, we do not only examine the episodic reward reached at the end of the training cycle but also the stability of the solution. For instance, the values $\tau = 5.10^{-1}$ and $\tau = 5.10^{-2}$ presented an instability around the final episodes of the training phase, as shown in Figure 5.12. Only the values of $\tau = 5.10^{-3}$ and $\tau = 5.10^{-4}$ presented stable learning curves at the end of the training. Based on the results of the final episodic reward illustrated in Table 5.6, we conclude that the best value obtained for the soft update parameter is $\tau = 5.10^{-3}$.

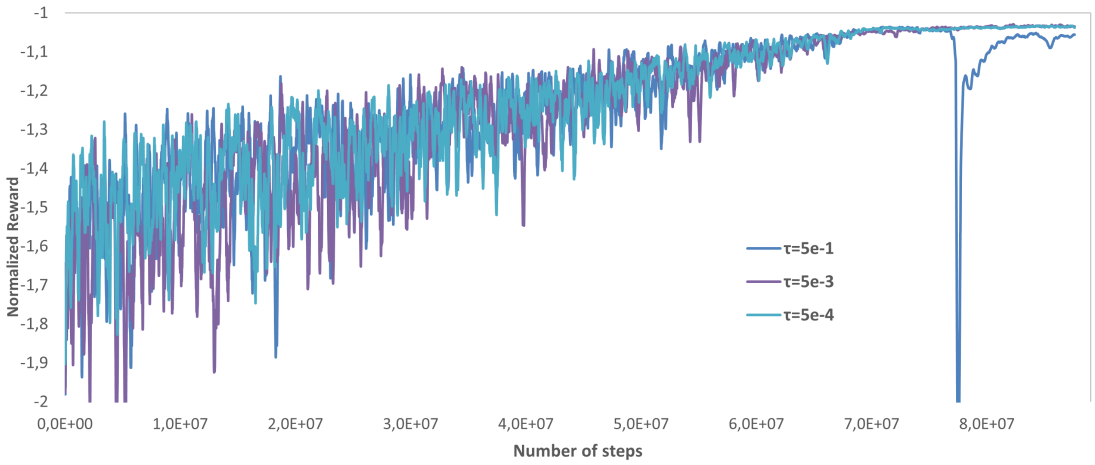


Figure 5.12: Learning curve of the DDPG agent for three values of the soft update parameter τ

Soft update parameter (τ)	0.5	0.05	**0.005**	0.0005
Normalized Reward	-1.06	-1.04	**-1.03**	-1.05

Table 5.6: Normalized final reward obtained by the DDPG agent for different values of the soft update parameter τ

Tuning of the buffer size The experience replay buffer plays the role of a memory where previous state-action transitions are collected as the agent interacts with the environment and are stored in the form (state, action, reward, next state). By randomly sampling from past experiences, the buffer breaks the temporal correlations

in data and allows the agent to learn from its past interactions with the environment while avoiding the risk of over-fitting to the most recent experiences. In fact, if the buffer is too small, this may prevent the agent from capturing a wide range of scenarios and thus increase the risk of over-fitting to recent data. This may lead to a poor performance and an unstable learning process. Conversely, a too large replay buffer will contain too many experiences that the agent will be likely to spend much time revisiting old data, and may miss more recent and potentially relevant information. This can significantly slow down the learning process. To tune the size of the replay buffer, we tested values ranging from 10^3 to 10^7. The results of these experiments illustrated in Figure 5.13 and Table 5.7 show that the best obtained value for the size of the replay buffer is 10^6.

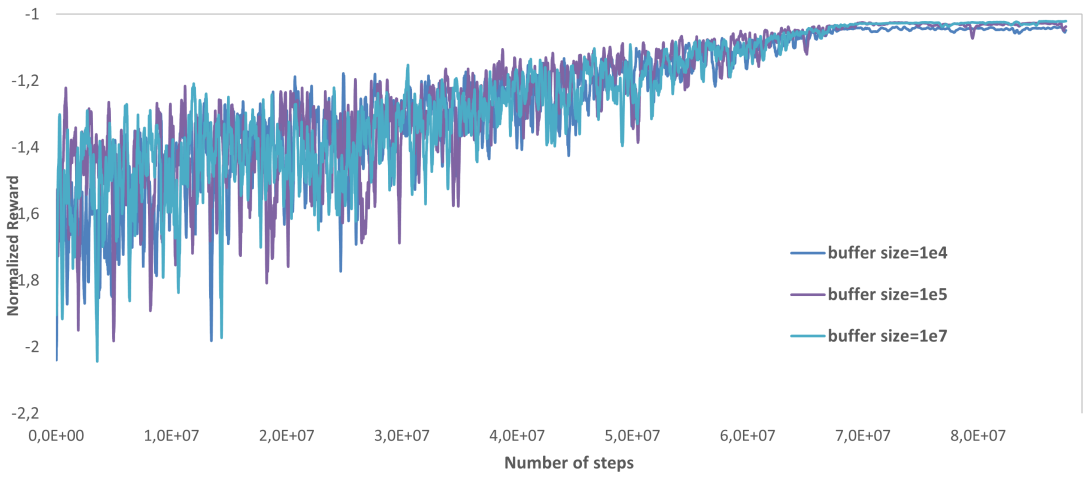


Figure 5.13: Learning curve of the DDPG agent for different values of the buffer size

Buffer Size	10^3	10^4	10^5	$\mathbf{10^6}$	10^7
Normalized Reward	-1.05	-1.04	-1.02	**-1.02**	-1.03

Table 5.7: Normalized final reward obtained by the DDPG agent for different values of the buffer size

Tuning of the batch size For the tuning of the batch size, one has to consider a trade-off between the quality and stability of the solution, and the generalization performance of the agent. Actually, a larger batch size can increase the stability of the training process but can also increase the risk of over-fitting. We tested four different values of the batch size, namely 64, 128, 192 and 256, which fall within

the range of typical batch size values that are commonly chosen by practitioners in the field of DRL. As presented in Figure 5.14 and Table 5.8, we find that the value that yields the best final reward is 128.

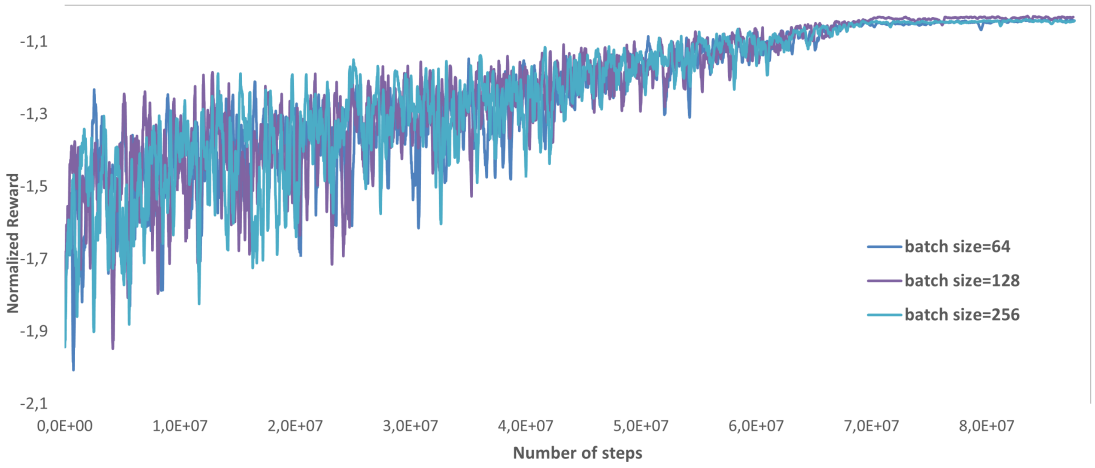


Figure 5.14: Learning curve of the DDPG agent for different values of the batch size

Batch Size	64	**128**	192	256
Normalized Reward	-1.05	**-1.03**	-1.05	-1.05

Table 5.8: Normalized final reward obtained by the DDPG agent for different values of the batch size

Tuning of the exploration noise The experiments that we conducted for the hyper-parameter tuning of the DDPG algorithm highlighted that the type and parameters of the exploration strategy are one of the most influential parameters on the performance and the learning dynamics of the agent. In this work, we experimented two types of action noises, namely Ornstein-Uhlenbeck (OU) and normal action noise, as well as one parameter noise. For each of these exploration noise types, we tested training cycles with five different values of standard deviation.

The action noise is added to the actions selected by the RL agent at each time step of each learning epoch. This noise is multiplied by a scaling factor ϵ that determines the magnitude of the noise added to the actions. It starts at the initial value $\epsilon = 1$ and decreases through the learning cycle by being incrementally multiplied by an exploration rate exp such that:

$$exp = -\alpha.\frac{1}{n_{ep}.n_{st}} \tag{5.11}$$

where n_{ep} is the total number of episodes of the training cycle, n_{st} is the total number of time steps per episode and α is a constant that we fixed at the value $\alpha = 1.25$ in order to have exploration only at the first 80% of the training steps. Moreover, to ensure that the value of ϵ remains positive, it takes the value $max(0, exp+\epsilon)$ at each time step. This exploration strategy aims at balancing exploration and exploitation throughout the learning process. At the beginning of the training cycle, the value of ϵ is high and the noise has, as a result, a more significant impact leading to more exploration. As the training progresses, the value of ϵ decreases and thus allows the agent to exploit more the policy that it learned.

– **Ornstein-Uhlenbeck (OU) action noise:** in these experiments, a noise of the type OU is added to the actions selected by the RL agent at each time step of each learning epoch. Training tests with five different values of the volatility σ that scales the magnitude of the noise, ranging from 5% to 25%, revealed that the value of $\sigma = 10\%$ gives the best results, as illustrated in Figure 5.15 and Table 5.9.

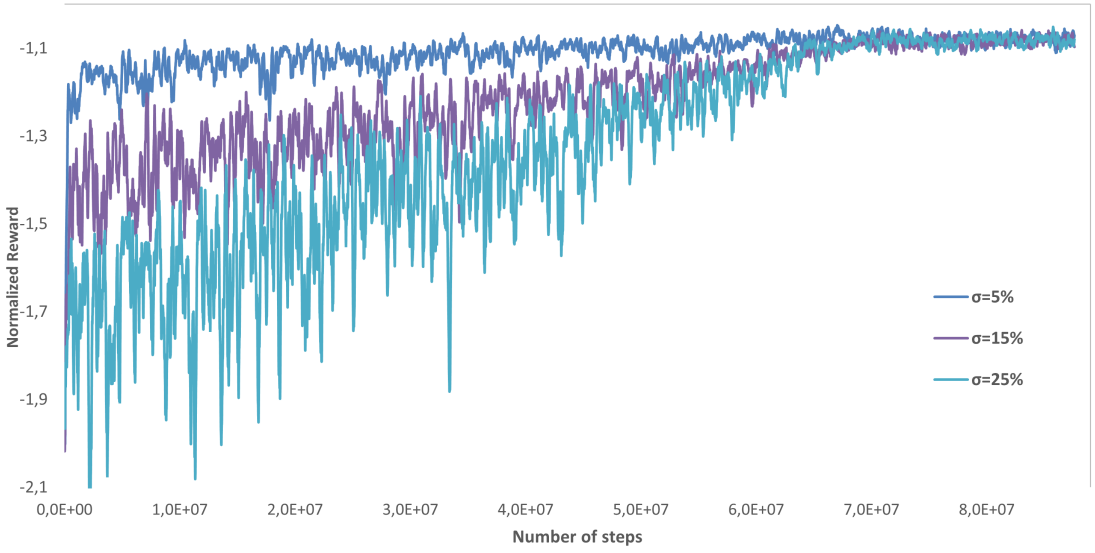


Figure 5.15: Learning curve of the DDPG agent with OU action noise for different values of the standard deviation σ of the OU noise.

OU noise volatility σ	5%	**10%**	15%	20%	25%
Normalized Reward	-1.11	**-1.01**	-1.02	-1.13	-1.07

Table 5.9: Normalized final rewards obtained by the DDPG agent for different values of the standard deviation σ of the OU noise.

– **Normal action noise:** similar experiments are carried out for a normal distribution action noise with different values of the standard deviation σ. The results of these experiments (Figure 5.16 and Table 5.10) give the best value of $\sigma = 15\%$.

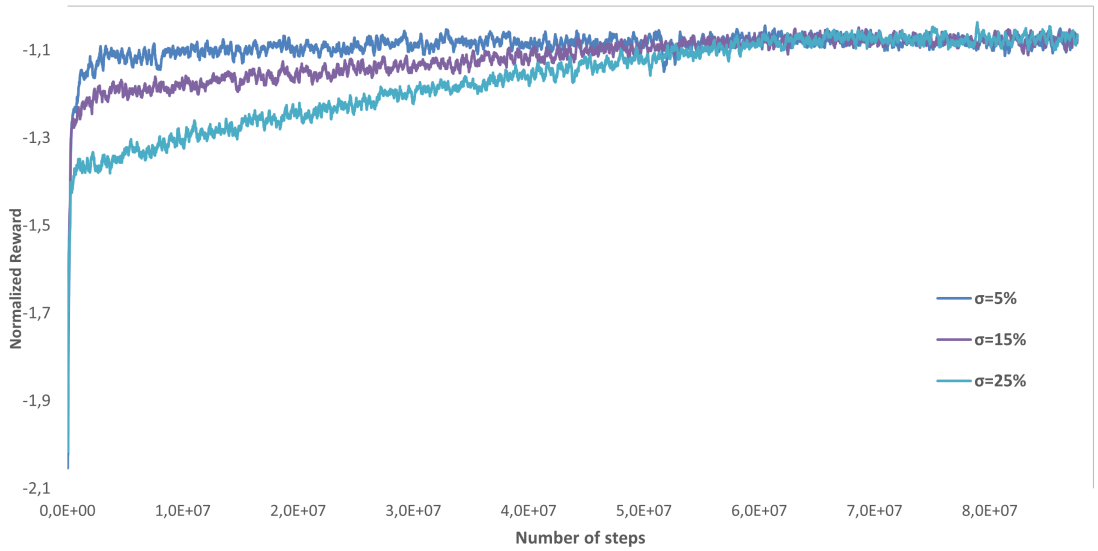


Figure 5.16: Learning curve of the DDPG agent with normal action noise, for different values of the standard deviation σ of the normal noise.

Normal noise volatility σ	5%	10%	**15%**	20%	25%
Normalized Reward	-1.20	-1.08	**-1.02**	-1.10	-1.10

Table 5.10: Normalized final rewards obtained by the DDPG agent for different values of the standard deviation σ of the normal noise

– **Parameter noise:** similarly to the previous two experiments, we ran training cycles using parameter noise instead of the action noise with five different values of the standard deviation σ. These experiments showed that the best value of σ for this case study is 5% (Figure 5.17 and Table 5.11).

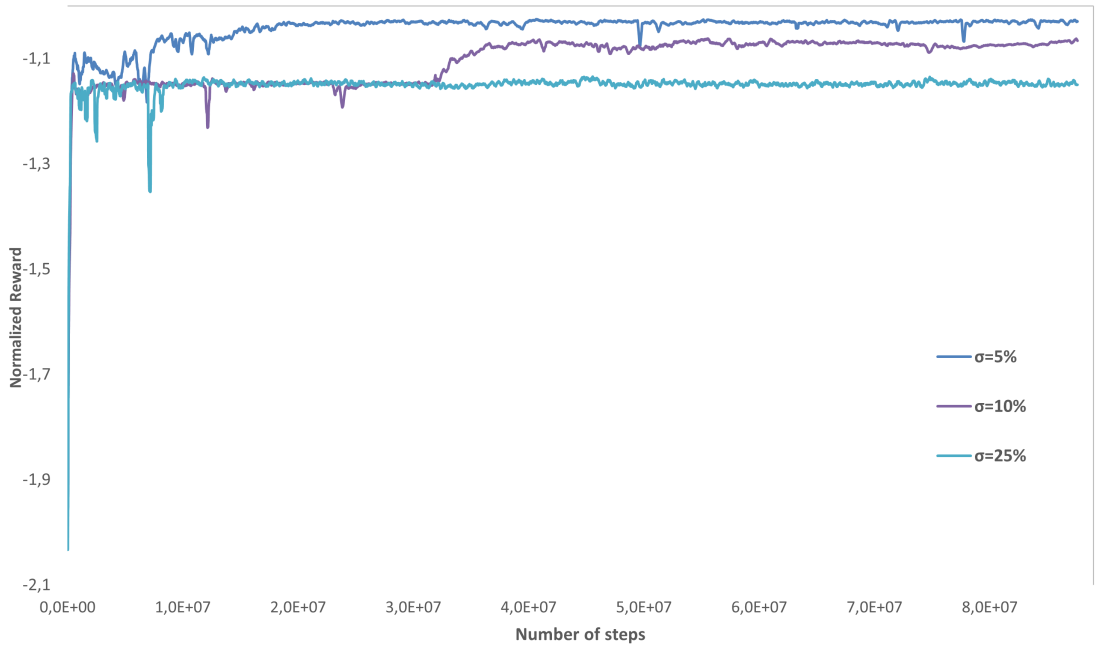


Figure 5.17: Learning curve of the DDPG agent with parameter noise, for different values of the standard deviation σ of the parameter noise.

Parameters noise volatility σ	**5%**	10%	15%	20%	25%
Normalized Reward	**-1.00**	-1.05	-1.07	-1.16	-1.15

Table 5.11: Normalized final rewards obtained by the DDPG agent with parameter noise, for different values of the standard deviation σ of the parameter noise.

– **Comparison of different exploration noises:** in these experiments, we compare the learning curves for the three different exploration noises previously investigated, with the best values of standard deviation found for each noise type. These learning curves, illustrated in figure 5.18, show that RL agents with parameter noise achieved a faster convergence and better stability as shown in Figure 5.18. These results align with research findings in the field of RL, such as those of Plappert et al. [331].

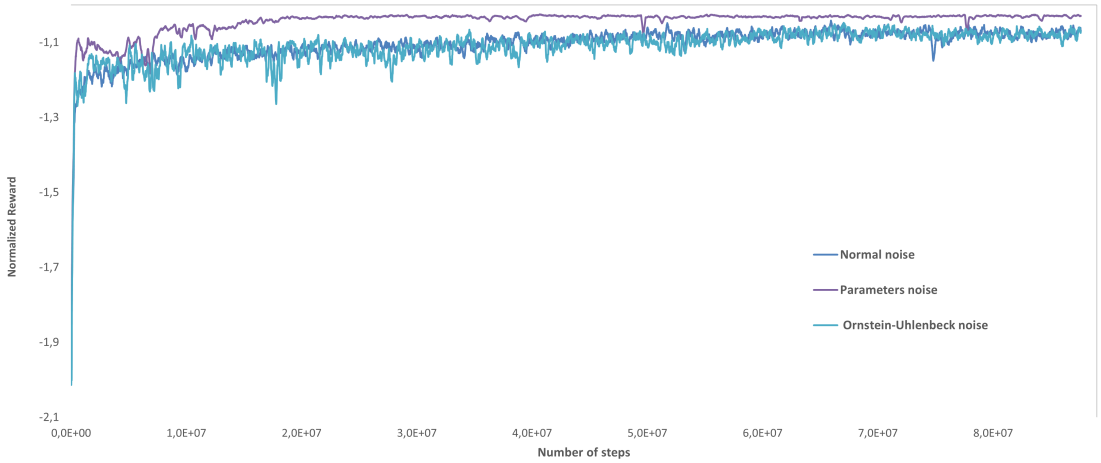


Figure 5.18: Learning curves of the DDPG agent for the three types of exploration noises: parameter noise, OU action noise and normal action noise.

Table 5.12 summarizes the optimal hyper-parameters found for the DDPG agent in this single-action setup.

Parameter	best value found
Activation function	ReLU
Number of hidden layer for the actor network	2
Number of hidden layers for the critic network	4
Size of the hidden layers	128
Learning rate of the actor	10^{-4}
Learning rate of the critic	2.10^{-4}
Discount factor (γ)	0.975
Soft-update parameter (τ)	5.10^{-3}
size of the replay buffer	10^6
Batch size	128
OU noise standard deviation	10%
Normal noise standard deviation	15%
Parameter noise standard deviation	5%
Best found exploration noise	Parameter noise

Table 5.12: Summary table of the parameter tuning results for the single-action setup

5.6.2 Multiple-action-environment results

5.6.2.1 Learning results

In this section, we expand the previously defined environment to include control over the heat and cold storage systems in addition to the control over the battery

storage system. As in the previous setup, the training cycles involve episodes that cover a full year with hourly time steps. To evaluate the effectiveness of our DRL agent, we benchmark the results against those of a theoretical MPC controller. Figure 5.19 illustrates the learning curve of the DDPG agent that showcases the evolution of the total reward signal through the learning process. As explained previously, this reward signal involves two components: the overall energy consumption costs of the multi-energy system and a penalty component that activates for each time step in which the DRL agent suggests actions that violate any of the storage systems' constraints.

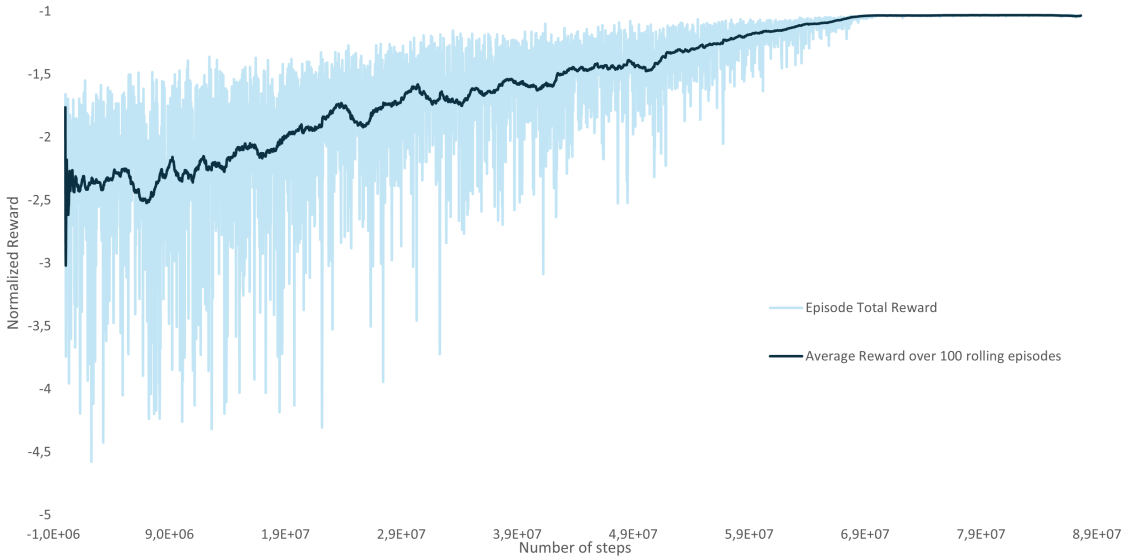


Figure 5.19: Learning curve of the DDPG agent for the multiple-action environment: evolution of the total reward signal and the average reward over 100 rolling episodes throughout a training a cycle.

In Figures 5.20 and 5.21 we separately plot the evolution of the penalty component and the cost component of the reward signal respectively. Remarkably, the penalty component progressively converges to zero indicating that the DRL agent succeeds in handling the boundary constraints of the storage systems by the end of the learning process. Besides, the cost component of the reward signal converges to a final value that closely approximates 98.5% of the theoretical MPC optimum, presented in green in Figure 5.21. These results demonstrate the capability of the DRL agent in orchestrating the operation of the multi-energy storage systems while adhering to their operational constraints. It is worth noting that the results presented in this this

section were obtained using the best-obtained hyper-parameter after tuning. The results of this parameter tuning will be discussed in section 5.6.2.3.

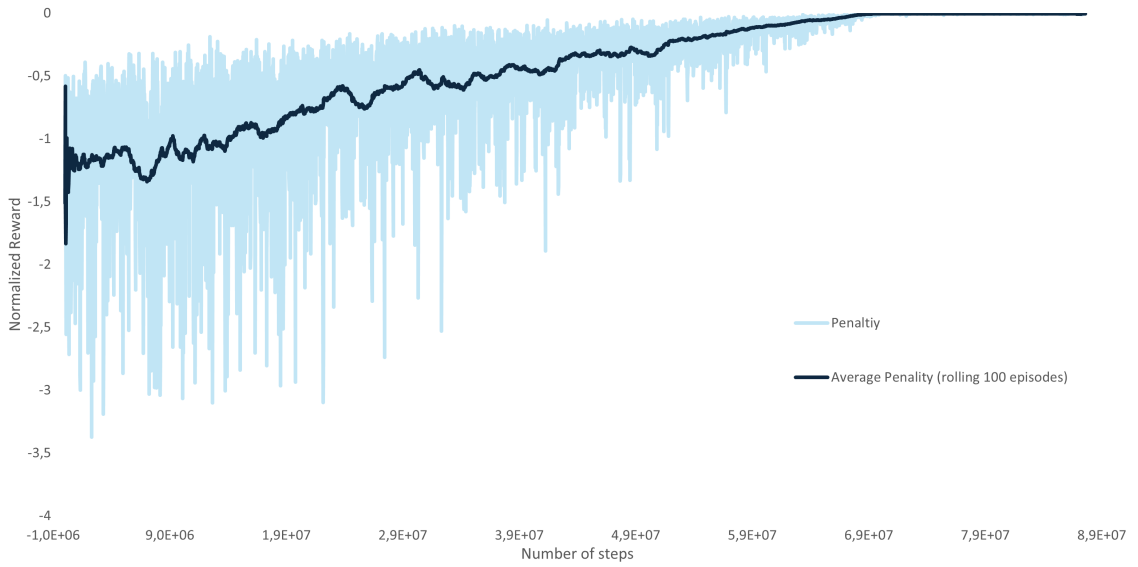


Figure 5.20: Learning curve of the DDPG agent: evolution of the penalty component of the reward signal, as well as its average over a rolling horizon of 100 episodes, throughout the training cycle

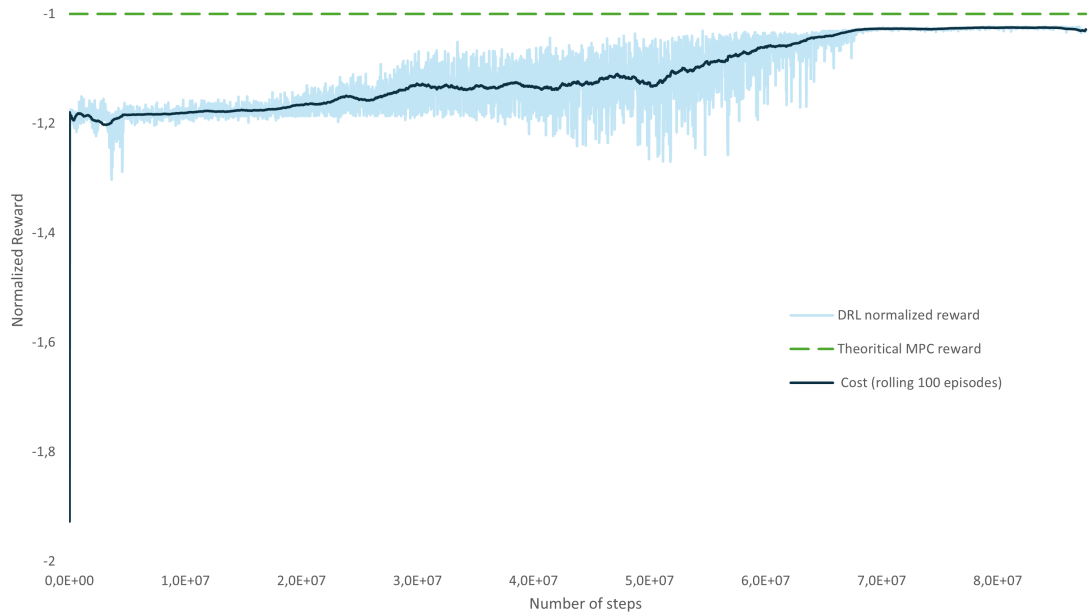


Figure 5.21: Learning curve of the DDPG agent: evolution of the cost component of the reward signal, as well as its average over a rolling horizon of 100 episodes, throughout the training cycle, and comparison with the theoretical optimal cost obtained by the MPC controller for the same time frame.

To further evaluate the DRL agent's performance, we extended the benchmark by

introducing a level of noise to the forecasts F provided to the MPC controller such that:

$$F = F(1 + \epsilon) \tag{5.12}$$

where the noise ϵ is modeled as a normal distribution $\epsilon \sim \mathcal{N}(0, \sigma^2)$, for which we vary the standard deviation from 10% to 30%. This leads to more realistic forecasts instead of the perfect forecasts and thus provides a benchmark MPC controller that is closer to real-world conditions (realistic forecast). The results are depicted in the histogram of Figure 5.22 where we compare the final cost component of the DRL agent's reward against the total cost obtained by the MPC with different levels of noise. We associate a percentage of 100% to the MPC optimum under ideal, noise-free forecasts. The MPC controller maintained a slightly stronger performance with noise standard deviations of 10% and 20%. Yet, as the noise standard deviation increased to 25% and 30%, the performance of the MPC controller declined to 97.8% and 95.2% respectively, going gradually below the performance of the DRL agent of 98.5%. Noticeably, the DDPG agent consistently delivered results that are close to the theoretical MPC optimum, and in some cases outperformed the MPC with realistic forecasts. These results underscore the resilience of the DRL approach as compared with the MPC approach that presents a variability in real-world forecasting scenarios.

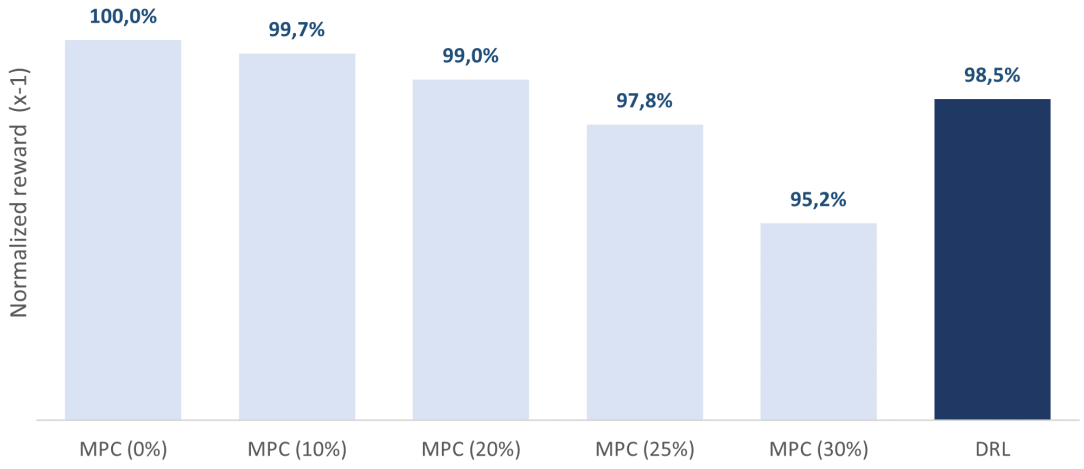


Figure 5.22: Normalized reward obtained by the DRL agent and by variants of the MPC with perfect and realistic forecasts

5.6.2.2 Validation results

After training, the DRL agent is subjected to a validation phase where we use data that were not previously utilized for the training phase. Figures 5.23 and 5.24 illustrate the strategy proposed by the DDPG agent for the management of the battery, the heat storage and the cold storage respectively for one randomly selected winter week. This strategy was overlaid with the one obtained by the theoretical MPC controller for the same time frame. Even though the strategies proposed by the DRL agent and by the MPC controller are not exactly identical, they show remarkable performance parity in terms of overall energy consumption costs. Actually, the DRL agent constantly achieves levels approaching 98% of the theoretical optimum of the MPC. One notable difference between the MPC and the DRL strategies arises when it comes to the management of the cold storage during winter. Indeed, the DRL agent opts for a gradual and very slow charging of the cold storage during winter, while the MPC does not. This divergence likely stems from the long time horizon of the DRL agent which can be traced back to the discount factor γ of 0.975 and to its training over a span of a full year, whereas the control horizon of the MPC is narrower (24 hours).

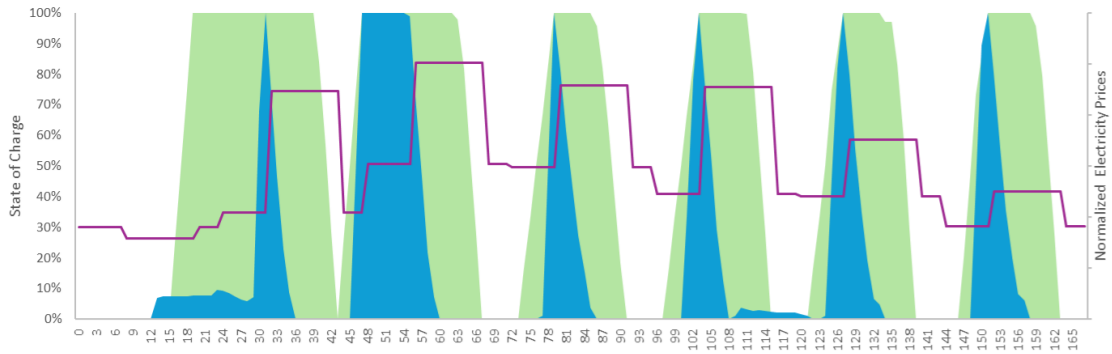


Figure 5.23: Illustration of the strategy proposed by the DDPG and the MPC agents for the management of the battery for one randomly selected winter week.

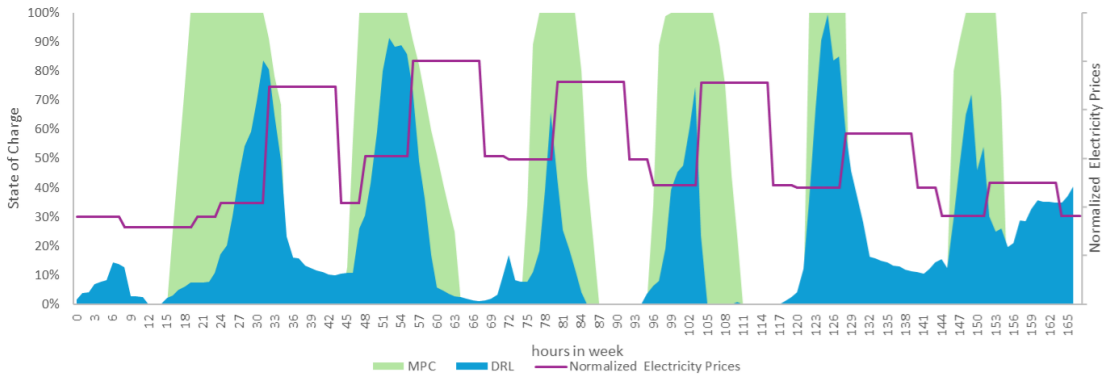


Figure 5.24: Illustration of the strategy proposed by the DDPG and the MPC agents for the management of the heat storage for one randomly selected winter week.

Figures 5.25 and 5.26 illustrate the energy management strategies obtained by the MPC and DRL agents for the battery and the cold storage systems respectively over the course of one randomly selected week of the summer. Both agents did not opt for the usage of the heat storage system during this summer week.

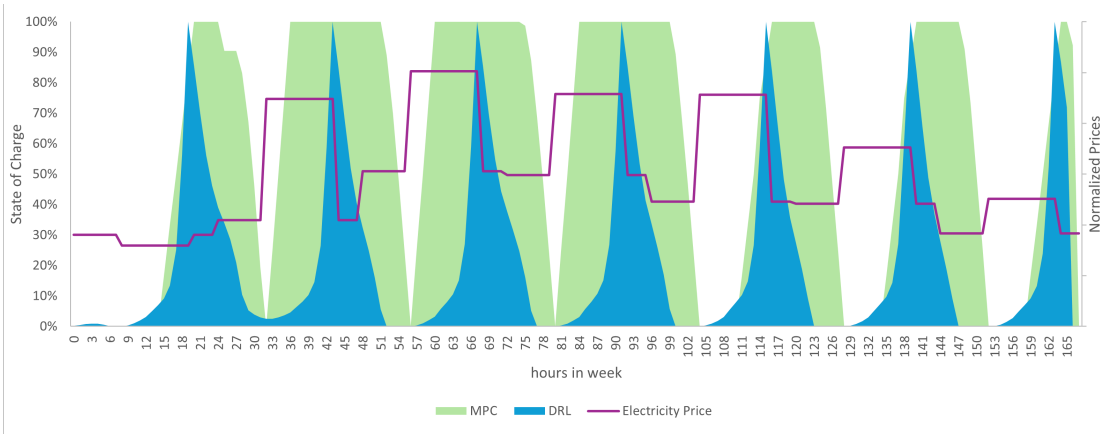


Figure 5.25: Illustration of the strategy proposed by the DDPG and the MPC agents for the management of the battery for one randomly selected summer week.

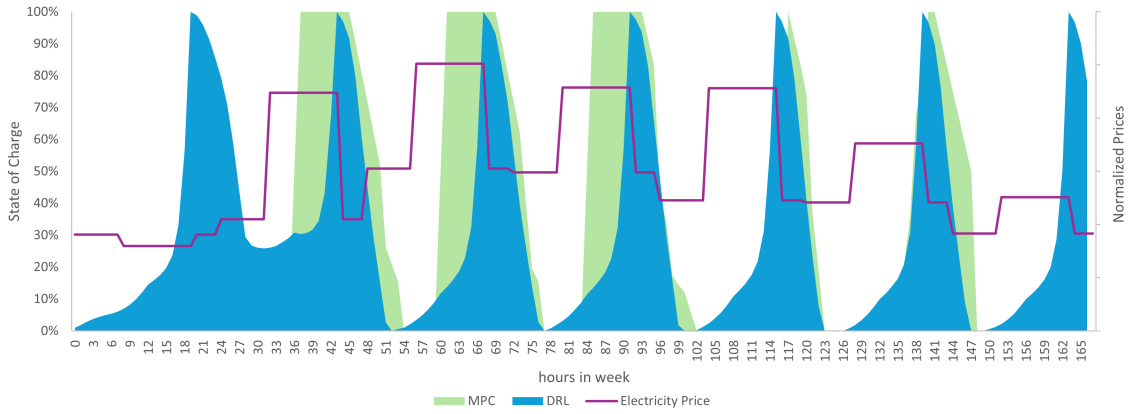


Figure 5.26: Illustration of the strategy proposed by the DDPG and the MPC agents for the management of the cold storage for one randomly selected summer week.

5.6.2.3 Hyper-parameter tuning

In line with the approach we adopted for the hyper-parameter tuning of the single-action DDPG agent, we carried similar tuning for the multiple-action environment, where we vary one parameter at a time. The findings of this hyper-parameter tuning are presented below. Overall, most of the best obtained parameter values closely align with those found in the single-action setup, except for the exploration noise. Actually, unlike the single-action environment for which parameter noise considerably outperformed the two types of action noise considered, in the multiple-action scenario, we observed that the parameter noise is no longer the best choice. Instead, normal distribution action noise slightly outperformed parameter noise.

Tuning of the DNNs:

- **Type of the activation functions:** results of the fine-tuning of the activation function used in the actor and critic neural networks for the multiple-action environment exhibit similar trends to those obtained with the mono-action environment. These results (illustrated in Figure 5.27 and Table 5.13) consistently indicate that the ReLU (Rectified Linear Unit) is the optimal activation function for this case-study.

Figure 5.27: Learning curve of the DDPG agent for three different types of activation functions for the actor and the critic

Activation function	Sigmoid	Sotmax	Softplus	Selu	Elu	**Relu**
Normalized Reward	-1.29	-1.52	-1.08	-1.01	-1.02	**-1.01**

Table 5.13: Normalized final episodic rewards obtained for six different types of the activation functions used in the actor and the critic neural networks - multiple actions environment

– **Size of the hidden layers:** results of the fine-tuning for the size of the hidden layer of the neural networks for the multiple-action environment also closely align with those obtained for the mono-action setup. They suggest a requirement of minimum 128 nodes for each hidden layer for an effective learning (Figure 5.28 and Table 5.14). Remarkably, we observe that sizes of 32 and 64 yielded significantly poorer results for the multiple-action setup than their counterparts for the single action setup.

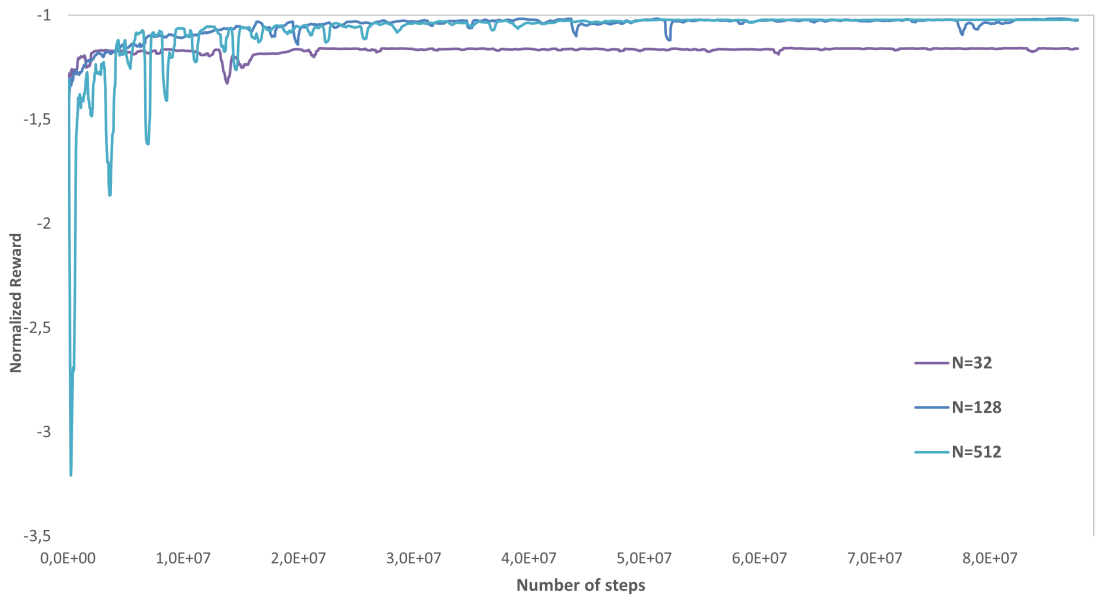


Figure 5.28: Learning curve of the DDPG agent for three different sizes of the hidden layers for the actor and the critic neural networks

Hidden Layers Size	N=32	N=64	**N=128**	N=256	N=512
Normalized Reward	-1.16	-1.16	**-1.02**	-1.02	-1.02

Table 5.14: Normalized final episodic rewards obtained for five different values of the size of the hidden layers of the actor and the critic neural networks - multiple actions environment

Tuning of the learning rates: Regarding the fine-tuning of the learning rates, results also closely align with those observed for experiments with the single-action agent. Its outcomes consistently suggest a requirement of a learning rate of the actor of 10^{-4} while maintaining the value of the critic's learning rate as twice its value for the actor. Figure 5.29 presents the learning curves of the DDPG for different values of the learning rate of the actor (LRA) and Table 5.15 illustrates the final normalized episodic rewards obtained for six different values of the learning rate.

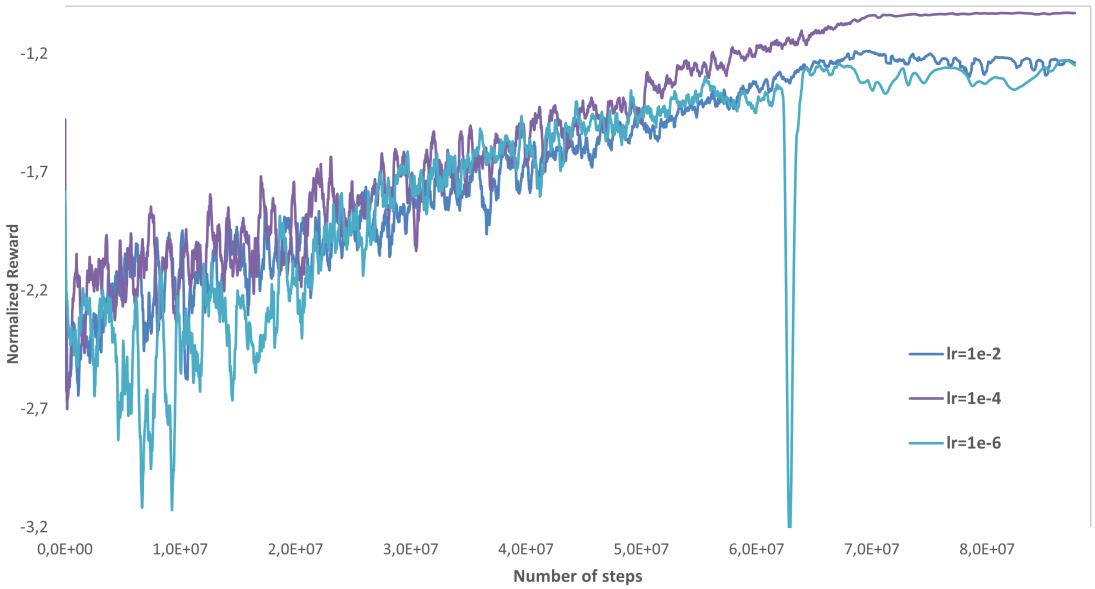


Figure 5.29: Learning curve of the DDPG agent for different values of the learning rate of the actor

Actor Learning Rate	10^{-1}	10^{-2}	10^{-3}	$\mathbf{10^{-4}}$	10^{-5}	10^{-6}
Normalized Reward	-5.33	-1.24	-1.06	**-1.03**	-1.16	-1.25

Table 5.15: Normalized final episodic reward obtained by the DDPG agent for different values of the learning rate of the actor - multiple actions environment

Tuning of the discount factor: the best obtained value of the discount factor is also the same as for the single-action agent: $\gamma = 0.975$ as depicted in Figure 5.30 and Table 5.16.

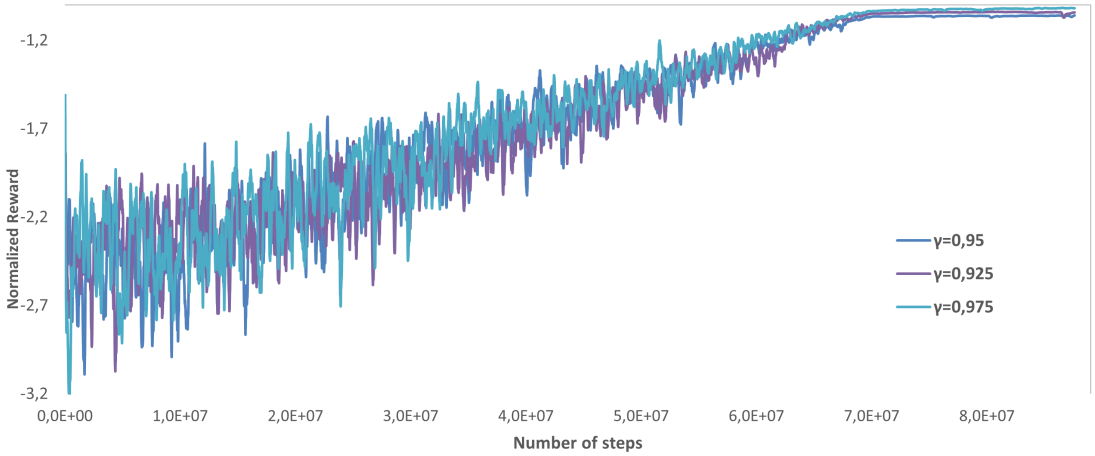


Figure 5.30: Learning curve of the DDPG agent for different values of the discount factor γ - multiple action environment

Discount Factor (γ)	0.9	0.925	0.95	**0.975**	0.99
Normalized Reward	-1.06	-1.04	-1.04	**-1.02**	-1.02

Table 5.16: Normalized final reward obtained by the DDPG agent for different values of the discount factor γ

Tuning of the soft update parameter: the outcomes of the fine-tuning of the soft-update parameter τ (Figures 5.31 and Table 5.17) also closely resemble the single-action agent results (optimal obtained value is $\tau = 5.10^{-3}$).

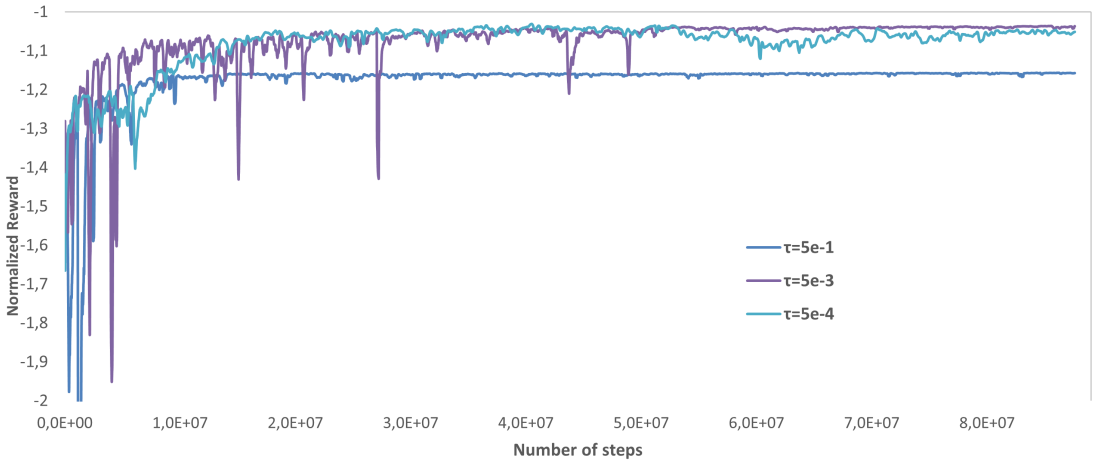


Figure 5.31: Learning curve of the DDPG agent for different values of soft update parameter τ

Soft update parameter (τ)	$5\,10^{-1}$	$5\,10^{-2}$	$\mathbf{5\,10^{-3}}$	$5\,10^{-4}$	$5\,10^{-5}$	$5\,10^{-6}$
Normalized Reward	-1.16	-1.16	**-1.04**	-1.05	-1.16	-1.16

Table 5.17: Normalized final episodic reward obtained by the DDPG agent for six different values of the soft update parameter τ - multiple actions environment

Tuning of the buffer size: regarding the tuning of the buffer size for the multiple-action environment, the outcomes globally align with those of the single-action environment. The optimal value remains consistent at 10^6. However, for the multiple-action setup we observe a notable deviation for the values 10^3 where the results are significantly inferior to those of the single-action environment, as shown in Figure 5.32 and Table 5.18. These symptoms may reflect a *catastrophic forgetting* that occurs towards the end of the training cycle. Catastrophic forgetting actually refers to a phenomenon where a neural network loses its ability to retain knowledge that it acquired during earlier phases of training or to remember past experiences while adapting to new

ones. A limited experience replay may be responsible for this phenomenon. The agent may not be able to retain a sufficient variety of past experiences because of very small buffer size and thus present a poor generalization capability.

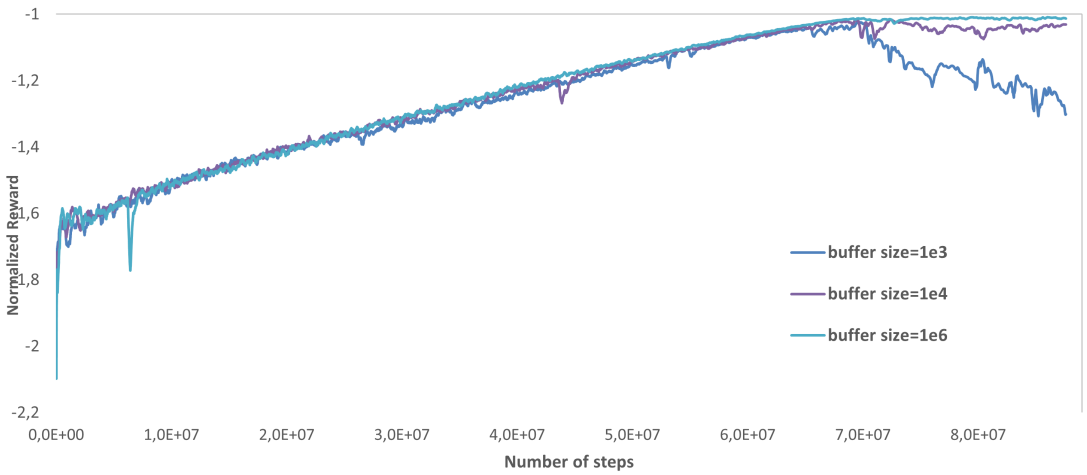


Figure 5.32: Learning curve of the DDPG agent for different values of the buffer size

Buffer Size	10^3	10^4	10^5	$\mathbf{10^6}$	10^7
Normalized Reward	-1.28	-1.03	-1.03	**-1.01**	-1.02

Table 5.18: Normalized final reward obtained by the DDPG agent for different values of the buffer size - multiple actions environment

Tuning of the batch size: the best obtained value of the batch size remains in line with the outcomes of the single-action agent tuning, consistently at 128, as depicted in Figures 5.33 and 5.19. However, a notable disparity with the results of the single-action environment arises with smaller batch sizes (64) where the multiple-action agent's performance is notably inferior to those obtained with the single-action setting.

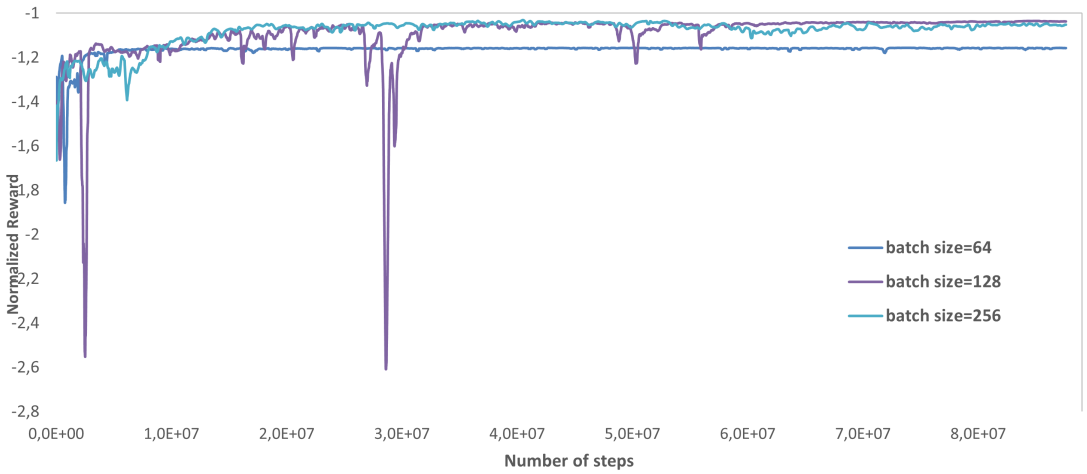


Figure 5.33: Learning curve of the DDPG agent for different values of the batch size

Batch Size	64	**128**	192	256
Normalized Reward	-1.16	**-1.02**	-1.04	-1.05

Table 5.19: Normalized final reward obtained by the DDPG agent for different values of the batch size - multiple actions environment

Tuning of the exploration noise To fine-tune the type and parameters of the exploration noise, we conducted experiments with three types of exploration noises, similarly to the approach taken for the single-action environment. Specifically, we investigated OU and normal distribution action noises as well as parameter noise.

* **Ornstein-Uhlenbeck (OU) action noise:** we explored seven standard deviation values for the OU action noise. The results of these tests, presented in Figure 5.34 and Table 5.20 determined the best value of the standard deviation at 15%.

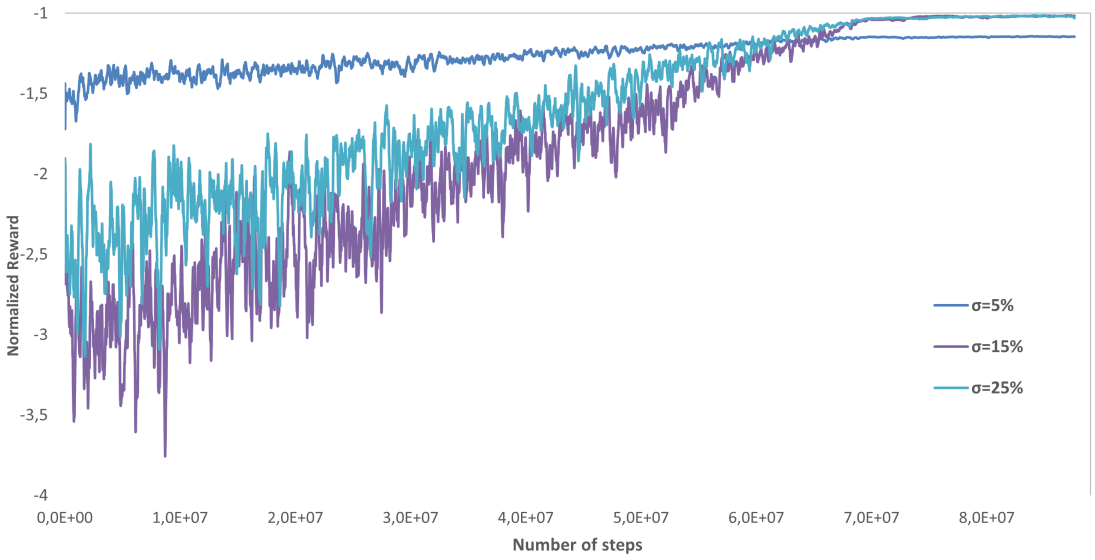


Figure 5.34: Learning curve of the DDPG agent with OU action noise, for different values of the standard deviation σ of the OU action noise.

OU noise volatility σ	5%	10%	**15%**	20%	25%	30%	35%
Normalized Reward	-1.15	-1.02	**-1.01**	-1.02	-1.02	-1.03	-1.16

Table 5.20: Normalized final rewards obtained by the DDPG agent for different values of the standard deviation σ of the OU noise - multiple actions environment

* **Normal action noise:** we also examined seven values for the standard deviation of the normal distribution action noise, with the best-performing standard deviation being at 20% as depicted in Figures 5.35 and Table 5.21.

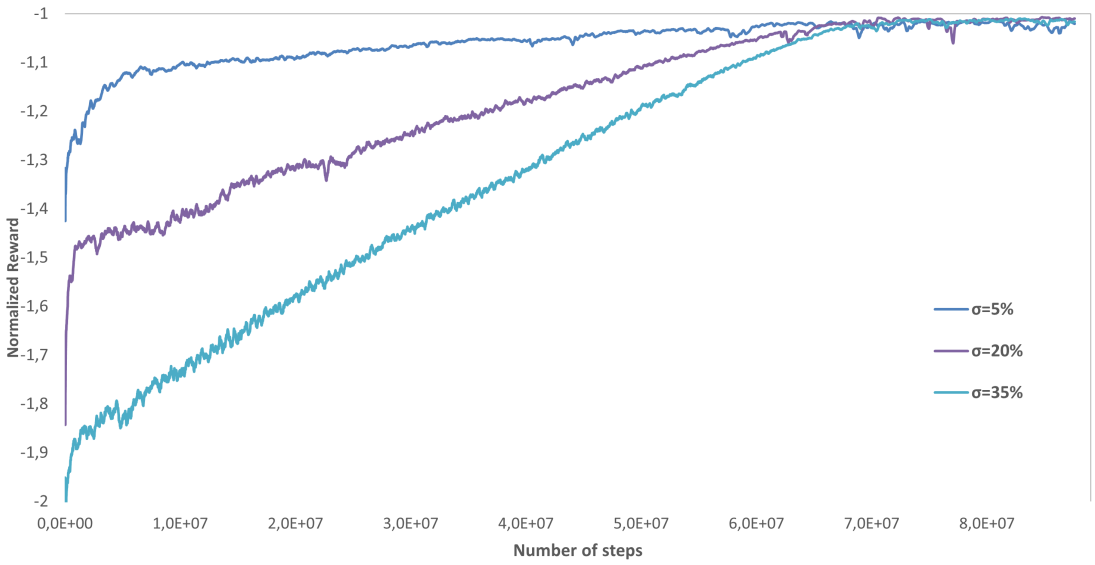


Figure 5.35: Learning curve of the DDPG agent with normal action noise, for different values of the standard deviation σ of the normal action noise.

Normal noise volatility σ	5%	10%	15%	**20%**	25%	30%	35%
Normalized Reward	-1.02	-1.04	-1.04	**-1.01**	-1.02	-1.02	-1.02

Table 5.21: Normalized final episodic reward obtained by the DDPG agent for seven different values of the standard deviation σ of the normal action noise - multiple actions environment

* **Parameter noise:** to assess the optimal value for the standard deviation of the parameter noise, we experimented five different values as presented in Figures 5.36 and 5.22. We found tha the most effective value for the standard deviation is 1%.

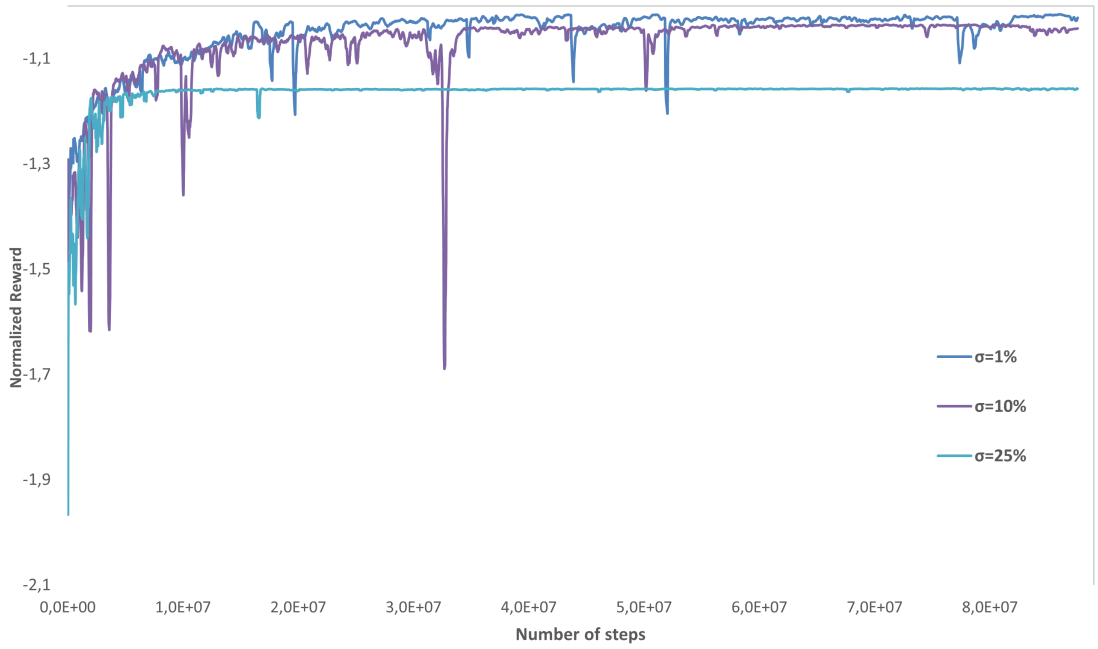


Figure 5.36: Learning curve of the DDPG agent with parameter noise, for different values of the standard deviation σ of the parameter noise.

Parameters noise volatility σ	**1%**	5%	10%	15%	20%
Normalized Reward	**-1.02**	-1.04	-1.16	-1.16	-1.16

Table 5.22: Normalized final episodic reward obtained by the DDPG agent for five different values of the standard deviation σ of the parameter noise - multiple actions environment

* **Comparison of different exploration noises:** when comparing the learning curves for the three different exploration noises, with the best obtained values of standard deviation for each noise type we observe that, unlike the single-action environment, the parameter noise is no longer the top performer with the multiple-action agent. Instead, normal distribution action noise exhibited a slightly better performance than parameter noise as can be observed in Figure 5.37. This observation suggests that the parameter tuning of the DDPG algorithm is likely to vary according to the complexity of the environment and the dimension of the action space.

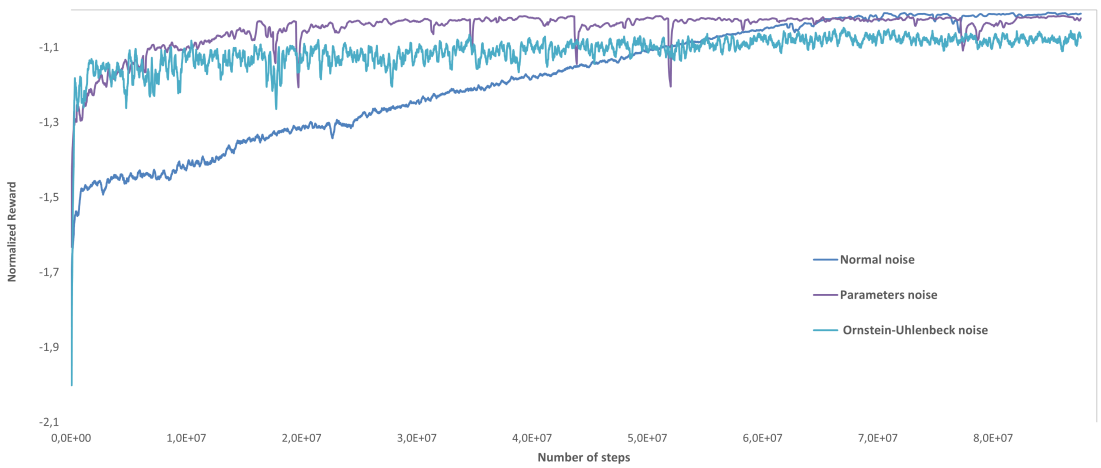


Figure 5.37: Learning curves of the DDPG agent for the three types of exploration noises: parameter noise, OU action noise and normal action noise.

Finally, the outcomes of the hyper-parameter tuning for both the single-action and the multiple-action setting are summarised in Table 5.23.

Parameter	Single-action setup	Multiple-action setup
Activation function	ReLU	ReLU
Number of hidden layer for the actor network	2	2
Number of hidden layers for the critic network	4	4
Size of the hidden layers	128	128
Learning rate of the actor	10^{-4}	10^{-4}
Learning rate of the critic	2.10^{-4}	2.10^{-4}
Discount factor (γ)	0.975	0.975
Soft-update parameter (τ)	5.10^{-3}	5.10^{-3}
size of the replay buffer	10^6	10^6
Batch size	128	128
OU noise standard deviation	**10%**	**15%**
Normal noise standard deviation	**15%**	**20%**
Parameter noise standard deviation	**5%**	**1%**
Best found exploration noise	**Parameter noise**	**Normal action noise**

Table 5.23: Summary table of the parameter tuning results for the single-action and multiple-action setups

5.6.3 Computational time

The experiments presented above were run on a computer with an Intel(R) Xeon(R) Gold 6138 CPU running at 2.00 GHz using 128 GB of RAM and running Windows 10. The training of the DRL agent takes around 37 seconds per episode for the single-action setup and 39 seconds for the multiple-action

setup	single-action setup	multiple-action setup
Mean	36.78	39.23
standard deviation	15.62	18.25
Q 25 %	28.16	29.02
Q 75 %	42.04	45.47

Table 5.24: Training time statistics per one year of training episode for the DDPG agent

setup as summarized in Table 5.24. We run training cycles of 10^4 episodes, which means that the overall computational time of a training phase is of about 100 hours. Nevertheless, once trained, the DRL agent becomes capable of running a one year simulation in about 16 seconds for the multiple-action environment. Meanwhile, the benchmark MPC algorithm takes 1135 seconds to simulate one year of the same environment.

5.7 Conclusion

This chapter presented the simulation results that we obtained by applying the DRL and the MPC approaches simultaneously on the same simulation model of multi-energy system case study 1 that was drawn from the MSE eco-district use case. These results showed that the DDPG agent presented, after learning and meticulous hyper-parameter tuning, a comparable performance to that of the MPC controller in terms of cost reduction, and a better computational performance and therefore demonstrated on a first simplified case study that DRL is a promising approach for the optimized multi-energy management of smart energy systems. In the remainder of this manuscript, this conclusion will be validated on a more complex case study 2 where the DRL approach is applied to solve the optimized multi-energy management problem of a digital twin that we developed under Modelica language for the MSE smart energy system.

DRL in Smart Multi-Energy Systems: the Meridia Smart Energy case-study

Introduction of Part II

This second part of the manuscript is devoted to the MSE case study. We first describe the MSE project and details the multi-energy systems that it involves. Then, the Modelia digital twin of MSE is presented and the modeling approach that we adopted for developing it is detailed. This digital twin, once exported as functional mock-up unit and wrapped into an OpenAI Gym environment, plays the role of the environment for the deep reinforcement learning agent for this case-study. Simulation results of applying our DRL framework for the optimal operation of the MSE digital twin are then discussed in the thrid chapter of this part.

Chapter 6

The Meridia Smart Energy case study

Résumé

Ce chapitre introduit le projet Méridia Smart energie (MSE) qui définit le contexte dans lequel ce travail de recherche doctoral a été mené. MSE est un éco-quartier dont les systèmes énergétiques représentent un démonstrateur de systèmes multi-énergétiques intelligents. Notre mission au sein de ce projet est de développer des systèmes de pilotage multi-énergétique intelligent qui assurent un fonctionnement optimisé des systèmes énergétiques flexibles au sein de MSE tout en optimisant un ensemble d'objectifs stratégiques. Le premier chapitre de cette deuxième partie du manuscrit décrit le projet MSE, les systèmes multi-énergétiques qu'il intègre ainsi que ses objectifs stratégiques.

6.1 Introduction

This chapter introduces the case-study project that constitutes the context in which this PhD research work was conducted. In fact, this work was part of the Meridia Smart Energy (MSE) project that aims at constructing an eco-district whose energy systems represent a real-life case-study of Smart Multi-Energy Systems. Our mission within this project is to develop Smart Multi-Energy Management Systems that ensure the optimal operation of the flexible energy systems within MSE while optimizing strategic objectives. The first chapter

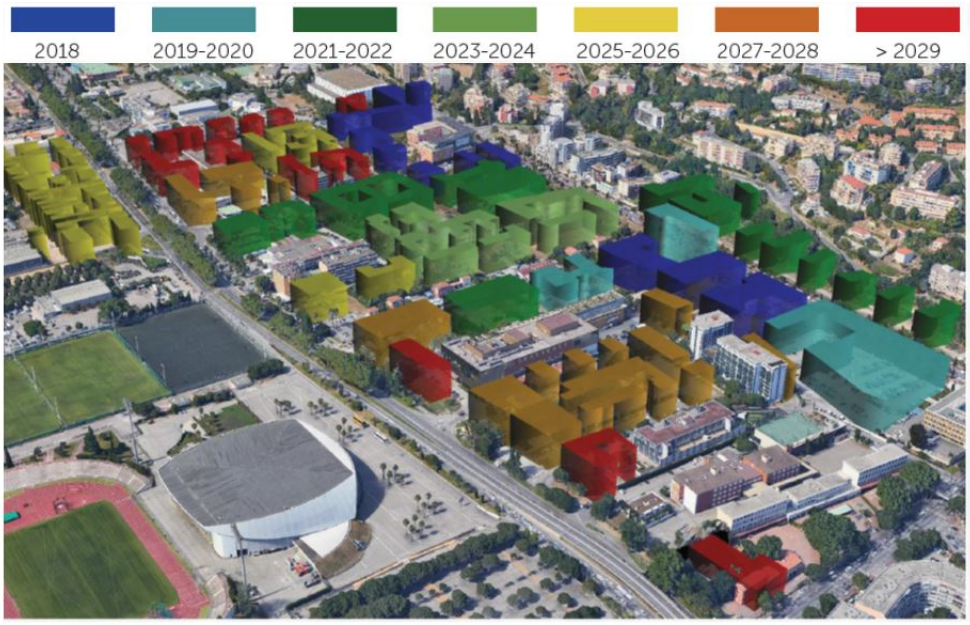


Figure 6.1: Development plan of the MSE eco-district buildings between 2018 and 2029 [338].

of this second part of the manuscript describes the MSE project, the multi-energy systems that it integrates and the strategic objectives that mostly drive the energy management systems that we develop for these systems.

6.2 The Meridia Smart Energy project

The case study under consideration in this work is part of the Merdia Smart Energy (MSE) eco-district currently under construction since 2018 in the Nice Meridia joint development zone located in the city of Nice, south of France. By the time this manuscript is being written, 8 buildings have already been connected to the District Heating and Cooling networks of MSE and a total of 50 buildings will be connected by 2033, which corresponds to almost 3500 households. The development planning of these 50 buildings is illustrated in Figure 6.1. The area covered by this neighborhood presents a density of new buildings of various types, including residential buildings, shops, showrooms, offices, healthcare facilities, university campuses, laboratories as well as other tertiary activities.

The energy systems of this eco-district include renewable energy generation via photovoltaic panels installed on the rooftops of office and residential build-

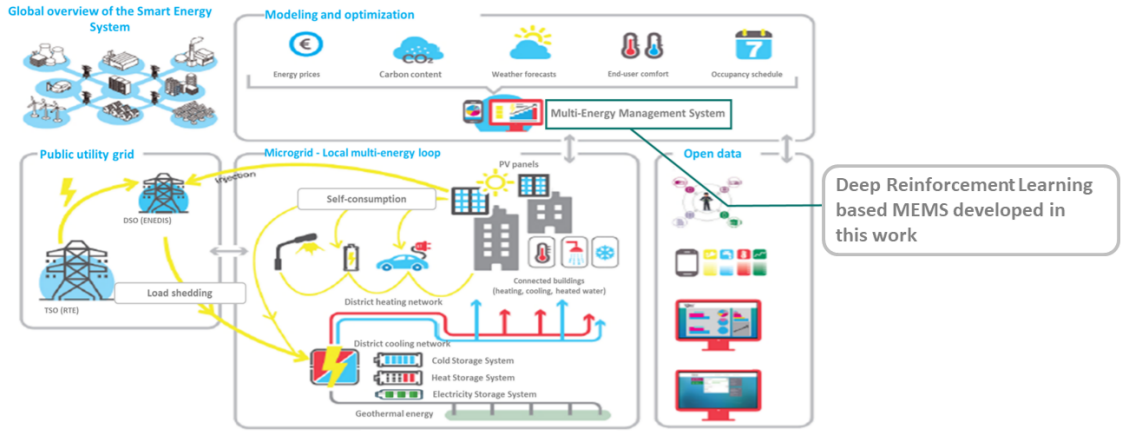


Figure 6.2: Overview of the main energy systems integrated in the MSE smart multi-energy system as well as their coordination process (adapted from [338]).

ings, a geothermal district heating and cooling system, ice storage tanks, heated water storage tanks, heat storage by phase change materials as well as battery energy storage systems. These multi-energy production, consumption and storage systems, together with the electric vehicle charging stations, the additional energy storage capacity that will later be provided by the electric vehicles (via Vehicle-to-grid systems) and the public lighting of the district need to be optimally scheduled and operated so as to achieve the strategic objectives laying behind this smart territory project. These optimization objectives are detailed in section 6.4.

6.3 Energy systems in the MSE Smart Multi-energy System

An overview of the energy systems involved in the MSE smart multi-energy system is presented in Figure 6.2 and the main components considered in the present work are divided into production, storage and consumption units and described in this section.

6.3.1 Energy generation systems

6.3.1.1 Geothermal District Heating and Cooling System

The Meridia Smart Energy eco-district is a cutting-edge sustainable urban development that includes a district heating and cooling system using geother-

mal Thermo-Refrigerating Heat Pumps (TRHP). This system provides buildings with both heat for heating as well as heated water storage tanks and cold for air conditioning, while also reducing energy consumption and greenhouse gas emissions. The geothermal District Heating and Cooling Network, together with the photovoltaic panels planned to be installed on the rooftops of the buildings, will ensure a share of renewable and waste energy of more than 70% in the eco-district.

The district heating and cooling system consists of a network of pipes that circulate water between a central heating and cooling plant and the substations of the district's buildings. The system is designed to simultaneously provide heating and cooling for the end-users, using a combination of renewable energy sources and energy-efficient technologies.

The central heating and cooling plant is powered by six geothermal thermo-refrigerating heat pumps, which extract heat from the aquifer during the winter months and reject heat back into the aquifer during the summer months. The geothermal heat pumps consist of a ground loop system that circulates a fluid, such as water or antifreeze, through a series of underground pipes. The fluid absorbs heat from the aquifer in the winter and transfers it to the heating system, while in the summer, the process is reversed and the system rejects heat into the aquifer. The technical characteristics of the geothermal thermo-refrigerating heat pumps of the MSE eco-district are given in appendix A.

Overall, the district heating and cooling network of MSE lays over 3.5 kilometers, will reach 5.6 kilometers by 2026, and comprises 12 geothermal wells ranging between 30 and 40 meters of depth each. A map of the heating and cooling network showing the locations of the geothermal wells as well as the power plant is given in figure 6.3. A simplified diagram of the heating and cooling power plant is illustrated in Figure 6.4 and a simplified P&ID (Process and Instrumentation Diagram) is presented in appendix A. The Thermo-Refrigerating Heat Pumps (TRHPs) and the chillers form the core of this heating and cooling power plant, and are specifically composed of:

* 2 tandems of 2 TRHPs each: Tandem 1 is composed of TRHPA and TRHPB and tandem 2 is composed of TRHPC and TRHPD.

* 1 positive chiller group
* 1 Positive-negative chiller group composed of 2 TRHPs.
* 3 adiabatic aero-refrigerant towers
* Access to the aquifer through a geothermal installation consisting mainly of extraction and injection wells.
* Various heat exchangers and auxiliary components.

In addition to these heat and cold production systems, the MSE power plant also involves thermal and power storage systems, namely:

* A heat storage by phase-change materials (PCM),
* A cold storage system by ice-on-coil,
* A battery energy storage system.

All the aforementioned energy systems will be described in further details in the next sections of this chapter. By the end of its development in 2033, the heating and cooling system is expected to have a total heat demand of 14765 useful MWh for a contracted power of 13491 kW, and a total cooling demand of 16832 MWh for a contracted power of 16741 kW. The network will reach 45 subscribers for 94 substations and a total of 50 buildings. The layout of the heating and cooling network presenting the locations of the substations is given in Figure 6.5.

The connection of the buildings gradually started between 2020 and 2023, and will carry on until 2033. The public service delegate, Idex, conducted a simulation of the heat and cold power demands of the network up to 2033. The modeling and analysis of the heating and cooling needs of the eco-district led to an evaluation of the combined heating and cooling needs of 6 MW and 10.5 MW respectively. The evolution of the heating and cooling needs from 2020 to 2033 is shown in the graphs 6.6 and 6.7.

6.3.1.2 Solar energy generation

Over the past decade, the overall cost of renewable energy sources has significantly decreased, making them competitive against traditional fossil fuels. In particular, the average cost of photovoltaic solar generation has decreased by $80\,\%$ since 2008 [339] and keeps on this trend. This significant decrease

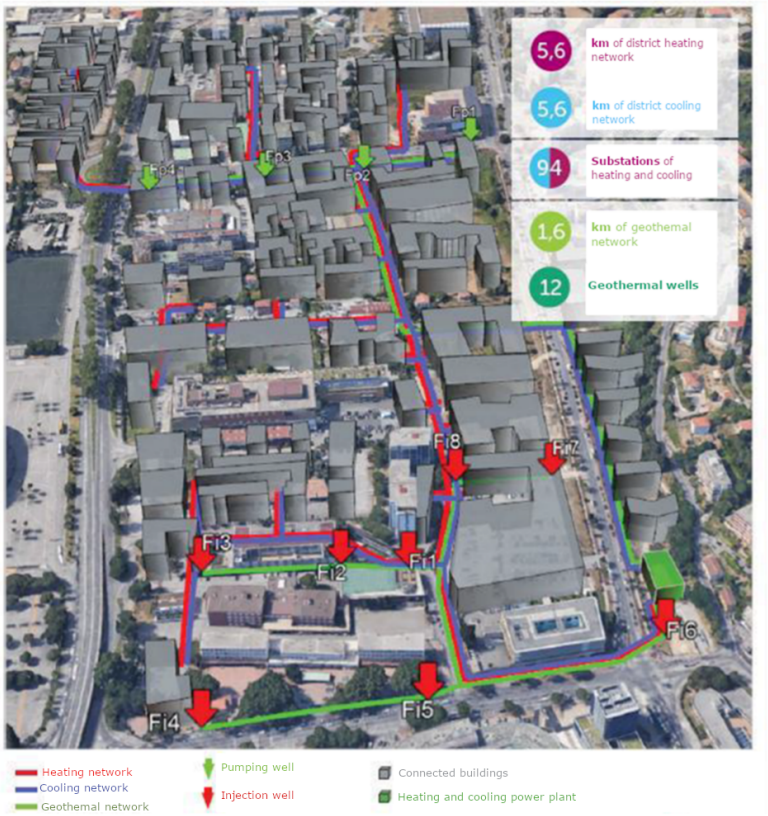


Figure 6.3: Map of the MSE eco-district showing the locations of pumping and injection geothermal wells and the heating and cooling power plant (source: Idex document).

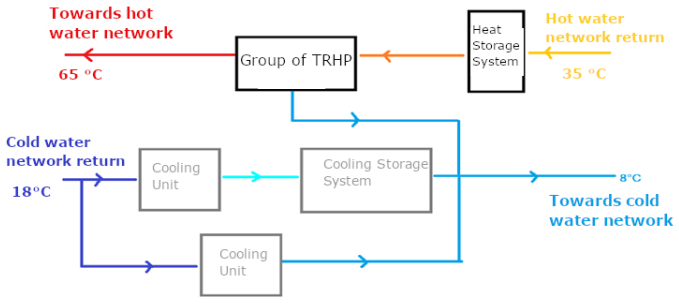


Figure 6.4: Simplified diagram showing the main components of the heating and cooling power plant of the MSE eco-district.

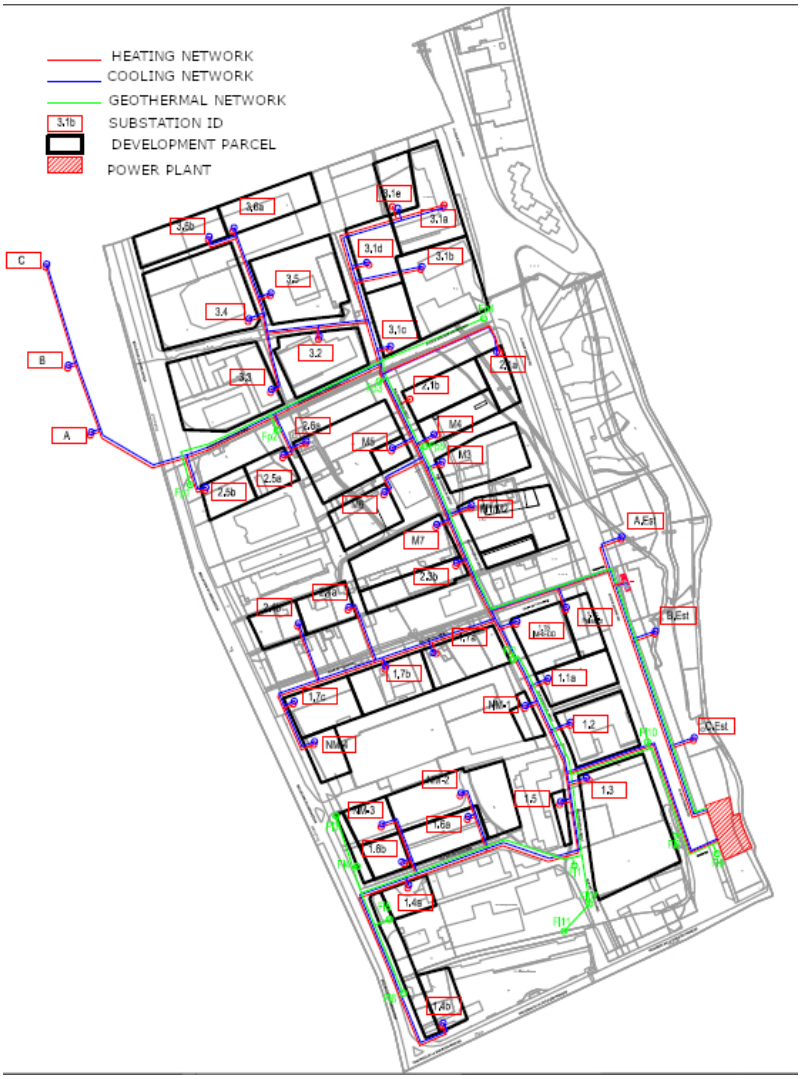


Figure 6.5: Layout of the geothermal heating and cooling network of the MSE eco-district illustrating the locations of substations. Each substation ID corresponds to two substations: one for heating and one for cooling (source: Idex document).

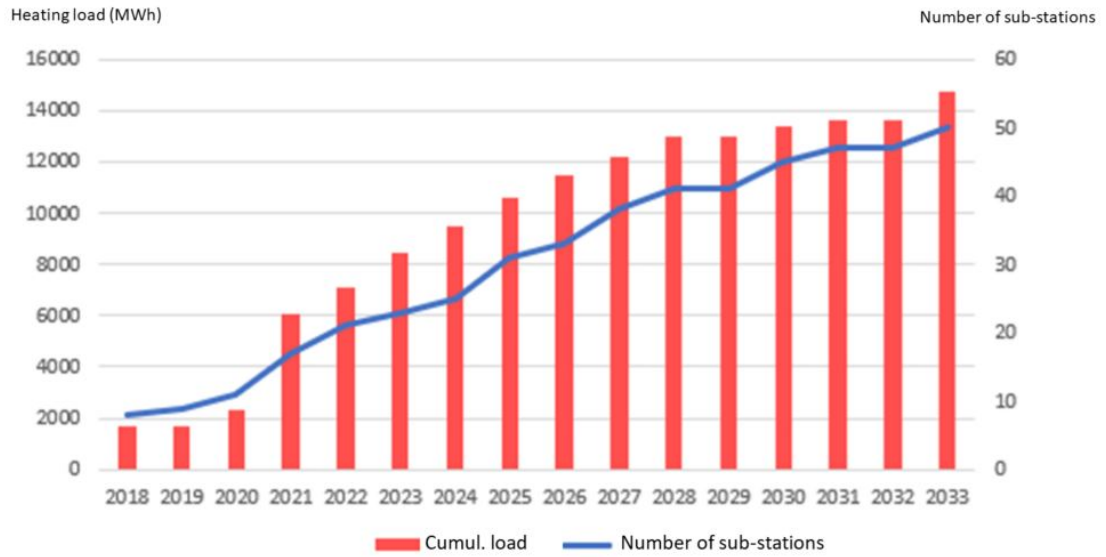


Figure 6.6: Evolution of the heat load and the number of heat substations for the district heating and cooling network of the MSE eco-district (source: Idex document).

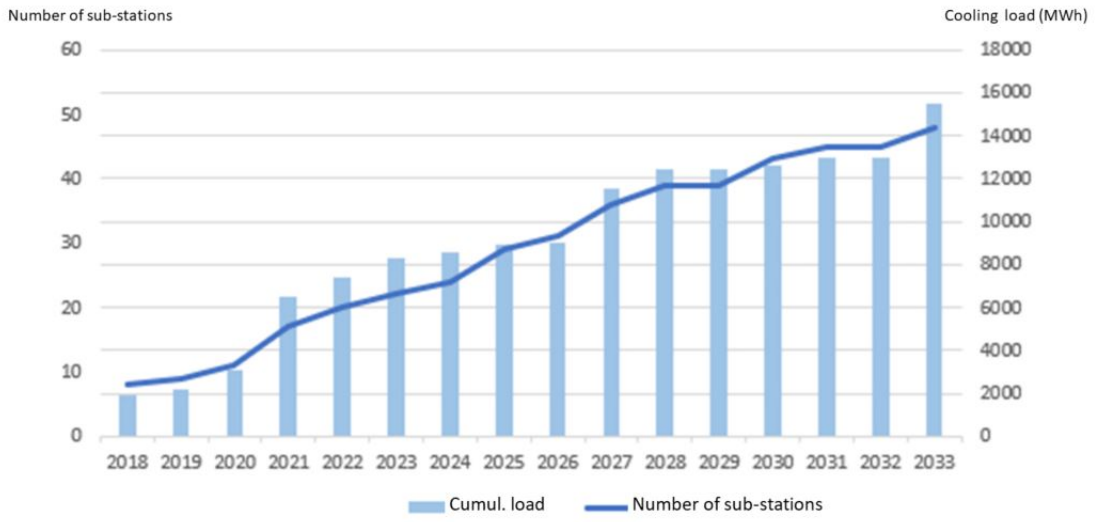


Figure 6.7: Evolution of the cooling load and the number of cooling substations for the district heating and cooling network of the MSE eco-district (source: Idex document).

has contributed to the growing deployment of renewable energy generation equipments. Between 2012 and 2015, the total installed renewable capacity worldwide increased by 60%, giving rise to a growing number of small-scale and decentralized energy production installations.

At the local level in the city of Nice, the "Plaine du Var" territory has a strong potential for photovoltaic production, ranging from approximately $1,400$ to $2,000$ kWh/m².year. That is why PV panels will be installed within the MSE eco-district. It is planned that almost 100% of the tertiary buildings will be equipped with PV panels on their rooftops. This is expected to allow for the installation of up to $5,000$ kWp, resulting in an approximate annual production of 6 GWh [340]. It is also possible for residential buildings to be equipped with PV panels on their rooftops and to have these equipments managed by the public service delegate Idex. These PV panels can be used to enable individual or collective self-consumption within the eco-district, for example, to heat domestic water in hot water tanks.

Finally, it is also envisaged that PV panels will be installed at the level of the heating and cooling production plant of the eco-district, primarily to add PV self-consumption through the battery as electrical flexibility alongside heat and cold storage at the plant level, thus enabling multi-energy management at the level of the power plant.

Nonetheless, as the eco-district is not yet fully constructed, the total surface of PV panels that will be installed, the resulting PV power generation, the brand or type of the PV modules, and the share of PV whose management will be delegated to Idex are not yet known by the time this manuscript is being written. Therefore, in order to account for this renewable electrical production in this case study, some assumptions had to be taken when modeling solar production at the eco-district level. Thus, a solar panel model has been validated, which allows for the generation of PV power based on given weather conditions, with a peak power of 1200 kWp. Once the total peak power is known, simulations can be performed using the appropriate module type and multiplying the obtained time series by a coefficient. In order to validate the PV model, simulations were conducted using four different weather patterns from

Nice. More details on these models will be presented in the next chapter.

6.3.2 Energy storage systems

6.3.2.1 Thermal energy storage systems

In DHCS, two main gaps exist between thermal energy demand and supply. The first gap arises from the time difference between thermal energy generation and consumption which can be attributed to physical factors such as the intermittent nature and the dynamic characteristics of solar generations, as well as economic factors related to the variability of the thermal energy cost during the day. The second gap is related to the geographical distance between thermal generation plants and the locations where heat and cold are consumed. These gaps may lead to wasting the thermal energy that is not consumed and to increasing inefficiencies if the thermal generation has to follow the thermal demand. Thus Thermal Energy Storage Systems (TESS) appear as a promising solution for the smart management of the gap between supply and demand. Not only can they be used as a buffer between load and generation by storing heat and cold for later use, but they also contribute in integrating renewable energy sources into DHCS and maximizing their flexibility and performance [61]. Therefore, their integration in the industrial and building sector in the European union has the potential to yield annual energy savings of about $7.8\,\%$ (approximately 1.4 million GWh of energy annually [341]), together with a CO2 emission reduction by $5.5\,\%$ [342].

Heat storage by phase-change materials

Phase Change Materials (PCM) constitute a promising option for a cost-effective and energy-efficient heat storage due to their ability to absorb and release a large amount of thermal energy at a constant temperature [343] during the phase change (melting/ solidification) process. The PhD thesis of Martellini [344] investigated the potential of thermal energy storage using PCM for DHS by exploring various aspects of PCM-based thermal energy storage including PCM selection, storage system design and integration within district heating networks as well as operation strategies optimization based on numerical simulations.

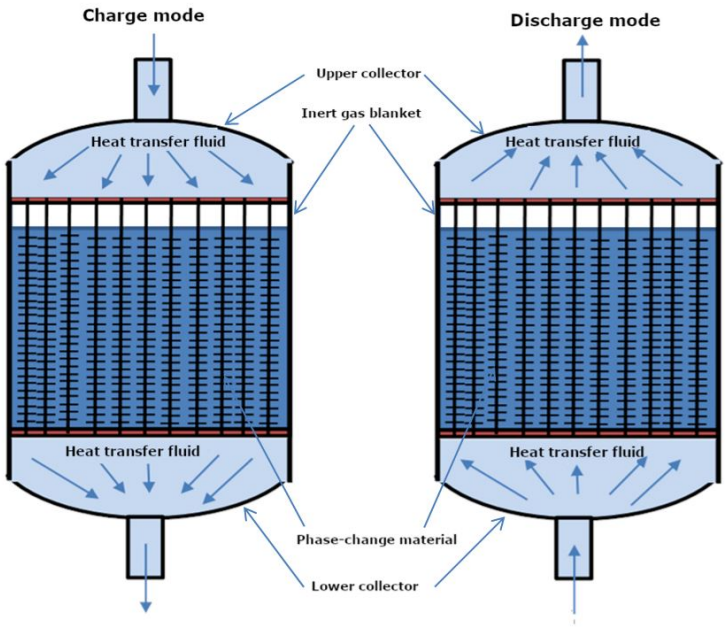


Figure 6.8: The design and main components of the Phase-change material heat storage system of the MSE eco-district (adapted from a CEA document).

Based on the insights from the research work of Martellini et al. [345], the french Alternative Energies and Atomic Energy Commission (CEA), member of the MSE consortium and a key actor of this demonstrator project designed, constructed and installed a PCM-based Heat Storage system within the MSE eco-district. This heat storage of about 1.2 MWh capacity is the first of its kind and aims to demonstrate the feasibility and benefits of using PCM-based thermal energy storage in district heating networks, with the ultimate goal of reducing the overall peak energy consumption and carbon footprint of the eco-district.

The design of this storage system, presented in Figure 6.8 is based on a tube and shell heat exchanger technology of 12 meter tall and 2 meter diameter, with a bundle of about 512 tubes in the middle and the PCM flowing around the tubes. Unlike most of the components of the power plant which are located indoors, the heat storage system is located outside due to its large size, as illustrated in Figure 6.9. The system was installed within the power plant in 2023 and the first tests and experiments on this new systems were conducted

Figure 6.9: The PCM heat storage system of the MSE eco-district installed outside the power plant.

starting from march 2023. The selected PCM is a glycoled water solution that has a melting temperature of around 58°C. The storage system operates in "charge" mode when the heat transfer fluid (water) enters the storage at around 65°C with the PCM being in the solid phase and in "discharge" mode when the water passing through it is at a temperature of 35°C and the PCM being in liquid phase. The operation of the storage depends on that of the geothermal TRHPs of the power plant introduced in the previous section.

Cold storage by ice on coil

When it comes to cold storage, several technologies are used within DHCS, including nodular storage systems and tube-based technologies known as ice on coil systems. In the MSE project, an ice on coil cold storage system has been integrated into the heating and cooling network at the power plant. Unlike the heat storage, the cold storage system was installed indoors. The principle of the ice on coil storage system involves submerging tubes in a water pool or tank. A glycoled water solution flows through the tubes, lowering its temperature to around -5°C until freezing the water surrounding the pool tubes. This process results in a large block of ice within the pool when the storage systems is fully charged. During discharge, warmer glycoled water solution circulates through the tubes, melting the ice and thus releasing stored

Table 6.1: Properties of the cold storage system.

Technical properties	Unit	1 tank	2 tanks
Total storage capacity	kWh	3795	7590
Latent storage capacity (ice)	kWh	3458	6916
Sensible storage capacity (0°C to +5°C))	kWh	337	674
Maximum operating temperature	°C	40	40
Maximum operating pressure	bar	3	3
Total water volume	m^3	58	116
Total ice volume	m^3	37	74
Total glycoled water solution volume	m^3	1.88	3.76
Connectors, NP10 flanges	ND	150	150

energy. To enhance the system's performance, air agitation can be incorporated by diffusing air bubbles through compressors at the bottom of the pool. This helps accelerate the transformation process and provide a higher power output, which can be particularly useful during peak cold demand periods on the DHCS.

The ice storage tank occupies a significant space within the heating and cooling power plant, with a total water volume of $124\,m^3$ and a latent capacity of $6916\,kWh$. During winter, both the ice storage and the cooling units would remain inactive due to low cooling demand. However, activating the system during the summer season proves beneficial as it may allow avoiding the excess heat that would otherwise be rejected into the geothermal wells or through the adiabatic aerorefrigerant towers installed on the rooftop of the heating and cooling power plant. Hence, the cold storage allows to produce a large quantity of cold without the need to reject excess heat. A cross-sectional view of the cold storage system installed in the heating and cooling plant of the MSE eco-district is presented in Figure 6.10 and its main technical details are summarized in Table 6.1.

According to the manufacturer's estimates, this storage system could provide up to 2 MW of power, and when air agitation is activated, the power output could reach up to $10\,\%$ of the storage's capacity.

During the charging phase, the negative cold group operates to cool the water entering the ice storage to -6°C. During discharge, through an exchange sys-

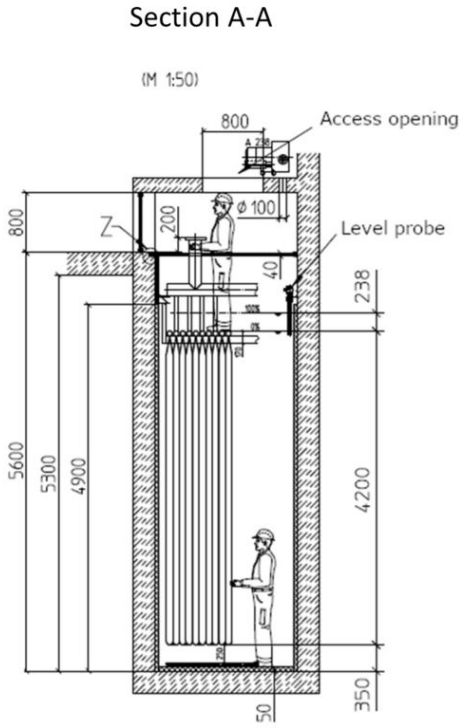


Figure 6.10: Cross-sectional view of the ice on coil cold storage system installed inside the power plant of the MSE eco-district (source: Idex document).

tem, the water reaches a temperature of 16°C as it enters the ice storage. A complete charge cycle typically takes around 8 hours.

6.3.2.2 Battery Energy Storage Systems

The battery energy storage system installed within the power plant of the MSE eco-district is one of the key energy systems offering an additional flexibility potential to the MSE smart energy system. Similarly to the heat storage system, the battery storage system was also installed outdoors, mainly due to its large size, as illustrated in Figure 6.11. This battery storage system consists in a lithium-ion battery that boasts a capacity of $616\,kWh$.

This storage system offers several versatile applications. Among the intended usages in the MSE project, we firstly consider enabling self-consumption for instance by allowing excess energy generated from local renewable sources such as PV panels to be stored and consumed later during periods of low generation or high load. Maximizing self-consumption not only contributes in efficiently integrating renewable energy sources but also reduces dependency

Figure 6.11: Picture of the battery energy storage system installed near to the power plant of the MSE eco-district.

on the power grid and thus reinforces the energy self-sufficiency of the smart energy system.

Additionally to self-consumption, the BESS can be used for providing frequency regulation services to the transmission system operator. Indeed, BESSs are deemed to be efficient in providing such services mainly due to their fast ramp rate as explained in [15]. By dynamically adjusting their charge or discharge rates, they can therefore help mitigate frequency deviations of the grid by injecting or absorbing power.

In addition to self-consumption and frequency regulation, the BESS can also be used for energy arbitrage, i.e for buying and selling electricity by profiting from the wholesale electricity price fluctuations. This application allows for cost savings by optimizing the electricity usage of the overall smart energy system during periods of high demand. If injecting power on the main utility grid is allowed, the purpose of arbitrage can go beyond cost saving and extend to revenue maximization through participation in energy markets.

Peak shaving is another valuable application of the BESS that consists in reducing electricity demand during peak hours when electricity prices are at their highest mainly by storing energy during off-peak hours and discharging it during peak periods to meet the power demand. Peak shaving does not only allow optimizing energy costs similarly to energy arbitrage, it can also enable

reducing carbon footprint.

6.3.3 Energy usages

6.3.3.1 Sub-stations, buildings and end-users

The MSE eco-district encompasses a comprehensive range of energy usages comprising heating, cooling and electricity. Heating plays a vital role in maintaining comfortable indoor temperatures for the buildings' residents mainly during cold seasons. This includes heating systems for the space heating and the production of domestic hot water. Most of heating within the eco-district is produced by the geothermal DHCS. Only some of the domestic hot water storage tanks are "combined", which means that they offer the possibility of using either heat coming from the DHCS or electricity to heat the domestic water. This offers an additional possibility of arbitrage that can be exploited by the energy management system.

When it comes to cooling usages, they mainly consist in air conditioning used to encounter higher temperatures during hot weather periods and therefore contribute to creating a comfortable living and working environment for the residents of the eco-district. Similarly to heating, most of the cooling usages within the eco-district will be provided by the geothermal DHCS. This sustainable solution harnesses renewable energy from geothermal sources and also helps mitigate the urban heat island effect that is often triggered by conventional individual air conditioning systems [346].

Lastly, electricity usages include various needs within the buildings such as lighting, electronic devices and other electricity-dependant appliances. In addition to buildings, one key electricity usage within the MSE eco-district is given by the power consumption of the district heating and cooling power plant which consumes electricity in the thermo-refrigerating heat pumps and other delivery pumps to produce and distribute heating and cooling all over the district network.

Within the heating and cooling network of MSE, each building is provided with an average of one sub-station. A sub-station is a delivery point located in a technical room next to the building and one can distinguish two types

of sub-stations: single heating sub-stations and combined heating and cooling sub-stations. The bloc diagram in appendix A illustrates the different sub-stations that will be connected to the MSE DHCN throughout its development. The portion of the heating and cooling network located upstream of the substation is referred to as the primary network and the part located downstream of the sub-station is called the secondary network. After each heat exchanger of the secondary network, temperature regulation is ensured by a communicating controller that is connected to the supervision system of the public service delegate.

Regarding the technical architecture, heating substations are composed of a skid including the following components:

* A hot water exchanger that supplies buildings with heat for space heating, domestic water heating as well as any other process that requires a negative enthalpy transfer from the network. The maximum temperature regime on the primary heating network is 65°C on the supply side and 35°C on the return side.
* A two-way control valve on the primary side with various temperature sensors on both the primary and secondary sides.
* A set of pipelines, isolation valves and draining points.
* A set of thermometers with immersion sleeves on both the primary and secondary sides.
* A thermal energy meter.
* An electrical cabinet.

On the other hand, sub-stations also include a cold water skid that includes the following components:

* A cold water exchanger that supplies buildings with cold for space cooling, servers and any other process that requires a positive enthalpy transfer from the network. The maximum temperature regime on the primary cooling network is 8°C on the supply side and 18°C on the return side.
* A two-way control valve on the primary side with temperature sensors on both the primary and the secondary sides.

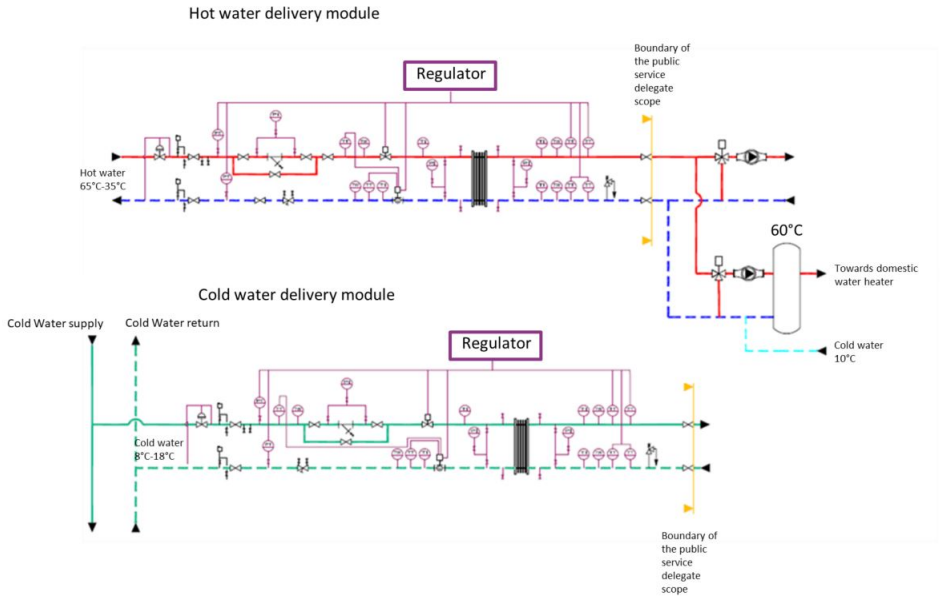


Figure 6.12: Schematic diagram of the primary side of a combined heating and cooling substation.

* A set of thermometers with immersion sleeves on both the primary and secondary sides.
* A thermal energy meter.

An example of schematic diagram of a combined heating and cooling substation is illustrated in Figure 6.12.

6.3.3.2 Electric vehicles and electric Vehicle charging stations

As the eco-district evolves, electric vehicles will be deployed and public and private electric vehicle parking lots and charging stations will be installed to support the electrification of transportation within the eco-district. Besides, advanced technologies such as smart charging and Vehicle-to-grid (V2G) will be pursued and actively integrated within the multi-energy management system. Smart charging will be used to optimize the charging process of electric vehicles by dynamically adjusting their charging rates regarding various factors like electricity demand, grid capacity and renewable energy generation patterns. On the other hand, the V2G concept enables electric vehicles to not only be energy consumers but also act as mobile power storage systems. Thus, electric vehicles can for example discharge their stored energy back to the grid

during periods of high demand or grid instability. This bi-directional power flow between electric vehicles and the grid can thus offer an additional flexibility in mitigating peak loads, balancing supply and demand and reinforcing the resilience of MSE smart energy system.

Similarly to most of the equipments that belong to the multi-energy infrastructure of the MSE eco-district, the operators of these systems have the possibility to delegate their management to Idex. If so, they should have these equipments meet a Smart Grid Ready charter defined by Idex. In particular, each Smart Grid Ready equipment should be able to offer services that belong to one of the three following levels:

* Level 1: ability to transmit information to the local energy manager of the eco-district.
* Level 2: ability to receive demand response orders from the local energy manager of the eco-district. For the particular case of electric vehicle charging stations, demand response orders consist in smart charging orders. In order to be able to offer this service, they should comply with the OCPP[1] v 1.6 or v 2.0 standard and their supervision platforms have to be compatible with the OSCP [2] standard [340].
* Level 3: ability to inject power back into the grid. For the particular case of electric vehicle charging stations, this consists in V2G. However, since there are no technical standards and corresponding offers currently available within MSE, this level is not expected for several years.

While these systems are not yet implemented in the current phase of the eco-district's construction, they are expected to be progressively added in future phases of the project development. They will then be seamlessly added to the energy management systems of the eco-district but they are not yet taken into account in the current model of the system since technical information on these systems is not yet available.

[1]OCPP (Open Charge Point Protocol) is the open communication protocol between electric vehicle charging stations and their management system [347].

[2]OSCP (Open Smart Charging Protocol) is the open communication protocol between the management systems of a charging station and an energ management system. Both OSCP and OCPP protocols are managed by the Open Charge Alliance OCA) consortium.

6.3.3.3 Public lighting

Similarly to electric vehicles and charging stations, the public lighting within the MSE eco-district can be optimally managed by the smart multi-energy management system. Public lighting can in fact be modulated by adjusting the intensity of streetlights based on various factors like the time of the day, the natural ambient brightness, the presence and movement of people, as well as other predefined parameters. This aims at reducing energy consumption within the eco-district and associated costs while keeping adequate lighting levels to ensure safety and comfort for users, and can be achieved for example by using ambient light sensors and programmable timers.

It is worth noting that such operations on the public lighting require special concessions from the city or the metropolitan authority (the Nice Côte d'Azur Metropolis in the case of MSE). Each of these entities should be equipped with a supervision platforms that the local energy manager should be able to communicate with. Depending to the operator's choice, the public lighting equipment and the aforementioned platforms should be compliant with the Smart Grid Ready charter and thus be able to provide one of the following service levels:

* Level 1: ability to transmit information to the local energy manager of the eco-district.
* Level 2: ability to receive demand response orders from the local energy manager of the eco-district. These orders consist for instance in modulation actions aiming at adjusting the lighting intensity of the streetlights.
* Level 3: ability to inject power back into the grid. Due to the absence of storage systems within the public lighting network, this service level is not expected in the MSE eco-district but can be considered in other similar eco-districts.

Finally, it's important to note that the management of electric vehicle charging and public lighting is not yet operational within the eco-district. Therefore, the integration of control for these elements into the energy management system developed in this work is planned for future phases as the eco-district evolves.

6.4 Strategic optimization objectives in MSE

The key strategic objectives behind the Meridia Smart Energy (MSE) project are fixed by the public service delegate Idex, the delegating territorial authority Metropole Nice Cote d'Azur, as well as the ADEME funder of the project through the program "Investissements d'Avenir" that they operate. Among these major strategic objectives we note:

* Maximizing the share of renewable energy and waste in the eco-district (to reach more than $70\,\%$) and maximizing its self-consumption and energy autonomy,
* Minimizing the overall energy consumption and operational costs of the Smart multi-energy systems of the eco-district, reducing load peaks and minimizing the energy bills of the end users,
* Minimizing Green House Gases emissions and fossil fuel consumption,
* Valuing the flexibility potential provided for instance by the multi-energy storage systems of the eco-district by demand Side Management of the heating and cooling power plant and participation in certain ancillary service markets (e.g. frequency regulation market).

6.5 Conclusion

This chapter provided a comprehensive overview of the multi-energy systems integrated within the MSE eco-district, that form together the main smart multi-energy system case-study for the current research work. The strategic optimization objectives that lay behind developing a smart multi-energy management system for MSE were then outlined.

In the up-comping chapter, we will focus on the digital twin that we developed for the MSE eco-district and that served as a simulation environment for the reinforcement learning algorithms and enabled the training, testing and performance evaluation of the energy management systems developed in this work.

Chapter 7

Building a simulation model for the Meridia Smart Energy eco-district

"All models are wrong, but some are useful"

Georges Box

Résumé

Ce chapitre détaille le développement d'un modèle de simulation pour l'ensemble des systèmes multi-énergétiques de l'éco-quartier MSE présenté dans le chapitre précédent. Ce modèle de simulation a été construit en langage Modelica (sur le logiciel Dymola) afin de rendre compte avec précision du comportement dynamique du système. Le principal objectif de ce jumeau numérique est de servir d'environnement pour le système de gestion de l'énergie basé sur l'apprentissage par renforcement profond (DRL) présenté précédemment et développé sous Python: le jumeau numérique est encapsulé en tant qu'unité de modélisation fonctionnelle (Functional Mock-up Unit, FMU) puis en tant qu'environnement OpenAI Gym pour permettre son interaction avec les algorithmes DRL développés sous Python. L'interaction entre l'agent DRL et le FMU du jumeau numérique s'effectue grâce à la co-simulation en utilisant la librairie FMPy. Cette approche de modélisation globale est détaillée dans ce chapitre.

7.1 Intrdoduction

This chapter presents the development of a simulation model for the multi-energy systems within the MSE eco-district case-study presented in the previous chapter. The simulation model is built using mainly the Modelica language, through the Dymola software, in order to accurately account for the dynamic behaviour of the system. The main purpose of this digital twin is to serve as an environment for the deep reinforcement learning based energy management system previously presented and developed using Python: the digital twin is encapsulated as a Functional Mock-up Unit (FMU) to enable its interaction with the Python framework through co-simulation. This overall modeling approach is detailed in this chapter.

7.2 The modeling approach

7.2.1 The modeling purpose and structure

Modeling plays a crucial role in understanding and analyzing complex systems such as smart multi-energy systems. In general, modeling approaches can be classified into three categories: white box, black box and grey box modeling [348]. White box models, also referred to as physical models, aim at providing a detailed understanding of the underlying system by incorporating its fundamental principles and equations. Black box models, on the other hand, focus solely on the input-output relationship of the system and do not explicitly consider its internal processes and dynamics. Finally, grey box models combine elements of both white box and black box modeling by incorporating some knowledge about the system while abstracting certain details.
On the other hand, the purposes of the system modeling can be classified into two types: simulation and optimization [348]. Simulation models aim at replicating the behaviour of the real system, capturing its dynamics and giving insights about its behaviour and performance under different operating conditions as well as forecasting its future behaviour. That is why they are also referred to as forecasting models for instance by Klemm and Vennemann [349]. On the other hand, optimization models aim at minimizing or max-

imizing some specific criteria or objectives in order to find optimal system configurations, sizing, or operating strategies.

The model of the multi-energy systems described in this chapter was built in order to replicate the behaviour and dynamics of the actual systems so as to be integrated into the optimized energy management frameworks. Once the modeling purpose is defined, one has to choose the structure of the multi-energy system models to be developed. Modeling structures also commonly fall into two types: top-down and bottom-up approaches. The top-down approach usually starts with a broad, high-level representation of the system, then gradually refines it by decomposing the system into its individual subsystems or components. In contrast, the bottom-up approach generally begins by modeling individual components or subsystems which are then progressively integrated and interconnected to construct a comprehensive representation of the entire system. In this work, we embraced a bottom-up approach for modeling the multi-energy systems within the MSE eco-district. By identifying key components of the overall system, carrying a detailed modeling of each of these components, and subsequently interconnecting them, we aimed to construct a comprehensive simulation model that captures the behavior and dynamics of the system more accurately. Additionally, the individual models built for the various defined subsystems provide us with a valuable repository of reusable subsystem models. This collection of subsystem models can be effectively used in the future for modeling and analyzing similar systems, thereby enhancing the reproducibility of the developed solution for comparable smart multi-energy systems.

7.2.2 The modeling tool

Modeling tools designed for multi-energy systems frequently adopt the aforementioned bottom-up approach, wherein the system's components are modeled using diverse libraries, and subsystem models are subsequently interconnected. Notable examples of such modeling tools include EnergyPlus [350] for building energy simulation, TRNSYS [351] for simulating transient systems behaviour, encompassing both electrical and thermal energy systems,

and the Modelica [352] language for the object-oriented modeling of intricate and multi-physics systems [353]. Modelica is supported by both an open-source modeling and simulation environment, OpenModelica [354], and a commercial software, Dymola [355]. For an in-depth review of modeling and simulation tools tailored for multi-energy systems, we refer the interested readers to the works of Allegrini et al. [356] and Klemm and Vennemann [349]. These works provide comprehensive reviews of modeling, simulation, and optimization tools specifically designed for district-scale energy systems. Moreover, Gronier et al. [357] offer a review of literature references highlighting energy modeling tools.

In this case-study, we opted for the use of the Modelica language, predominantly using the Dymola software tool, for constructing the simulation model of the MSE case-study. Notably, Dymola stands out as a robust commercial software, recognized for its capability to handle models with a substantial number of equations and to deliver simulation results with enhanced computational efficiency compared to other Modelica compilers [348], [358]. Additionally, Dymola ranks among the tools and Modelica compilers that exhibit a high level of compatibility with the Functional Mock-up Interface (FMI) standard (refer to Figure 7.1). FMI is a standard that serves as a common framework, facilitating the exchange and co-simulation of dynamic models across diverse software platforms. A focus on this standard and how we use it to manage the interaction between the energy management system framework and the simulation model is presented in section 7.4.1.

7.3 Aggregate simulation model

7.3.1 Model of the district heating and cooling network

7.3.1.1 The power plant

In order to build an initial comprehensive model of the district heating and cooling network, we first adopted a simplified model of the heating and cooling power plant that simultaneously produces heat and cold to meet the eco-district's buildings needs. Then, we modeled the heating and cooling network,

	FMU Export				FMU Import			
	Co-Simulation		Model-Exchange		Co-Simulation		Model-Exchange	
	1.0	2.0	1.0	2.0	1.0	2.0	1.0	2.0
Dymola								
Open Modelica								
JModelica								
Matlab								
SimulatorTo FMU								
PyFMI								

Figure 7.1: Compatibility of Dymola and other alternative tools with the FMI standard 1.0 and 2.0 for FMU export and import with Co-Simulation (CS) and Model-Exchange (ME) (adapted from [359]).

including the sub-stations in order to obtain an initial model of the complete network. Subsequently, we replaced the simplified model of the power plant with a more detailed model, which will be presented in section 7.3.2.

In the initial model, the heating and cooling power plant was represented as a black box that simulates the heating and cooling processes of the fluid to meet the setpoint temperatures and calculates the power transferred to the fluid. The parameters of this model include temperature setpoints, nominal mass flow rate, and nominal pressure drop (which can be optional). To develop this model, we used the *PrescribedOutlet* model from the Modelica Buildings library [360]. This model consists of a flow source with a prescribed temperature on the primary side of the network. It acts as an infinite reservoir that is capable of absorbing or generating as much energy as needed to maintain the temperature at the specified value. The return side was only defined by the pressure of the heat transfer fluid, since mass flow rate, enthalpy, and temperature vary during the simulation.

7.3.1.2 The heating and cooling network

A detailed topological simulation of the piping system of the district heating and cooling network is beyond the scope of this work. Such simulations are rather performed for very large networks where long pipelines can lead to

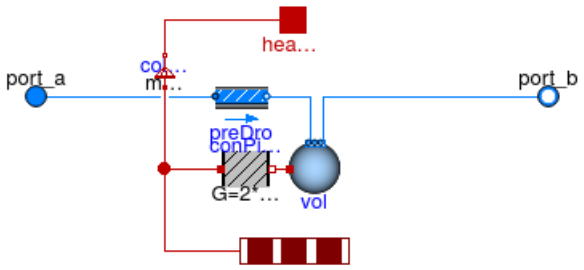


Figure 7.2: Dymola model of a pipe.

complex flow distributions, particularly in bidirectional networks.

The model developed in our work takes as input the complete time series for heating, cooling, and domestic hot water (DHW) demands. Each load is connected to the heating and cooling power plant through pipes at different heat exchange substations. The pipe parameters include the nominal mass flow rate per section. The regular pressure drop for each section is calculated using a simplified model like:

$$\dot{m} = k\sqrt{\Delta p} \tag{7.1}$$

Where $\dot{m}$ is the mass flow rate, Δp is the pressure drop and k is constant that is defined based on the nominal values of the mass flow rate and the pressure drop $\dot{m}_{nominal}$ and $\Delta p_{nominal}$.

The parameters of the fluid supply and return pipes include their length, the thickness and thermal conductivity of the insulation, the nominal mass flow rate and the velocity of the fluid. The pipe diameter can be automatically determined by the component following the equation:

$$d = \sqrt{\frac{4\dot{m}_{nominal}}{\rho\pi\Delta p_{nominal}}} \tag{7.2}$$

The developed Dymola model of a pipe is illustrate in figure 7.2 and is based on the following equations (Note that in Dymola, the variables and parameters contained in a sub-model are referred to by using the sub-model's name followed by a dot and the name of the input or output. For example, $port\_a.m\_flow$ represents the mass flow rate m_flow at $port\_a$):

* Mass flow conservation:

$$port\_a.m\_flow = -port\_b.m\_flow \tag{7.3}$$

* Equilibrium of potentials with pressure drops:

$$port\_a.p = port\_b.p + \Delta p \tag{7.4}$$

* Energy conservation with thermal losses, in the flow direction and the opposite direction respectively:

$$port\_a.m\_flow * (inStream(port\_a.h\_outflow) - port\_b.h\_outflow) = -Q\_flow \tag{7.5}$$

$$port\_a.m\_flow * (inStream(port\_b.h\_outflow) - port\_a.h\_outflow) = Q\_flow \tag{7.6}$$

The thermal power dissipated along the pipe is calculated, based on available geometric data and the outside temperature, using the following classical formula:

$$Q = G.\Delta T \tag{7.7}$$

Where G is the conductance of the material and $\Delta T = T_{internal} - T_{external}$.

7.3.1.3 The heat and cold substations

The substations are elements of the heating and cooling network that extract a certain amount of heat or cold based on the needs of the buildings to which they are connected. On average, each building is connected to one heat substation and one cold substation. The heat substations cater to the heating and domestic hot water needs of the buildings and the cold substations address the cooling needs.

The primary functionality of a substation is to connect the energy production source with the consumer point of that energy. We get the mass flow rate in each branch of the network by applying mass conservation at each node,

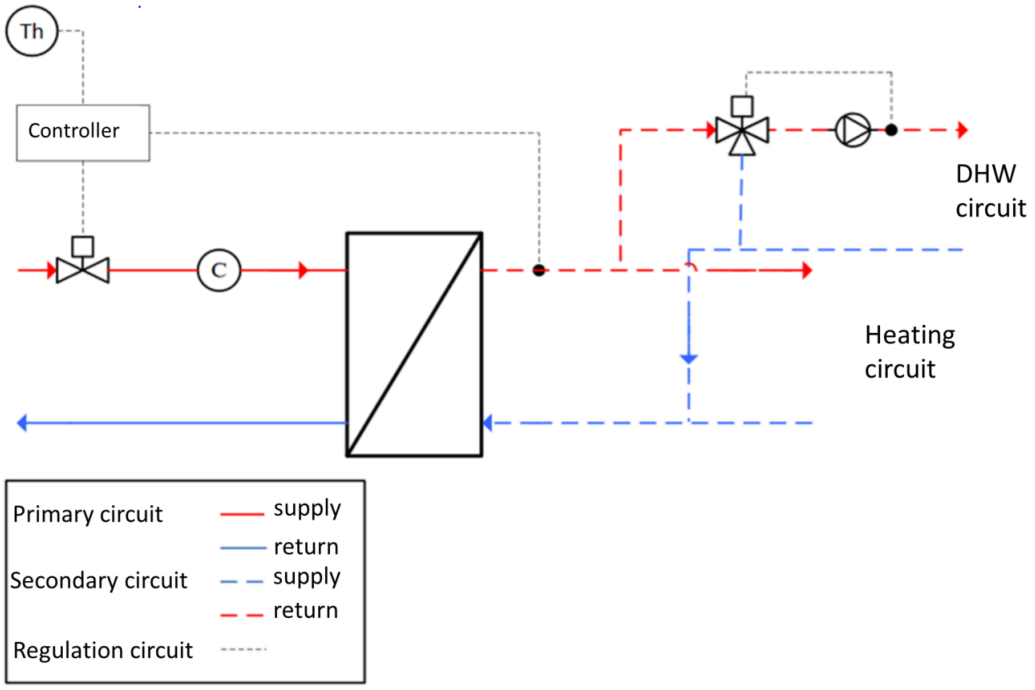


Figure 7.3: Illustrative diagram of a substation.

combined with the energy balance within each substation:

$$\dot{m}_{n.i} - \dot{m}_{ss} = \dot{m}_{n.o} \tag{7.8}$$

$$\dot{m}_{ss} = \frac{\dot{Q}_{ss}}{c_{p,w} * \Delta T_{ss}} \tag{7.9}$$

Where $m_{n.i}$ is the mass flow rate at the node inlet, $m_{o.i}$ the mass flow rate at the node outlet, $\dot{m}_{ss}$ the mass flow rate at the substation, $\dot{Q}_{ss}$ is the heat transfer rate of the substation, $c_{p,w}$ the water specific heat capacity and ΔT_{ss} the temperature difference at the substation.

An illustrative diagram of a substation is given in figure 7.3 and the model we developed for a substation in Dymola is illustrated in figure 7.4.

Our heating and cooling network Dymola model represents 17 heat substations and 15 cold substations that are all interconnected, forming a part of the heating and cooling network.

One of the key elements of the district heating and cooling system is the heat exchanger located inside each substation and that ensures the link between the primary and the secondary network by transferring the heat generated by the

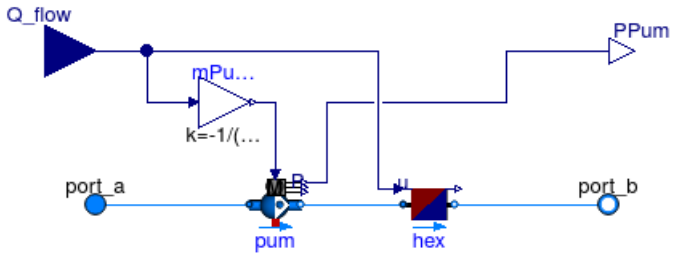


Figure 7.4: Dymola model of a substation (example of a heat skid).

power plant from the primary network to the secondary network serving the end users connected to the substation. A heat exchanger is modeled such that a heat quantity

$$Q_{flow} = u.Q_{flow\_nominal} \tag{7.10}$$

is added to the stream, where the input signal $u, |u| < 1$, and the nominal heat flow $Q_{flow\_nominal}$ can have either positive or negative values (positive for heating and negative for cooling). The mass and energy balance of the fluid flow are solved by the component *MixingVolume* of the Buildings library, illustrated as a blue sphere in figure 7.5. The kinetic and potential energies as well as the pressure drop are not considered, and the volume exchanges heat through a dedicated port.

The fluid dynamics at the heat exchanger are governed by two fundamental physical principles: mass and energy conservation. The mass balance equation at the heat exchanger states that the net mass flux across the surface S of the exchanger is equal to zero and can be written as follows:

$$\int_S \rho \overrightarrow{u}.\overrightarrow{n}\, dS = 0 \tag{7.11}$$

Where ρ is the fluid density, $\overrightarrow{u}$ its velocity vector and $\overrightarrow{n}$ the outward unit normal vector to the surface S.

The energy balance equation can be written as:

$$\int_S \rho c T \overrightarrow{u}.\overrightarrow{n}\, dS = \int_S \overrightarrow{q}.\overrightarrow{n}\, dS \tag{7.12}$$

Finally, the pressure drop at the heat exchanger is considered using the same formula as for pipes. The Dymola model of the heat exchanger is illustrated

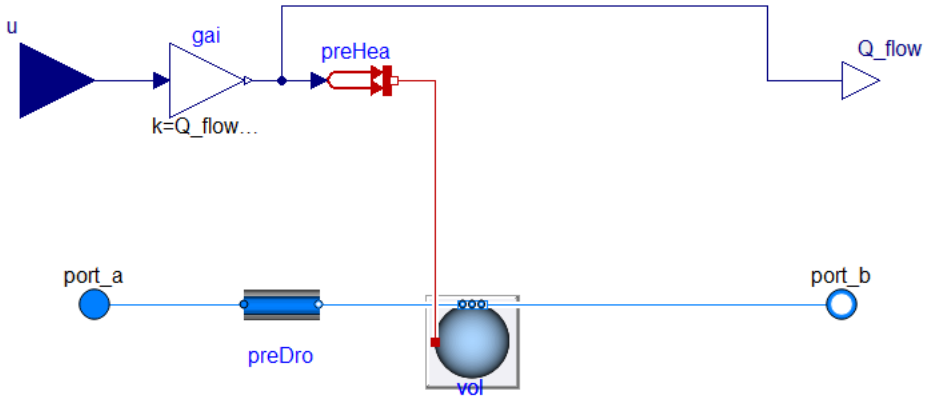


Figure 7.5: Dymola model of a heat exchanger.

in figure 7.5.

The model of the substations is completed by a model of a circulation pump from the Buildings library to ensure the flow of the heat transfer fluid. This model called *FlowControlled_m_flow* consists of a pump for which the mass flow rate is ideally controlled by an input signal. This means that the model prescribes a mass flow rate, which is typically provided by a Modelica block of type *constant*.

The models of the substations have then been integrated in the global model of the district heating and cooling network, illustrated in figure 7.6.

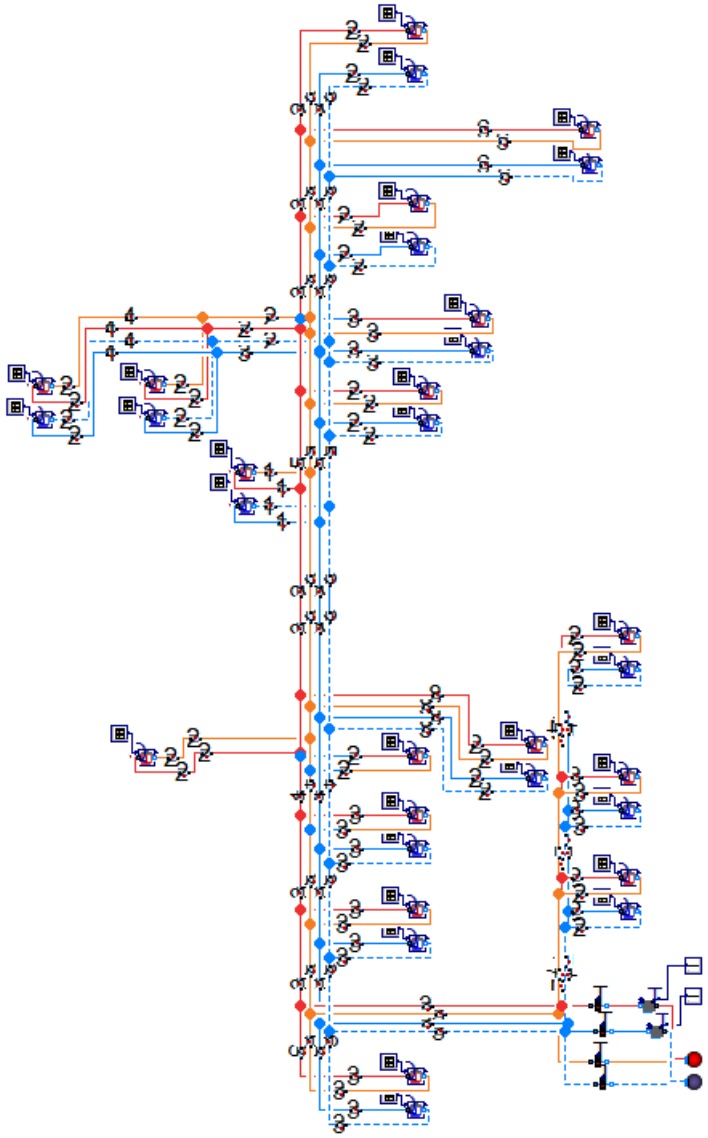


Figure 7.6: Dymola model developed for the MSE district heating and cooling network.

7.3.2 Model of the heating and cooling power plant

7.3.2.1 Thermo-refrigerating heat pumps

The four thermo-refrigerating heat pumps (TRHP) A, B, C and D were combined to form a model of what we refer to as TRHP system. The model of an individual TRHP was approached using the *Carnot_Tcon* heat pump model available in Dymola. The technical characteristics of each of these four TRHPs were specified based on the corresponding data provided by the manufacturer. We then added an estimation of the equivalent electrical power consumed by the TRHP based on an estimation of the Coefficient of Performance

(CoP). In fact, the CoPs of the TRHP vary according to the system configuration and load. However, the CoP values were not all available, that is why we used nominal CoPs of the TRHPs provided by the manufacturer and adopted a quasi-linear variation based on known equivalent CoPs. The documentation sheet for the model of individuals TRHP is provided in appendix B.

The global TRHP system includes the two tandems of TRHPs A/B and C/D as well as the regulation needed to operate their various configurations. Overall, the real TRHP system involves 14 configurations, detailed in Appendix B, but for the sake of simplicity, only 7 configurations were considered in the model. Indeed, our simplification consists in assembling two or three configurations defined as distinct by the manufacturer into one unique configuration whenever they are identical from the TRHP system's point of view. The 7 configurations we obtained are the following (note that we talk about configurations for cold when the TRHP system's production is adjusted according to the cooling loads). If there is a surplus of heat production, it is either injected into the geothermal wells or sent to the adiabatic aero-refrigerants according to the system configuration. These kind of scenarios basically happen during summer periods. Conversely, we talk about configurations for heat when the TRHP system's production is adjusted according to the heating loads. If there is a surplus of cold production, cold water is injected back into the geothermal wells.):

* Configuration A for cold: this configuration incorporates three different scenarios for cold defined by the manufacturer and referred to as cold scenarios A01, A02 and A03. We assemble these three scenarios in one configuration since they are all identical from the TRHP system's point of view.
* Configuration B for cold: this configuration incorporates three other scenarios for cold referred to as B01, B02 and B03 that we assemble into one unique configuration since they are all identical from the TRHP system's point of view.
* Configuration C for cold: this configuration incorporates three other scenarios for cold referred to as C01, C02 and C03 that we group under one

configuration since they are all identical from the TRHP system's point of view.

* Configuration D01 for cold: this configuration represents the configuration for cold referred to by the manufacturer as D01 for the TRHP system.
* Configuration D02-03 for cold: it merges the two configurations denoted D02 and D03 for cold since they are identical from the TRHP system's point of view.
* Configuration A01 for heat: this configuration corresponds to the heat scenario A01 defined by the manufacturer.
* Configuration A02 for heat: this configuration corresponds to the heat scenario A02 defined by the manufacturer.

For the flow rates regulation, we calculated the coefficient that links the flow rate on the condenser side of a TRHP to the flow rate on the evaporator side of the TRHP based on the ratio of the heat flux provided by the TRHP at the side of the condenser by the heat flux provided at the side of the evaporator. This coefficient is constantly calculated to achieve the desired temperature difference on both sides of the TRHP. Besides, we imposed maximum flow rates to manage the TRHPs' saturation. These flow rates are calculated based on data provided by the manufacturer and are as follows:

* Tandem 1 (TRHPA and TRHPB) on the condenser side ($35°C/65°C$):
  - $24.21\,kg/s$ for the cold configurations A, B, D01, D02-03, and the heat configuration A01,
  - $46.46\,kg/s$ for the cold configuration C and the heat configuration A02.
* Tandem 1 (TRHPA and TRHPB) on the evaporator side ($18°C/8°C$):
  - $60.18\,kg/s$ for all the configurations,
* Tandem 2 (TRHPC and TRHPD) on the condenser side ($35°C/65°C$):
  - $69.68\,kg/s$ for all the configurations,
* Tandem 2 (TRHPC and TRHPD) on the condenser side ($17°C/27°C$):
  - $69.68\,kg/s$ for all the configurations,
* Tandem 2 (TRHPC and TRHPD) on the evaporator side ($18°C/8°C$):

· $52.83\,kg/s$ for all the configurations.

Furthermore, some other adaptations on the model of the TRHP system were made in order to manage the different configurations as well as the transitions between them in a single TRHP system model. The electrical power consumption of the pumps of the TRHPs is also calculated and output by the model. A documentation sheet for the individual TRHP model as well as for the global TRHP system is presented in sections B.4 and B.5 of appendix B.

7.3.2.2 Chiller

The chiller was modeled following the same approach as for the unitary TRHP model, including flow regulation. Similarly to TRHP models, the electrical power consumed by the pumps of the chiller was calculated and maximum flow rates based on the manufacturer's data were imposed to manage the saturation of the chiller. Their values are as follows:

* $52.58\,kg/s$ on the condenser side ($17°C/27°C$)
* $47.78\,kg/s$ on the evaporator side ($18°C/8°C$)

Overall, the whole system model can be considered as an equivalent unitary TRHP.

7.3.2.3 Positive-negative chiller

The positive negative chiller is basically composed of two TRHPs forming a positive-negative chiller tandem. Due to the lack of available manufacturer data concerning this system, it was modeled based on the model of a single TRHP whose technical characteristics were matched with those of the chiller. The flow regulation was also matched with that of the unitary TRHP and the chiller and the electrical power consumed by the pumps were calculated.
The positive-negative chiller has some additional specific features on the evaporator side. For instance, the cold storage system charging exclusively relies on the cold produced by the positive-negative chiller. This chiller system can also be used to assist the TRHP system and the chiller in producing cold in order to meet the cooling loads. Such synergies normally involve intermediate networks of $6°C/16°C$ and $-6°C/-3°C$ along with heat exchangers.

However, to simplify the modeling process, we chose to work with the basic $8°C/18°C$ network and account for these interactions by means of power equivalences. Overall, the whole positive-negative chiller system model can be assimilated to that of the chiller, with additional synergies taken into consideration.

7.3.2.4 Adiabatic aerorefrigerant towers

The real system incorporates three adiabatic aero-refrigerant towers (referred to as DRY) to handle excess heat when the geothermal maximal capacities are reached. In our model we consider one single equivalent DRY system. Even though this model does not encompass the whole operational complexity of such system, it is sufficient for our modeling purposes since it accounts for the electric power consumption of the DRY system's pumps and motorized fans in a simplified way. A documentation sheet of the developed DRY Dymola model is provided in section B.6 of appendix B.

7.3.2.5 Geothermal systems

The geothermal systems provide the heat and cold production system with the possibility to dissipate the excess heat or cold production that results from certain configurations. Once the geothermal capacity is reached, the TRHP system is designed to transition to configurations of type D that allow usage of adiabatic aero-refrigerant towers to evacuate excess heat. However, since we have no data available concerning the thresholds that correspond to the concept of saturation of the geothermal reservoir, an approximate reasoning was adopted to account for this saturation concept in the management of the different TRHP system configurations into the complete model of the power plant presented in the next section. The electrical consumption of the injection and extraction pumps was also taken into consideration in this model.

7.3.2.6 Global model of the heat and cold production system

The complete heat and cold production system model includes all the elements presented above. These elements were assembled following the P&ID presented in appendix as well as the different TRHP configurations defined

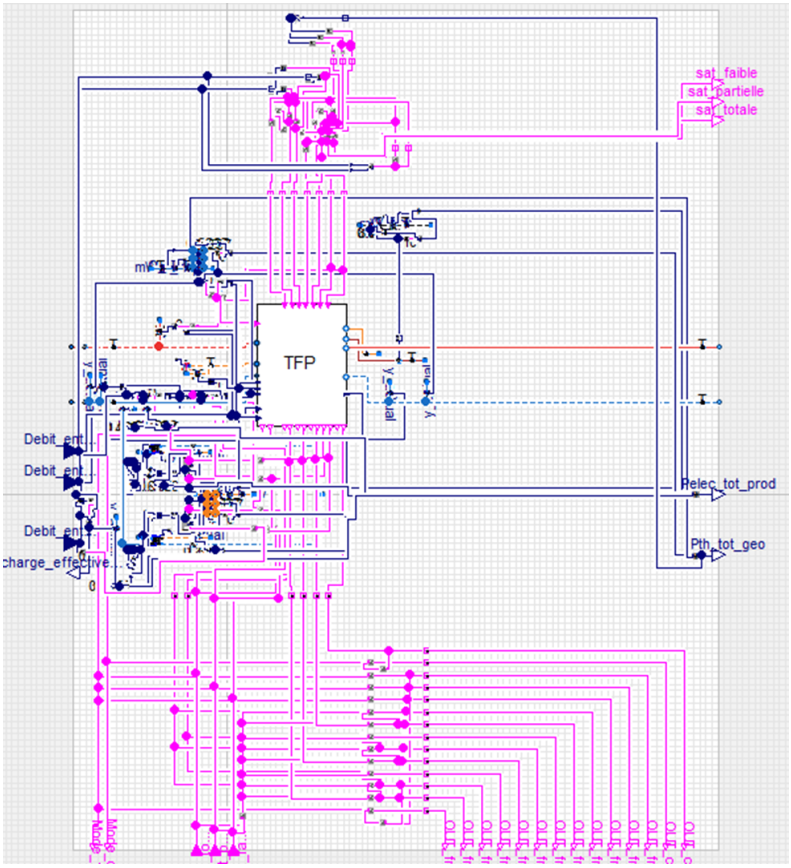


Figure 7.7: An overview of the global model of the heat and cold production system with regulation.

in the functional analysis. Besides the simplifications made within each of the previous models, some additional simplifications and adaptations had to be made during their integration within the global power plant system model. For instance, several auxiliary components were not taken into account in the global model.

Figures 7.7 and 7.8 present an overview of the global model of the heat and cold production system respectively with and without regulation, and figure 7.9 illustrates the model that controls the different configurations. This comprehensive heat and cold production system model was integrated into the district heating and cooling network model and its performance was tested over a full year.

7.3.3 Model of the heat storage system

For the modeling of the Phase-Change Material (PCM) heat storage system, it was decided to start from an existing model of the Buildings library that re-

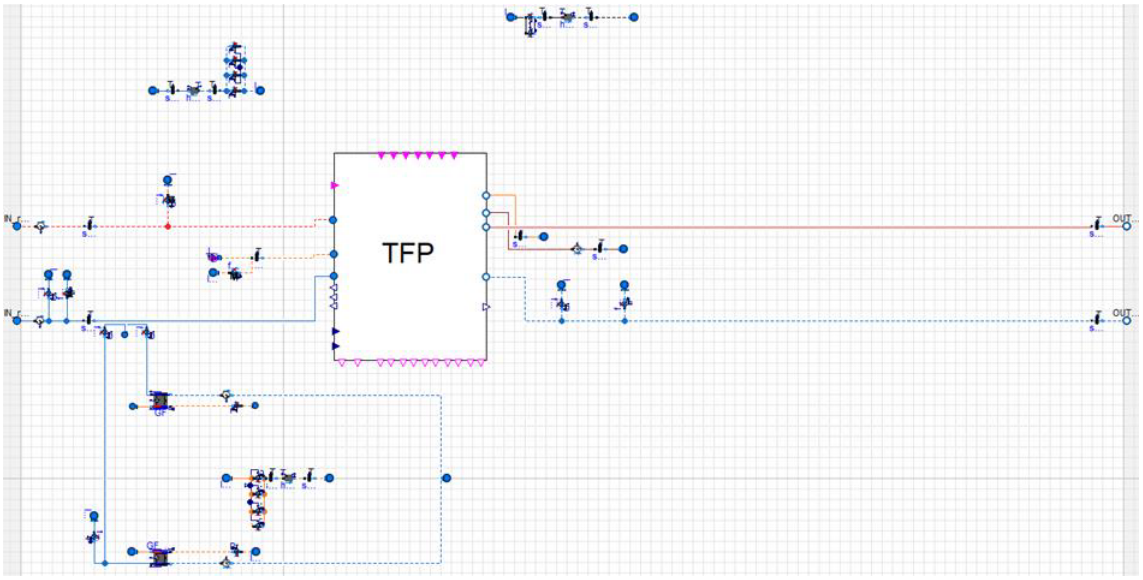


Figure 7.8: An overview of the global model of the heat and cold production system without regulation.

Figure 7.9: An overview of the model that manages the different configurations of the heat and cold production system.

Parameters

Type	Name	Default	Description
Length	x		Material thickness [m]
ThermalConductivity	k		Thermal conductivity [W/(m.K)]
SpecificHeatCapacity	c		Specific heat capacity [J/(kg.K)]
Density	d		Mass density [kg/m3]
Real	R	x/k	Thermal resistance of a unit area of material [m2.K/W]
Integer	nStaRef	3	Number of state variables in a reference material of 0.2 m concrete
Boolean	steadyState	(c < Modelica.Constants.eps ...	Flag, if true, then material is computed using steady-state heat conduction
Properties for phase change material			
Temperature	TSol		Solidus temperature, used only for PCM. [K]
Temperature	TLiq		Liquidus temperature, used only for PCM [K]
SpecificInternalEnergy	LHea		Latent heat of phase change [J/kg]
Advanced			
Integer	nSta	max(1, integer(ceil(nStaReal...	Actual number of state variables in material
Real	piRef	331.4	Ratio x/sqrt(alpha) for reference material of 0.2 m concrete
Real	piMat	if steadyState then piRef el...	Ratio x/sqrt(alpha)
Real	nStaReal	nStaRef*piMat/piRef	Number of states as a real number

Figure 7.10: Parameters to be specified for the Dymola model of the PCM heat storage.

quires two classes: an icon and a model. The first is used to gather data where the PCM parameters are specified and the latter takes the icon's name as input and uses data from the icon to perform calculations and enable the computation of the storage's outlet temperature and power. The table of figure 7.10 summarizes the parameters that have to be specified for the icon. The values of these parameters actually used in our model are those of the octadecanol as a PCM material. They were selected based on the work of Martinelli [344] and are detailed in table 7.1. Figure 7.11 illustrates an example of discharge scenario where the model denoted *lay* uses the data from *matPCM* (phase-change material) to calculate the outlet temperature and power of the storage system. During the discharge phase, water enters the storage at $35°C$ and is heated by the storage system having its temperature approximately equal to that of the PCM. Details on this PCM heat storage Dymola model are given in the documentation sheet in section B.8 of appendices B. This model has been validated using simulation data provided by the CEA who is in charge of the design and construction of the PCM heat storage system in the MSE project.

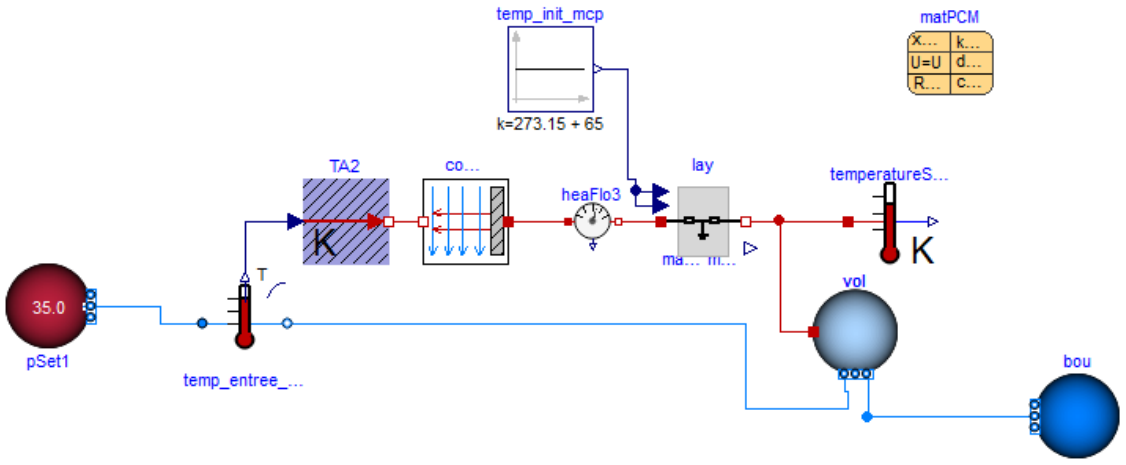


Figure 7.11: Illustration of a discharge scenario of the PCM heat storage model under Dymola.

Table 7.1: Properties of the octadecanol used in the model of the PCM heat storage.

Properties	Values
thermal conductivity	$0.2W/m.K$
Specific heat capacity	$2611J/kg.K$
Mass density	$0.77g/cm^3$
Melting temperature	$58°C$
Latent heat of phase change	$246900J/kg$

7.3.4 Model of the cold storage system

Even though the ice on coil cold storage system is already installed within the heat and cold power plant of the MSE eco-district, this system is not yet operational. Besides, unlike the PCM heat storage system for which we were provided with design and simulation data from the CEA, it was not possible for us to obtain charge and discharge profiles of the ice storage system from the manufacturer. We were provided with some typical charge and discharge curves for the cold storage. However, for the charge curve given in figure 7.12 for instance, the inlet water temperature ranges between $-3°C$ and $-6°C$ which is not the case for the MSE cold storage system for which the inlet water temperature will remain constant at $-6°C$. Besides, no further information could be obtained regarding the flow rates considered in these charge curves.

Under these conditions, it was difficult to validate a model with specific charge and discharge scenarios. One can also abstract from the physical

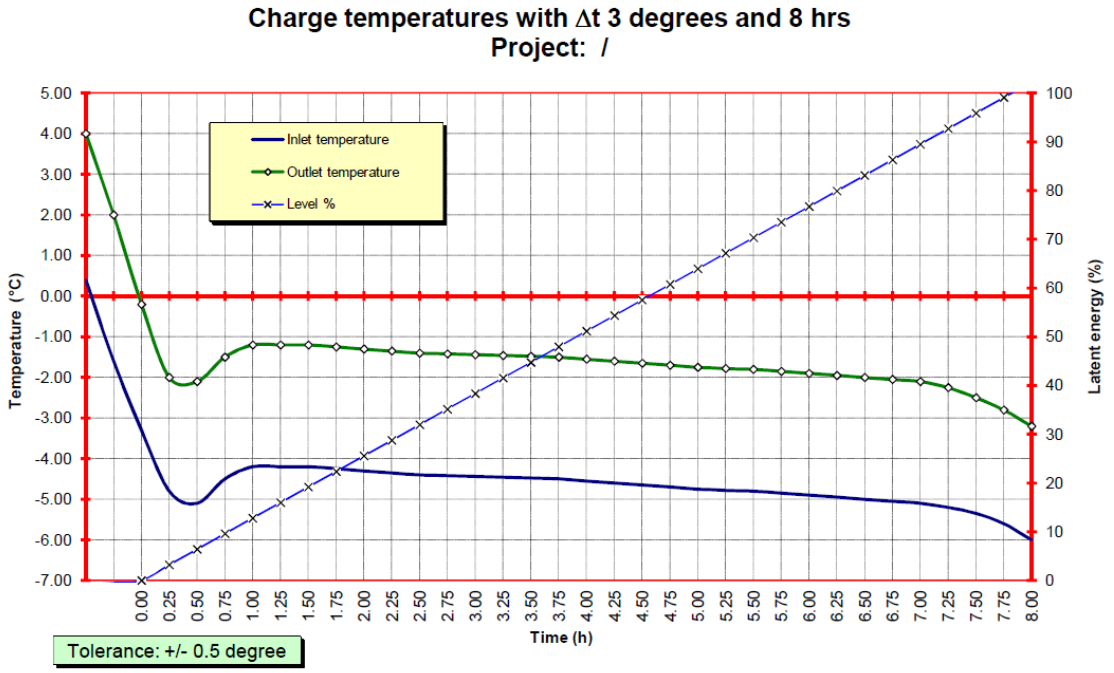


Figure 7.12: Typical charge curve of the cold storage system provided by the manufacturer.

details of the model since our primary focus is to study the water temperature at the storage system inlet and outlet. Therefore, we can consider a black box model that simulates the dynamics of the actual system without explicitly considering its physical intricacies. A simplified Dymola model could consist of a *Heat Capacitor* from the base Modelica library *Modelica.Thermal.HeatTransfer.Components.HeatCapacitor*. Its heat capacity would be calculated based on the temperature of the glycoled water and it would be conneced to the fluid representing the glycoled water through heat transfer.

Overall, modeling the heat and cold storage systems of this project can be complex for various reasons. For the cold storage system, the lack of simulation data together with the absence of operational data pose challenges since the storage system is not yet functional. As for the heat storage system, the complexity comes basically from it being an innovative demonstrator. It is indeed the first of its kind and size installed in a real project. That is why it is essential to have data from the initial commissioning tests of both thermal storage systems to calibrate the simulation models or use data-driven approaches to identify the systems dynamics.

At the time of writing this dissertation, experiments on the PCM heat storage

system have just began, while tests on the cold storage system have not yet started. Therefore, we implemented, as a temporary solution, simplified Dymola models of these two storage systems to incorporate their dynamics into the overall simulation model of the eco-district. These models are presented in section 7.3.6. Once operational data from the real systems become available, these simplified models will be replaced with more accurate data-driven models.

7.3.5 Model of the electrical systems

The electrical systems of the eco-district were aggregated and integrated in a simplified way into the global simulation model. Their model encompasses all power prodcution and consumption systems as well as electrical components from all the previously presented systems. This entire setup represents a model of electrical grid that was included in the comprehensive simulation model in parallel with the heating and cooling network. All the produced and consumed powers are summed at each time step to determine the amount of power to be withdrawn or injected to The public utility grid.
the power consumption considered in this model includes:

* Consumption of the distribution pumps of the heating and cooling network,
* Consumption of the pumps of the heat and cold storage systems,
* Consumption of all the elements of the heat and cold production system (including electric consumption of the thermo-refrigerating heat pumps, chillers, pumps, adiabatic aero-refrigerant tower, ect.),
* Electric load due to the battery charging,
* Electric loads of the buildings of the eco-district, aggregated at the substation level.

When it comes to the electrical power generation systems considered, they include:

* Power generation provided by the battery when discharging
* Power generation from the PV panels.

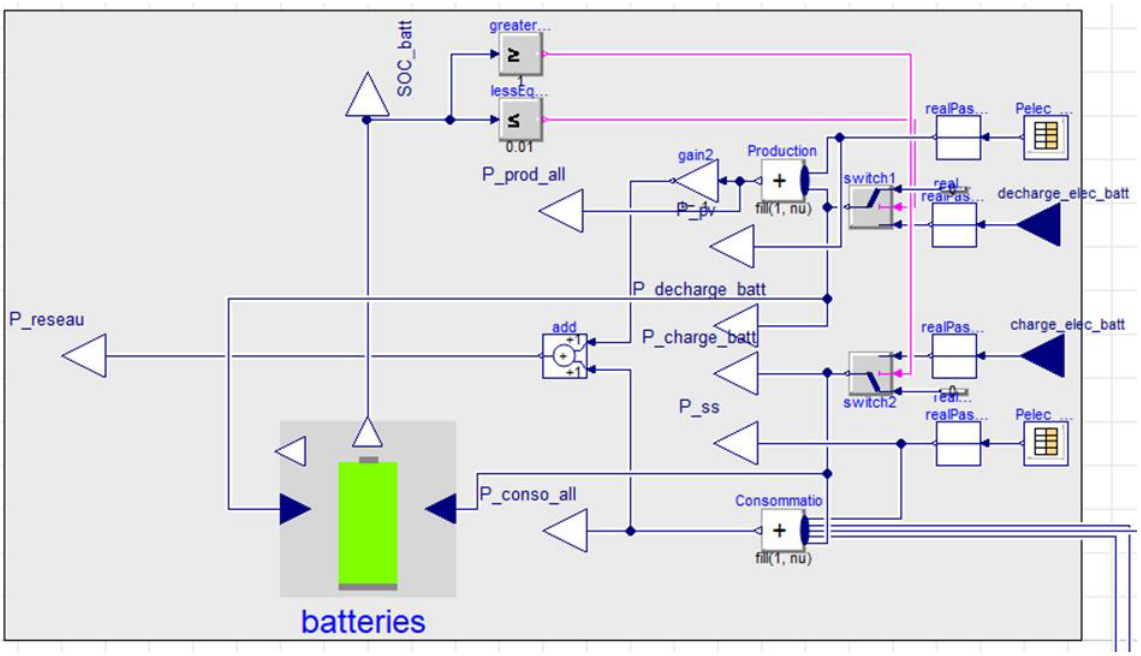


Figure 7.13: Dymola model of the electric systems of the MSE eco-district.

The latter power generation item was modeled based on the construction of time series of PV power generation based on historical data available for the MSE project's location on PVGIS [361]. Some details regarding this PV system model are also provided in section B.9 of appendix B.

Finally, a Dymola model of the electric storage system is developed based on a battery model available in the *BuildingSystems* library. The storage system is considered as a single equivalent lithium-ion battery block. The characteristics of this block were defined based on available data provided by the manufacturer. This model enables outputting the state of charge of a battery energy storage system with configurable characteristics for a given charge or discharge input power. Additional details regarding this model of the electricity storage system are provided in section B.10 of appendix B.

7.3.6 Model simplification and final sub-models

7.3.6.1 Simplified model of the heat and cold storage systems

Simplified models of the thermal storage systems were integrated between the heat and cold production system and the distribution system. Even though this implementation does not exactly reflect the real installation of these two stor-

ages, it can be considered as equivalent given the simplifying assumptions. The simplified model of the heat storage system involves adding or withdrawing a given flow rate at the inlet of the heat and cold production system. This flow rate then directly undergoes a positive or negative heat exchange, depending on whether the storage system is being charged or discharge, and is finally re-injected. This model also includes the calculation of the electric power consumed by the storage pumps as well as the state of charge (SoC) of the storage system, taking into account daily losses and various regulation elements. More details on the heat storage model are presented in appendix B.

For practical reasons, the cold storage model is grouped in the same Dymola model as the heat storage system. It is modeled similarly to the heat storage system, only considering an inverse charge and discharge operation in terms of heat flow and allowing charge solely using the Positive-Negative chiller. Similarly to the heat storage, further details on the model of the cold storage systems are presented in appendix B section B.11. An overview of the integration of this simplified heat and cold storage model into the overall power plant is presented in figure 7.14.

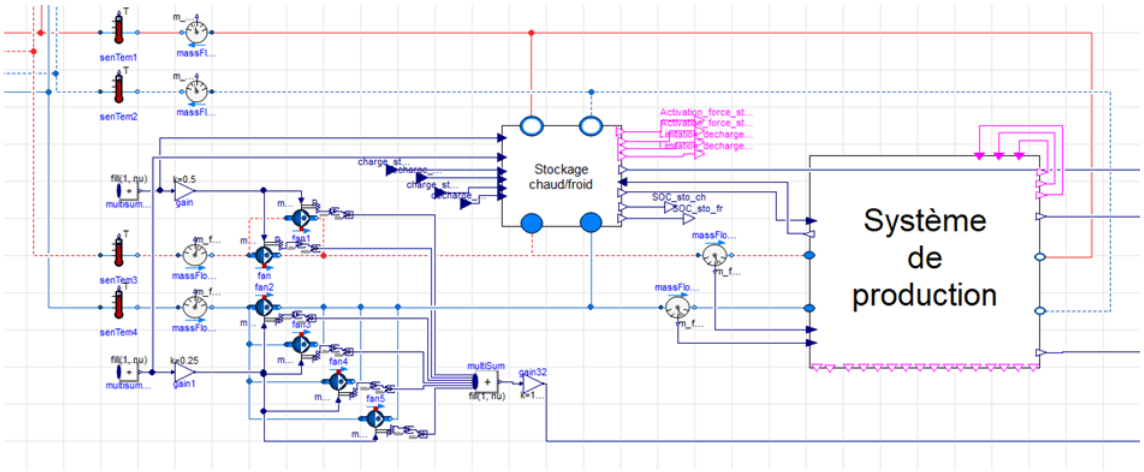


Figure 7.14: Dymola model of the heat and cold storage systems integrated within the power plant.

7.3.6.2 Global aggregate model

The compete aggregate model of the MSE eco-district's multi-energy systems incorporates in a coherent way the four major blocks presented above, namely:

* District heating and cooling network,
* District heating and cooling production system composed of the TRHP system, chiller, positive-negative chiller, adiabatic aero-refrigerants and geothermal system,
* Heat and cold storage systems,
* Electrical systems including PV generation, electrical load and battery energy storage system.

This compartmentalized approach allowed for testing each of these components individually before connecting them and validating the operation of the overall system over a full year in the base scenario. Moreover, this approach also meets standardization requirements since most of the models developed were accompanied by documentation sheets that describe the modeling process and hypothesis.

This simulation model, illustrated in figure 7.15 can thus be considered as a preliminary version of the MSE digital twin since it still contains certain flaws and some missing elements. Actually, in order to advance this digital twin and bring it closer to the actual system, additional design as well as operational data are required. Some of these needed data include:

* Technical specifications of the positive-negative chiller,
* Operational characteristics of the concept of geothermal saturation,
* Technical specifications of the geothermal pumps,
* CoPs of each of the TRHPs of the heat and cold power plant for each system configuration and load rate,
* Functional analysis, technical characteristics and experimental data of the cold storage system,
* Complementary functional analysis, technical characteristics and experimental data of the heat storage system as well as its associated pumps.

Thanks to the commissioning tests and experiments being currently carried on mainly for the heat storage system, the TRHP system and the battery energy storage system, most of these required data will soon be available and will provide valuable insights and allow us to advance this simulation model and

enhance its accuracy.

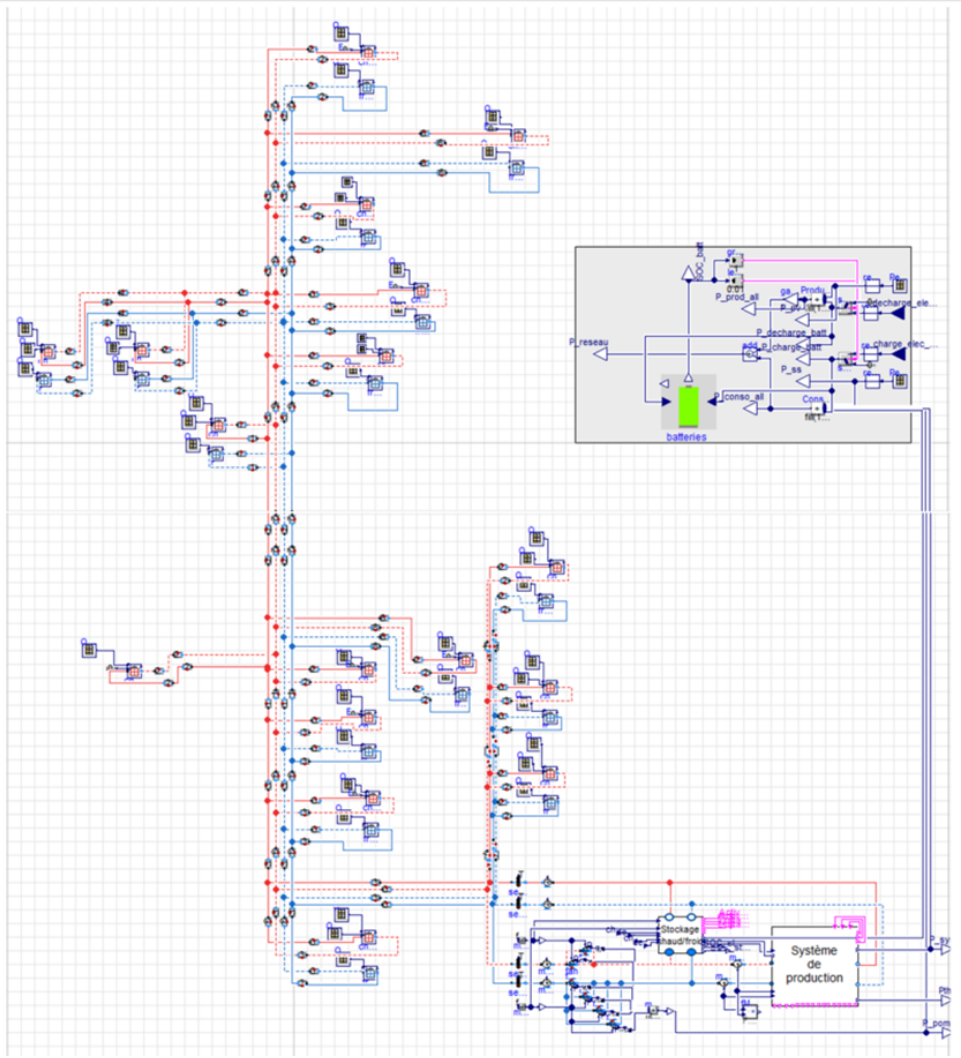


Figure 7.15: Overall aggregate Dymola model of the multi-energy system of the MSE eco-district.

7.4 Disaggregate simulation model

7.4.1 Interoperability, portability and co-simulation

The previously described aggregated model is considered as the initial version of the simulation model that we built for the MSE case study. It can thus represent a simulation environment for the smart multi-energy management framework developed in Python. To this end, it needs to be able to interface with this framework and an efficient way to achieve this is by exporting the model as a Functional Mock-up Unit (FMU). FMUs are self-contained en-

tities that encapsulate a simulation model and foster interoperability across a spectrum of platforms and tools such as Dymola, Matlab-Simulink, Catia, and Adams. This capability is enabled by a standardized interface known as the Functional Mock-up Interface (FMI), which guarantees the seamless integration and utilization of FMUs across various software environments.

The FMI standard offers two methods for exchanging FMUs between various tools and platforms, namely Model-Exchange (ME) and Co-simulation (CS), as depicted in Figure 7.16. Both model-exchange and co-simulation can be employed for exporting an FMU from one tool and importing it into another. Subsequently, the imported FMU can be simulated to determine the evolution of the system state over time. However, the key difference between model-exchange and co-simulation lies in how the importing tool progresses the FMU forward in time during the simulation:

* Co-Simulation (CS): also referred to as cooperative simulation or coupled system simulation [125], [362], co-simulation involves providing the numerical solver by the exporting tool and embedding it within the FMU. In this approach, the importing tool configures the inputs, instructs the FMU to progress in time, and then retrieves the outputs.
* Model-Exchange (ME): in model-exchange, the numerical solver is supplied by the importing tool. The FMU offers functions to establish the states and inputs, as well as compute the state derivatives. The solver within the importing tool determines the time steps and calculates the state at the subsequent time step. Generally, if the model to export involves a simulation time on the order of minutes or more, model-exchange may pose challenges, and it is advisable to consider co-simulation instead [363].

In this work, we opted for the use of co-simulation with the library FMPy [365]. Note that other libraries such as PyFMI [366] can also be used for loading FMUs in Python frameworks and interacting with them for both co-simulation and model-exchange.

Various tests carried on the global model FMU in co-simulation revealed some limitations and simulation issues related mainly to the default solver (Cvode)

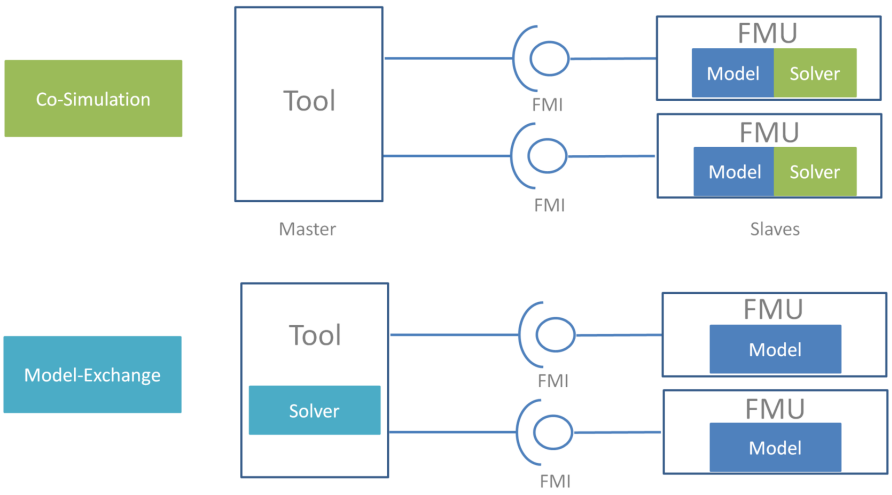


Figure 7.16: Usage of the FMI standard for Co-Simulation (CS) and Model-Exchange (ME) (adapted from [364]).

and to the lack of flexibility due to the aggregated approach. This led us to introduce a second version of the simulation model based on a disaggregate approach. In fact, this second version follows the same methodology as the aggregate simulation model previously presented and uses the same sub-system models. However, these models were dissociated into elementary blocks and adapted to allow them to work independently and to be individually exported as separate FMUs. The resulting disaggregate version of the simulation model is presented in the next section.

7.4.2 Components of the disaggregate model

The elementary blocks that form the new disaggregate version of the MSE simulation model are as follows:

* Heat substations: a model that aggregates all the heat substations of the eco-district,
* Cold substations: a model that aggregates all the cold substations of the eco-district,
* Heat distribution system: a model of the distribution system of the district heating and cooling network that carries the heat flow between substations and the heat and cold power plant,
* Cold distribution system: a model of the distribution system of the dis-

trict heating and cooling network that carries the cold flow between substations and the heat and cold power plant,

* Heat and cold production system: a model that involves all the energy systems of the heat and cold production plant (TRHPs, chiller, negative-positive chiller,etc.), except for the storage systems,
* A model for each of the storage systems: the heat storage system, the cold storage system and the battery energy storage system,
* Electric grid: a model that incorporates all the electricity demand end generation devices as well as the connection to the main utility grid.

These elementary blocks as well as a graphical representation of the connections between them are illustrated in figure 7.17. Each of these individual models was exported as an FMU and dedicated inputs and outputs were defined to establish connections between them. These coupling variables, i.e. the input and output variables that the individual FMUs exchange during a co-simulation, are summarized in table 7.2.

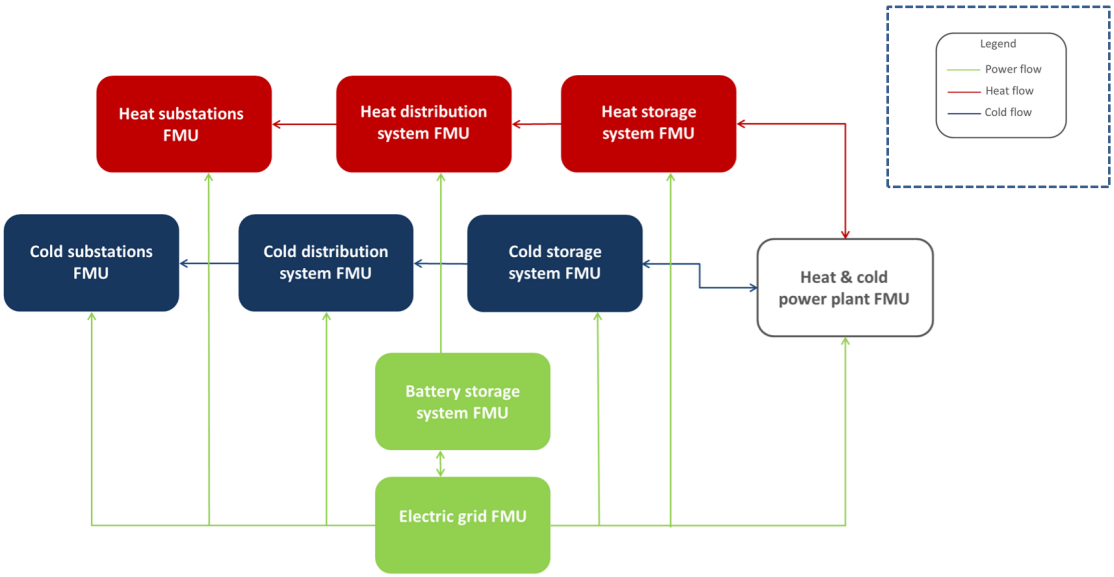


Figure 7.17: Components of the disaggregate model.

Overall, these individual elementary FMUs interact with each other during a co-simulation time-step by exchanging coupling variables. They collectively form the disaggregate complete model - a second version that is more robust and flexible- of the MSE simulation model that we integrated within the

Table 7.2: Coupling variables between the FMUs of the disaggregate model.

Coupling variable name	Description	Source FMU (output)	Target FMU (input)
$debit\_chaud\_ss$	Global flow rate of heat sub-stations (aggregated)	Heat substations FMU	Heat distribution system FMU
$debit\_froid\_ss$	Global flow rate of cold sub-stations (aggregated)	Cold substations FMU	Cold distribution system FMU
$P\_distrib\_ch$	Electric consumption of the heat distribution system	Heat distribution system FMU	Electric grid FMU
$P\_distrib\_fr$	Electric consumption of the cold distribution system	Cold distribution system FMU	Electric grid FMU
$debit\_chaud\_distrib$	Heat flow rate distributed by the DHCN	Heat distribution system FMU	Heat storage system FMU
$debit\_froid\_distrib$	Cold flow rate distributed by the DHCN	Cold distribution system FMU	Cold storage system FMU
$P\_elec\_stock\_ch$	Electric consumption of the heat storage	Heat storage system FMU	Electric grid FMU
$debit\_ch$	Heat flow rate	Heat storage system FMU	Heat and cold production system FMU
$P\_elec\_stoch\_fr$	electric consumption of the cold storage	Cold storage system FMU	Electric grid FMU
$debit\_fr$	Cold flow rate	Cold storage system FMU	Heat and cold production system FMU
$debit\_GF\_NP$	Cold flow rate to be provided by the negative-positive chiller	Cold storage system FMU	Heat and cold production system FMU
P_elec_sp	electric consumption of the heat and cold production system	Heat and cold production system FMU	Electric grid FMU
P_charge_batt	Battery charge power	Battery storage system FMU	Electric grid FMU
$P\_decharge\_batt$	Battery discharge power	Battery storage system FMU	Electric grid FMU

Python framework and tested for co-simulation. This disaggregate version of the simulation model paves the way for coming advanced versions. Indeed, as soon as sufficient operational data for a given energy system of the MSE eco-district become available, we can use, for instance, data-driven approaches to identify more precisely the dynamics and behavior of this system. This way, we can create a new and improved FMU of this component. If this new component is closer to the real-system behaviour, the compartmentalized approach we adopted in the disaggregate model will allow us to easily replace the old component FMU by the new one. As we gather more data from different components of the energy systems, we can thus update their corresponding FMUs accordingly. This approach not only facilitates improving the accuracy of the digital twin but also ensures that it remains up-to-date and reflective of the current operational conditions in the eco-district.

7.5 Conclusion

This chapter presented the simulation tool that we developed for the multi-energy systems of the Meridia Smart Energy eco-district. This digital twin built using the Modelica language under the Dymola Software serves mainly as a test-bed for the DRL-based smart multi-energy management systems developed in this work. Indeed, this digital twin is converted to an FMU and integrated into the Python-based framework to play the role of the environment in the DRL-based algorithms by means of co-simulation. Using the developed DRL-based framework for the optimal energy management of the MSE digital twin constitutes the second case study of our research work and its simulation results are presented in the next chapter.

Chapter 8

The DRL approach applied on the Meridia Smart Energy case study

Résumé

Ce chapitre vise l'application de l'approche basée sur l'apprentissage par renforcement profond (DRL) proposée sur le jumeau numérique que nous avons développé pour le système multi-énergie intelligent MSE, tel que détaillé dans les chapitres précédents. Ce jumeau numérique est encapsulé sous la forme d'une Unité de Modélisation Fonctionnelle (FMU) et intégré sous la forme d'environnement Open AI Gym. Cette transformation permet l'intégration du jumeau numérique en tant qu'environnement au sein du cadre DRL développé sous Python. L'agent DRL interagit avec le FMU en prenant des actions de pilotage sur les trois systèmes de stockage d'énergie et en recevant des informations sur l'état et le signal de récompense, apprenant ainsi une stratégie optimale de gestion multi-énergies par essais et erreurs. Dans ce chapitre, on présente d'abord la méthodologie et l'architecture du cadre développé, puis on discute des résultats de simulations où on évalue l'efficacité de l'agent DRL dans le pilotage de systèmes multi-énergétiques intelligents complexes et dynamiques.

8.1 Introduction

This chapter focuses on the application of the DRL-based approach on the digital twin that we built for the MSE smart multi-energy system, as detailed in the preceding chapters. The digital twin is encapsulated as a Functional Mock-up Unit (FMU) and wrapped as an Open AI Gym environment. This transformative process allows the integration of the digital twin as an environment within the Python-based DRL framework. The DRL agent interacts with the FMU by taking managing actions on the three energy storage systems and receiving state and reward feedback from it and hence learning an optimal energy management strategy through trial and error. Within this chapter, we first elucidate the methodology and architecture of the framework developed and then discuss the simulation results where we evaluate the effectiveness of the DRL agent in operating complex and dynamic smart multi-energy systems.

8.2 Methodology and framework setup

The case-study that we consider in this chapter, referred to as case-study 2, is the disaggregate digital twin of the MSE smart-energy system presented in the preceding chapter. As detailed previously, this model is composed of nine elementary blocks, namely: heat substations, heat distribution systems, a heat storage system, cold substations, cold distribution systems, a cold storage system, a heat and cold production system, a battery storage system and an electricity grid. Each of these individual models is exported as a co-simulation FMU, and coupling input and output variables allow interaction between them. On the other hand, the interaction of the DRL agent with these simulation models within the Python framework is orchestrated through the use of the FMPy library. The architecture of the framework created to ensure this interaction is illustrated in Figure 8.1 and the architecture of the tool-chain used is presented in Figure 8.2. Basically, at each step of the training or validation cycle, the DRL agent provides its selected actions to the FMUs. These actions are applied to the digital twin using the **fmu.setReal()** function on the corresponding FMU. All the FMUs are then orchestrated and simulated for

one time step using the function **fmu.doStep()** and their subsequent updated state can be accessed through the **fmu.getReal()** function. Hence, an observation of the state of the digital twin can be provided as feedback to the DRL agent and allows computing the reward signal and moving to the next training or validation step. This observation of the state contains information like charge or discharge power and state of charge of the heat storage, cold storage and battery storage systems, electric, heating , domestic hot water and cooling demands of the buildings, PV generation in the district, electric consumption of the heat and cold production system and network distribution system, heat and cold power produced by the power plant, electricity prices, outdoor temperature and date time information.

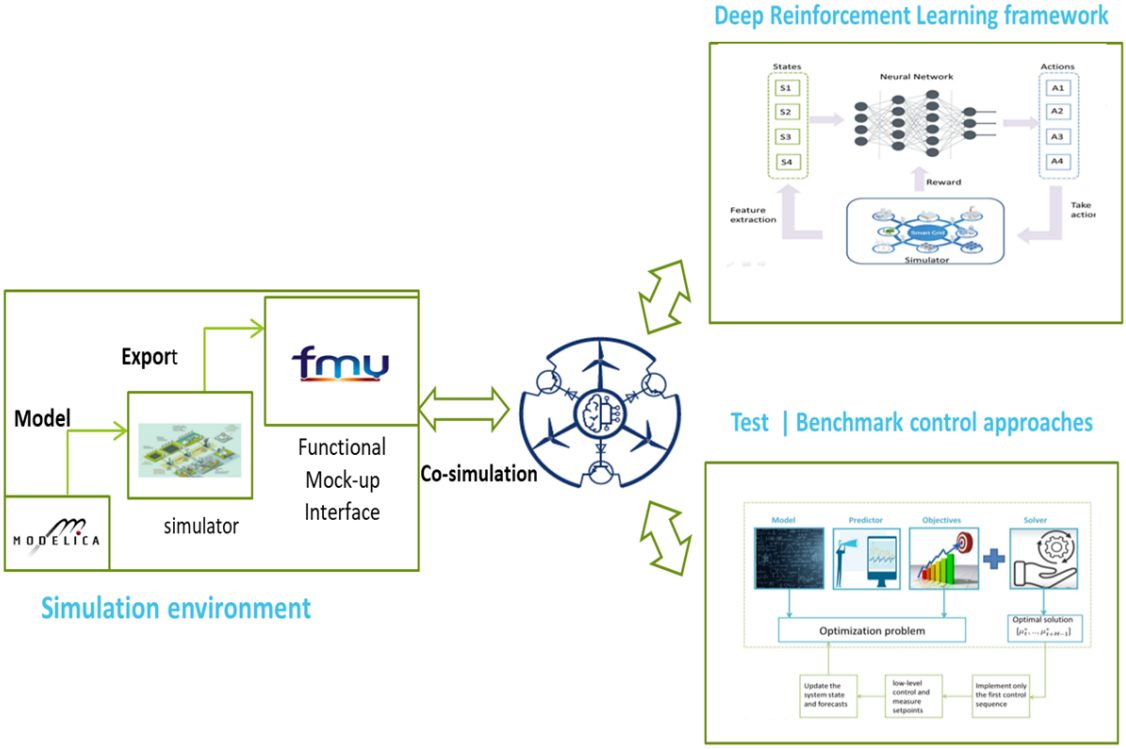


Figure 8.1: Architecture of the proposed framework.

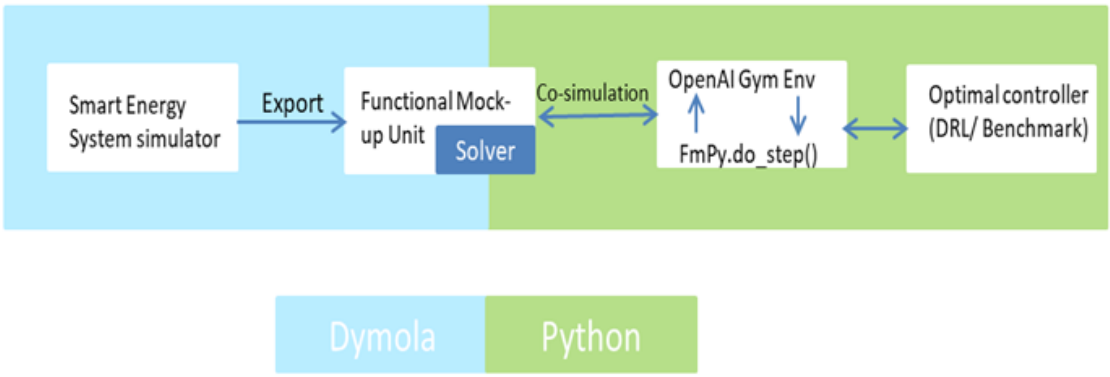


Figure 8.2: Architecture of the tool-chain used in the proposed framework (adapted from [37]).

8.3 The benchmark solutions

Unlike the case-study 1 where an MPC-based controller embedding a Linear Programming (LP) optimization problem proved effective due to the model's simplicity, the same approach could not be used as a benchmark in this case-study 2 due to significantly increased complexity and detail of the digital twin. Actually, trying to translate the problem into an LP (or a MILP) would inevitably lead to a significant loss of precision rendering the LMPC approach inadequate. As an alternative, a setup was attempted where MPC was used, but instead of an LP model, a Genetic Algorithm (GA) [367] was embedded to perform searches by simulating the digital twin and minimizing the output cost function resulting from the simulations. Unfortunately, the tests with this setup yielded inconclusive results. Despite multiple attempts, the GA approach failed to provide outcomes, primarily due to runtime errors. Efforts to mitigate this issue by extending the function timeout in the GA library [368] that we used proved ineffective. The reasons behind this behavior, whether attributed to the GA library used, the parameters we tested for the GA algorithm, or the suitability of the GA for this context, remained unexplored due to the excessive computational time required by the GA for this digital twin-based case-study. Further investigation is needed to determine the root cause and potential solutions for this computational challenge.

The alternative benchmark strategy that we used in this case study relies on a rule-based approach commonly employed in the industry for the manage-

ment of energy storage systems. This strategy aims at optimizing cost savings by taking advantage from fluctuating energy prices throughout the day. The main principle of this rule-based approach involves charging the energy storage systems during periods of low prices, typically observed at night and during off-peak hours, often around 2 and 3 PM, and discharging the storage systems during periods of higher prices, notably in the morning around 9 AM, mid-day, and in the evening around 7 PM, based on the price signal used in this case study. This rule-based strategy aligns with prevailing industry practices, leveraging market price fluctuations to minimize energy consumption costs.

8.4 Exogenous data used

The data used while training or testing the DRL agent are classified into two categories that account for the dual deterministic-stochastic nature of the smart energy systems [268]:

* Exogenous data (fixed inputs): they involve variables that stem from external factors to the dynamics of the smart energy system. This includes renewable energy generation, outdoor temperatures, electricity prices, as well as electric, heating and cooling loads of the buildings that are largely influenced by meteorological and consumer behavior factors. Hence, these data embody the stochastic aspect of the smart energy system. Since they are independant of the system's energy dynamics, they are supplied to the digital twin and to the DRL agent across predefined scenarios, and are divided into distinct sets for training, validation and testing cycles to prevent over-fitting.
* Dynamically adapted data: these data embody the deterministic aspect of the smart energy system. They include information such as the state of charge of the various storage systems, the output of the heat and cold power plant and the heat and cold flow in different locations of the district heating and cooling network. They are called "dynamically-adaped" data because their values dynamically adapt in response to actions taken. Thus, they can not be provided as pre-defined scenarios and are rather

computed based on the dynamic response of the digital twin to actions selected by the DRL agent.

In practise, exogenous data used for simulations in this work are derived from estimations using various tools. These estimations as well as the tools used for each type of exogenous data will be presented in this section. Nevertheless, as the development of the real-life MSE eco-district progresses, these data will be gradually replaced with real-time as well as historical data sourced from the MSE project's Datalake. This Datalake is supplied with data coming from the sensors deployed within the eco-district's infrastructure.

8.4.1 Heating and cooling demands

In order to generate realistic heating and cooling load profiles that are representative of specific criteria like the MSE building's types, characteristics, occupants and local weather, we propose the use of TEASER (Tool for Energy Analysis and Simulation for Efficient Retrofit) [369]. TEASER is an-open source tool developed by researchers from the RWTH Aachen University. It allows the generation of building archetypes with limited input requirements as well as the export of individual simulation models for various Modelica libraries. It provides hence an alternative to more complex building models generated by more complete Dynamic Thermal Simulation (DTS) software that require more inputs such as Compfie-Pleiade [370]. The methodology and package structure of the TEASER tool are provided by Remmen et al. [371]. In this work, we used the Aixlib [372] library within the Modelica language, together with the TEASER package to generate the heat and cold load profiles of the MSE eco-district's buildings. To generate Aixlib building models, we supplied the TEASER tool with buildings' information including orientation, number of floors, height of floors, window areas, wall, roof and intermediate floor areas and intended uses. Most of these information came from available documents of the MSE project including Revit models of the buildings, Autodesk plans, Dynamic Thermal Simulation reports, energy performance diagnostics, a well as Google Earth [373]. Finally, in order to validate the models developed, we checked that the obtained total heat and cold

consumption are in line with the reference values provided in the initial development program realised by Idex, specifically, a total annual heating demand of 9 MWh and an annual cooling demand of 10.8 MWh.

8.4.2 Domestic hot water demands

Domestic Hot Water (DHW) demands are estimated for each individual substation and injected as inputs to the digital twin of the district heating and cooling network together with the heating and cooling demands. The development plan of the MSE project provided by Idex outlines that substations of the network do not all have integrated DHW needs. Thus, we evaluate the hourly DHW requirements only for the valid substations. To do so, we propose the use of the *LoadProfileGenerator* tool developed by Pflugradt et al. [374]. This open source software enables the generation of household models based on the assignment of weekly activities to the occupants with specific ages, gender and profiles. Each activity is associated with an energy vector among heating, cooling and electricity. We selected specific profiles on the *LoadProfileGenerator* tool based on building types including students residence, social housing, families with or without children, couples working mostly from home or not, etc. The output DHW requirements of households were then aggregated to obtain the total DHW for each building, with hourly time steps. The aggregation of these time series consistently revealed two main demand peaks, one occurring in the morning and the other in the evening at around 7 PM. We also checked that the annual DHW requirements per person per household align with the french average of 80 liters per person, with a variation of around ± 35 liters, as outlined by the ADEME in [375]. A visualisation of the obtained heat and cold flow demands evolution throughout the year, including heating, cooling and DHW demands, is presented in Figures 8.3 and 8.4.

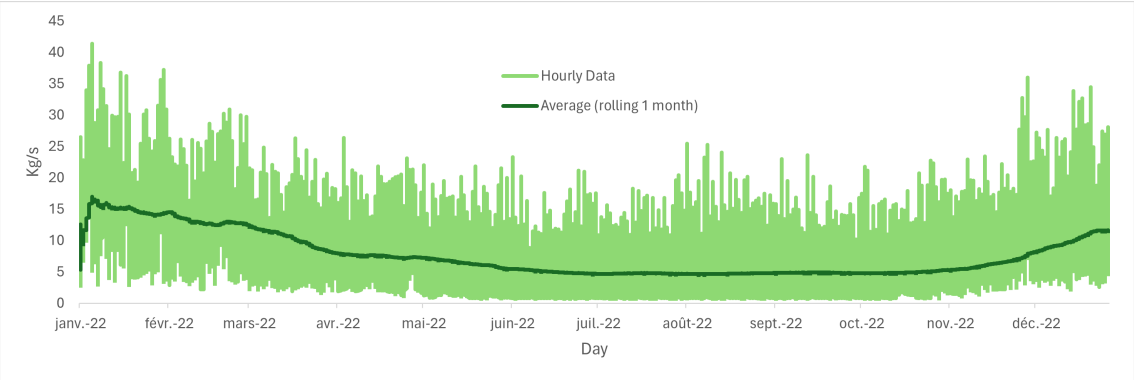


Figure 8.3: Visualization of annual heat flow dynamics in the district heating and cooling network substations (aggregated for all substations, including space heating and DHW demands).

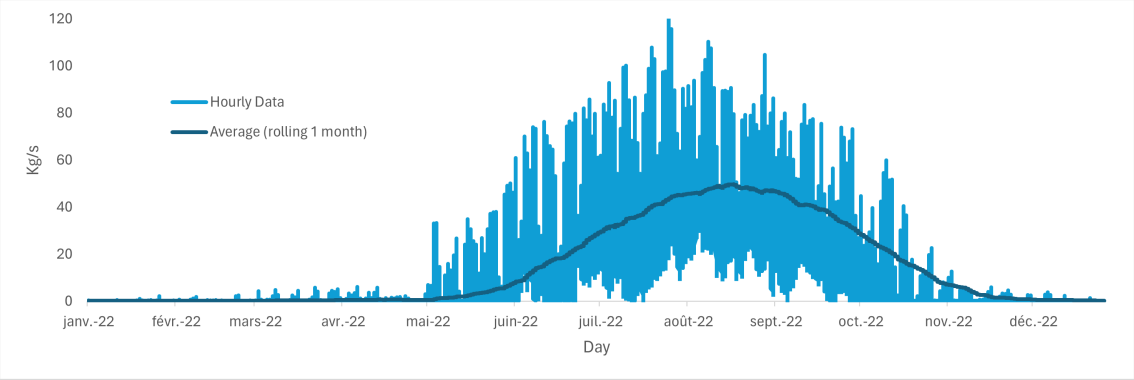


Figure 8.4: Visualization of annual cold flow dynamics in the district heating and cooling network substations (aggregated for all substations).

8.4.3 Electric load demands, PV power generation and electricity prices

To generate annual electric load time series, excluding heating and cooling related consumption, we considered distinct methodologies for office buildings and for residential buildings. For office buildings, electric loads are mainly driven by computer work-stations, lighting and auxiliary devices. The annual consumption of a computer work-station ranges from 120 to 250 kWh per year. We used hourly consumption data for screens and central processing units provided by the ADEME-ENERTECH study [376], and we aggregated these data to obtain average daily consumption per work-station, taking into account an annual holiday period of five weeks. The same study was also used to estimate lighting electric consumption. Thus, the total electric consumption

profile of an office building is obtained by summing work-station and lighting consumption.

For residential buildings, electric consumption is mainly driven by lighting, kitchen appliances (including oven, microwave, stovetop, dishwasher, refrigerator, etc.) washing-machines and various other power-consuming devices like computers, TVs and mobile phones. We generated hourly electrical load profiles for buildings by selecting specific devices and estimating hourly consumption values for each equipment type based on average annual consumption data provided by RTE [377]. Daily profiles for these equipment were established by selecting their weekly usage frequency and randomly selecting their days of usage. Similarly to office buildings, we considered an annual holiday period of five weeks where only refrigerators' consumption is present. Finally, we obtained the electrical load time series for each building by aggregating the consumption profile of the different households that it involves based on its type.

Regarding the PV power generation, time series were generated based on historical data available in PVGIS for the location of the MSE eco-district, as explained in the preceding Chapter 7 and in Appendix B.

For the hourly electricity price signal, we explored various price structures to ensure that the DRL-based approach is effective across various price scenarios, since they directly influence the optimization objective in our energy consumption cost minimization problem. The simulation results presented in the following section correspond to a generated price structure based on peak and off-peak hours. Variations on both the timing and pricing of these peak and off-peak periods are introduced to bring stochasticity into the price structure.

8.5 Simulation results

8.5.1 Training and parameter tuning

Similarly to the approach adopted in case-study 1, the DDPG agent was trained in this digital twin-based case-study 2 through a series of learning episodes of

one-year simulation, with hourly time-steps. The training objective was to acquire an effective operational strategy for the heat, cold, and battery energy storage systems, that allows minimization of the overall energy consumption costs within the smart energy system. The DRL agent successfully learnt a management strategy for the storage systems that surpassed the performance of the rule-based benchmark strategy, resulting in a noteworthy 5% reduction in annual energy consumption costs. The learning curve of the DDPG agent, depicted in Figure 8.5, illustrates the progression of the total reward signal throughout the learning phase. Once again, rewards were normalized with reference to those generated by the benchmark approach to facilitate a direct and meaningful comparison between the two approaches.

In Figure 8.6, we depict the penalty component of the reward signal, triggered when the agent selects actions that could breach the constraints of a storage system. This visual representation illustrates that the penalty component of the reward steadily converges to zero, meaning that the agent successfully learnt to navigate and manage the boundary constraints by the end of the training phase. Additionally, Figure 8.7 outlines the trajectory of the energy consumption cost component within the reward signal, demonstrating the DRL's superiority over the rule-based benchmark approach.

It should be noted that a more streamlined hyper-parameter tuning was conducted in this case-study given the prolonged training time of the DDPG agent combined with the Dymola licensing restrictions allowing one simulation at a time for each license (we own two licenses). Actually, a training episode of case-study 2 takes an average 5 minutes, whereas a learning episode of case-study 1 takes around 50 seconds. This computational time difference is mainly due to the interaction and simulation time of each of the FMUs for each hourly time-step of the training episode. That is why, preliminary hyper-parameter adjustments were carried with a particular focus on the most influential factors, primarily the exploration noise. We observed that, consistently with the results of case-study 1, normal action noise yielded slightly better results than Ornstein uhlenbeck action noise and parameter noise.

Remarkably, we obtained the simulation results presented in Figures 8.5, 8.6

and 8.7 by applying the optimal parameters obtained from case-study 1 and we found out that they performed effectively in this more complex case-study 2. This observation is in line with the conclusions drawn by Ceusters et al. [37] which suggest that the optimal hyper-parameters of a DRL agent are mainly task-specific and not entirely environment-specific. Nonetheless, further investigations need to be performed in future research to validate this relationship.

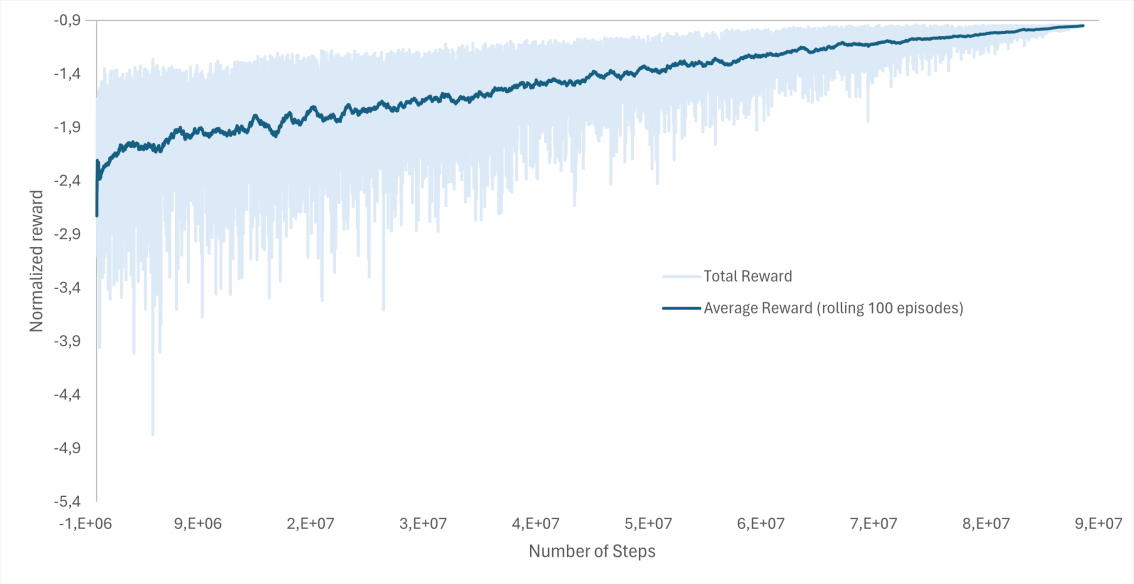


Figure 8.5: Learning curve of the DDPG agent for the case-study 2 environment: evolution of the total reward signal and the average reward over 100 rolling episodes throughout a training a cycle.

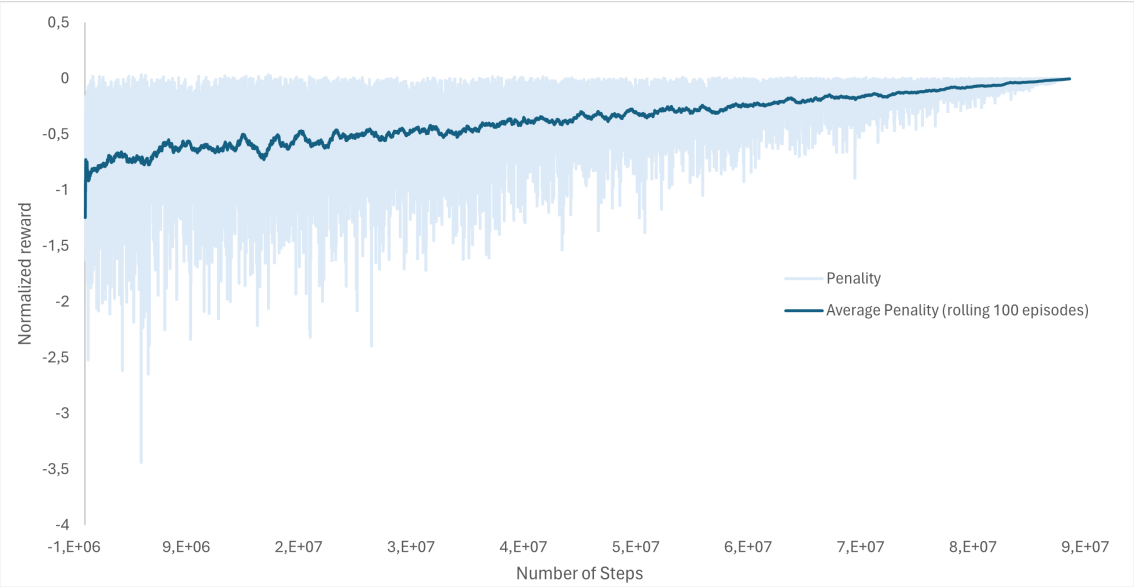


Figure 8.6: Learning curve of the DDPG agent for the case-study 2 environment: evolution of the penalty component of the reward signal and the average penalty over 100 rolling episodes throughout a training a cycle.

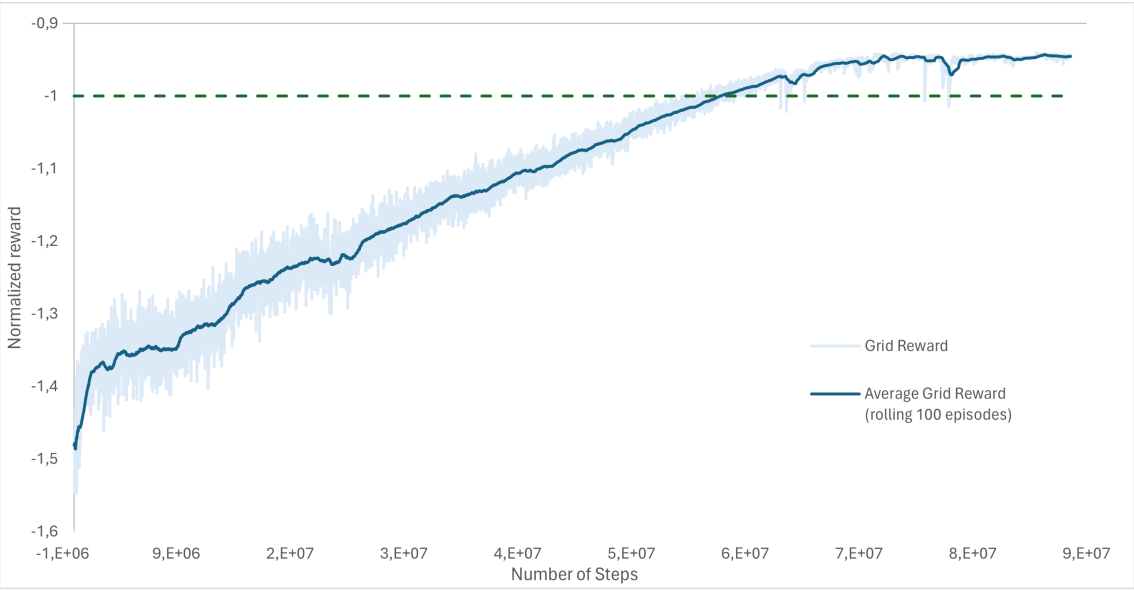


Figure 8.7: Learning curve of the DDPG agent: evolution of the cost component of the reward signal, as well as its average over a rolling horizon of 100 episodes, throughout the training cycle, and comparison with the cost obtained by the benchmark approach (presented by the dotted line in green).

8.5.2 Validation results

Once the DRL agent trained, we validate its acquired approach by applying it on the digital twin, using a distinct dataset for the exogenous data. The

simulation results presented in the following figures illustrate the strategies yielded by the DRL agent and by the predefined rule-based strategy for one randomly selected winter week and one randomly selected summer week. In each case, we present the evolution of the difference between normalized cumulative costs of the DRL-based strategy and the rule-based strategy throughout the week. In both cases, this evolution showed the superior performance of the DRL-based strategy with respect to the rule-based one. We also show a visualization of the state of charge of each of the storage systems, the aggregated heat and cold flow demand of the DHCN's substations, the heat and cold produced by the power plant, as well as the PV generation, electric loads of the buildings and the overall power withdrawn from the public utility grid.

Visualization of the DRL strategy for a winter week

Figure 8.8 presents the evolution of the difference between normalized cumulative costs of the DRL-based strategy and the rule-based strategy throughout the week, for one random winter week.

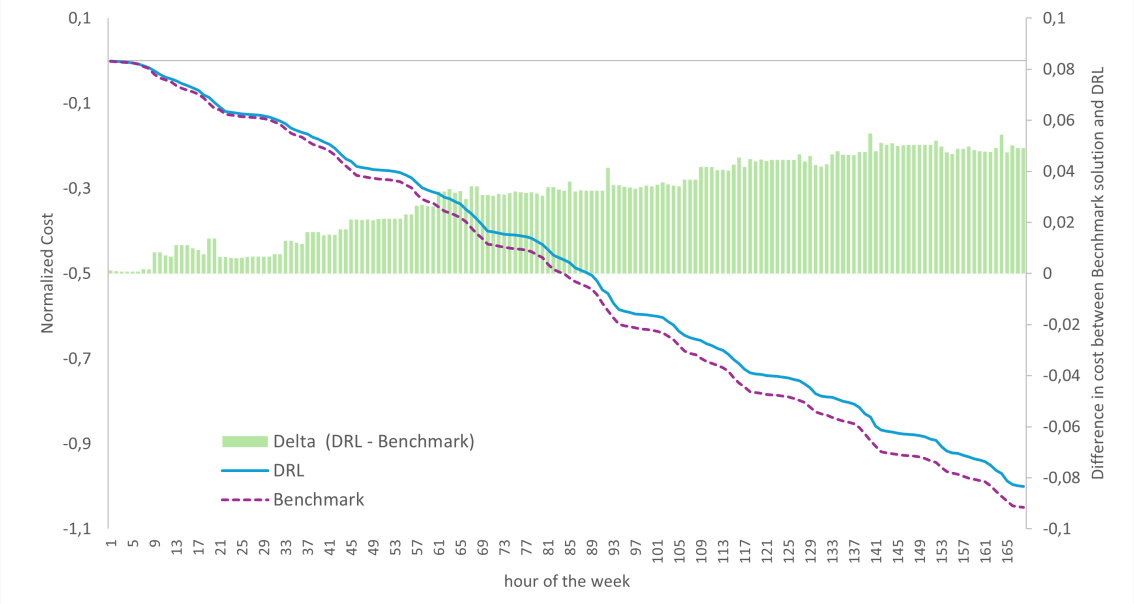


Figure 8.8: Difference between normalized cumulative costs over one random week of the winter obtained by the DRL agent and the benchmark approach.

Figures 8.9, 8.10 and 8.11 show the strategy adopted by the trained DRL agent for the selected winter week: Figures 8.9 shows the evolution of the state of

charge of each of the storage systems, as well as the electricity prices signal. Figure8.10 presents the aggregated heat flow demand of the DHCN's substations and the heat flow produced by the power plant, and Figure 8.11 shows the PV generation, electric loads of the buildings and the overall power withdrawn from the public utility grid denoted P_{grid}. The latter value is computed based on the total power demand of the district, including the electric power consumption of the thermo-refrigerating heat pumps, the electric power consumption of the heat and cold distribution system's pumps, as well as the auxiliary electric power consumption heat and cold storage systems.

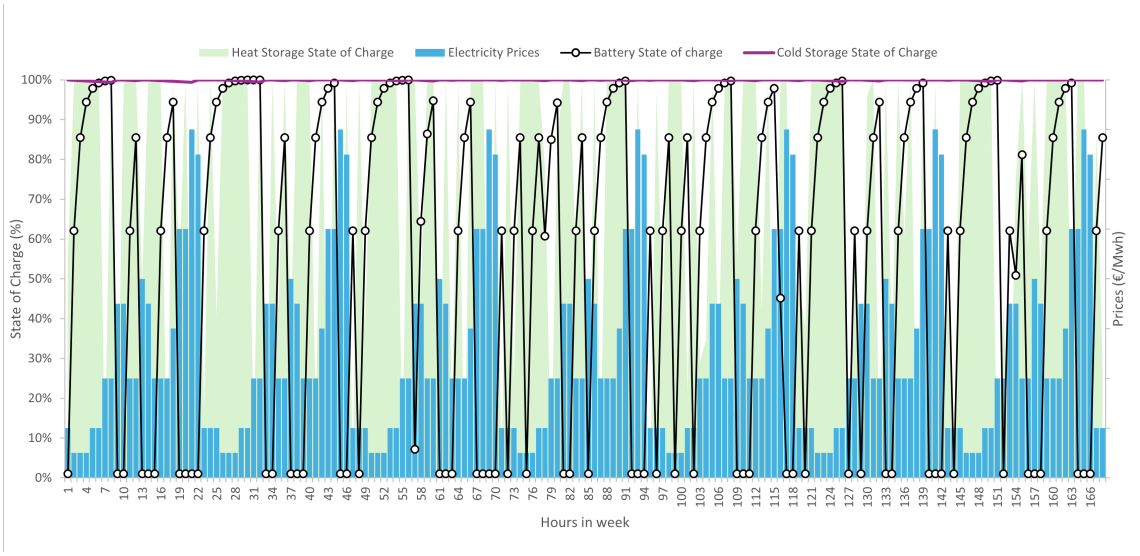


Figure 8.9: Visualization of the state of charge of the heat storage, cold storage and battery storage systems following the operational strategy of the DRL agent, together with the electricity price signal for a randomly selected winter week.

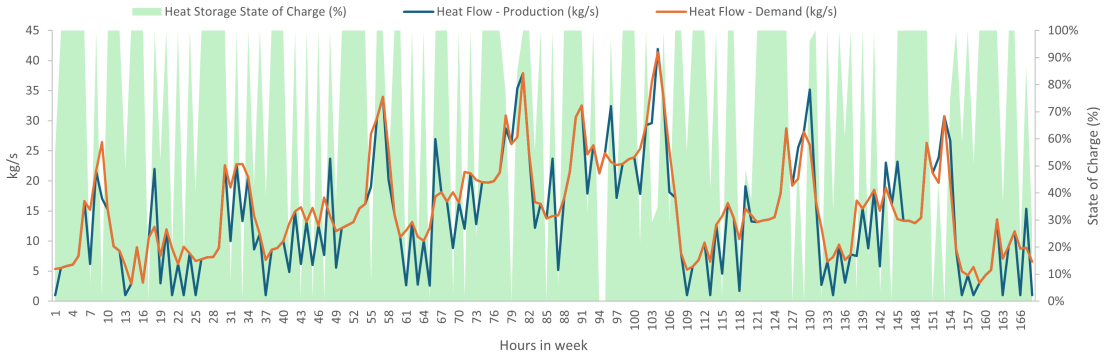


Figure 8.10: Visualization of the state of charge of the heat storage system, together with the aggregated heat flow demand of the DHCN's substations and the heat flow provided by the power plant for the DRL-based strategy.

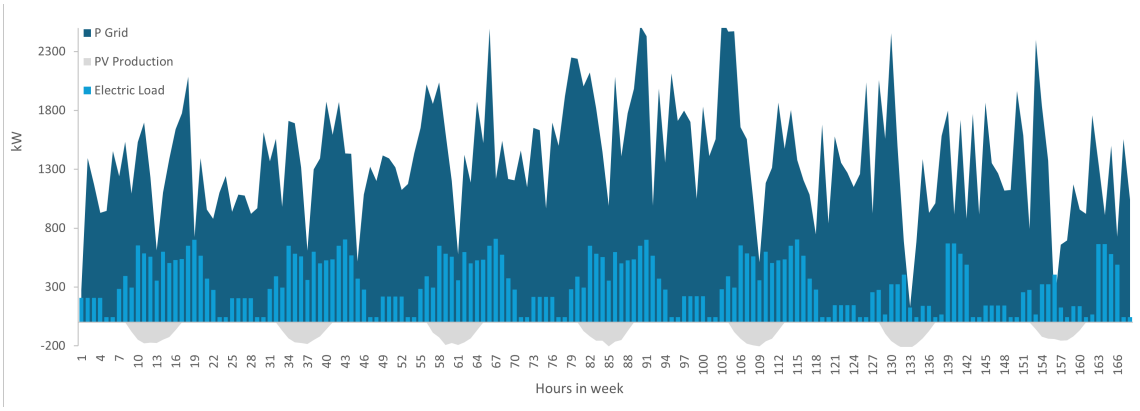


Figure 8.11: Visualization of the PV power generation, the aggregated buildings' electric loads and the overall power withdrawn from the public utility grid for the DRL-based strategy.

Visualization of the rule-based strategy for the same winter week

Figures 8.12, 8.13 and 8.14 show the strategy obtained by applying the rule-based benchmark strategy for same the selected winter week.

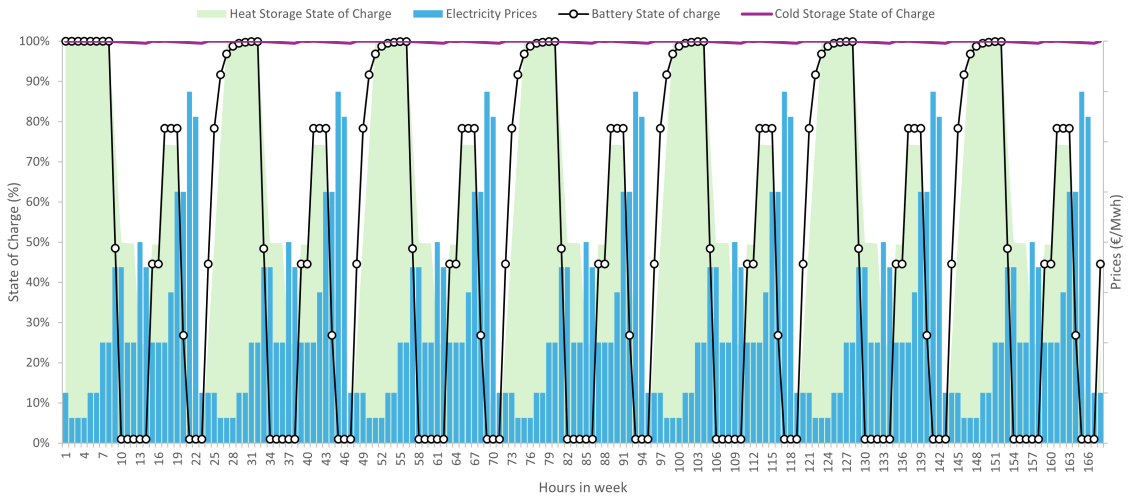


Figure 8.12: Visualization of the state of charge of the heat storage, cold storage and battery storage systems following the rule-based strategy, together with the electricity price signal for the same winter week.

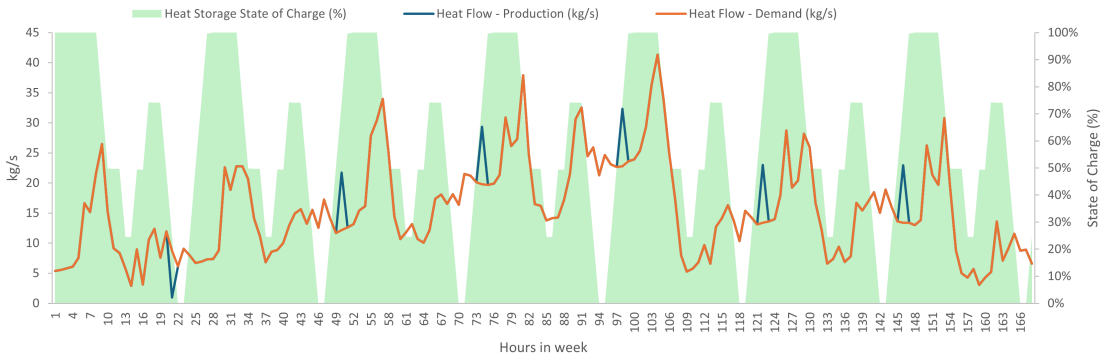


Figure 8.13: Visualization of the state of charge of the heat storage system, together with the aggregated heat flow demand of the DHCN's substations and the heat flow provided by the power plant for the rule-based strategy.

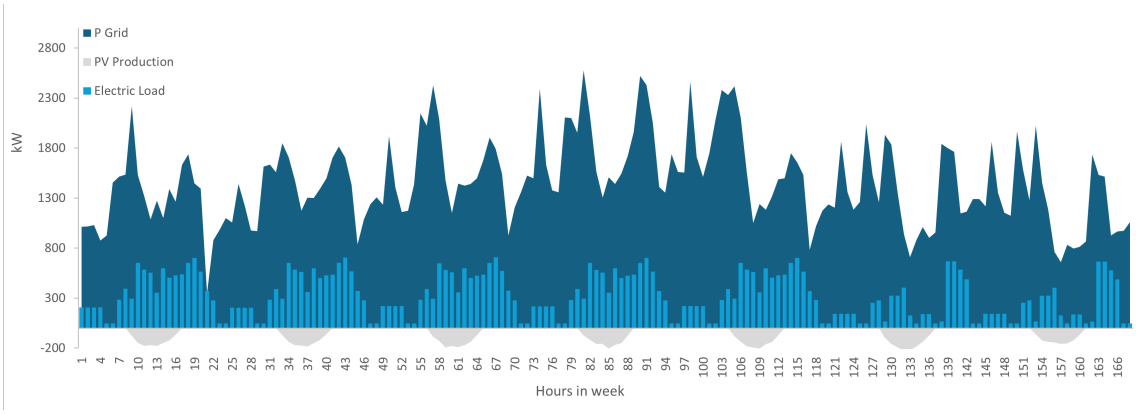


Figure 8.14: Visualization of the PV power generation, the aggregated buildings' electric loads and the overall power withdrawn from the public utility grid for the rule-based strategy.

Visualization of the DRL strategy for a summer week

Figure 8.15 presents the evolution of the difference between normalized cumulative costs of the DRL-based strategy and the rule-based strategy throughout the week, for one random summer week.

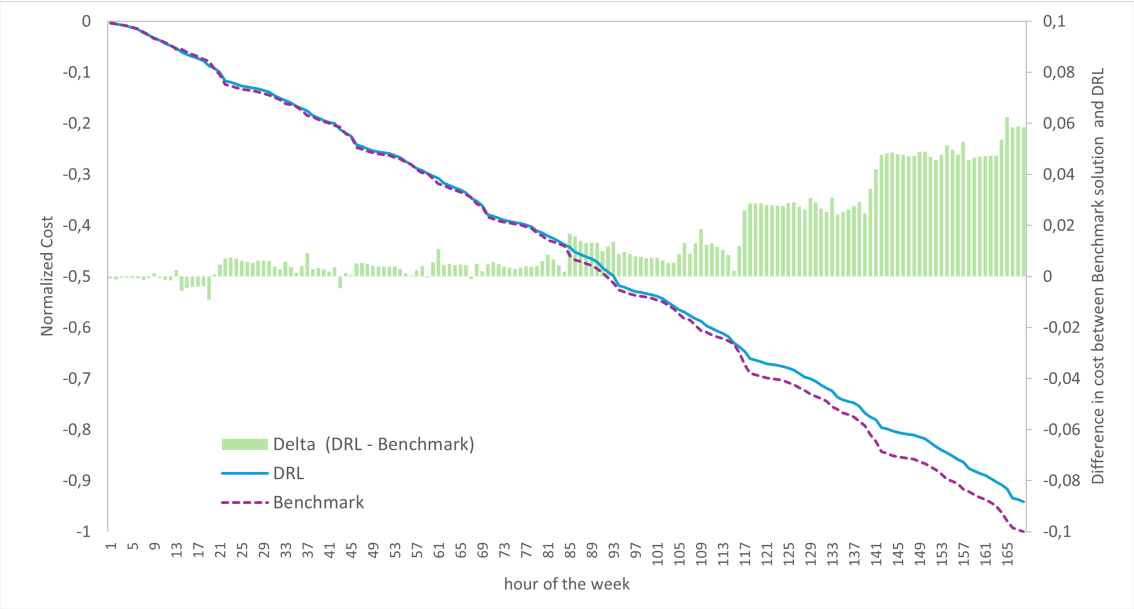


Figure 8.15: Difference between normalized cumulative costs over one random week of the summer obtained by the DRL agent and the benchmark approach.

Figures 8.16, 8.17, 8.18 and 8.19 show the strategy adopted by the trained DRL agent for the selected summer week: Figure 8.16 shows the evolution of the state of charge of each of the storage systems, as well as the electricity prices signal. Figure 8.17 presents the aggregated cold flow demand of the DHCN's substations and the heat cold produced by the power plant, Figure 8.18 presents the aggregated heat flow demand of the DHCN's substations and the heat flow produced by the power plant, and Figure 8.19 shows the PV generation, electric loads of the buildings and the overall power withdrawn from the public utility grid denoted P_{grid}.

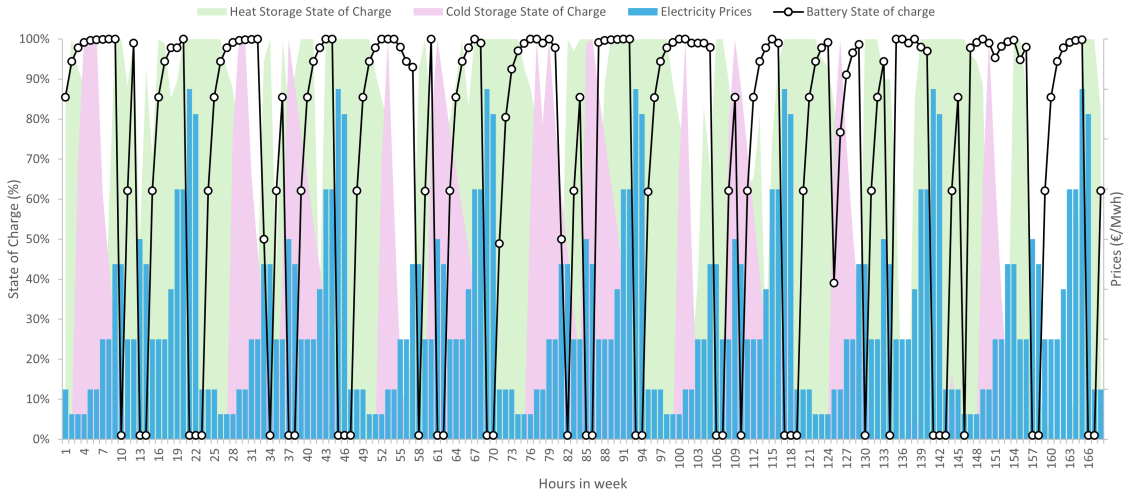


Figure 8.16: Visualization of the state of charge of the heat storage, cold storage and battery storage systems following the operational strategy of the DRL agent, together with the electricity price signal for a randomly selected summer week.

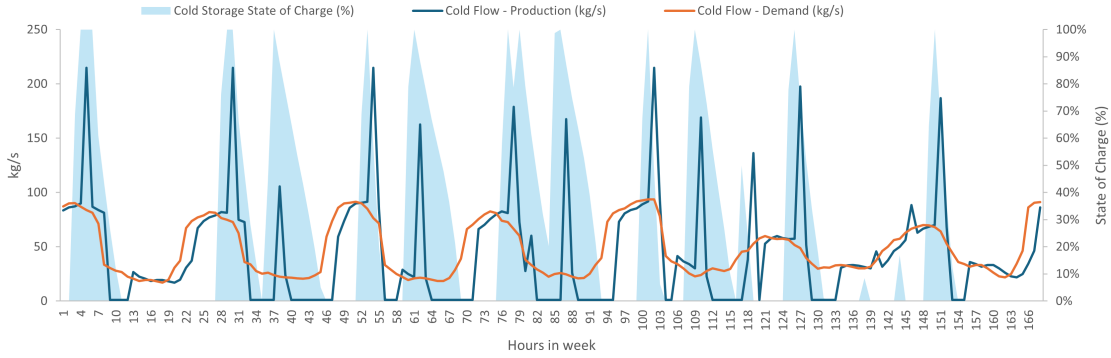


Figure 8.17: Visualization of the state of charge of the cold storage system, together with the aggregated cold flow demand of the DHCN's substations and the cold flow provided by the power plant for the DRL-based strategy.

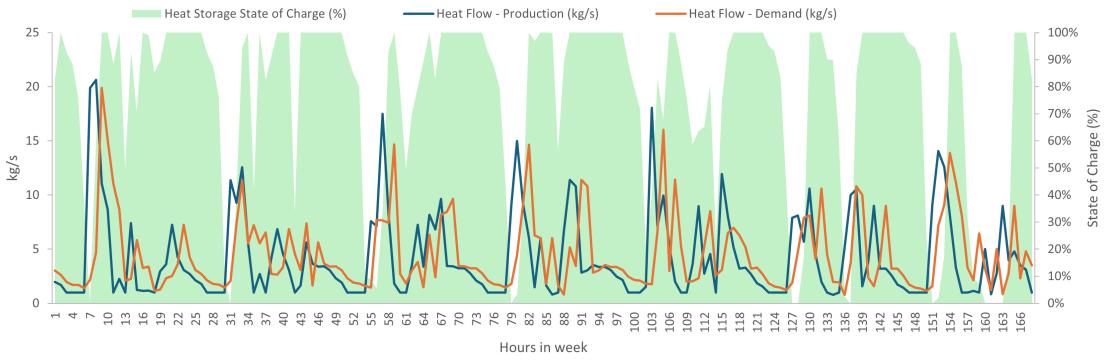


Figure 8.18: Visualization of the state of charge of the heat storage system, together with the aggregated heat flow demand of the DHCN's substations and the heat flow provided by the power plant for the DRL-based strategy.

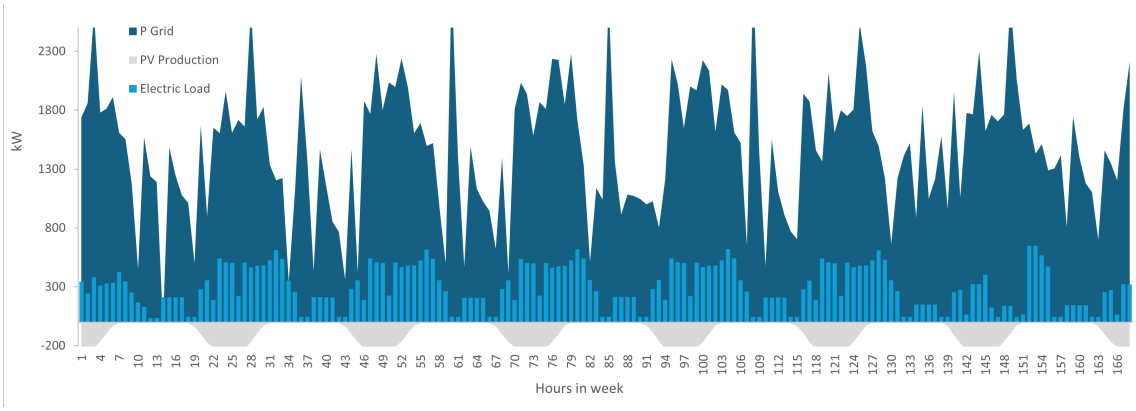


Figure 8.19: Visualization of the PV power generation, the aggregated buildings' electric loads and the overall power withdrawn from the public utility grid for the DRL-based strategy.

Visualization of the rule-based strategy for the same summer week

Figures 8.20, 8.21, 8.22 and 8.23 show the strategy obtained by applying the rule-based benchmark strategy for same the selected summer week.

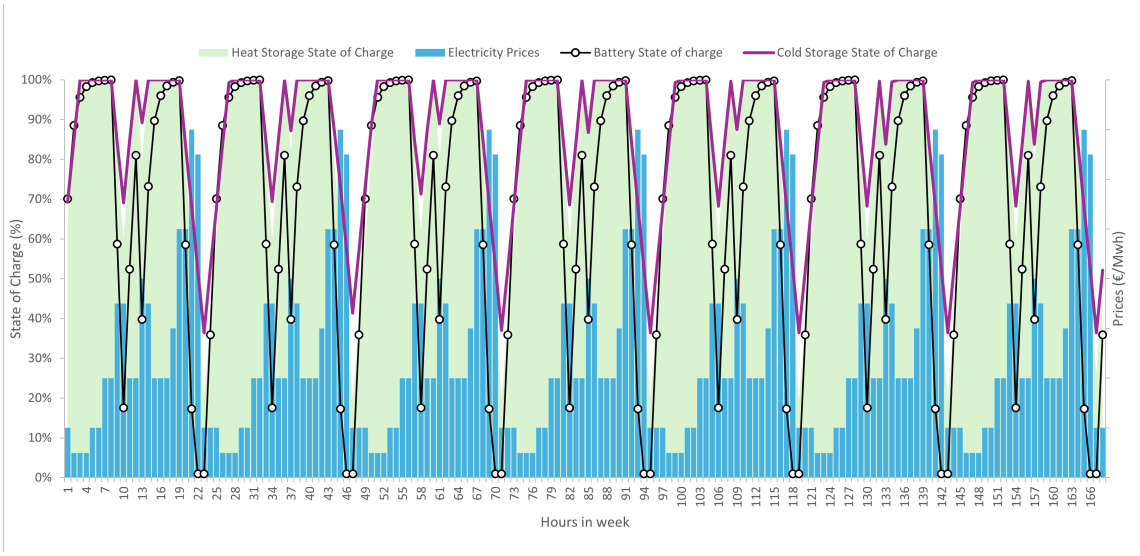


Figure 8.20: Visualization of the state of charge of the heat storage, cold storage and battery storage systems following the rule-based strategy, together with the electricity price signal for the same summer week.

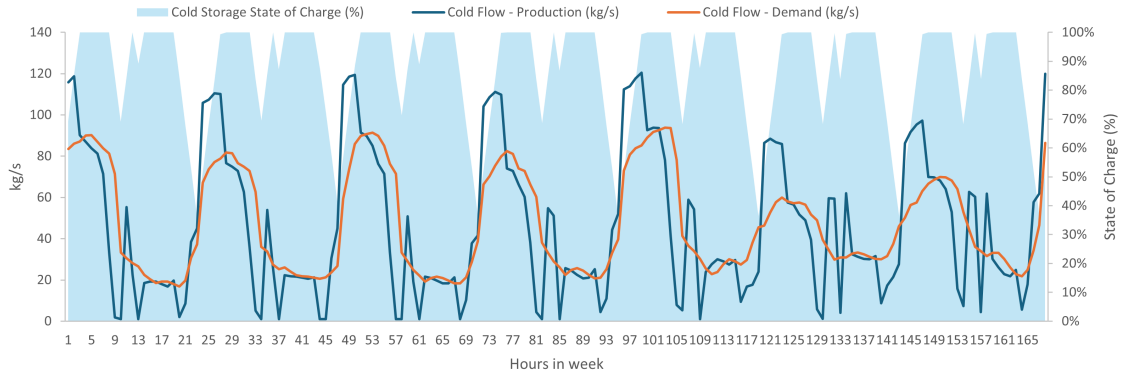


Figure 8.21: Visualization of the state of charge of the cold storage system, together with the aggregated cold flow demand of the DHCN's substations and the cold flow provided by the power plant for the rule-based strategy, over the same summer week.

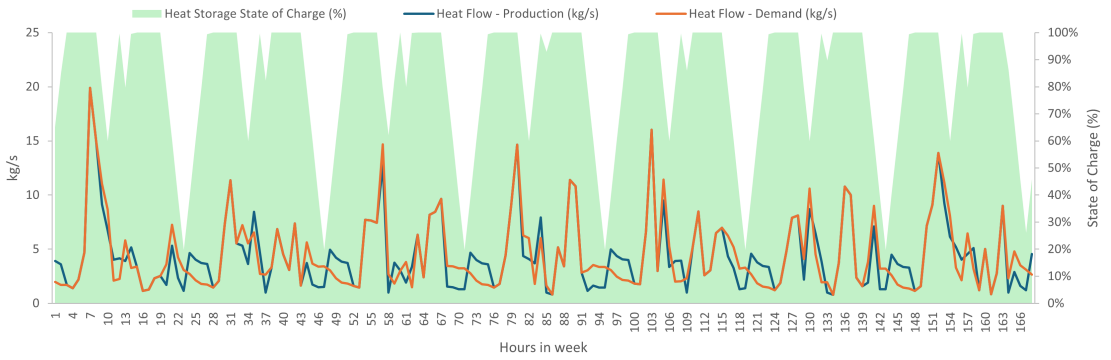


Figure 8.22: Visualization of the state of charge of the heat storage system, together with the aggregated heat flow demand of the DHCN's substations and the heat flow provided by the power plant for the rule-based strategy, over the same summer week.

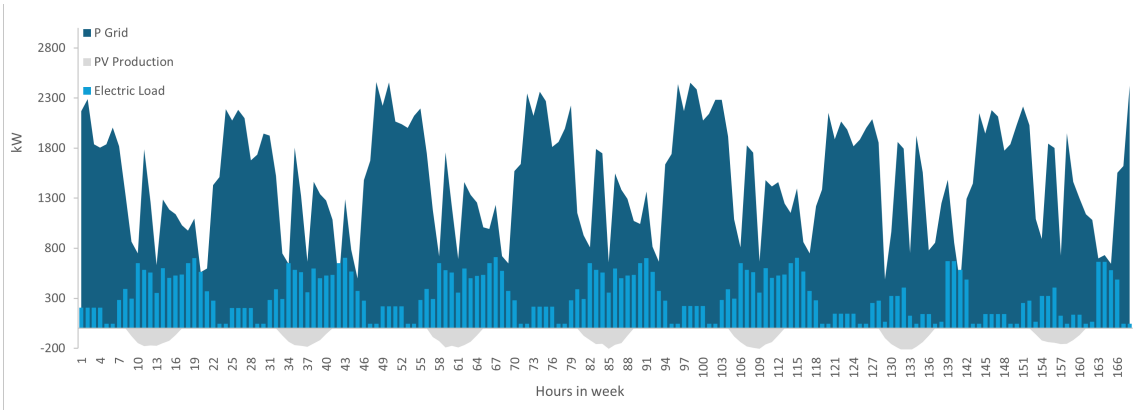


Figure 8.23: Visualization of the PV power generation, the aggregated buildings' electric loads and the overall power withdrawn from the public utility grid for the rule-based strategy over the same summer week.

A notable distinction between the energy management strategies obtained for the winter and for the summer periods basically lies in the utilization of the different energy storage systems. While during winter, we observe a deployment of the battery and heat storage systems, with limited to no utilization of the cold storage, the three storage systems come into play during winter. This seasonal variation can be attributed to the absence of cooling demand during winter months. Conversely, in the summer time, the cooling demand is coupled with a heating demand mainly for domestic hot water.

This intricate interplay between the three storage systems demonstrates the adaptability of the DRL-based approach in learning to manage multi-energy systems according to seasonal requirements and prices signals. It is worth not-

ing that the DRL agent successfully learns these dynamic strategies without being provided with any prior guidance regarding the actions to take. This emphasizes the capability of the DRL agent to adapt and make informed decisions based on evolving conditions and system requirements.

Overall, the results obtained from applying the proposed DRL-based approach on the digital twin-based case study, and benchmarking it with a rule-based approach, corroborate the results from case-study 1 presented in Chapter 5 suggesting that the DRL approach holds a significant promise for the optimized energy management in smart multi-energy systems. Particularly, the application of this approach on the more complex setting of case-study 2 further showcases the adaptability of DRL and its potential in addressing the intricacies of real-world smart energy systems' optimization challenges.

8.6 Conclusion

In this chapter, we implemented the proposed DRL-based approach on the MSE digital twin-based case study. The strategy learnt by the DDPG agent aims at simultaneously operating the storage systems in a way that minimizes the energy consumption costs within the smart multi-energy system. This acquired strategy resulted in cost savings of at least $5\,\%$ when compared to a predefined rule-based strategy.

Future research works pertaining to this case-study will focus on reducing the simulation time of the digital twin. This effort will not only diminish the DRL agent's training duration and therefore facilitate a more comprehensive parameter tuning, but it will also facilitate the identification and execution of an optimization-based benchmark approach that closely approximates the theoretical optimum, instead of the rule-based approach.

Further prospective work within this case-study will also expand the scope of the optimization objective to include intricate and multi-objective applications such as peak shaving and demand response, together with energy cost minimization.

General conclusion and perspectives

Summary and contributions

This PhD research work proposes the application of Deep Reinforcement Learning to address the problem of optimally operating smart multi-energy systems: a DRL-based framework has been developed and applied on two simulated smart multi-energy system case-studies. **Chapter 1** of this manuscript provided an overview of the smart energy systems concept. It defined key concepts such as smart electrical grids, district heating and cooling systems and smart thermal grids and set the stage for the broader emerging concepts of smart multi-energy grids and smart (multi) energy systems. Then, **chapter 2** sheds light on the optimal control and energy management in smart energy systems and focuses on optimization techniques and established methodologies that can be used to address this problem like Model Predictive Control. These optimization-based techniques are classified into three categories: exact methods, approximate methods and hybrid methods. While exact mathematical methods often guarantee the exact optimum, they generally suffer from costly optimization procedures in terms of computational requirement. They are thus hardly adequate to solve real-time optimal control of complex systems. On the other hand, an RL agent, once it learnt how to act, is able to take actions within a few milliseconds. Besides, most of model-based approaches, such as MPC, require accurate system models and predictors as well as appropriate solvers. This intricate model development demands domain expertise and requires a continual adaptation and a re-design of these components

whenever a change is made to the architecture or scale of the smart energy system at hand. These limitation may significantly restrict the adaptability and scalability of these methods in complex and dynamically evolving systems. In response to these challenges, we propose a paradigm shift towards data-based and model-free approaches: we introduce a deep reinforcement learning based approach for the optimized energy management in smart energy systems and evaluate its performance through comparison with an MPC controller. In **chapter 3** we propose a comprehensive examination of the RL paradigm as well as its combination with deep neural networks for function approximation by explaining its principle and distinctive features and tracing its historical evolution from its earliest pioneers to its contemporary academic and industrial applications. In **chapter 4**, we propose an exploration of the MPC approach from theoretical and practical points of view. Following this, we simultaneously benchmark DRL and MPC approaches through a simulated smart energy systems case study featuring three storage systems and drawn from the MSE real-life smart energy system. The problem of managing these three storage systems while minimizing total energy consumption costs is formulated both as a Markov Decision Process and as a linear programming optimization problem. The first formulation allows solving the problem using an RL-based approach and the latter is embedded in an MPC controller. The DRL framework developed is based on an actor-critic architecture, namely a DDPG algorithm, that combines the advantages of both value-based and policy-based RL algorithms and allows dealing with continuous action spaces. On the other hand, the MPC framework developed is based on a linear MPC in order to align with the prevailing practices, despite the potential for superior performance of a non linear MPC. Simulation results presented in **chapter 5** showed that the trained DRL agent performs close (within 98%) to the theoretical MPC controller (that was provided with perfect forecasts) and performs even better than some variants of the MPC controller that involve more realistic forecasts. To sum up, the first part of this thesis represented one of the first studies in the literature to simultaneously benchmark DRL and MPC on smart multi-energy system case studies and suggests that DRL is a promising

approach for the optimal energy management in smart multi-energy systems. The second part of this thesis is devoted to validating the proposed DRL-based approach on the Meridia smart Energy (MSE) case-study: **chapter 6** presents the MSE smart multi-energy system and details the various energy systems that it involves and **chapter 7** introduces the digital twin that we developed under Dymola to better account for the dynamics of the energy systems of the MSE eco-district. This digital twin is exported as a Functional Mock-up unit (FMU) and plays the role of the environment that the DRL agent interacts with during the training, validation and test processes. Finally, simulation results of applying the DRL approach to the MSE digital twin are presented in **chapter 8** and showcased that the DRL agent succeeds in learning a near-optimal energy management strategy and outperforms widely used rule-based approaches.

This work paved the way for applying the DRL-based approach on the real-world MSE smart energy system by seamlessly transitioning from simulation to reality. Indeed, future works will include training the DRL agent on other optimization objectives that integrate peak shaving, load shedding as well as participation in various ancillary service and energy markets, before focusing on the learning transfer from simulated environments to the real-world MSE systems and data. This involves further investigating the generalizability of the DRL agent, i.e. its ability to leverage the knowledge that it learnt on given environments to perform well in a wider range of environments and situations.

Challenges, limitations and future works

While this PhD research work presented a successful implementation of a DRL-based approach for the optimal energy management in smart multi-energy system case studies, it also revealed some challenges and limitations that require further refinement and investigation in future research works. These challenges and shortcomings are briefly discussed below.

Constraint handling and learning legal actions

One difficulty that is encountered by the DRL agent during training is learning legal actions and effectively handling system constraints. A specific reward shaping, such as the integration of hand-crafted penalty components in the reward signal, might drive the RL agent towards learning legal actions while also speeding up the training time, since the agent would spend less training steps in exploring actions that do not yield a change of the state or reward. Nevertheless, achieving a robust constraint handling remains a quite complex task. That is why future research should explore advanced approaches for constraint handling such as constrained reinforcement learning to improve the DRL agent's ability to adhere to legal actions and system constraints.

Variability of training results and catastrophic forgetting

Training a DRL agent also often involves a variability of the results, which can be traced back, for instance, to the stochastic nature of the exploration noises. Moreover, the problem of catastrophic forgetting also presents a significant challenge while training a DRL Agent. The term catastrophic forgetting is used to refer to the tendency of an agent to forget knowledge that has previously been learned during training, when adapting to new information and situations [378]. Thus, future research efforts should also focus on mitigating variability and catastrophic forgetting issues, for example by developing more stable exploration and training strategies.

Challenges of real-world applications of RL

Even though the DRL approach has proven its success on a bunch of applications, only a few real world applications are beginning to show. Actually, the transition of DRL from simulation environments to extensive real-world deployment is not trivial [379] and is still limited by several challenges. This idea was outlined by Dulac-Arnold et al. [380] who identified nine independent challenges that limit common deployment of RL in real-world systems and formalized them in the context of Markov Decision Processes. The authors stated that an approach that addresses these nine challenges would be

ready to be widely implemented in real-world systems.

Among the challenges of applying Reinforcement Learning approaches in real-world problems: the RL agent often does not have the ability to freely interact with the real environment mainly because of safety and/or cost reasons. In such situations, the agent may not have access to the actual environment but only to a simulation of it as it was the case for the present work. One of the main challenges in this kind of problems is that one has to deal with what is called *the reality gap* between the simulator and the true environment as denoted by [19]. In order to reduce this gap, one can first aim to make the digital twin as accurate as possible. Second, the Deep Reinforcement Learning algorithm can be designed in a way that aims at improving generalizability. In fact, despite the numerous recent successful applications of RL, generalization across different unseen scenarios remains one of its fundamental challenges for real-life applications [381]. These challenges can be mitigated for instance by incorporating transfer learning techniques [382], [383] in order to seamlessly adapt to real-world scenarios. Besides, increasing the diversity and the size of the training set is known to improve the generalization of RL agents.

Training time and hyper-parameter tuning

The training phases of a DRL agent are generally time-consuming. For instance, a single training episode for case study 2 in our study requires an average of 5 minutes to complete. Considering that a training cycle involves thousands of episodes, this extensive training time can be challenging in the research and development process. In fact, the extended training time itself is typically not problematic once the initial training phase is completed, since the DRL agent, once trained, is capable of taking actions within a matter of milliseconds. However, the lengthy training duration poses challenges for initial experimentation and comprehensive hyper-parameter tuning where the ability to iterate through various settings can be crucial for fine-tuning and optimizing the agent. Typically, due the prolonged training duration of the DRL algorithm, we opted for a hyper-parameter tuning that relies on varying one

parameter at a time. Yet, we believe that additional and exhaustive parameter-tuning may further enhance the DRL agent's performance and generalizability. That is why, future research works should also investigate methods to reduce the training duration, potentially through parallelization. This option could not be investigated in our study due to inherent limitations (our current implementation of the DRL agent does not support parallelization) and due to licensing constraints of the Dymola software. Therefore, alternative approaches to build the digital twin should as well be considered in future works such as developing surrogate models derived from the Dymola digital twin or building data-driven models using historical data from the actual systems.

Other future research directions

The following points outline other potential research directions that aim at enhancing the versatility and applicability of the DRL approach for the optimization of smart multi-energy systems:

* Integrating additional optimization objectives: future research works will extend the considered smart multi-energy system use-cases to integrate a wider spectrum of optimization objectives, beyond energy cost minimization. Typically, for the MSE case-study, future use-cases would include collective self-consumption within the eco-district and participation in various markets such as frequency regulation and demand response programs. Besides, it is also crucial to account for the aging processes within the different storage systems while dealing with their energy management in order to ensure their long-term efficiency and performance.
* Expanding the scope of the smart energy systems considered: as the MSE eco-district evolves, it will incorporate further energy systems and functionalities such as electric vehicles, smart charging, Vehicle-to-grid systems, electric vehicle charging stations management, optimization of the district's public lighting and even building-level demand response, etc. Thus, including the management of these additional flexibility potentials in the proposed DRL-based energy management systems becomes increasingly relevant and emerges as a promising avenue for future re-

search.

* Exploring application on similar smart energy systems: other eco-districts, heating and cooling networks and smart energy systems managed by Idex offer opportunities for extending the methodology developed in this work, evaluating its performance and assessing its generalization capabilities on diverse environments. This future research would provide valuable insights for a broader applicability of the DRL-based approach.
* Integrating other energy vectors: while the current research work has primarily focused on the integration and management of electricity, heating and cooling systems, it is worth noting that other energy vectors such as gas and hydrogen are integral components of future comprehensive smart multi-energy systems. While their integration in the DRL-based smart multi-energy management approach was beyond the scope of the present research work, this holds a great potential for future research.
* Exploring alternative DRL and MPC variants: future research can explore alternative DRL algorithms such as Twin Delayed Deep Deterministic Policy Gradient (TD3), Proximal Policy Optimization (PPO) and Soft Actor Critic (SAC) in order to evaluate and enhance the robustness of the proposed approach. Furthermore, benchmarking against other variants of MPC like Nonlinear MPC (NMPC) and stochastic MPC to consider forecast uncertainties could also be considered.
* Investigating imitative Learning: imitative or imitation learning involves using expert-generated actions as a form to guide the reinforcement learning agent during the training phase. This can help the RL agent learn and converge faster and more effectively by leveraging the knowledge introduced by the expert's actions. In future research work, we can for instance use optimized control actions generated by a Model Predictive Controller as expert demonstrations for training the DRL agents.

These suggested future research directions extend the application of the DRL-based multi-energy management in smart energy systems. Exploring these research directions holds the potential to address a wider range of optimization objectives, enhance the adaptability of the proposed methodology to evolving

energy landscapes and improve its effectiveness in real-world scenarios. As these areas continue to evolve, they contribute to the advancement of sustainable, cost-effective, and resilient energy management solutions for the future.

Bibliography

Bibliography

[1] X. Fang, S. Misra, G. Xue, and D. Yang, "Smart grid—the new and improved power grid: a survey," *IEEE communications surveys & tutorials*, vol. 14, no. 4, pp. 944–980, 2011.

[2] M. L. Tuballa and M. L. Abundo, "A review of the development of smart grid technologies," *Renewable and Sustainable Energy Reviews*, vol. 59, pp. 710–725, 2016.

[3] L. Yang, E. Entchev, A. Rosato, and S. Sibilio, "Smart thermal grid with integration of distributed and centralized solar energy systems," *Energy*, vol. 122, pp. 471–481, 2017.

[4] M. van den Ende, Z. Lukszo, and P. M. Herder, "Smart thermal grid," in *2015 IEEE 12th International Conference on Networking, Sensing and Control*, IEEE, 2015, pp. 432–437.

[5] C. Stănişteanu, "Smart thermal grids–a review," *The Scientific Bulletin of Electrical Engineering Faculty*, vol. 1, no. ahead-of-print, 2017.

[6] P. Mancarella, "Smart multi-energy grids: concepts, benefits and challenges," in *2012 IEEE Power and Energy Society General Meeting*, IEEE, 2012, pp. 1–2.

[7] H. Lund, A. N. Andersen, P. A. Østergaard, B. V. Mathiesen, and D. Connolly, "From electricity smart grids to smart energy systems–a market operation based approach and understanding," *Energy*, vol. 42, no. 1, pp. 96–102, 2012.

[8] T. Ma, J. Wu, L. Hao, W.-J. Lee, H. Yan, and D. Li, "The optimal structure planning and energy management strategies of smart multi energy systems," *Energy*, vol. 160, pp. 122–141, 2018.

[9] C. E. Garcia, D. M. Prett, and M. Morari, "Model predictive control: theory and practice—a survey," *Automatica*, vol. 25, no. 3, pp. 335–348, 1989.

[10] M. Morari and J. H. Lee, "Model predictive control: past, present and future," *Computers & Chemical Engineering*, vol. 23, no. 4-5, pp. 667–682, 1999.

[11] R. Gelleschus, M. Böttiger, P. Stange, and T. Bocklisch, "Comparison of optimization solvers in the model predictive control of a pv-battery-heat pump system," *Energy Procedia*, vol. 155, pp. 524–535, 2018.

[12] T. Wang, H. Kamath, and S. Willard, "Control and optimization of grid-tied photovoltaic storage systems using model predictive control," *IEEE Transactions on Smart Grid*, vol. 5, no. 2, pp. 1010–1017, 2014.

[13] A. Parisio, E. Rikos, and L. Glielmo, "A model predictive control approach to microgrid operation optimization," *IEEE Transactions on Control Systems Technology*, vol. 22, no. 5, pp. 1813–1827, 2014.

[14] P. Pflaum, M. Alamir, and M. Y. Lamoudi, "Comparison of a primal and a dual decomposition for distributed mpc in smart districts," in *2014 IEEE international conference on smart grid communications (SmartGridComm)*, IEEE, 2014, pp. 55–60.

[15] D. Bousnina, W. d. Oliveira, and P. Pflaum, "A stochastic optimization model for frequency control and energy management in a microgrid," in *International Conference on Machine Learning, Optimization, and Data Science*, Springer, 2020, pp. 177–189.

[16] Y. Ji, J. Wang, J. Xu, X. Fang, and H. Zhang, "Real-time energy management of a microgrid using deep reinforcement learning," *Energies*, vol. 12, no. 12, p. 2291, 2019.

[17] T. Sogabe, D. B. Malla, S. Takayama, *et al.*, "Smart grid optimization by deep reinforcement learning over discrete and continuous action space," in *2018 IEEE 7th World Conference on Photovoltaic Energy Conversion (WCPEC)(A Joint Conference of 45th IEEE PVSC, 28th PVSEC & 34th EU PVSEC)*, IEEE, 2018, pp. 3794–3796.

[18] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[19] V. François-Lavet, "Contributions to deep reinforcement learning and its applications in smartgrids," Ph.D. dissertation, Université de Liège, Liège, Belgique, 2017.

[20] R. Bellman, "Dynamic programming," *Press Princeton, New Jersey*, 1957.

[21] A. L. Samuel, "Some studies in machine learning using the game of checkers," *IBM Journal of research and development*, vol. 3, no. 3, pp. 210–229, 1959.

[22] B. Zhang, W. Hu, D. Cao, Q. Huang, Z. Chen, and F. Blaabjerg, "Deep reinforcement learning–based approach for optimizing energy conversion in integrated electrical and heating system with renewable energy," *Energy Conversion and Management*, vol. 202, p. 112 199, 2019.

[23] V. Mnih, K. Kavukcuoglu, D. Silver, *et al.*, "Playing atari with deep reinforcement learning," *arXiv preprint arXiv:1312.5602*, 2013.

[24] V. François-Lavet, P. Henderson, R. Islam, M. G. Bellemare, and J. Pineau, "An introduction to deep reinforcement learning," *arXiv preprint arXiv:1811.12560*, 2018.

[25] V. Mnih, K. Kavukcuoglu, D. Silver, *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[26] D. Silver, A. Huang, C. J. Maddison, *et al.*, "Mastering the game of go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, pp. 484–489, 2016.

[27] T. de Bruin, J. Kober, K. Tuyls, and R. Babuška, "Improved deep reinforcement learning for robotics through distribution-based experience retention," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2016, pp. 3947–3952.

[28] M. Vecerik, T. Hester, J. Scholz, *et al.*, "Leveraging demonstrations for deep reinforcement learning on robotics problems with sparse rewards," *arXiv preprint arXiv:1707.08817*, 2017.

[29] R. Liaw, S. Krishnan, A. Garg, D. Crankshaw, J. E. Gonzalez, and K. Goldberg, "Composing meta-policies for autonomous driving using hierarchical deep reinforcement learning," *arXiv preprint arXiv:1711.01503*, 2017.

[30] A. E. Sallab, M. Abdou, E. Perot, and S. Yogamani, "Deep reinforcement learning framework for autonomous driving," *Electronic Imaging*, vol. 2017, no. 19, pp. 70–76, 2017.

[31] S. Carta, A. Ferreira, A. S. Podda, D. R. Recupero, and A. Sanna, "Multi-dqn: an ensemble of deep q-learning agents for stock market forecasting," *Expert systems with applications*, vol. 164, p. 113 820, 2021.

[32] M. Lopez-Martin, B. Carro, and A. Sanchez-Esguevillas, "Application of deep reinforcement learning to intrusion detection for supervised problems," *Expert Systems with Applications*, vol. 141, p. 112 963, 2020.

[33] H. Liu, C. Yu, H. Wu, Z. Duan, and G. Yan, "A new hybrid ensemble deep reinforcement learning model for wind speed short term forecasting," *Energy*, vol. 202, p. 117 794, 2020.

[34] W. Knight, *Google just gave control over data center cooling to an ai*, 2018.

[35] OpenAI, *ChatGPT*. [Online]. Available: https://chat.openai.com.

[36] K. I. Roumeliotis and N. D. Tselikas, "Chatgpt and open-ai models: a preliminary review," *Future Internet*, vol. 15, no. 6, p. 192, 2023.

[37] G. Ceusters, R. C. Rodríguez, A. B. García, *et al.*, "Model-predictive control and reinforcement learning in multi-energy system case studies," *Applied Energy*, vol. 303, p. 117 634, 2021.

[38] G. Brockman, V. Cheung, L. Pettersson, *et al.*, "Openai gym," *arXiv preprint arXiv:1606.01540*, 2016.

[39] S. M. Amin and B. F. Wollenberg, "Toward a smart grid: power delivery for the 21st century," *IEEE power and energy magazine*, vol. 3, no. 5, pp. 34–41, 2005.

[40] European Commission. "Smart grid regional group." (2022), [Online]. Available: https://energy.ec.europa.eu/topics/infrastructure/projects-common-interest/selection-process/smart-grid-regional-group_en (visited on 11/28/2022).

[41] Y. Bamberger, J. Baptista, R. Belmans, *et al.*, "Vision and strategy for europe's electricity networks of the future: european technology platform smart grids," 2006.

[42] M. Moretti, S. N. Djomo, H. Azadi, *et al.*, "A systematic review of environmental and economic impacts of smart grids," *Renewable and Sustainable Energy Reviews*, vol. 68, pp. 888–898, 2017.

[43] M. Mohammadi, Y. Noorollahi, B. Mohammadi-ivatloo, M. Hosseinzadeh, H. Yousefi, and S. T. Khorasani, "Optimal management of energy hubs and smart energy hubs–a review," *Renewable and Sustainable Energy Reviews*, vol. 89, pp. 33–50, 2018.

[44] M. E. El-Hawary, "The smart grid—state-of-the-art and future trends," *Electric Power Components and Systems*, vol. 42, no. 3-4, pp. 239–250, 2014.

[45] R. H. Lasseter and P. Paigi, "Microgrid: a conceptual solution," in *2004 IEEE 35th annual power electronics specialists conference (IEEE Cat. No. 04CH37551)*, IEEE, vol. 6, 2004, pp. 4285–4290.

[46] H. Jiayi, J. Chuanwen, and X. Rong, "A review on distributed energy resources and microgrid," *Renewable and Sustainable Energy Reviews*, vol. 12, no. 9, pp. 2472–2483, 2008.

[47] D. Kolokotsa, N. Kampelis, A. Mavrigiannaki, *et al.*, "On the integration of the energy storage in smart grids: technologies and applications," *Energy Storage*, vol. 1, no. 1, e50, 2019.

[48] D.-I. Stroe, V. Knap, M. Swierczynski, A.-I. Stroe, and R. Teodorescu, "Operation of a grid-connected lithium-ion battery energy storage system for primary frequency regulation: a battery lifetime perspective," *IEEE transactions on industry applications*, vol. 53, no. 1, pp. 430–438, 2016.

[49] Y. Shi, B. Xu, D. Wang, and B. Zhang, "Using battery storage for peak shaving and frequency regulation: joint optimization for superlinear gains," *IEEE Transactions on Power Systems*, vol. 33, no. 3, pp. 2882–2894, 2017.

[50] O. Vilppo, A. Rautiainen, J. Rekola, J. Markkula, K. Vuorilehto, and P. Järventausta, "Profitable multi-use of battery energy storage in outage mitigation and as frequency reserve," *International Review of Electrical Engineering*, vol. 13, no. 3, pp. 185–194, 2018.

[51] E. Ghiani, S. Mocci, G. Celli, and F. Pilo, "Increasing the flexible use of hydro pumping storage for maximizing the exploitation of res in sardinia," 2014.

[52] A. J. van MEERWIJK, R. M. Benders, A. Davila-Martinez, and G. A. Laugs, "Swiss pumped hydro storage potential for germany's electricity system under high penetration of intermittent renewable energy," *Journal of Modern Power Systems and Clean Energy*, vol. 4, no. 4, pp. 542–553, 2016.

[53] T. Anilkumar, S. P. Simon, and N. P. Padhy, "Residential electricity cost minimization model through open well-pico turbine pumped storage system," *Applied energy*, vol. 195, pp. 23–35, 2017.

[54] P. Chaudhary and M. Rizwan, "Energy management supporting high penetration of solar photovoltaic generation for smart grid using solar forecasts and pumped hydro storage system," *Renewable Energy*, vol. 118, pp. 928–946, 2018.

[55] E. Muh and F. Tabet, "Comparative analysis of hybrid renewable energy systems for off-grid applications in southern cameroons," *Renewable energy*, vol. 135, pp. 41–54, 2019.

[56] M. G. Molina, "Energy storage and power electronics technologies: a strong combination to empower the transformation to the smart grid," *Proceedings of the IEEE*, vol. 105, no. 11, pp. 2191–2219, 2017.

[57] C. S. Hearn, M. C. Lewis, S. B. Pratap, *et al.*, "Utilization of optimal control law to size grid-level flywheel energy storage," *IEEE Transactions on Sustainable Energy*, vol. 4, no. 3, pp. 611–618, 2013.

[58] R. Latha, S. Palanivel, and J. Kanakaraj, "Frequency control of microgrid based on compressed air energy storage system," *Distributed Generation & Alternative Energy Journal*, vol. 27, no. 4, pp. 8–19, 2012.

[59] S. M. Patel, P. T. Freeman, and J. R. Wagner, "An electrical microgrid: integration of solar panels, compressed air storage, and a micro-cap gas turbine," in *Dynamic Systems and Control Conference*, American Society of Mechanical Engineers, vol. 46193, 2014, V002T22A003.

[60] G. Alva, Y. Lin, and G. Fang, "An overview of thermal energy storage systems," *Energy*, vol. 144, pp. 341–378, 2018.

[61] E. Guelpa and V. Verda, "Thermal energy storage in district heating and cooling systems: a review," *Applied Energy*, vol. 252, p. 113 474, 2019.

[62] J. Zheng, D. W. Gao, and L. Lin, "Smart meters in smart grid: an overview," in *2013 IEEE Green Technologies Conference (GreenTech)*, IEEE, 2013, pp. 57–64.

[63] V. C. Gungor, D. Sahin, T. Kocak, *et al.*, "Smart grid technologies: communication technologies and standards," *IEEE transactions on Industrial informatics*, vol. 7, no. 4, pp. 529–539, 2011.

[64] F. Saffre and R. Gedge, "Demand-side management for the smart grid," in *2010 IEEE/IFIP Network Operations and Management Symposium Workshops*, IEEE, 2010, pp. 300–303.

[65] L. Gelazanskas and K. A. Gamage, "Demand side management in smart grid: a review and proposals for future direction," *Sustainable Cities and Society*, vol. 11, pp. 22–30, 2014.

[66] Q. Qdr, "Benefits of demand response in electricity markets and recommendations for achieving them," *US Dept. Energy, Washington, DC, USA, Tech. Rep*, vol. 2006, 2006.

[67] A. Conchado and P. Linares, "The economic impact of demand-response programs on power systems. a survey of the state of the art," *Handbook of networks in power systems I*, pp. 281–301, 2012.

[68] M. Lee, O. Aslam, B. Foster, *et al.*, "Assessment of demand response and advanced metering," *Federal Energy Regulatory Commission, Tech. Rep*, 2013.

[69] M. D. Galus, M. G. Vayá, T. Krause, and G. Andersson, "The role of electric vehicles in smart grids," *Wiley Interdisciplinary Reviews: Energy and Environment*, vol. 2, no. 4, pp. 384–400, 2013.

[70] R. Das, Y. Wang, G. Putrus, *et al.*, "Multi-objective techno-economic-environmental optimisation of electric vehicle for energy services," *Applied Energy*, vol. 257, p. 113 965, 2020.

[71] P. Pflaum, M. Alamir, and M. Y. Lamoudi, "Probabilistic energy management strategy for ev charging stations using randomized algorithms," *IEEE Transactions on Control Systems Technology*, vol. 26, no. 3, pp. 1099–1106, 2017.

[72] X. L. Dang, M. Petit, and P. Codani, "Energy optimization in an eco-district with electric vehicles smart charging," in *2015 IEEE Eindhoven PowerTech*, IEEE, 2015, pp. 1–6.

[73] C. Corchero, M. Cruz-Zambrano, F.-J. Heredia, *et al.*, "Optimal energy management for a residential microgrid including a vehicle-to-grid system," *IEEE transactions on smart grid*, vol. 5, no. 4, pp. 2163–2172, 2014.

[74] P. Siano, "Demand response and smart grids—a survey," *Renewable and sustainable energy reviews*, vol. 30, pp. 461–478, 2014.

[75] K. Wang, Z. Ouyang, R. Krishnan, L. Shu, and L. He, "A game theory-based energy management system using price elasticity for smart grids," *IEEE Transactions on Industrial Informatics*, vol. 11, no. 6, pp. 1607–1616, 2015.

[76] M. Javadi, M. Marzband, M. Funsho Akorede, R. Godina, A. Saad Al-Sumaiti, and E. Pouresmaeil, "A centralized smart decision-making hierarchical interactive architecture for multiple home microgrids in retail electricity market," *Energies*, vol. 11, no. 11, p. 3144, 2018.

[77] F. Katiraei, R. Iravani, N. Hatziargyriou, and A. Dimeas, "Microgrids management," *IEEE power and energy magazine*, vol. 6, no. 3, pp. 54–65, 2008.

[78] A. A. Khan, M. Naeem, M. Iqbal, S. Qaisar, and A. Anpalagan, "A compendium of optimization objectives, constraints, tools and algorithms for energy management in microgrids," *Renewable and Sustainable Energy Reviews*, vol. 58, pp. 1664–1683, 2016.

[79] D. Espín-Sarzosa, R. Palma-Behnke, and O. Núñez-Mata, "Energy management systems for microgrids: main existing trends in centralized control architectures," *Energies*, vol. 13, no. 3, p. 547, 2020.

[80] T. Stocker, *Climate change 2013: the physical science basis: Working Group I contribution to the Fifth assessment report of the Intergovernmental Panel on Climate Change*. Cambridge university press, 2014.

[81] Y. Zhang, P. Johansson, and A. S. Kalagasidis, "Assessment of district heating and cooling systems transition with respect to future changes in demand profiles and renewable energy supplies," *Energy Conversion and Management*, vol. 268, p. 116 038, 2022.

[82] D. Connolly, H. Lund, B. V. Mathiesen, *et al.*, "Heat roadmap europe: combining district heating with heat savings to decarbonise the eu energy system," *Energy policy*, vol. 65, pp. 475–489, 2014.

[83] D. Connolly, "Heat roadmap europe: quantitative comparison between the electricity, heating, and cooling sectors for different european countries," *Energy*, vol. 139, pp. 580–593, 2017.

[84] B. Rezaie and M. A. Rosen, "District heating and cooling: review of technology and potential enhancements," *Applied energy*, vol. 93, pp. 2–10, 2012.

[85] ADEME, *Fonds chaleur ademe*. [Online]. Available: https://fondschaleur.ademe.fr/reseau-de-chaleur/ (visited on 12/13/2022).

[86] M. Soltani, F. M. Kashkooli, A. Dehghani-Sanij, *et al.*, "A comprehensive study of geothermal heating and cooling systems," *Sustainable Cities and Society*, vol. 44, pp. 793–818, 2019.

[87] H. H. Thorsteinsson and J. W. Tester, "Barriers and enablers to geothermal district heating system development in the united states," *Energy Policy*, vol. 38, no. 2, pp. 803–813, 2010.

[88] M. A. Alkan, A. Keçebaş, and N. Yamankaradeniz, "Exergoeconomic analysis of a district heating system for geothermal energy using specific exergy cost method," *Energy*, vol. 60, pp. 426–434, 2013.

[89] R. Zeh, B. Ohlsen, D. Philipp, *et al.*, "Large-scale geothermal collector systems for 5th generation district heating and cooling networks," *Sustainability*, vol. 13, no. 11, p. 6035, 2021.

[90] I. B. Hassine and U. Eicker, "Impact of load structure variation and solar thermal energy integration on an existing district heating network," *Applied Thermal Engineering*, vol. 50, no. 2, pp. 1437–1446, 2013.

[91] T. Urbaneck, T. Oppelt, B. Platzer, *et al.*, "Solar district heating in east germany–transformation in a cogeneration dominated city," *Energy Procedia*, vol. 70, pp. 587–594, 2015.

[92] N. Perez-Mora, F. Bava, M. Andersen, *et al.*, "Solar district heating and cooling: a review," *International Journal of Energy Research*, vol. 42, no. 4, pp. 1419–1441, 2018.

[93] T. Ge, R. Wang, Z. Xu, *et al.*, "Solar heating and cooling: present and future development," *Renewable Energy*, vol. 126, pp. 1126–1140, 2018.

[94] D. Su, Q. Zhang, G. Wang, and H. Li, "Market analysis of natural gas for district heating in china," *Energy Procedia*, vol. 75, pp. 2713–2717, 2015.

[95] T. J. Lindroos, E. Mäki, K. Koponen, I. Hannula, J. Kiviluoma, and J. Raitila, "Replacing fossil fuels with bioenergy in district heating–comparison of technology options," *Energy*, vol. 231, p. 120 799, 2021.

[96] K. Ericsson and L. J. Nilsson, "Assessment of the potential biomass supply in europe using a resource-focused approach," *Biomass and bioenergy*, vol. 30, no. 1, pp. 1–15, 2006.

[97] J. E. Nielsen and P. A. Sørensen, "Renewable district heating and cooling technologies with and without seasonal storage," in *Renewable heating and cooling*, Elsevier, 2016, pp. 197–220.

[98] S. Werner, "District heating and cooling in sweden," *Energy*, vol. 126, pp. 419–429, 2017.

[99] O. Eriksson, G. Finnveden, T. Ekvall, and A. Björklund, "Life cycle assessment of fuels for district heating: a comparison of waste incineration, biomass-and natural gas combustion," *Energy policy*, vol. 35, no. 2, pp. 1346–1362, 2007.

[100] M. Cordioli, S. Vincenzi, and G. A. De Leo, "Effects of heat recovery for district heating on waste incineration health impact: a simulation study in northern italy," *Science of the total environment*, vol. 444, pp. 369–380, 2013.

[101] A. S. Pratiwi and E. Trutnevyte, "Decision paths to reduce costs and increase economic impact of geothermal district heating in geneva, switzerland," *Applied Energy*, vol. 322, p. 119 431, 2022.

[102] M. J. Castaldi, J. LeBlanc, and A. Licata, "The case for waste to energy," *Mechanical Engineering*, vol. 144, no. 4, pp. 34–39, 2022.

[103] K. Holmgren, "Role of a district-heating network as a user of waste-heat supply from various sources–the case of göteborg," *Applied energy*, vol. 83, no. 12, pp. 1351–1367, 2006.

[104] H. Fang, J. Xia, K. Zhu, Y. Su, and Y. Jiang, "Industrial waste heat utilization for low temperature district heating," *Energy policy*, vol. 62, pp. 236–246, 2013.

[105] M. Morandin, R. Hackl, and S. Harvey, "Economic feasibility of district heating delivery from industrial excess heat: a case study of a swedish petrochemical cluster," *Energy*, vol. 65, pp. 209–220, 2014.

[106] J. Ivner and S. B. Viklund, "Effect of the use of industrial excess heat in district heating on greenhouse gas emissions: a systems perspective," *Resources, Conservation and Recycling*, vol. 100, pp. 81–87, 2015.

[107] H. Fang, J. Xia, and Y. Jiang, "Key issues and solutions in a district heating system using low-grade industrial waste heat," *Energy*, vol. 86, pp. 589–602, 2015.

[108] A. Lake, B. Rezaie, and S. Beyerlein, "Review of district heating and cooling systems for a sustainable future," *Renewable and Sustainable Energy Reviews*, vol. 67, pp. 417–425, 2017.

[109] M. G. Salameh, "Oil crises, historical perspective," 2004.

[110] R. Thévenot, "History of refrigeration throughout the world," 1979.

[111] S. Werner, "District heating and cooling," 2013.

[112] H. Lund, S. Werner, R. Wiltshire, *et al.*, "4th generation district heating (4gdh): integrating smart thermal grids into future sustainable energy systems," *Energy*, vol. 68, pp. 1–11, 2014.

[113] U. N. E. P. O. Secretariat, *Handbook for the Montreal protocol on substances that deplete the ozone layer*. UNEP/Earthprint, 2006.

[114] F. Polonara, L. Kuijpers, and R. Peixoto, "Potential impacts of the montreal protocol kigali amendment to the choice of refrigerant alternatives," *Int J Heat Technol*, vol. 35, no. 1, pp. 1–8, 2017.

[115] D. Prando, A. Prada, F. Ochs, A. Gasparella, and M. Baratieri, "Analysis of the energy and economic impact of cost-optimal buildings refurbishment on district heating systems," *Science and Technology for the Built Environment*, vol. 21, no. 6, pp. 876–891, 2015.

[116] A. Jodeiri, M. Goldsworthy, S. Buffa, and M. Cozzini, "Role of sustainable heat sources in transition towards fourth generation district heating–a review," *Renewable and Sustainable Energy Reviews*, vol. 158, p. 112 156, 2022.

[117] P. A. Østergaard, S. Werner, A. Dyrelund, *et al.*, "The four generations of district cooling-a categorization of the development in district cooling from origin to future prospect," *Energy*, vol. 253, p. 124 098, 2022.

[118] H. Lund, N. Duic, P. A. Østergaard, and B. V. Mathiesen, *Smart energy systems and 4th generation district heating*, 2016.

[119] S. Fabozzi, G. De Luca, and L. Vanoli, "Fourth generation district heating and cooling," in *Polygeneration Systems*, Elsevier, 2022, pp. 323–350.

[120] B. van der Heijde, A. Vandermeulen, R. Salenbien, and L. Helsen, "Integrated optimal design and control of fourth generation district heating networks with thermal energy storage," *Energies*, vol. 12, no. 14, p. 2766, 2019.

[121] E. Commission, *Low temperature, high exergy district heating and cooling networks*, 2015.

[122] S. Buffa, M. Cozzini, M. D'Antoni, M. Baratieri, and R. Fedrizzi, "5th generation district heating and cooling systems: a review of existing cases in europe," *Renewable and Sustainable Energy Reviews*, vol. 104, pp. 504–522, 2019.

[123] I. P. Pattijn and A. Baumans, "Fifth-generation thermal grids and heat pumps: a pilot project in leuven, belgium," *HPT Mag*, vol. 35, no. 2, pp. 53–57, 2017.

[124] F. Bünning, M. Wetter, M. Fuchs, and D. Müller, "Bidirectional low temperature district energy systems with agent-based control: performance comparison and operation optimization," *Applied Energy*, vol. 209, pp. 502–515, 2018.

[125] M. Abugabbara, S. Javed, H. Bagge, and D. Johansson, "Bibliographic analysis of the recent advancements in modeling and co-simulating the fifth-generation district heating and cooling systems," *Energy and Buildings*, vol. 224, p. 110 260, 2020.

[126] H. Lund, P. A. Østergaard, T. B. Nielsen, *et al.*, "Perspectives on fourth and fifth generation district heating," *Energy*, vol. 227, p. 120 520, 2021.

[127] E. Commission, *Energy 2020: A strategy for competitive, sustainable and secure energy*. 2011.

[128] D. Connolly, H. Lund, B. V. Mathiesen, *et al.*, "Smart energy systems: holistic and integrated energy systems for the era of 100% renewable energy," 2013.

[129] P. Mancarella, "Mes (multi-energy systems): an overview of concepts and evaluation models," *Energy*, vol. 65, pp. 1–17, 2014.

[130] S. Howell, Y. Rezgui, J.-L. Hippolyte, B. Jayan, and H. Li, "Towards the next generation of smart grids: semantic and holonic multi-agent management of distributed energy resources," *Renewable and Sustainable Energy Reviews*, vol. 77, pp. 193–214, 2017.

[131] J. P. Catalão, *Electric power systems: advanced forecasting techniques and optimal generation scheduling*. CRC press, 2017.

[132] P. Favre-Perrod, "A vision of future energy networks," in *2005 IEEE power engineering society inaugural conference and exposition in Africa*, IEEE, 2005, pp. 13–17.

[133] W. Leontief, *Input-output economics*. Oxford University Press, 1986.

[134] M. Geidl and G. Andersson, "Optimal power flow of multiple energy carriers," *IEEE Transactions on power systems*, vol. 22, no. 1, pp. 145–155, 2007.

[135] M. Geidl, G. Koeppel, P. Favre-Perrod, B. Klockl, G. Andersson, and K. Frohlich, "Energy hubs for the future," *IEEE power and energy magazine*, vol. 5, no. 1, pp. 24–30, 2006.

[136] M. Mohammadi, Y. Noorollahi, B. Mohammadi-Ivatloo, and H. Yousefi, "Energy hub: from a model to a concept–a review," *Renewable and Sustainable Energy Reviews*, vol. 80, pp. 1512–1527, 2017.

[137] A. Sheikhi, M. Rayati, S. Bahrami, A. M. Ranjbar, and S. Sattari, "A cloud computing framework on demand side management game in smart energy hubs," *International Journal of Electrical Power & Energy Systems*, vol. 64, pp. 1007–1016, 2015.

[138] M. Rayati, A. Sheikhi, and A. M. Ranjbar, "Optimising operational cost of a smart energy hub, the reinforcement learning approach," *International Journal of Parallel, Emergent and Distributed Systems*, vol. 30, no. 4, pp. 325–341, 2015.

[139] H. Lund, *Renewable energy systems: a smart energy systems approach to the choice and modeling of 100% renewable solutions*. Academic Press, 2014.

[140] H. Lund, P. A. Østergaard, D. Connolly, and B. V. Mathiesen, "Smart energy and smart energy systems," *Energy*, vol. 137, pp. 556–565, 2017.

[141] G. Guerassimoff and L. Adegnon, *L'hydrogène: un vecteur pour la transition énergétique*, 2020.

[142] W. Kempton and J. Tomić, "Vehicle-to-grid power implementation: from stabilizing the grid to supporting large-scale renewable energy," *Journal of power sources*, vol. 144, no. 1, pp. 280–294, 2005.

[143] G. Mendes, C. Ioakimidis, and P. Ferrão, "On the planning and analysis of integrated community energy systems: a review and survey of available tools," *Renewable and Sustainable Energy Reviews*, vol. 15, no. 9, pp. 4836–4854, 2011.

[144] L.-N. Liu and G.-H. Yang, "Distributed optimal energy management for integrated energy systems," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 10, pp. 6569–6580, 2022.

[145] J. Wu, "Drivers and state-of-the-art of integrated energy systems in europe," *Automation of Electric Power Systems*, vol. 40, no. 5, pp. 1–7, 2016.

[146] B. P. Koirala, E. Koliou, J. Friege, R. A. Hakvoort, and P. M. Herder, "Energetic communities for community energy: a review of key issues and trends shaping integrated community energy systems," *Renewable and Sustainable Energy Reviews*, vol. 56, pp. 722–744, 2016.

[147] Y. Xu, C. Yan, H. Liu, J. Wang, Z. Yang, and Y. Jiang, "Smart energy systems: a critical review on design and operation optimization," *Sustainable Cities and Society*, vol. 62, p. 102 369, 2020.

[148] W. Gu, Z. Wu, R. Bo, *et al.*, "Modeling, planning and optimal energy management of combined cooling, heating and power microgrid: a review," *International Journal of Electrical Power & Energy Systems*, vol. 54, pp. 26–37, 2014.

[149] S. D. Beigvand, H. Abdi, and M. La Scala, "A general model for energy hub economic dispatch," *Applied Energy*, vol. 190, pp. 1090–1111, 2017.

[150] I. Sarbu, M. Mirza, and E. Crasmareanu, "A review of modelling and optimisation techniques for district heating systems," *International Journal of Energy Research*, vol. 43, no. 13, pp. 6572–6598, 2019.

[151] B. Talebi, P. A. Mirzaei, A. Bastani, and F. Haghighat, "A review of district heating systems: modeling and optimization," *Frontiers in Built Environment*, vol. 2, p. 22, 2016.

[152] M. Sameti and F. Haghighat, "Optimization approaches in district heating and cooling thermal network," *Energy and Buildings*, vol. 140, pp. 121–130, 2017.

[153] A. Benonysson, B. Bøhm, and H. F. Ravn, "Operational optimization in a district heating system," *Energy conversion and management*, vol. 36, no. 5, pp. 297–314, 1995.

[154] S. Grosswindhager, A. Voigt, and M. Kozek, "Predictive control of district heating network using fuzzy dmc," in *2012 Proceedings of International Conference on Modelling, Identification and Control*, IEEE, 2012, pp. 241–246.

[155] G. Sandou, S. Font, S. Tebbani, *et al.*, "Predictive control of a complex district heating network," in *IEEE conference on decision and control*, Citeseer, vol. 44, 2005, p. 7372.

[156] S. Idowu, C. Åhlund, and O. Schelen, "Machine learning in district heating system energy optimization," in *2014 IEEE International Conference on Pervasive Computing and Communication Workshops (PERCOM WORKSHOPS)*, IEEE, 2014, pp. 224–227.

[157] M. Vesterlund and J. Dahl, "A method for the simulation and optimization of district heating systems with meshed networks," *Energy conversion and management*, vol. 89, pp. 555–567, 2015.

[158] E. Guelpa, C. Toro, A. Sciacovelli, R. Melli, E. Sciubba, and V. Verda, "Optimal operation of large district heating networks through fast fluid-dynamic simulation," *Energy*, vol. 102, pp. 586–595, 2016.

[159] L. Giraud, M. Merabet, R. Baviere, and M. Vallée, "Optimal control of district heating systems using dynamic simulation and mixed integer linear programming," in *Proceedings of the 12th International Modelica Conference, Prague, Czech Republic, May 15-17, 2017*, Linköping University Electronic Press, 2017, pp. 141–150.

[160] M. Vesterlund, A. Toffolo, and J. Dahl, "Optimization of multi-source complex district heating network, a case study," *Energy*, vol. 126, pp. 53–63, 2017.

[161] B. Morvaj, R. Evins, and J. Carmeliet, "Optimising urban energy systems: simultaneous system sizing, operation and district heating network layout," *Energy*, vol. 116, pp. 619–636, 2016.

[162] Y. Li, Y. Rezgui, and H. Zhu, "District heating and cooling optimization and enhancement–towards integration of renewables, storage and smart grid," *Renewable and Sustainable Energy Reviews*, vol. 72, pp. 281–294, 2017.

[163] T. Zhang, J. Luo, P. Chen, and J. Liu, "Flow rate control in smart district heating systems using deep reinforcement learning," *arXiv preprint arXiv:1912.05313*, 2019.

[164] G. Schweiger, P.-O. Larsson, F. Magnusson, P. Lauenburg, and S. Velut, "District heating and cooling systems–framework for modelica-based simulation and dynamic optimization," *Energy*, vol. 137, pp. 566–578, 2017.

[165] M. Leśko, W. Bujalski, and K. Futyma, "Operational optimization in district heating systems with the use of thermal energy storage," *Energy*, vol. 165, pp. 902–915, 2018.

[166] K. M. Powell, J. S. Kim, W. J. Cole, *et al.*, "Thermal energy storage to minimize cost and improve efficiency of a polygeneration district energy system in a real-time electricity market," *Energy*, vol. 113, pp. 52–63, 2016.

[167] S. J. Cox, D. Kim, H. Cho, and P. Mago, "Real time optimal control of district cooling system with thermal energy storage using neural networks," *Applied energy*, vol. 238, pp. 466–480, 2019.

[168] L. I. Minchala-Avila, L. E. Garza-Castañón, A. Vargas-Martínez, and Y. Zhang, "A review of optimal control techniques applied to the energy management and control of microgrids," *Procedia Computer Science*, vol. 52, pp. 780–787, 2015.

[169] S. Twaha and M. A. Ramli, "A review of optimization approaches for hybrid distributed energy generation systems: off-grid and grid-

connected systems," *Sustainable Cities and Society*, vol. 41, pp. 320–331, 2018.

[170] M. Pourbehzadi, T. Niknam, J. Aghaei, G. Mokryani, M. Shafie-khah, and J. P. Catalão, "Optimal operation of hybrid ac/dc microgrids under uncertainty of renewable energy resources: a comprehensive review," *International Journal of Electrical Power & Energy Systems*, vol. 109, pp. 139–159, 2019.

[171] P. Mancarella, "Multi-energy systems: an overview of concepts and evaluation models," *Invited Paper, Energy, under Review*,

[172] M. C. Bozchalui, C. A. Cañizares, and K. Bhattacharya, "Optimal energy management of greenhouses in smart grids," *IEEE transactions on smart grid*, vol. 6, no. 2, pp. 827–835, 2014.

[173] C. Weber, F. Maréchal, and D. Favrat, "Design and optimization of district energy systems," in *Computer aided chemical engineering*, vol. 24, Elsevier, 2007, pp. 1127–1132.

[174] V. Hosseinnezhad, M. Rafiee, M. Ahmadian, and P. Siano, "Optimal day-ahead operational planning of microgrids," *Energy Conversion and Management*, vol. 126, pp. 142–157, 2016.

[175] W. Ma, S. Fang, and G. Liu, "Hybrid optimization method and seasonal operation strategy for distributed energy system integrating cchp, photovoltaic and ground source heat pump," *Energy*, vol. 141, pp. 1439–1455, 2017.

[176] R. Bellman, "A markovian decision process," *Journal of mathematics and mechanics*, pp. 679–684, 1957.

[177] A. U. Udom, "A markov decision process approach to optimal control of a multi-level hierarchical manpower system," *CBN Journal of Applied Statistics*, vol. 4, no. 2, pp. 31–49, 2013.

[178] M. Wiering and M. Van Otterlo, *Reinforcement learning*. Springer, 2012, vol. 12.

[179] E. A. Feinberg and A. Shwartz, *Handbook of Markov decision processes: methods and applications*. Springer Science & Business Media, 2012, vol. 40.

[180] F. Lauri, G. Basso, J. Zhu, R. Roche, V. Hilaire, and A. Koukam, "Managing power flows in microgrids using multi-agent reinforcement learning," *Agent Technologies in Energy Systems (ATES)*, 2013.

[181] M. L. Puterman, *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

[182] C. Boutilier, T. Dean, and S. Hanks, "Decision-theoretic planning: structural assumptions and computational leverage," *Journal of Artificial Intelligence Research*, vol. 11, pp. 1–94, 1999.

[183] O. Sigaud and O. Buffet, *Markov decision processes in artificial intelligence*. John Wiley & Sons, 2013.

[184] R. G. Di Wu, F. lavet Vincent, P. Doina, and B. Benoit, "Optimizing home energy management and electric vehicle charging with reinforcement learning," *Proceedings of the 16th Adaptive Learning Agents*, 2018.

[185] J. Silvente Saiz, "Improving the tactical and operational decision making procedures in chemical supply chains," 2016.

[186] A. C. L. Hernández, "Energy management systems for microgrids equipped with renewable energy sources and battery units," Ph.D. dissertation, Aalborg Universitetsforlag, 2017.

[187] F. Y. Melhem, "Optimization methods and energy management in" smart grids"," Ph.D. dissertation, 2018.

[188] S. Teleke, M. E. Baran, S. Bhattacharya, and A. Q. Huang, "Rule-based control of battery energy storage for dispatching intermittent renewable sources," *IEEE Transactions on Sustainable Energy*, vol. 1, no. 3, pp. 117–124, 2010.

[189] K. Büdenbender, M. Braun, A. Schmiegel, D. Magnor, and J. C. Marcel, "Improving pv-integration into the distribution grid-contribution of multifunctional pv-battery systems to stabilised system operation," in *25th European Photovoltaic Solar Energy Conference and Exhibition*, 2010, pp. 4839–4845.

[190] F. Braam, R. Hollinger, M. L. Engesser, S. Müller, R. Kohrs, and C. Wittwer, "Peak shaving with photovoltaic-battery systems," in *IEEE PES Innovative Smart Grid Technologies, Europe*, IEEE, 2014, pp. 1–5.

[191] G. Dantzig, "Linear programming and extensions," in *Linear programming and extensions*, Princeton university press, 2016.

[192] S. J. Wright, *Primal-dual interior-point methods*. SIAM, 1997.

[193] D. Alkaya, S. Vasantharajan, and L. T. Biegler, *Successive quadratic programming: applications in the process industry.* 2009.

[194] A. H. Land and A. G. Doig, "An automatic method for solving discrete programming problems," in *50 Years of Integer Programming 1958-2008*, Springer, 2010, pp. 105–132.

[195] R. E. Gomory, "Outline of an algorithm for integer solutions to linear programs and an algorithm for the mixed integer problem," in *50 Years of Integer Programming 1958-2008*, Springer, 2010, pp. 77–103.

[196] M. Tawarmalani and N. V. Sahinidis, "A polyhedral branch-and-cut approach to global optimization," *Mathematical programming*, vol. 103, no. 2, pp. 225–249, 2005.

[197] J. Silvente and L. G. Papageorgiou, "An milp formulation for the optimal management of microgrids with task interruptions," *Applied energy*, vol. 206, pp. 1131–1146, 2017.

[198] S. Aslam, A. Khalid, and N. Javaid, "Towards efficient energy management in smart grids considering microgrids with day-ahead energy forecasting," *Electric Power Systems Research*, vol. 182, p. 106 232, 2020.

[199] L. Giraud, R. Bavière, M. Vallée, and C. Paulus, "Presentation, validation and application of the districtheating modelica library," in *Proceedings of the 11th International Modelica Conference, Versailles, France, September 21-23, 2015*, Linköping University Electronic Press, 2015, pp. 79–88.

[200] T. Arnold, R. Henrion, A. Möller, and S. Vigerske, "A mixed-integer stochastic nonlinear optimization problem with joint probabilistic constraints," 2013.

[201] D. P. Bertsekas, *Dynamic programming and optimal control*. Athena scientific Belmont, MA, 1995, vol. 1.

[202] S. M. Ross, *Introduction to stochastic dynamic programming*. Academic press, 2014.

[203] K. K. Haugen, *Stochastic Dynamic Programming*. Scandinavian University Press (Universitetsforlaget), 2016.

[204] F. W. Glover and G. A. Kochenberger, *Handbook of metaheuristics*. Springer Science & Business Media, 2006, vol. 57.

[205] P. Cortés, J. Muñuzuri, I. Domínguez, *et al.*, "Genetic algorithms to optimize the operating costs of electricity and heating networks in buildings considering distributed energy generation and storage," *Computers & Operations Research*, vol. 96, pp. 157–172, 2018.

[206] A. Lorestani and M. Ardehali, "Optimal integration of renewable energy sources for autonomous tri-generation combined cooling, heating and power system based on evolutionary particle swarm optimization algorithm," *Energy*, vol. 145, pp. 839–855, 2018.

[207] M. R. Mozafar, M. H. Moradi, and M. H. Amini, "A simultaneous approach for optimal allocation of renewable energy sources and electric vehicle charging stations in smart grids based on improved ga-pso algorithm," *Sustainable cities and society*, vol. 32, pp. 627–637, 2017.

[208] E. Mocanu, "Machine learning applied to smart grids," *Energy*, vol. 2, no. 3, p. 4, 2017.

[209] C. Szepesvári, "Algorithms for reinforcement learning," *Synthesis lectures on artificial intelligence and machine learning*, vol. 4, no. 1, pp. 1–103, 2010.

[210] J. R. Vázquez-Canteli and Z. Nagy, "Reinforcement learning for demand response: a review of algorithms and modeling techniques," *Applied energy*, vol. 235, pp. 1072–1089, 2019.

[211] C. J. Watkins and P. Dayan, "Q-learning," *Machine learning*, vol. 8, no. 3-4, pp. 279–292, 1992.

[212] F.-D. Li, M. Wu, Y. He, and X. Chen, "Optimal control in microgrid using multi-agent reinforcement learning," *ISA transactions*, vol. 51, no. 6, pp. 743–751, 2012.

[213] Y. Zhao, D. Zeng, M. A. Socinski, and M. R. Kosorok, "Reinforcement learning strategies for clinical trials in nonsmall cell lung cancer," *Biometrics*, vol. 67, no. 4, pp. 1422–1433, 2011.

[214] S. V. Bruno, L. A. Moraes, and W. de Oliveira, "Optimization techniques for the brazilian natural gas network planning problem," *Energy Systems*, vol. 8, no. 1, pp. 81–101, 2017.

[215] E. N. Pistikopoulos, "Perspectives in multiparametric programming and explicit model predictive control," 2009.

[216] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. Scokaert, "Constrained model predictive control: stability and optimality," *Automatica*, vol. 36, no. 6, pp. 789–814, 2000.

[217] M. Y. Lamoudi, "Distributed model predictive control for energy management in buildings," Ph.D. dissertation, Université de Grenoble, 2012.

[218] W. H. Kwon and S. H. Han, *Receding horizon control: model predictive control for state models*. Springer Science & Business Media, 2006.

[219] G. M. Kopanos and E. N. Pistikopoulos, "Reactive scheduling by a multiparametric programming rolling horizon framework: a case of a network of combined heat and power units," *Industrial & Engineering Chemistry Research*, vol. 53, no. 11, pp. 4366–4386, 2014.

[220] A. Shapiro, D. Dentcheva, and A. Ruszczyński, *Lectures on stochastic programming: modeling and theory*. SIAM, 2009.

[221] W. de Oliveira, *A minicourse on stochastic programming*, http://www.oliveira.mat.br/teaching, 2018.

[222] S. Ahmed, A. Shapiro, and E. Shapiro, "The sample average approximation method for stochastic programs with integer recourse," *Submitted for publication*, pp. 1–24, 2002.

[223] A. Ben-Tal, L. El Ghaoui, and A. Nemirovski, *Robust optimization*. Princeton University Press, 2009, vol. 28.

[224] R. E. Bellman and L. A. Zadeh, "Decision-making in a fuzzy environment," *Management science*, vol. 17, no. 4, B–141, 1970.

[225] H.-J. Zimmermann, *Fuzzy set theory—and its applications*. Springer Science & Business Media, 2011.

[226] N. V. Sahinidis, "Optimization under uncertainty: state-of-the-art and opportunities," *Computers & Chemical Engineering*, vol. 28, no. 6-7, pp. 971–983, 2004.

[227] R. R. Negenborn, B. De Schutter, M. A. Wiering, and H. Hellendoorn, "Learning-based model predictive control for markov decision processes," *IFAC Proceedings Volumes*, vol. 38, no. 1, pp. 354–359, 2005.

[228] F. Smarra, A. Jain, T. de Rubeis, D. Ambrosini, A. D'Innocenzo, and R. Mangharam, "Data-driven model predictive control using random forests for building energy optimization and climate control," *Applied energy*, vol. 226, pp. 1252–1272, 2018.

[229] D. Silver, *Deep reinforcement learning, a tutorial*, https://icml.cc/2016/tutorials/deep_rl_tutorial.pdf, 2016.

[230] B. Mahesh, "Machine learning algorithms-a review," *International Journal of Science and Research (IJSR).[Internet]*, vol. 9, pp. 381–386, 2020.

[231] T. O. Ayodele, "Types of machine learning algorithms," *New advances in machine learning*, vol. 3, pp. 19–48, 2010.

[232] Y. Li, "Deep reinforcement learning: an overview," *arXiv preprint arXiv:1701.07274*, 2017.

[233] H.-x. Zhao and F. Magoulès, "A review on the prediction of building energy consumption," *Renewable and Sustainable Energy Reviews*, vol. 16, no. 6, pp. 3586–3592, 2012.

[234] Y. Chakhchoukh, P. Panciatici, and L. Mili, "Electric load forecasting based on statistical robust methods," *IEEE Transactions on Power Systems*, vol. 26, no. 3, pp. 982–991, 2010.

[235] M. Mohandes, "Support vector machines for short-term electrical load forecasting," *International Journal of Energy Research*, vol. 26, no. 4, pp. 335–345, 2002.

[236] P.-F. Pai and W.-C. Hong, "Support vector machines with simulated annealing algorithms in electricity load forecasting," *Energy Conversion and Management*, vol. 46, no. 17, pp. 2669–2688, 2005.

[237] J. Yang, Y. Tang, and H. Duan, "Application of fuzzy support vector machine in short-term power load forecasting," *Journal of Cases on Information Technology (JCIT)*, vol. 24, no. 5, pp. 1–10, 2022.

[238] M. D. C. Ruiz-Abellón, A. Gabaldón, and A. Guillamón, "Load forecasting for a campus university using ensemble methods based on regression trees," *Energies*, vol. 11, no. 8, p. 2038, 2018.

[239] S. Kumar, S. Mishra, and S. Gupta, "Short term load forecasting using ann and multiple linear regression," in *2016 second international conference on computational intelligence & communication technology (cict)*, IEEE, 2016, pp. 184–186.

[240] S. Bouktif, A. Fiaz, A. Ouni, and M. A. Serhani, "Optimal deep learning lstm model for electric load forecasting using feature selection and genetic algorithm: comparison with machine learning approaches," *Energies*, vol. 11, no. 7, p. 1636, 2018.

[241] G. Suryanarayana, J. Lago, D. Geysen, P. Aleksiejuk, and C. Johansson, "Thermal load forecasting in district heating networks using deep learning and advanced feature selection methods," *Energy*, vol. 157, pp. 141–149, 2018.

[242] M. R. Alam, M. St-Hilaire, and T. Kunz, "Computational methods for residential energy cost optimization in smart grids: a survey," *ACM Computing Surveys (CSUR)*, vol. 49, no. 1, pp. 1–34, 2016.

[243] A. Vandermeulen, B. van der Heijde, and L. Helsen, "Controlling district heating and cooling networks to unlock flexibility: a review," *Energy*, vol. 151, pp. 103–115, 2018.

[244] E. Klaassen, "Demand response benefits from a power system perspective: methodologies and evaluation of field tests," 2016.

[245] X. Chen, G. Qu, Y. Tang, S. Low, and N. Li, "Reinforcement learning for selective key applications in power systems: recent advances and future challenges," *IEEE Transactions on Smart Grid*, 2022.

[246] Q. Yang, G. Wang, A. Sadeghi, G. B. Giannakis, and J. Sun, "Two-timescale voltage control in distribution grids using deep reinforcement learning," *IEEE Transactions on Smart Grid*, vol. 11, no. 3, pp. 2313–2323, 2019.

[247] J. Duan, D. Shi, R. Diao, *et al.*, "Deep-reinforcement-learning-based autonomous voltage control for power grid operations," *IEEE Transactions on Power Systems*, vol. 35, no. 1, pp. 814–817, 2019.

[248] D. Cao, W. Hu, J. Zhao, Q. Huang, Z. Chen, and F. Blaabjerg, "A multi-agent deep reinforcement learning based voltage regulation using coordinated pv inverters," *IEEE Transactions on Power Systems*, vol. 35, no. 5, pp. 4120–4123, 2020.

[249] Y. Bengio, I. Goodfellow, and A. Courville, *Deep learning*. MIT press Cambridge, MA, USA, 2017, vol. 1.

[250] A. LeNail, "Nn-svg: publication-ready neural network architecture schematics.," *J. Open Source Softw.*, vol. 4, no. 33, p. 747, 2019.

[251] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[252] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," in *Thirty-first AAAI conference on artificial intelligence*, 2017.

[253] I. Goodfellow, J. Pouget-Abadie, M. Mirza, *et al.*, "Generative adversarial networks," *Communications of the ACM*, vol. 63, no. 11, pp. 139–144, 2020.

[254] A. H. Klopf, *Brain function and adaptive systems: a heterostatic theory*. Air Force Cambridge Research Laboratories, Air Force Systems Command, United . . ., 1972.

[255] A. H. Klopf, "A comparison of natural and artificial intelligence," *ACM SIGART Bulletin*, no. 52, pp. 11–13, 1975.

[256] A. Turing, "Intelligent machinery (1948)," *B. Jack Copeland*, p. 395, 2004.

[257] D. Bertsekas and J. N. Tsitsiklis, *Neuro-dynamic programming*. Athena Scientific, 1996.

[258] D. Görges, "Relations between model predictive control and reinforcement learning," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 4920–4928, 2017.

[259] M. Minsky, "Steps toward artificial intelligence," *Proceedings of the IRE*, vol. 49, no. 1, pp. 8–30, 1961.

[260] P. J. Werbos, "Advanced forecasting methods for global crisis warning and models of intelligence," *General System Yearbook*, pp. 25–38, 1977.

[261] C. J. C. H. Watkins, "Learning from delayed rewards," 1989.

[262] A. Géron, "Hands-on machine learning with scikit-learn and tensorflow: concepts," *Tools, and Techniques to build intelligent systems*, 2017.

[263] B. Farley and W. d. Clark, "Simulation of self-organizing systems by digital computer," *Transactions of the IRE Professional Group on Information Theory*, vol. 4, no. 4, pp. 76–84, 1954.

[264] J. Schmidhuber, "Deep learning in neural networks: an overview," *Neural networks*, vol. 61, pp. 85–117, 2015.

[265] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller, "Deterministic policy gradient algorithms," in *International conference on machine learning*, Pmlr, 2014, pp. 387–395.

[266] E. Kuznetsova, Y.-F. Li, C. Ruiz, E. Zio, G. Ault, and K. Bell, "Reinforcement learning for microgrid energy management," *Energy*, vol. 59, pp. 133–146, 2013.

[267] D. Ernst, M. Glavic, F. Capitanescu, and L. Wehenkel, "Reinforcement learning versus model predictive control: a comparison on a power system problem," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 39, no. 2, pp. 517–529, 2008.

[268] V. François-Lavet, D. Taralla, D. Ernst, and R. Fonteneau, "Deep reinforcement learning solutions for energy microgrids management," in *European Workshop on Reinforcement Learning (EWRL 2016)*, 2016.

[269] T. Hirata, D. B. Malla, K. Sakamoto, K. Yamaguchi, Y. Okada, and T. Sogabe, "Smart grid optimization by deep reinforcement learning over discrete and continuous action space," *Bulletin of Networking, Computing, Systems, and Software*, vol. 8, no. 1, pp. 19–22, 2019.

[270] M. Glavic, R. Fonteneau, and D. Ernst, "Reinforcement learning for electric power system decision and control: past considerations and perspectives," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 6918–6927, 2017.

[271] D. Zhang, X. Han, and C. Deng, "Review on the research and practice of deep learning and reinforcement learning in smart grids," *CSEE Journal of Power and Energy Systems*, vol. 4, no. 3, pp. 362–370, 2018.

[272] J. Qin, X. Han, G. Liu, S. Wang, W. Li, and Z. Jiang, "Wind and storage cooperative scheduling strategy based on deep reinforcement learning algorithm," in *Journal of Physics: Conference Series*, IOP Publishing, vol. 1213, 2019, p. 032 002.

[273] S. Lin, H. Yu, and H. Chen, "On-line optimization of microgrid operating cost based on deep reinforcement learning," in *IOP Conference Series: Earth and Environmental Science*, IOP Publishing, vol. 701, 2021, p. 012 084.

[274] T. P. Lillicrap, J. J. Hunt, A. Pritzel, *et al.*, "Continuous control with deep reinforcement learning," *arXiv preprint arXiv:1509.02971*, 2015.

[275] G. Gao, J. Li, and Y. Wen, "Energy-efficient thermal comfort control in smart buildings via deep reinforcement learning," *arXiv preprint arXiv:1901.04693*, 2019.

[276] E. Mocanu, D. C. Mocanu, P. H. Nguyen, *et al.*, "On-line building energy optimization using deep reinforcement learning," *IEEE transactions on smart grid*, vol. 10, no. 4, pp. 3698–3708, 2018.

[277] V. R. Konda and J. N. Tsitsiklis, "Actor-critic algorithms," in *Advances in neural information processing systems*, 2000, pp. 1008–1014.

[278] L. Yu, W. Xie, D. Xie, *et al.*, "Deep reinforcement learning for smart home energy management," *IEEE Internet of Things Journal*, vol. 7, no. 4, pp. 2751–2762, 2019.

[279] Y. Kuang, X. Wang, H. Zhao, Y. Huang, X. Chen, and X. Wang, "Agent-based energy sharing mechanism using deep deterministic policy gradient algorithm," *Energies*, vol. 13, no. 19, p. 5027, 2020.

[280] D. Bousnina and G. Guerassimoff, "Deep reinforcement learning for optimal energy management of multi-energy smart grids," *Lecture Notes in Computer Science*, pp. 15–30, 2022.

[281] S. J. Qin and T. A. Badgwell, "An overview of industrial model predictive control technology," in *AIche symposium series*, New York, NY: American Institute of Chemical Engineers, 1971-c2002., vol. 93, 1997, pp. 232–256.

[282] A. Shapiro, D. Dentcheva, and A. Ruszczyński, *Lectures on stochastic programming: modeling and theory*. SIAM, 2014.

[283] A. Bemporad and M. Morari, "Robust model predictive control: a survey," in *Robustness in identification and control*, Springer, 1999, pp. 207–226.

[284] H.-G. Beyer and B. Sendhoff, "Robust optimization–a comprehensive survey," *Computer methods in applied mechanics and engineering*, vol. 196, no. 33-34, pp. 3190–3218, 2007.

[285] R. HALVGAARD, "Model predictive control for smart energy systems," Ph.D. dissertation, PhD thesis. Technical University of Denmark. Department of Applied …, 2014.

[286] E. C. Kerrigan and J. M. Maciejowski, "Designing model predictive controllers with prioritised constraints and objectives," in *Proceedings. IEEE International Symposium on Computer Aided Control System Design*, IEEE, 2002, pp. 33–38.

[287] R. E. Bixby, "A brief history of linear and mixed-integer programming computation," *Documenta Mathematica*, vol. 2012, pp. 107–121, 2012.

[288] A. Schirrer, M. Brandstetter, I. Leobner, S. Hauer, and M. Kozek, "Nonlinear model predictive control for a heating and cooling system of a low-energy office building," *Energy and Buildings*, vol. 125, pp. 86–98, 2016.

[289] R. Findeisen and F. Allgöwer, "An introduction to nonlinear model predictive control," in *21st Benelux meeting on systems and control*, Veldhoven, vol. 11, 2002, pp. 119–141.

[290] E. Žáčeková, Z. Váňa, and J. Cigler, "Towards the real-life implementation of mpc for an office building: identification issues," *Applied Energy*, vol. 135, pp. 53–62, 2014.

[291] D. Sturzenegger, D. Gyalistras, M. Morari, and R. S. Smith, "Model predictive climate control of a swiss office building: implementation, results, and cost–benefit analysis," *IEEE Transactions on Control Systems Technology*, vol. 24, no. 1, pp. 1–12, 2015.

[292] P. Ferreira, A. Ruano, S. Silva, and E. Conceicao, "Neural networks based predictive control for thermal comfort and energy savings in public buildings," *Energy and buildings*, vol. 55, pp. 238–251, 2012.

[293] K. Zhang, J. Wang, X. Xin, *et al.*, "A survey on learning-based model predictive control: toward path tracking control of mobile platforms," *Applied Sciences*, vol. 12, no. 4, p. 1995, 2022.

[294] J. Mattingley, Y. Wang, and S. Boyd, "Receding horizon control," *IEEE Control Systems Magazine*, vol. 31, no. 3, pp. 52–65, 2011.

[295] E. O'Dwyer, I. Pan, R. Charlesworth, S. Butler, and N. Shah, "Integration of an energy management tool and digital twin for coordination and control of multi-vector smart energy systems," *Sustainable Cities and Society*, vol. 62, p. 102 412, 2020.

[296] M. Killian and M. Kozek, "Ten questions concerning model predictive control for energy efficient buildings," *Building and Environment*, vol. 105, pp. 403–412, 2016.

[297] D. Mariano-Hernández, L. Hernández-Callejo, A. Zorita-Lamadrid, O. Duque-Pérez, and F. S. García, "A review of strategies for building energy management system: model predictive control, demand side management, optimization, and fault detect & diagnosis," *Journal of Building Engineering*, vol. 33, p. 101 692, 2021.

[298] J. Ma, S. J. Qin, B. Li, and T. Salsbury, *Economic model predictive control for building energy systems*. IEEE, 2011.

[299] M. Marzband, H. Alavi, S. S. Ghazimirsaeid, H. Uppal, and T. Fernando, "Optimal energy management system based on stochastic approach for a home microgrid with integrated responsive load demand and energy storage," *Sustainable cities and society*, vol. 28, pp. 256–264, 2017.

[300] C. Lv, H. Yu, P. Li, *et al.*, "Model predictive control based robust scheduling of community integrated energy system with operational flexibility," *Applied energy*, vol. 243, pp. 250–265, 2019.

[301] K. Shan, C. Fan, and J. Wang, "Model predictive control for thermal energy storage assisted large central cooling systems," *Energy*, vol. 179, pp. 916–927, 2019.

[302] A. Parisio, E. Rikos, and L. Glielmo, "Stochastic model predictive control for economic/environmental operation management of microgrids: an experimental case study," *Journal of Process Control*, vol. 43, pp. 24–37, 2016.

[303] G. Gambino, F. Verrilli, D. Meola, *et al.*, "Model predictive control for optimization of combined heat and electric power microgrid," *IFAC Proceedings Volumes*, vol. 47, no. 3, pp. 2201–2206, 2014.

[304] R. Tang, S. Wang, and L. Xu, "An mpc-based optimal control strategy of active thermal storage in commercial buildings during fast demand response events in smart grids," *Energy Procedia*, vol. 158, pp. 2506–2511, 2019.

[305] M. Wytock, N. Moehle, and S. Boyd, "Dynamic energy management with scenario-based robust mpc," in *2017 American Control Conference (ACC)*, IEEE, 2017, pp. 2042–2047.

[306] F. Jorissen, D. Picard, K. Six, and L. Helsen, "Detailed white-box nonlinear model predictive control for scalable building hvac control," in *Modelica Conferences*, 2021, pp. 315–323.

[307] G. Gambino, F. Verrilli, M. Canelli, *et al.*, "Optimal operation of a district heating power plant with thermal energy storage," in *2016 American Control Conference (ACC)*, IEEE, 2016, pp. 2334–2339.

[308] F. Verrilli, S. Srinivasan, G. Gambino, *et al.*, "Model predictive control-based optimal operations of district heating system with thermal energy storage and flexible loads," *IEEE Transactions on Automation Science and Engineering*, vol. 14, no. 2, pp. 547–557, 2016.

[309] F. Verrilli, A. Parisio, and L. Glielmo, "Stochastic model predictive control for optimal energy management of district heating power plants," in *2016 IEEE 55th Conference on Decision and Control (CDC)*, IEEE, 2016, pp. 807–812.

[310] F. Lie-Jensen, A. Aannø, E. Aleksandrova, A. Westli, M. Nielsen, and T. M. Komulainen, "Model predictive control of district heating system," 2018.

[311] I. Cupeiro Figueroa, M. Cimmino, and L. Helsen, "A methodology for long-term model predictive control of hybrid geothermal systems: the shadow-cost formulation," *Energies*, vol. 13, no. 23, p. 6203, 2020.

[312] C. Huang, Y. Zong, S. You, and C. Træholt, "Economic model predictive control for multi-energy system considering hydrogen-thermal-electric dynamics and waste heat recovery of mw-level alkaline electrolyzer," *Energy Conversion and Management*, vol. 265, p. 115 697, 2022.

[313] P. C. Blaud, P. Haurant, F. Claveau, B. Lacarrière, P. Chevrel, and A. Mouraud, "Modelling and control of multi-energy systems through

multi-prosumer node and economic model predictive control," *International Journal of Electrical Power & Energy Systems*, vol. 118, p. 105 778, 2020.

[314] M. Arnold, R. R. Negenborn, G. Andersson, and B. De Schutter, "Model-based predictive control applied to multi-carrier energy systems," in *2009 IEEE power & energy society general meeting*, IEEE, 2009, pp. 1–8.

[315] M. Arnold, R. R. Negenborn, G. Andersson, and B. De Schutter, "Distributed predictive control for energy hub coordination in coupled electricity and gas networks," *Intelligent Infrastructures*, pp. 235–273, 2010.

[316] G. Coccia, A. Mugnini, F. Polonara, and A. Arteconi, "Artificial-neural-network-based model predictive control to exploit energy flexibility in multi-energy systems comprising district cooling," *Energy*, vol. 222, p. 119 958, 2021.

[317] A. A. Aliu, *Stochastic Model Predictive Control for Multi-Energy Systems with High Penetration of Electric Vehicles*. The University of Manchester (United Kingdom), 2021.

[318] M. Alamir, "Learning against uncertainty in control engineering," *Annual Reviews in Control*, 2022.

[319] H. Zhang, S. Seal, D. Wu, F. Bouffard, and B. Boulet, "Building energy management with reinforcement learning and model predictive control: a survey," *IEEE Access*, vol. 10, pp. 27 853–27 862, 2022.

[320] D. Wang, W. Zheng, Z. Wang, Y. Wang, X. Pang, and W. Wang, "Comparison of reinforcement learning and model predictive control for building energy system optimization," *Applied Thermal Engineering*, p. 120 430, 2023.

[321] D. P. Bertsekas, "Dynamic programming and suboptimal control: a survey from adp to mpc," *European Journal of Control*, vol. 11, no. 4-5, pp. 310–334, 2005.

[322] D. Ernst, M. Glavic, F. Capitanescu, and L. Wehenkel, "Model predictive control and reinforcement learning as two complementary frameworks," *International Journal of Tomography and Statistics*, vol. 6, 2007.

[323] S. Fujimoto, H. Hoof, and D. Meger, "Addressing function approximation error in actor-critic methods," in *International conference on machine learning*, PMLR, 2018, pp. 1587–1596.

[324] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *International conference on machine learning*, PMLR, 2018, pp. 1861–1870.

[325] H. T. H. Giang, T. N. K. Hoan, P. D. Thanh, and I. Koo, "Hybrid noma/oma-based dynamic power allocation scheme using deep reinforcement learning in 5g networks," *Applied Sciences*, vol. 10, no. 12, p. 4236, 2020.

[326] G. Matheron, N. Perrin, and O. Sigaud, "The problem with ddpg: understanding failures in deterministic environments with sparse rewards," *arXiv preprint arXiv:1911.11679*, 2019.

[327] G. Matheron, N. Perrin, and O. Sigaud, "Understanding failures of deterministic actor-critic with continuous action spaces and sparse rewards," in *International Conference on Artificial Neural Networks*, Springer, 2020, pp. 308–320.

[328] A. Trott, S. Zheng, C. Xiong, and R. Socher, "Keeping your distance: solving sparse reward tasks using self-balancing shaped rewards," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[329] C. Colas, O. Sigaud, and P.-Y. Oudeyer, "Gep-pg: decoupling exploration and exploitation in deep reinforcement learning algorithms," in *International conference on machine learning*, PMLR, 2018, pp. 1039–1048.

[330] S. Amin, M. Gomrokchi, H. Aboutalebi, H. Satija, and D. Precup, "Locally persistent exploration in continuous control tasks with sparse rewards," *arXiv preprint arXiv:2012.13658*, 2020.

[331] M. Plappert, R. Houthooft, P. Dhariwal, *et al.*, "Parameter space noise for exploration," *arXiv preprint arXiv:1706.01905*, 2017.

[332] Open AI, *Better exploration with parameter noise*. [Online]. Available: https://openai.com/research/better-exploration-with-parameter-noise (visited on 09/14/2023).

[333] D. Wang and M. Hu, "Deep deterministic policy gradient with compatible critic network," *IEEE Transactions on Neural Networks and Learning Systems*, 2021.

[334] P. Dhariwal, C. Hesse, O. Klimov, *et al.*, *Openai baselines*, https://github.com/openai/baselines, 2017.

[335] A. Hill, A. Raffin, M. Ernestus, *et al.*, *Stable baselines*, https://github.com/hill-a/stable-baselines, 2018.

[336] S. Sharma, S. Sharma, and A. Athaiya, "Activation functions in neural networks," *Towards Data Sci*, vol. 6, no. 12, pp. 310–316, 2017.

[337] C. Samende, J. Cao, and Z. Fan, "Multi-agent deep deterministic policy gradient algorithm for peer-to-peer energy trading considering distribution network constraints," *Applied Energy*, vol. 317, p. 119 123, 2022.

[338] Idex, *Projet MSE, Annexe projet aux conditions particulieres*, 2019.

[339] IRENA, "Renewable power generation costs in 2020," 2021.

[340] CGI, *Délégation de service public du réseau de chaleur et de froid du quartier Méridia à Nice et optimisation énergétique, etude préalable, SI éco-quartier*, May 2019.

[341] IEA-ETSAP and IRENA, "Thermal energy storage, technology brief," Jan. 2013.

[342] P. Arce, M. Medrano, A. Gil, E. Oró, and L. F. Cabeza, "Overview of thermal energy storage (TES) potential energy savings and climate change mitigation in Spain and Europe," *Applied energy*, vol. 88, no. 8, pp. 2764–2774, 2011.

[343] S. Rigal, D. Haillot, E. Franquet, S. Gibout, F. Jay, and J.-P. Bédécarrats, "Stockage de l'énergie thermique par matériaux à changement de phase adapté à la valorisation de chaleur fatale: études expérimentales et numériques," in *25ème congrès de la Société Française de Thermique SFT; 30 Mai–2 Juin*, 2017.

[344] M. Martinelli, "Stockage d'énergie thermique par changement de phase–application aux réseaux de chaleur," Ph.D. dissertation, Université Grenoble Alpes, 2016.

[345] M. Martinelli, F. Bentivoglio, A. Caron-Soupart, R. Couturier, J.-F. Fourmigue, and P. Marty, "Experimental study of a phase change thermal energy storage with copper foam," *Applied thermal engineering*, vol. 101, pp. 247–261, 2016.

[346] B. Tremeac, P. Bousquet, C. de Munck, *et al.*, "Influence of air conditioning management on heat island in paris air street temperatures," *Applied Energy*, vol. 95, pp. 102–110, 2012.

[347] Z. Garofalaki, D. Kosmanos, S. Moschoyiannis, D. Kallergis, and C. Douligeris, "Electric vehicle charging: a survey on the security issues and challenges of the open charge point protocol (ocpp)," *IEEE Communications Surveys & Tutorials*, 2022.

[348] J. El Feghali, "Contribution à l'utilisation des modèles physiques en modelica à des fins de commande de systèmes multi-énergies," Ph.D. dissertation, université Paris-Saclay, 2023.

[349] C. Klemm and P. Vennemann, "Modeling and optimization of multi-energy systems in mixed-use districts: a review of existing methods and approaches," *Renewable and Sustainable Energy Reviews*, vol. 135, p. 110 206, 2021.

[350] U.S Departement of Energy and National Renewable Energy laboratory, *EnergyPlus*. [Online]. Available: https://energyplus.net/.

[351] Transient Systems Simulation Tool, *TRNSYS*. [Online]. Available: https://www.trnsys.com/.

[352] The Modeclia Association, *Modelica*. [Online]. Available: https://modelica.org/.

[353] P. Fritzson and V. Engelson, "Modelica—a unified object-oriented language for system modeling and simulation," in *ECOOP'98—Object-Oriented Programming: 12th European Conference Brussels, Belgium, July 20–24, 1998 Proceedings 12*, Springer, 1998, pp. 67–90.

[354] The Open Source Modelica Consortium, *OpenModelica*. [Online]. Available: https://openmodelica.org/.

[355] Dassault Systemes, *Dynamic Modeling laboratory, Dymola*. [Online]. Available: https://www.3ds.com/fr/produits-et-services/catia/produits/dymola/.

[356] J. Allegrini, K. Orehounig, G. Mavromatidis, F. Ruesch, V. Dorer, and R. Evins, "A review of modelling approaches and tools for the simulation of district-scale energy systems," *Renewable and Sustainable Energy Reviews*, vol. 52, pp. 1391–1404, 2015.

[357] T. Gronier, E. Franquet, and S. Gibout, "Platform for transverse evaluation of control strategies for multi-energy smart grids," *Smart Energy*, vol. 7, p. 100 079, 2022.

[358] J. Frenkel, C. Schubert, G. Kunze, P. Fritzson, M. Sjölund, and A. Pop, "Towards a benchmark suite for modelica compilers: large models," in *8th International Modelica Conference (Modelica'2011), Dresden, Germany, March 20-22, 2011*, Linköping University Electronic Press, 2011, pp. 143–152.

[359] Modelica Association Project, Dassault Systemes, *The Functional Mock-up Interface (FMI) standard tools*. [Online]. Available: https://fmi-standard.org/tools/.

[360] M. Wetter, W. Zuo, T. S. Nouidui, and X. Pang, "Modelica buildings library," *Journal of Building Performance Simulation*, vol. 7, no. 4, pp. 253–270, 2014.

[361] M. Šúri, T. A. Huld, and E. D. Dunlop, "Pv-gis: a web-based solar radiation database for the calculation of pv potential in europe," *International Journal of Sustainable Energy*, vol. 24, no. 2, pp. 55–67, 2005.

[362] K. Park and C. A. Felippa, "Partitioned analysis of coupled systems," 1983.

[363] Modelon. "FMI Standard: Understand the two types of Functional Mock-up Units – CS v. ME." (), [Online]. Available: https://modelon.com/blog/fmi-functional-mock-up-unit-types/ (visited on 07/20/2023).

[364] C. Noll, T. Blochwitz, T. Neidhold, and C. Kehrer, "Implementation of modelisar functional mock-up interfaces in simulationx," in *8th International Modelica Conference*, vol. 2, 2011.

[365] CATIA-Systems, *FMPy*. [Online]. Available: https://github.com/CATIA-Systems/FMPy.

[366] JModelica, *PyFMI*. [Online]. Available: https://jmodelica.org/pyfmi/.

[367] J. H. Holland, "Genetic algorithms," *Scientific american*, vol. 267, no. 1, pp. 66–73, 1992.

[368] S. Ryan Mohammad. "Geneticalgorithm library." (2020), [Online]. Available: https://github.com/rmsolgi/geneticalgorithm.

[369] RWTH Aaachen University, E.On Energy Research Center. "TEASER - Tool for Energy Analysis and Simulation for Efficient Retrofit." (2016), [Online]. Available: https://rwth-ebc.github.io/TEASER//master/docs/index.html (visited on 08/01/2023).

[370] T. Salomon, R. Mikolasek, and B. Peuportier, "Outil de simulation thermique du bâtiment, comfie," *Journée thématique SFT-IBPSA*, 2005.

[371] P. Remmen, M. Lauster, M. Mans, M. Fuchs, T. Osterhage, and D. Müller, "Teaser: an open tool for urban energy modelling of building stocks," *Journal of Building Performance Simulation*, vol. 11, no. 1, pp. 84–98, 2018.

[372] D. Müller, M. Lauster, A. Constantin, M. Fuchs, and P. Remmen, "Aixlib-an open-source modelica library within the iea-ebc annex 60 framework," *BauSIM 2016*, pp. 3–9, 2016.

[373] Google. "Google Earth." (2001), [Online]. Available: https://earth.google.com/web/@43.6938475,4.96098521,123.75462423a,617918.11573863d,35y,0h,0t,0r/data=OgMKATA (visited on 10/19/2023).

[374] N. Pflugradt, P. Stenzel, L. Kotzur, and D. Stolten, "Loadprofilegenerator: an agent-based behavior simulation for generating residential load profiles," *Journal of Open Source Software*, vol. 7, no. 71, p. 3574, 2022.

[375] L'ADEME, Le Costic. "Les besoins d'eau chaude sanitaire en habitat individuel et collectif." (2016), [Online]. Available: https://www.costic.com/sites/default/files/upload/telechargements/ECS_MJL_ADEME/besoin-eau-chaude-sanitaire-habitat-individuel-et-collectif.pdf (visited on 10/01/2023).

[376] L'ADEME, ENERTECH. "Technologies de l'information et éclairage." (2005), [Online]. Available: https://www.enertech.fr/wp-content/uploads/docs/R_FinTIE50.pdf (visited on 10/01/2023).

[377] RTE. "Bilan électrique." (2019), [Online]. Available: https://assets.rte-france.com/prod/public/2020-06/bilan-electrique-2019_1_0.pdf (visited on 10/16/2023).

[378] A. Asperti and M. Del Brutto, "Microracer: a didactic environment for deep reinforcement learning," *arXiv preprint arXiv:2203.10494*, 2022.

[379] G. Ceusters, L. R. Camargo, R. Franke, A. Nowé, and M. Messagie, "Safe reinforcement learning for multi-energy management systems with known constraint functions," *Energy and AI*, vol. 12, p. 100 227, 2023.

[380] G. Dulac-Arnold, N. Levine, D. J. Mankowitz, *et al.*, "Challenges of real-world reinforcement learning: definitions, benchmarks and analysis," *Machine Learning*, vol. 110, no. 9, pp. 2419–2468, 2021.

[381] Q. Li, Z. Peng, L. Feng, Q. Zhang, Z. Xue, and B. Zhou, "Metadrive: composing diverse driving scenarios for generalizable reinforcement learning," *IEEE transactions on pattern analysis and machine intelligence*, 2022.

[382] M. E. Taylor and P. Stone, "Transfer learning for reinforcement learning domains: a survey.," *Journal of Machine Learning Research*, vol. 10, no. 7, 2009.

[383] Z. Zhu, K. Lin, and J. Zhou, "Transfer learning in deep reinforcement learning: a survey," *arXiv preprint arXiv:2009.07888*, 2020.

[384] PiLogic, *Manuel nice méridia*, 2021.

Appendix A

Technical details of the energy systems in the Meridia Smart Energy case study

A.1 Thermo-refrigerating heat pumps

A Thermo-refrigerating Heat pump is denoted by TRHP and tandem of TRHPs refers to a pair of TRHPs. The technical specifications of the four TRHPs of the MSE power plant are given in table A.1.

Table A.1: Technical specifications of the geothermal TRHPs of the MSE eco-district.

Parameter	Unit	TRHPA	TRHPB	TRHPC	TRHPD	Total
Load	%	100	100	100	100	100
Cooling capacity	kW	1259.7	1381.8	1105.8	1178.9	4926.2
Heating capacity	kW	1520.3	1698.8	1458.5	1600.8	6278.4
Share in the tandem (AB or CD)	%	47	53	48	52	
Share in the A-B-C-D series configu-ration	%	24	27	23	25	

A.2 P&ID of the heating and cooling power plant

A simplified P&ID of the MSE heating and cooling power plant is given in figure A.1.

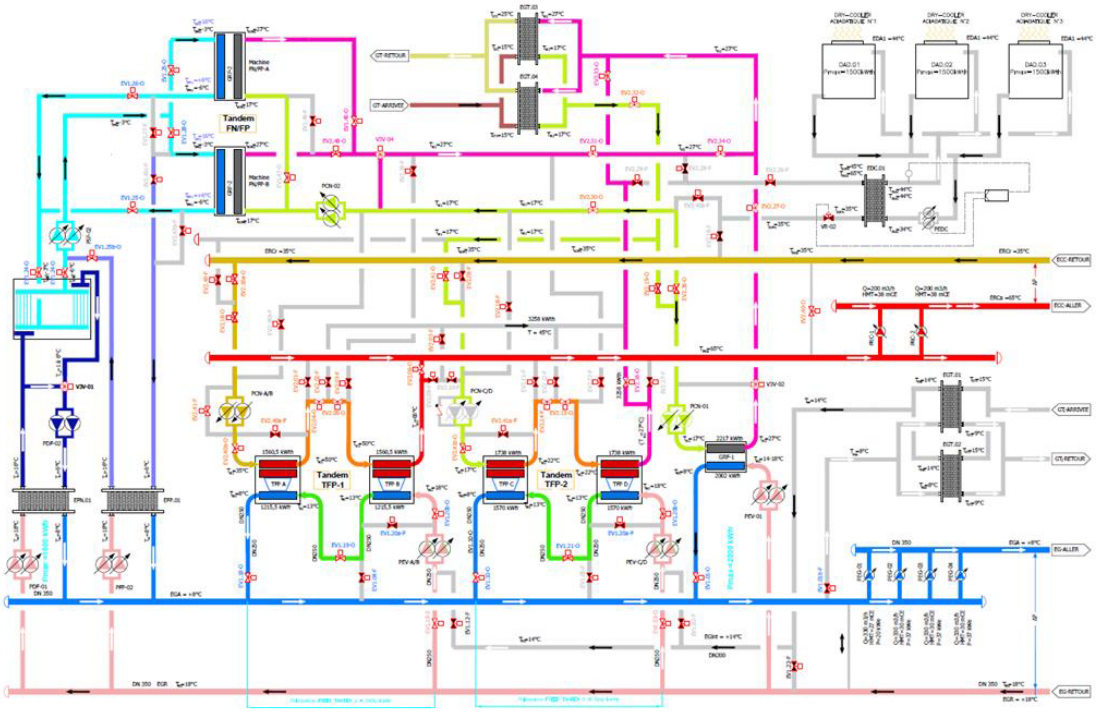


Figure A.1: Simplified process and instrumentation diagram of the MSE eco-district heating and cooling power plant.

A.3 Heating and cooling sub-stations

A schematic overview of the MSE district heating and cooling network illustrating its different sub-stations is given in figure A.2.

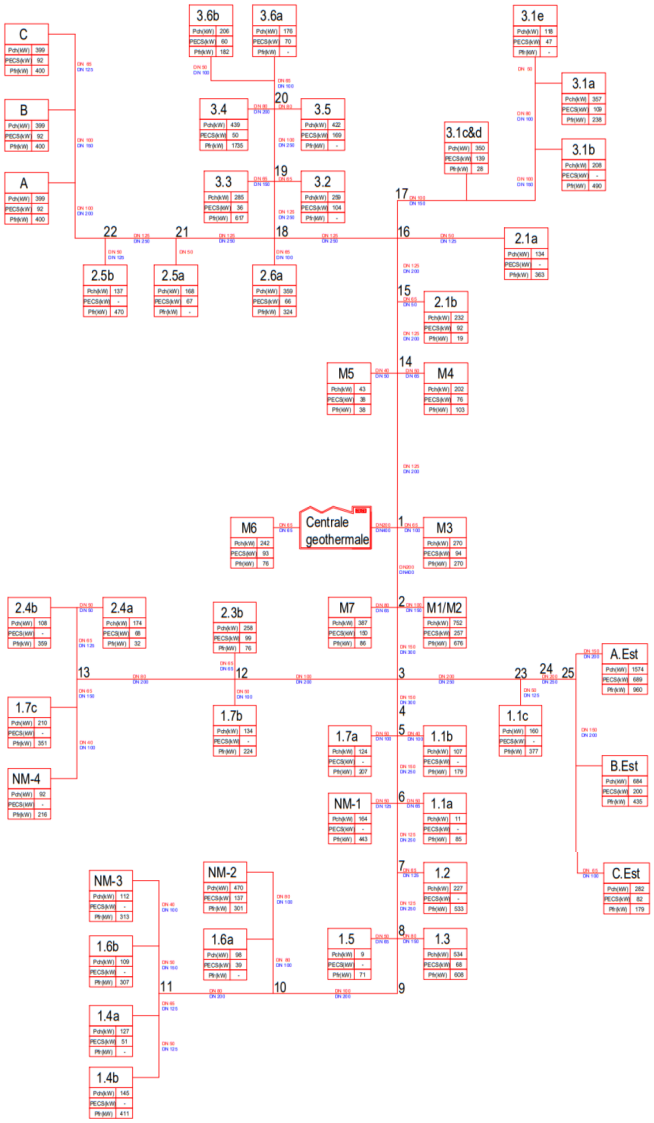


Figure A.2: Schematic overview of the MSE district heating and cooling network illustrating locations of the sub-stations and their heating power (denoted P_{ch}), domestic water heating power (denoted P_{ECS}) and cooling power (denoted P_{fr}).

Appendix B

Documentation sheets for the Dymola sub-models of the MSE simulation model

The following appendices contain documentation sheets for the sub-models developed in this study and that once connected together allowed building a simulation model for the MSE eco-district. We would like to bring to your attention that these documentation sheets are presented in French. In fact, these documents are part of a set of deliverables and reports that were submitted to the MSE project funding authority which requires all project reports to be submitted in French. Regarding the time that would be required for translating these extensive documentation sheets from french to english, we have opted to include them in their original language in these appendices.

B.1 Heat substation with valve

Objet : Sous-station de chaud avec vanne
Phénomènes : Transfert de chaleur entre la sous-station et le réseau de chaud
Hypothèses : Pas de perte thermique, non prise en compte des pertes de charge
Modèle : Heating_withoutPump_final

Schéma du composant

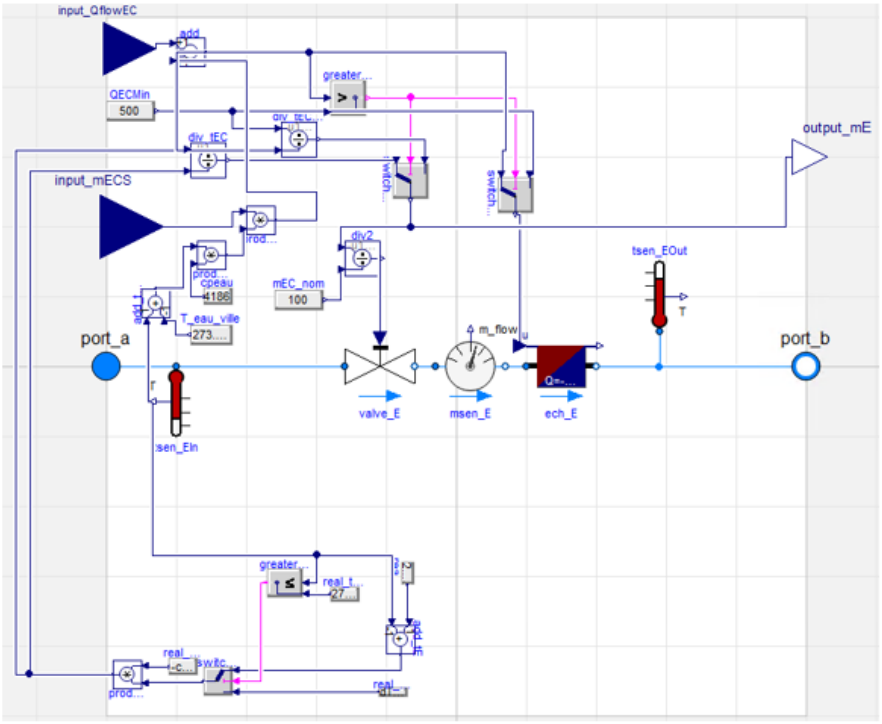
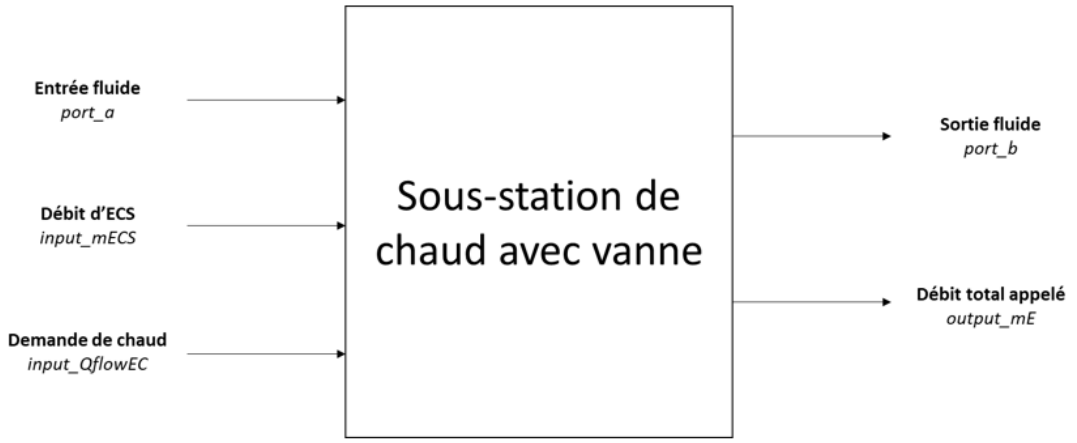


Schéma bloc



Connecteurs :

Nom	Description	Unité	Valeur
port_a	Port d'entrée du fluide dans la sous-station	-	-
port_b	Port de sortie du fluide dans la sous-station	-	-
input_QflowEC	Besoin en eau chaude	W	-
input_mECS	Besoin en ECS	kg/s	-
output_mE	Débit nécessaire à la sous-station	kg/s	-

Paramètres :

Nom	Description	Unité	Valeur par défaut
cpeau	Capacité thermique massique de l'eau	J/(kg.K)	4186
T_eau_ville	Température de l'eau de ville	K	273.15 + 10
QEC_min	Flux de chaleur minimum assuré par la sous-station	W	500
mEC_nom	Débit nominal de la vanne de régulation	kg/s	100
dTHex	Delta de température au niveau de l'échangeur	-	-30

Équations

Le débit d'ECS est ramené en flux de chaleur équivalent côté sous-station :

$$Qecs = input_{mECS} * cpeau * (Tin - T_{eau\_ville})$$

Ce flux de chaleur est sommé au flux de chaleur *input_QflowEC*.
Si cette somme est inférieure à *QEC_min* alors c'est *QEC_min* qui est assuré.
Cette valeur est transmise à l'échangeur à flux de chaleur paramétrable et à la régulation d'ouverture de la vanne.

Le débit correspondant au flux de chaleur total à traiter *Q* est déterminé :

$$mE = output_{mE} = \frac{Q}{cpeau * dTHex}$$

La vanne s'ouvre proportionnellement à son débit nominal *mEC_nom* :

$$ouverture_{vanne} = \frac{mE}{mEC_{nom}}$$

Une différence de pression importante est fixée au niveau de la vanne pour assurer le passage du débit souhaité.

Tableau des révisions	
Création	AR, 10/05/2022
Ajout du schéma bloc	AR, 13/07/2022

B.2 Cold substation with valve

Objet : Sous-station de froid avec vanne
Phénomènes : Transfert de chaleur entre la sous-station et le réseau
Hypothèses : Pas de perte thermique, non prise en compte des pertes de charge
Modèle : Cooling_withoutPump_final

Schéma du composant

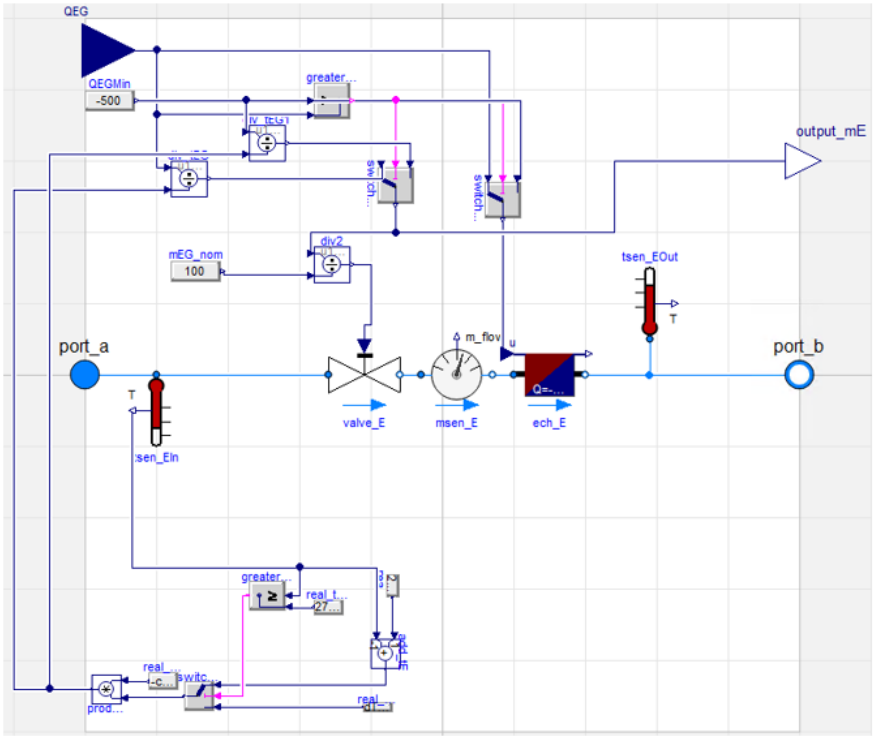
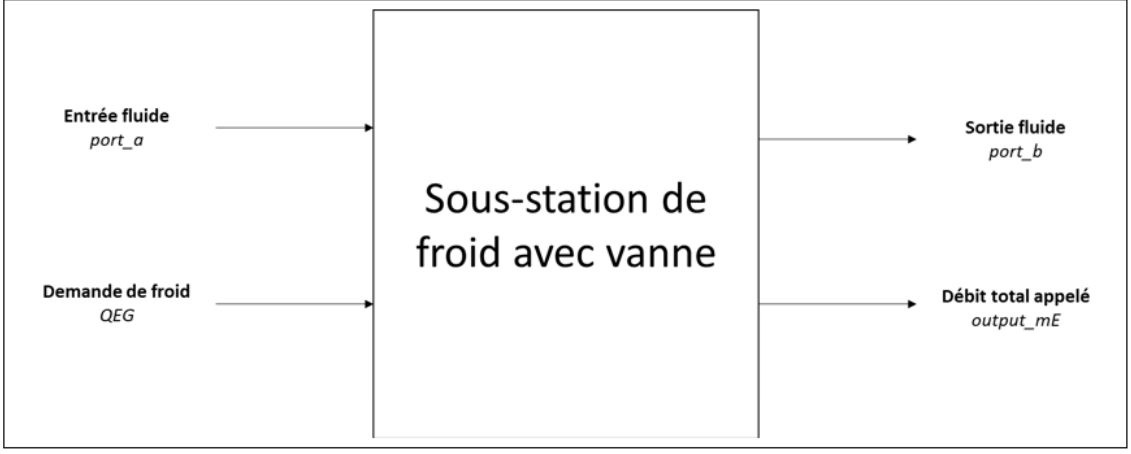


Schéma bloc



Connecteurs :

Nom	Description	Unité	Valeur
port_a	Port d'entrée du fluide dans la sous-station	-	-
port_b	Port de sortie du fluide dans la sous-station	-	-
QEG	Besoin en eau froide	W	-
output_mE	Débit nécessaire à la sous-station	kg/s	-

Paramètres :

Nom	Description	Unité	Valeur par défaut
cpeau	Capacité thermique massique de l'eau	J/(kg.K)	4186
QEG_min	Flux de chaleur minimum assuré par la sous-station	W	-500
mEG_nom	Débit nominal de la vanne de régulation	kg/s	100
dTHex	Delta de température au niveau de l'échangeur	-	+10

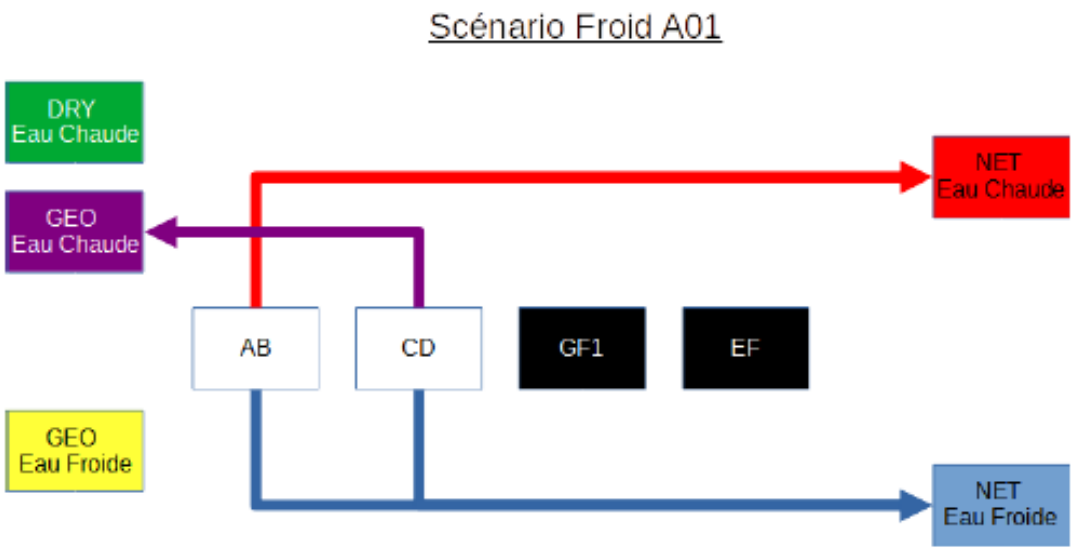
Équations
Si *QEG* est inférieur à *QEG_min* alors c'est *QEG_min* qui est assuré. Cette valeur est transmise à l'échangeur à flux de chaleur paramétrable et à la régulation d'ouverture de la vanne. Le débit correspondant au flux de chaleur à traiter est déterminé : $$mE = output_{mE} = \frac{Q}{cpeau * dTHex}$$ La vanne s'ouvre proportionnellement à son débit nominal *mEC_nom* : $$ouverture_{vanne} = \frac{mE}{mEC_{nom}}$$ Une différence de pression importante est fixée au niveau de la vanne pour assurer le passage du débit souhaité.

Tableau des révisions	
Création	AR, 10/05/2022
Ajout du schéma bloc	AR, 13/07/2022

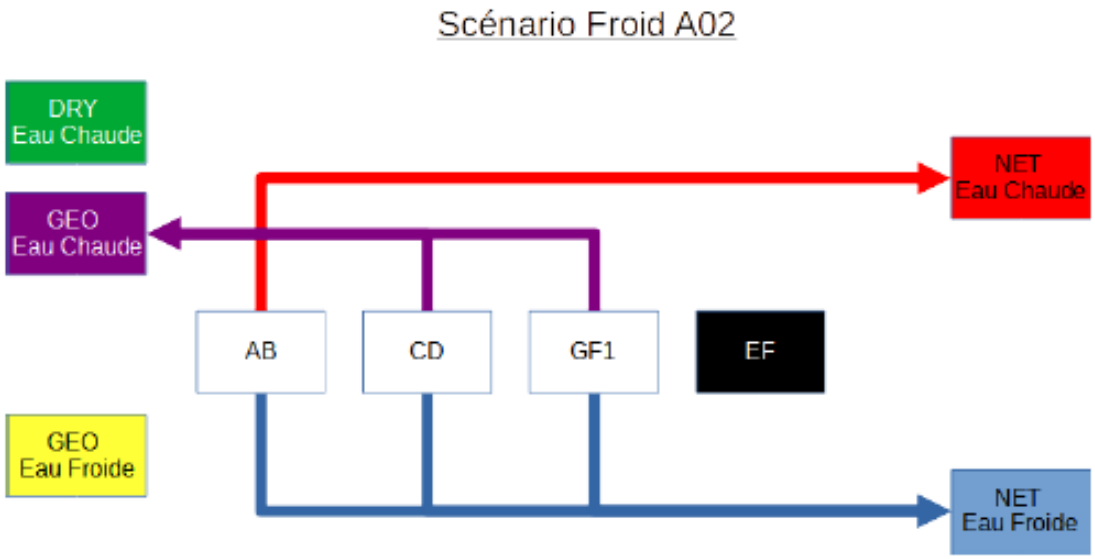
B.3 Configurations of the TRHP system

This appendix describes the 14 different configurations of MSE's Thermo-Refrigerating Heat Pumps as described in the functional analysis provided by the company PiLogic [384].

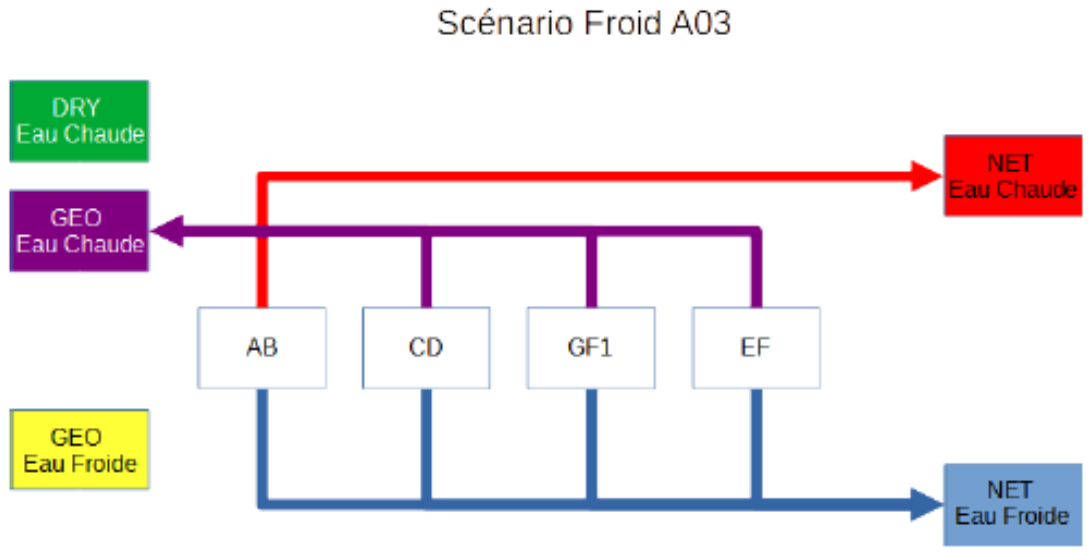
* Configuration de froid A01: Cette configuration est la configuration de base du fonctionnement en mode froid, c'est-à-dire quand les besoins de froid sont supérieurs aux besoins de chaud, classiquement en été.



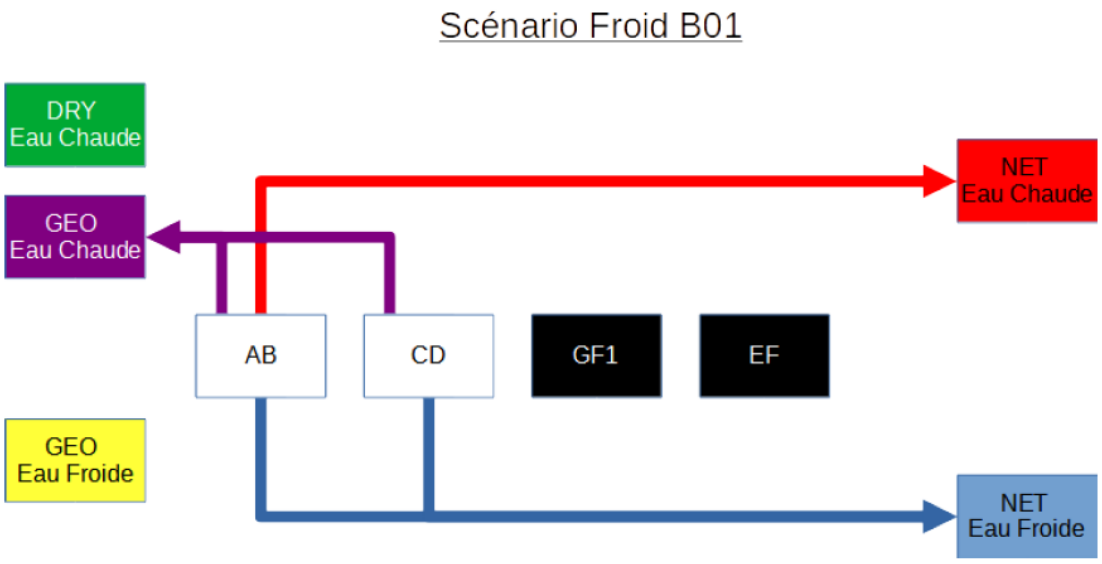
* Configuration de froid A02: Cette configuration fait suite à la configuration froid A01. Lorsque les besoins de froid ne peuvent être satisfaits par les deux tandems de TFP, le Groupe Froid (GF) est employé.



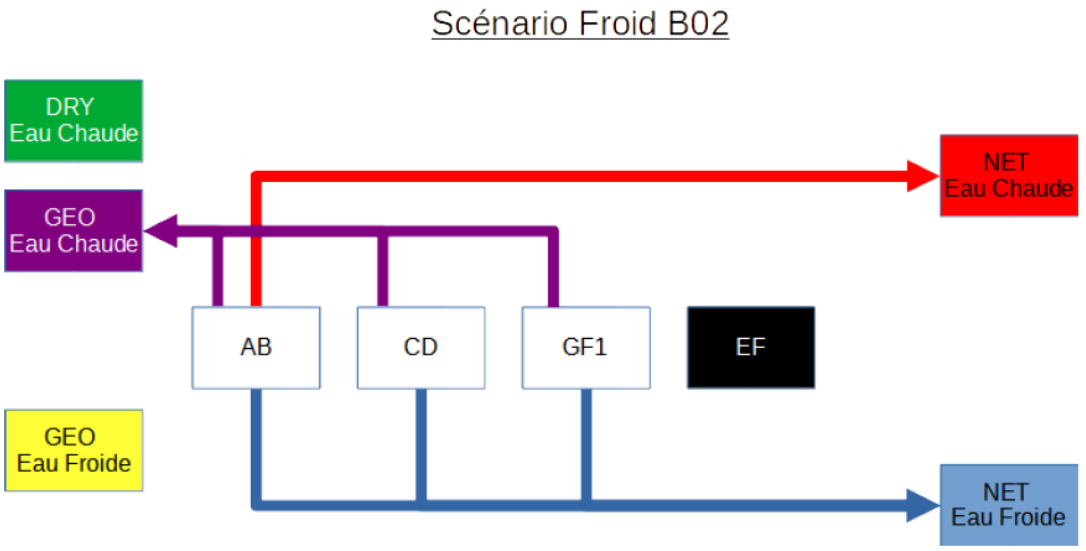
* Configuration de froid A03: Cette configuration fait suite à la configuration froid A02. Si les deux tandems de TFP et le GF ne peuvent répondre aux besoins de froid alors le Groupe Froid Negatif-Positif (GF-NP) est employé.



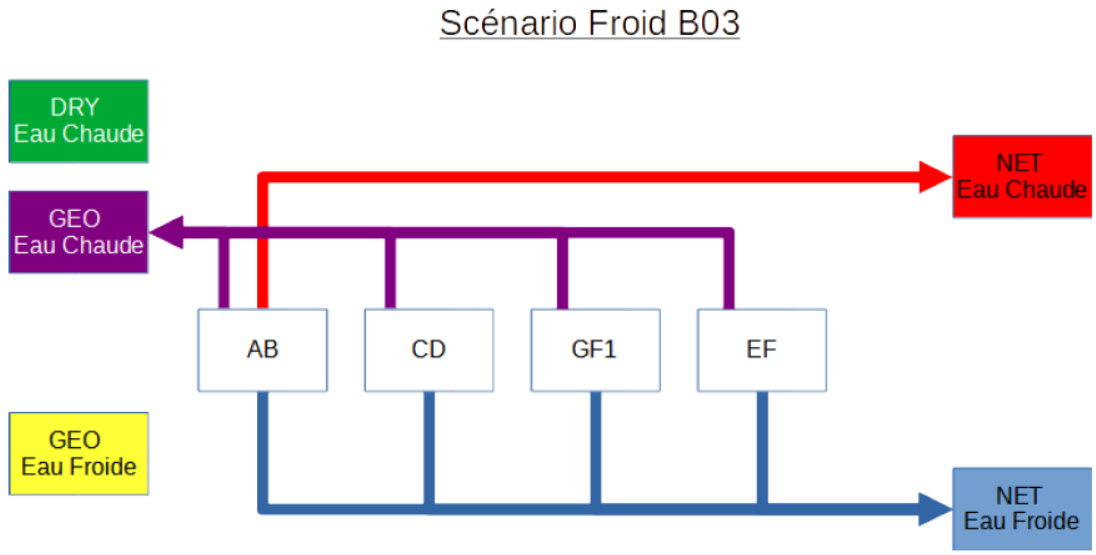
* Configuration de froid B01: Cette configuration fait suite à la configuration froid A03. Si l'ensemble tandems de TFP, GF et GF-NP ne peut pas répondre aux besoins de froid alors le système de production change de catégorie de configuration pour tenter d'y répondre.



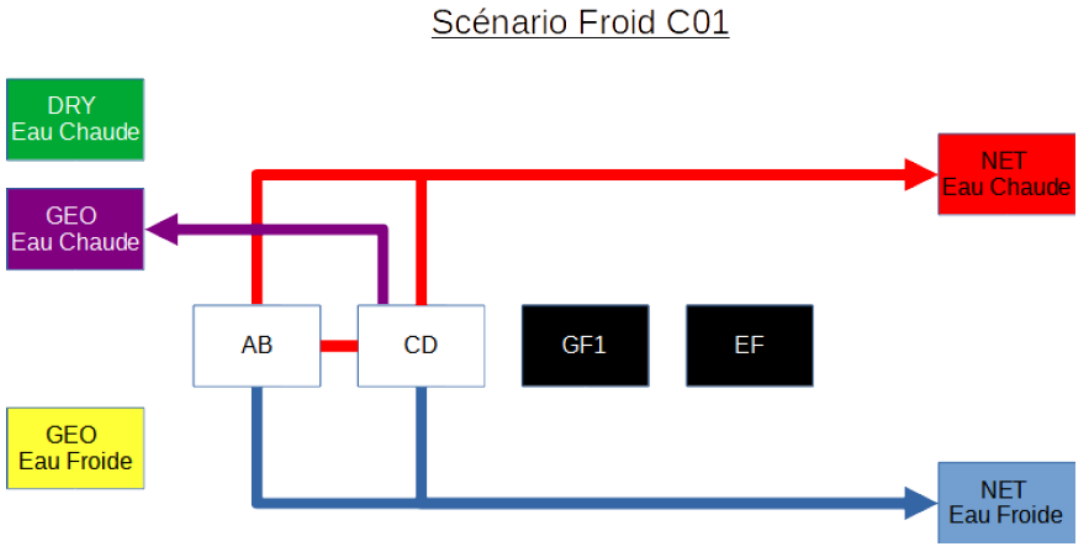
* Configuration de froid B02: La configuration froid B02 fait suite à la configuration froid B01. Lorsque les besoins de froid ne peuvent être satisfaits par les deux tandems de TFP, le GF est employé.



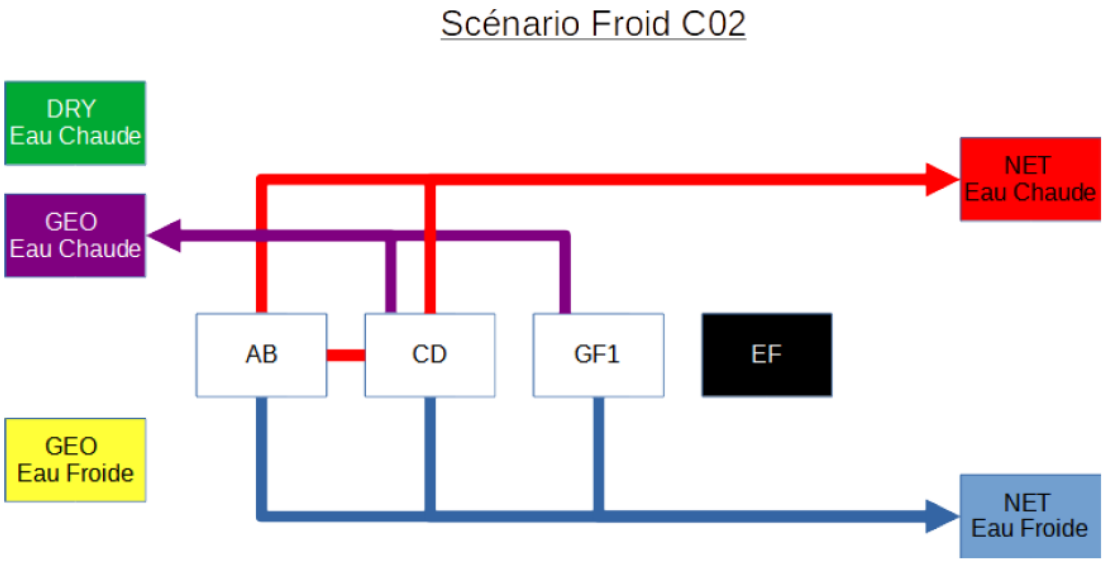
* Configuration de froid B03: La configuration froid B03 fait suite à la configuration froid B02. Si les deux tandems de TFP et le GF ne peuvent répondre aux besoins de froid alors le GF-NP est employé.



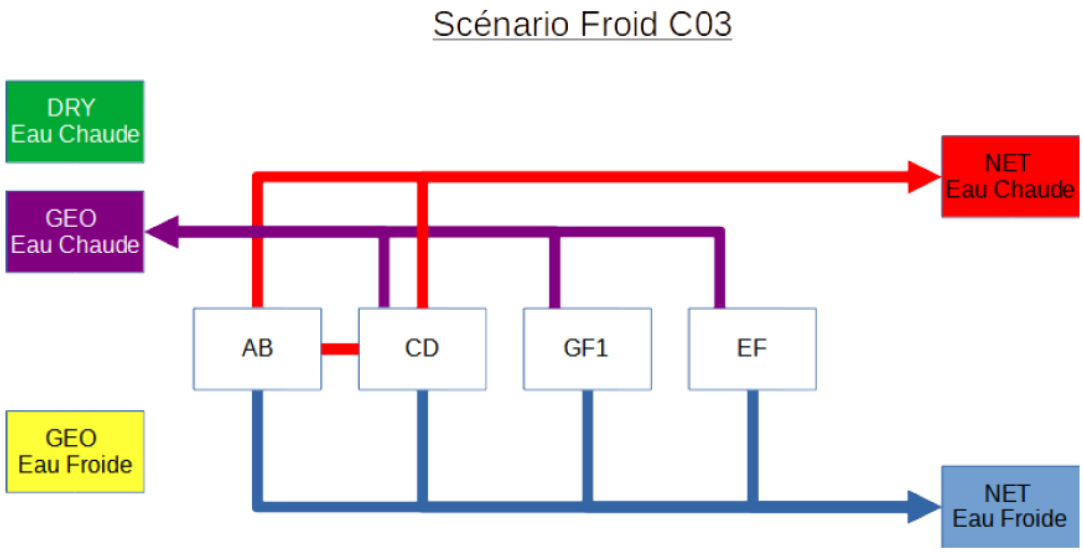
* Configuration de froid C01: Cette configuration ne fait pas suite aux configurations froid A ou B. Il s'agit de la configuration de base en mode froid lorsque les besoins de chaud ne peuvent être assurés par un seul tandem de TFP.



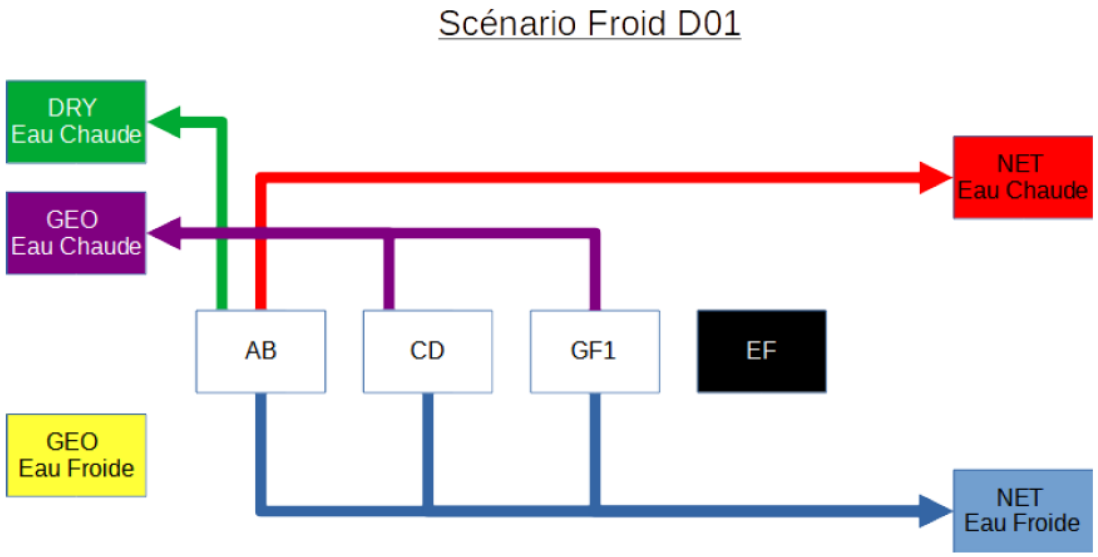
* Configuration de froid C02: La configuration froid C02 fait suite à la configuration froid C01. Lorsque les besoins de froid ne peuvent être satisfaits par les deux tandems de TFP, le GF est employé.



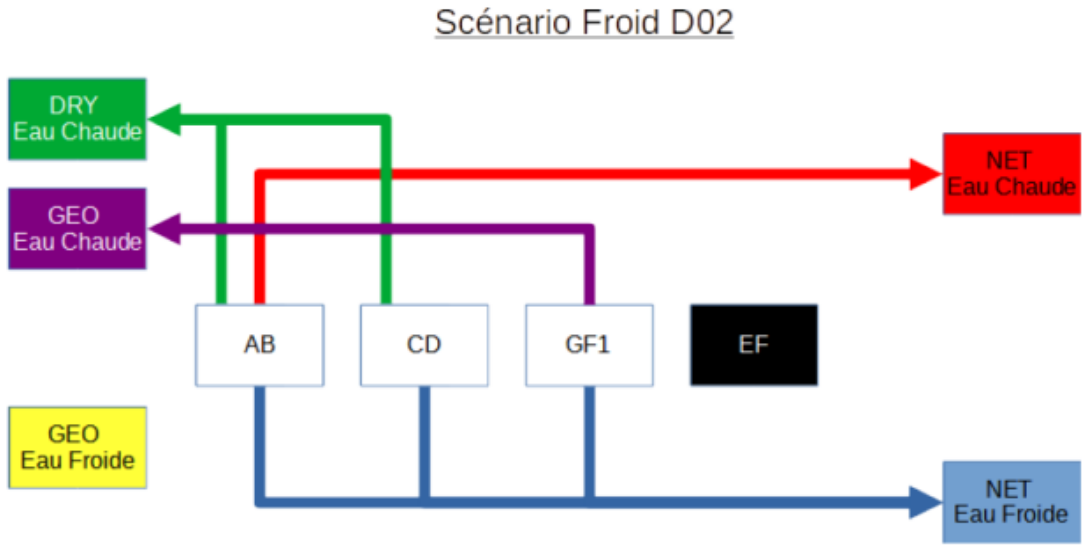
* Configuration de froid C03: La configuration froid C03 fait suite à la configuration froid C02. Si les deux tandems de TFP et le GF ne peuvent répondre aux besoins de froid alors le GF-NP est employé.



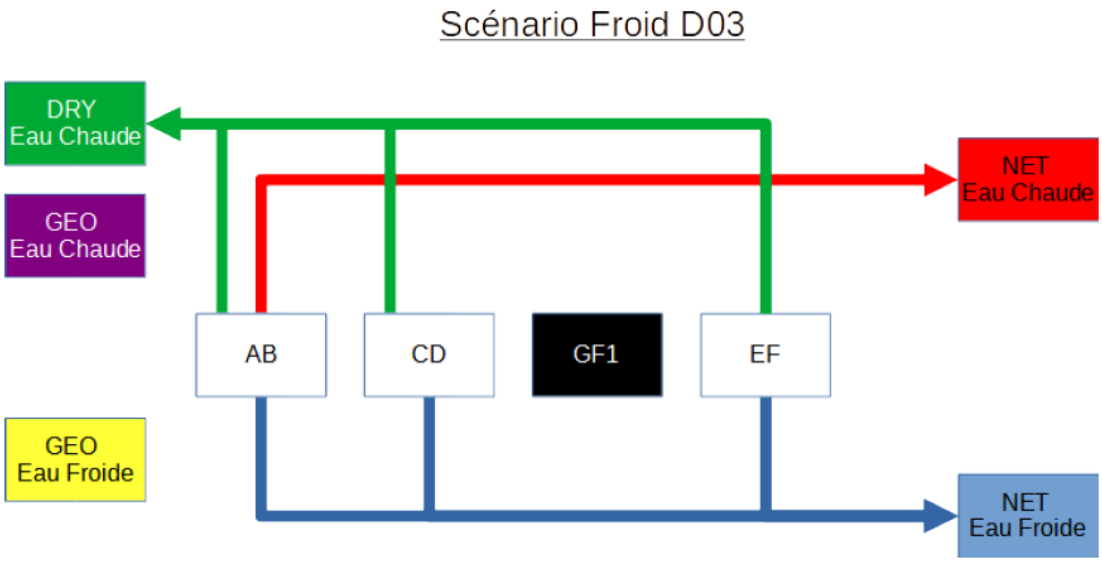
* Configuration de froid D01: Cette configuration ne fait pas suite aux configurations A, B ou C. Il s'agit de la configuration de base en mode froid lorsque les capacités d'absorption de la géothermie sont « saturées ». Cette configuration correspond à une faible saturation.



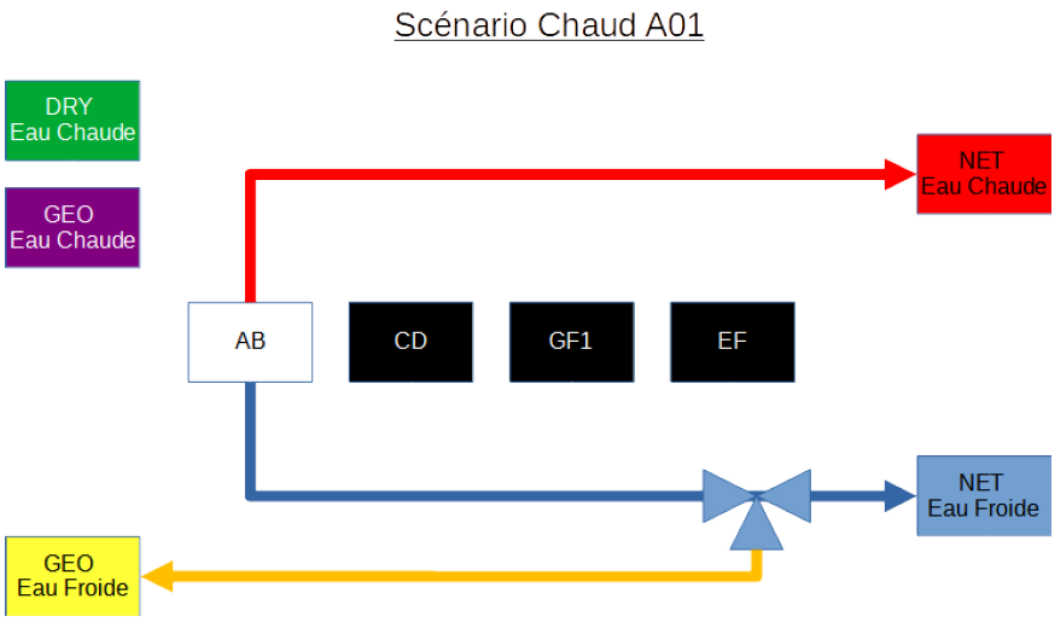
* Configuration de froid D02: La configuration froid D02 fait suite à la configuration froid D01. Le passage à cette configuration s'effectue si les capacités d'absorption de la géothermie sont moyennement saturées.



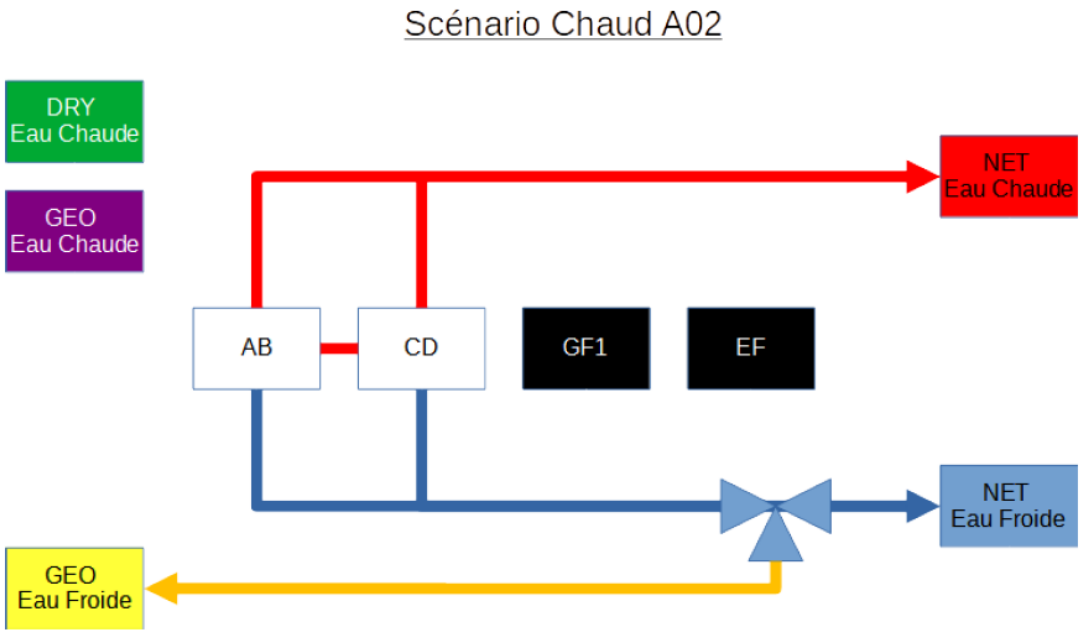
* Configuration de froid D03: La configuration froid D03 fait suite à la configuration froid D02. Le passage à cette configuration s'effectue si les capacités d'absorption de la géothermie sont totalement saturées.



* Configuration de chaud A01: Cette configuration est la configuration de base du fonctionnement en mode chaud, c'est-à-dire quand les besoins de chaud sont supérieurs aux besoins de froid, classiquement en hiver.



* Configuration de chaud A02: La configuration chaud A02 fait suite à la configuration chaud A01. Si les besoins de chaud ne peuvent être comblés par un seul tandem de TFP alors le deuxième entre en fonctionnement.


Scénario Chaud A02
DRY
Eau Chaude
GEO
Eau Chaude
NET
Eau Chaude
AB
CD
GF1
EF
NET
Eau Froide
GEO
Eau Froide

B.4 Individual thermo-refrigerating heat pump

Objet : Thermofrigopompe
Phénomènes : Cycle thermodynamique
Hypothèses : Pas de perte thermique, non prise en compte des pertes de charge
Méthode :
Modèle : Carnot_Tcon_TFP

Schéma du composant

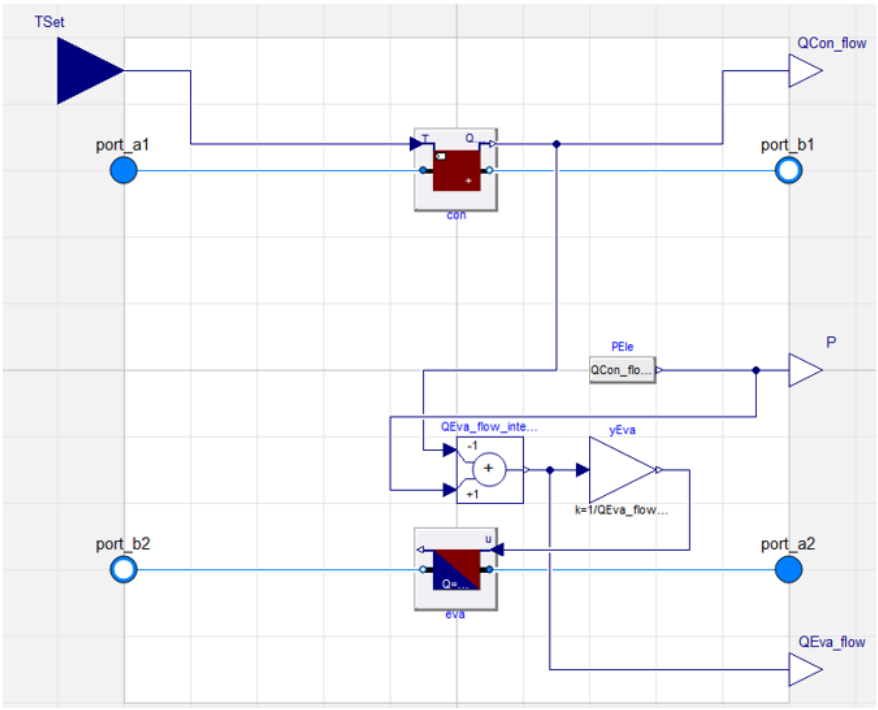
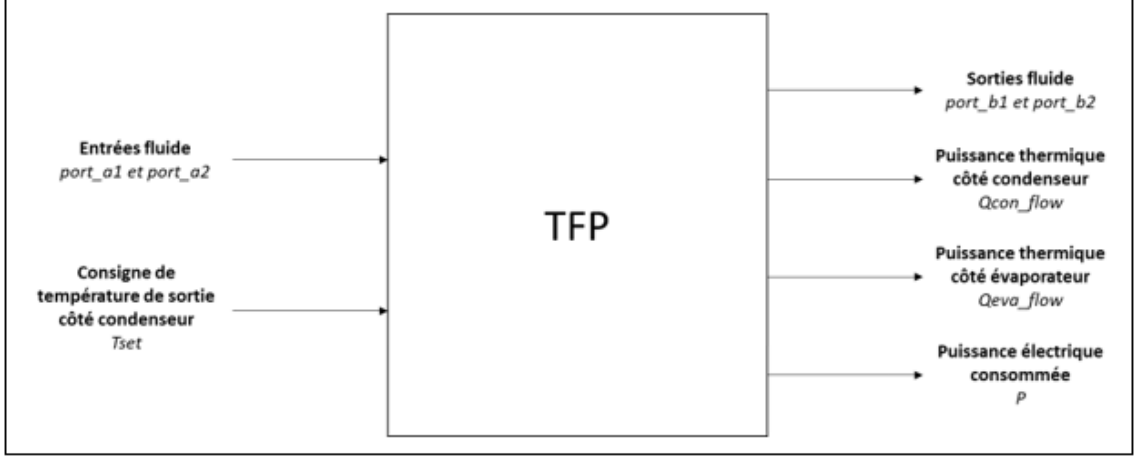


Schéma bloc



Connecteurs :

Nom	Description	Unité	Valeur
port_a1	Port d'entrée du fluide côté condenseur	-	-
port_b1	Port de sortie du fluide côté condenseur	-	-
port_a2	Port d'entrée du fluide côté évaporateur	-	-
port_b2	Port de sortie du fluide côté évaporateur	-	-
Tset	Consigne de température de sortie côté condenseur	K	-
QCon_flow	Flux de chaleur échangé au niveau de condenseur	W	-
QEva_flow	Flux de chaleur échangé au niveau de l'évaporateur	W	-
P	Puissance électrique calculée par le modèle	W	-

Paramètres :

Nom	Description	Unité	Valeur par défaut
COP	COP de la TFP considérée pour la configuration et la charge associées	-	-
P_elec (équivalent à P)	Puissance électrique équivalente recalculée	W	-

Équations
Le COP des TFP est variable pour chaque configuration et taux de charge du système. Malheureusement, ces COP ne sont pas tous connus. Pour pouvoir pallier ce manque de données, le COP nominal des TFP (selon les données constructeur) a uniquement été utilisé et leur variation en fonction du taux de charge du système a été calée sur la variation connue de COP équivalents.

$$COP_{nom}\,(TFP_i, \%_{charge})$$

$$QEva_{flow} = P - QCon_{flow}$$

$$P_{elec} = \frac{QCon_{flow}}{COP_{nom}\,(TFP_i, \%_{charge})}$$

Tableau des révisions	
Création	AR, 10/05/2022
Ajout du schéma bloc	AR, 13/07/2022

B.5 Thermo-refrigerating heat pump system

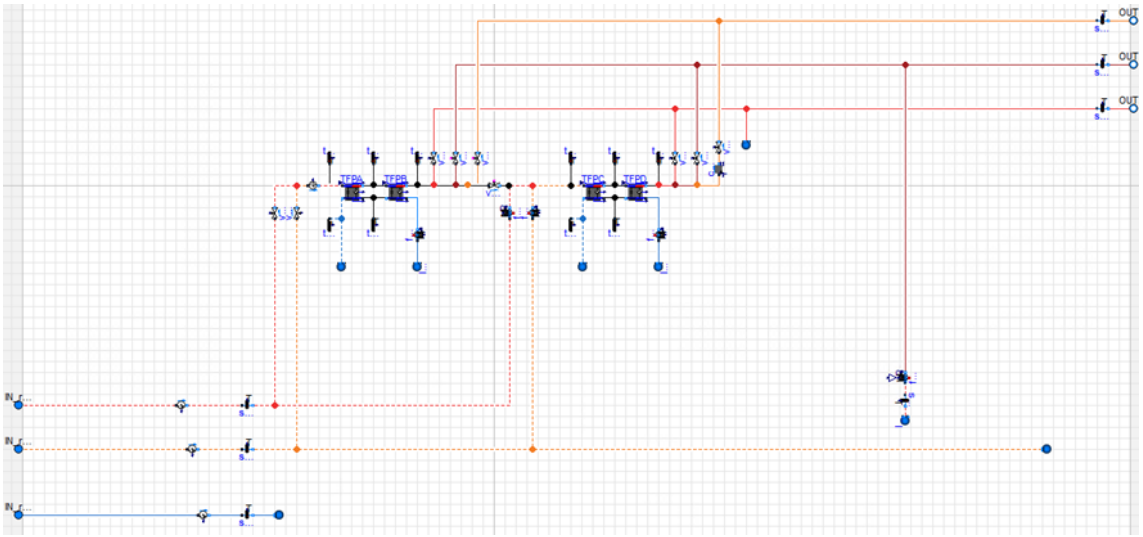


Figure B.1: An overview of the thermo-refrigerating heat pump system without regulation.

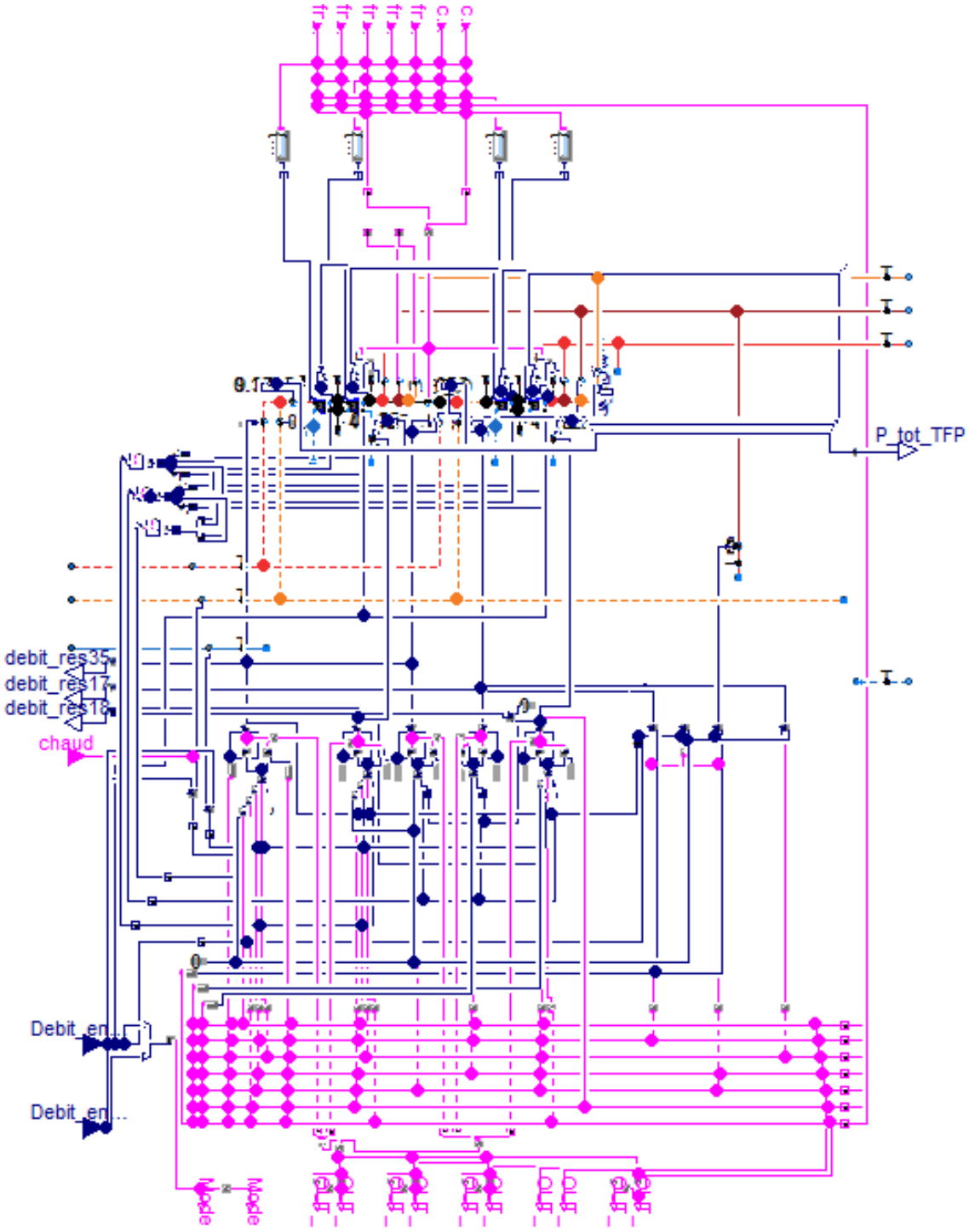


Figure B.2: An overview of the thermo-refrigerating heat pump system with regulation.

B.6 Adiabatic aero-refrigerant system (DRY)

Objet : DRY (Aéroréfrigérant adiabatique)
Phénomènes : Échange de chaleur avec l'air ambiant
Hypothèses : Pas de perte thermique, non prise en compte des pertes de charge
Méthode :
Modèle : Syst_pro_complet_final_V2

Schéma du composant

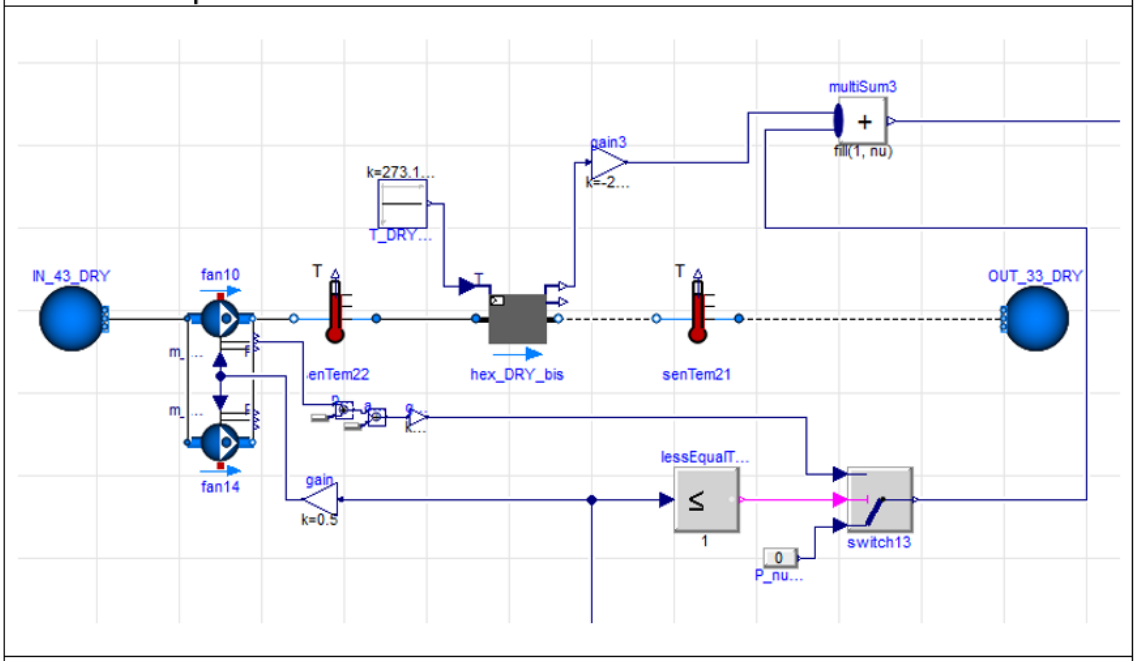
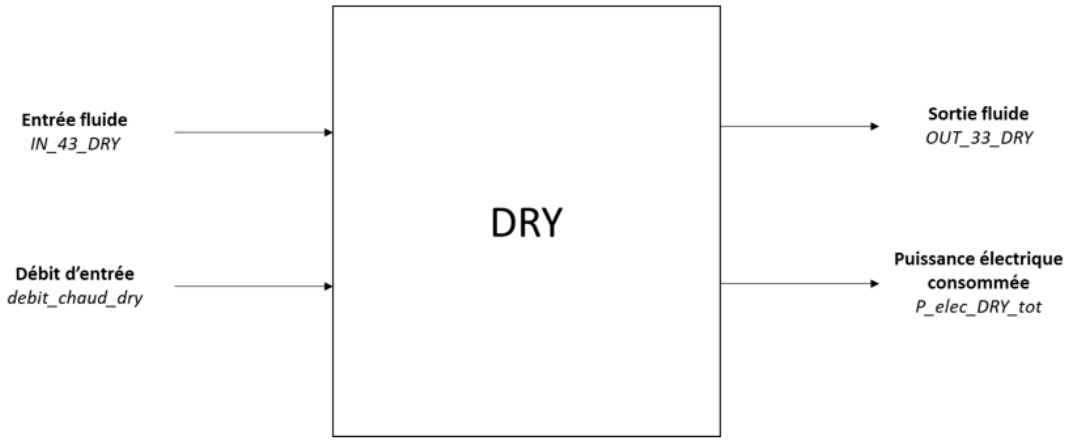


Schéma bloc



Connecteurs :

Nom	Description	Unité	Valeur
IN_43_DRY	Source fournissant le fluide à une température fixée	-	-
OUT_33_DRY	Frontière récupérant le fluide	-	-

Paramètres :

Nom	Description	Unité	Valeur par défaut
T_DRY	Température de sortie de l'échangeur à température paramétrable évacuant la chaleur vers les DRY	K	273.15 + 33
debit_chaud_dry	Débit de chaud à évacuer vers les DRY	kg/s	-
Q_flow	Flux de chaleur transmis au fluide au niveau des échangeurs à température paramétrable	W	-

Équations
Les pompes fan10 et fan14 traitent chacune la moitié du débit imposé sur le réseau 45°C/35°C, pour au total avoir un débit équivalent sur le réseau 33°C/43°C. La puissance électrique consommée par ces deux pompes du réseau 33°C/43°C est calculée d'après leurs données constructeur. L'échangeur à température paramétrable hex_DRY traite ce débit total selon le régime de température 33°C/43°C. Le flux de chaleur *Q_flow* transmis au fluide est alors récupéré au niveau de cet échangeur pour calculer la puissance électrique consommée par un système équivalent composé de 3 DRY de 1500 kWth : $$P_{elec_{DRY}} = Q_{flow} * X$$ $$X = \frac{-3 * 22 * 3.2}{3 * 1500} = \frac{-211.2}{4500}$$ Où X est un coefficient de proportionnalité calculé comme étant le rapport entre la puissance électrique installée du système équivalent composé de 3 DRY (en réalité celle des moto-ventilateurs) sur la puissance thermique totale pouvant être évacuée par ce système. Finalement, la puissance totale consommée par le système équivalent composé de 3 DRY est égale à la somme des consommations électriques déterminées. $$P_{elec_{DRY_{tot}}} = P_{elec_{DRY}} + P_{pompe_{tot}}$$

Tableau des révisions	
Création	AR, 11/05/2022
Simplification du modèle	AR, 03/06/2022
Ajout du schéma bloc	AR, 13/07/2022

B.7 Geothermal system

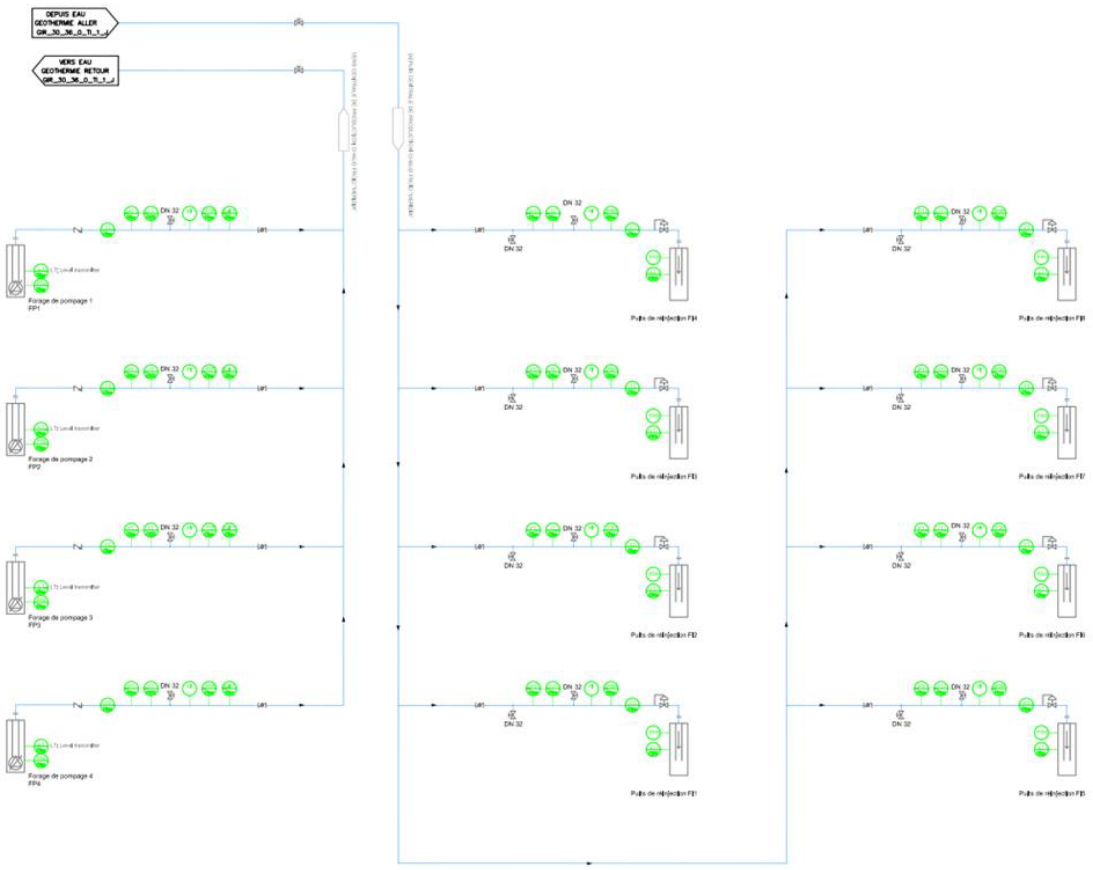
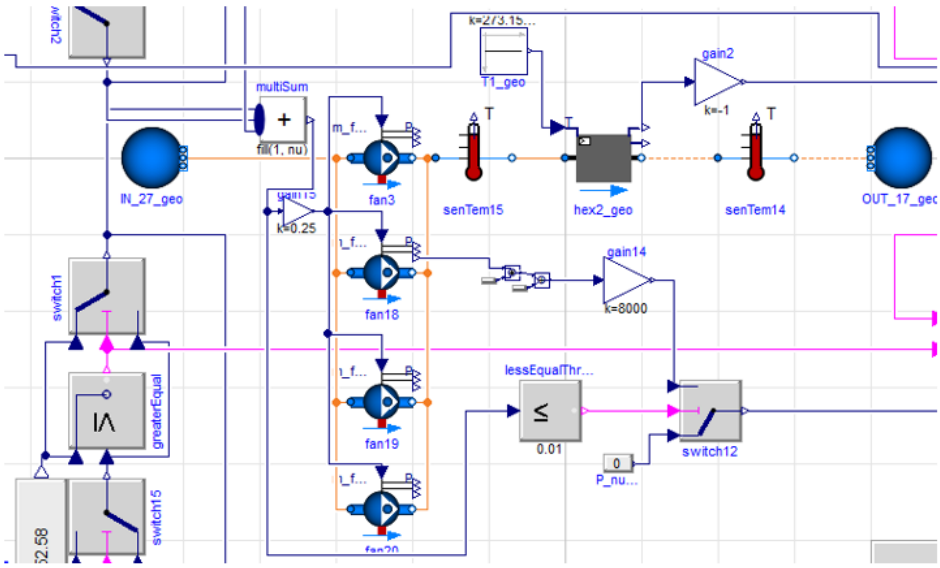


Figure B.3: Process and Instrumentation diagram for the geothermal drilling of MSE.

Objet : Géothermie
Phénomènes : Échanges de chaleur avec la nappe alluviale du Var
Hypothèses : Pas de perte thermique, non prise en compte des pertes de charge
Méthode :
Modèle : Syst_pro_complet_final_V2

Schéma du composant



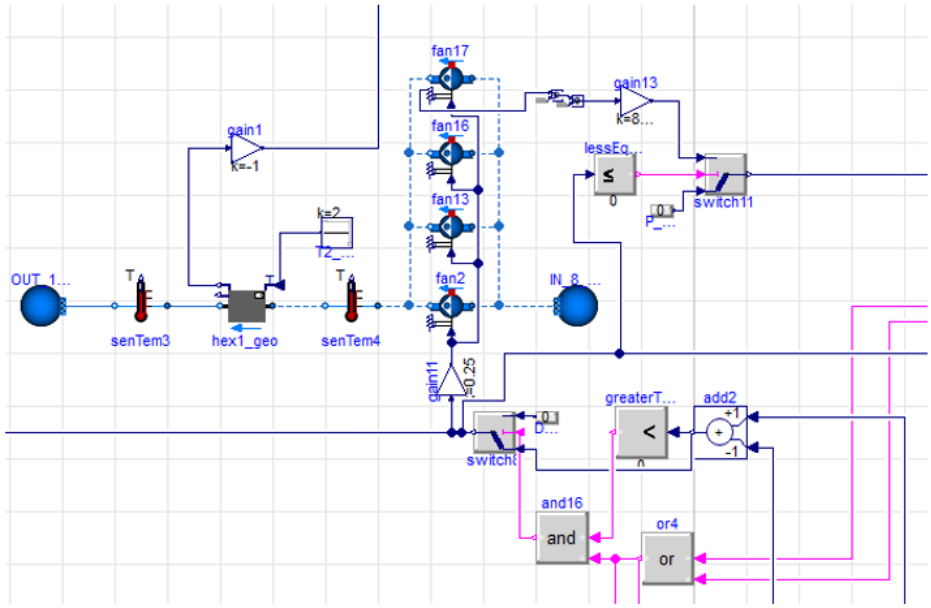
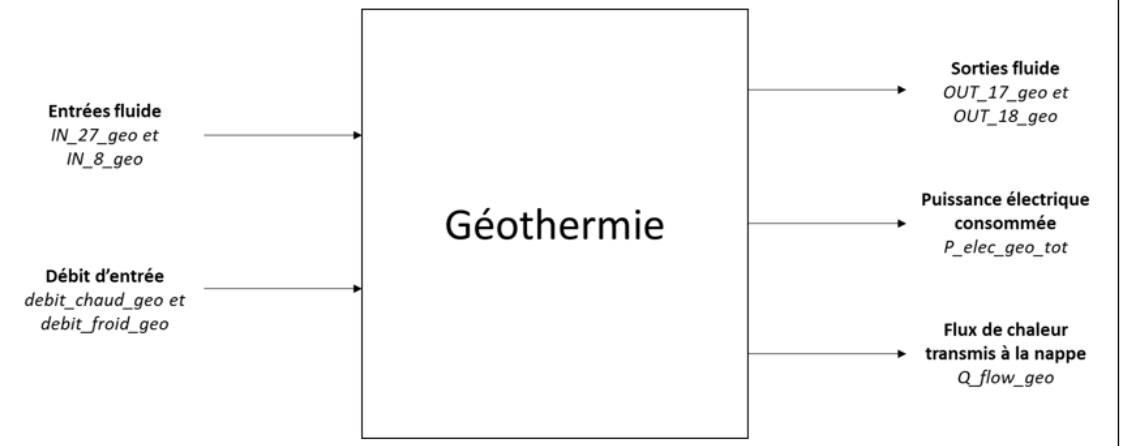


Schéma bloc



Connecteurs :

Nom	Description	Unité	Valeur
IN_27_geo	Source fournissant le fluide à une température fixée	-	-
OUT_17_geo	Frontière récupérant le fluide	-	-
IN_8_geo	Source fournissant le fluide à une température fixée	-	-
OUT_18_geo	Frontière récupérant le fluide	-	-

Paramètres :

Nom	Description	Unité	Valeur par défaut
T1_geo	Température de sortie de l'échangeur à température paramétrable évacuant la chaleur vers la géothermie	K	273.15 + 18
T2_geo	Température de sortie de l'échangeur à température paramétrable évacuant le froid vers la géothermie	K	273.15 + 17
debit_chaud_geo	Débit de chaud à évacuer vers la géothermie	kg/s	-
debit_froid_geo	Débit de froid à évacuer vers la géothermie	kg/s	-
Q_flow	Flux de chaleur transmis au fluide au niveau des échangeurs à température paramétrable	W	-

Équations

Les consignes des pompes (fan2, fan18, fan 19, fan 20 et fan3, fan13, fan 16, fan 17) sont imposées par *debit_chaud_geo* et *debit_froid_geo*, qui représentent respectivement la somme des débits de chaud à évacuer et la somme des débits de froid à évacuer.

La puissance électrique totale consommée par ces pompes ($P_{elec_{geo_{tot}}}$) est calculée d'après leurs données constructeur.

Les échangeurs à température paramétrable (hex1_geo et hex2_geo) traitent ces débits et fournissent en sortie un fluide à la température fixée par les consignes *T1_geo* et *T2_geo*.

Le flux de chaleur *Q_flow* transmis au fluide est alors récupérable au niveau des échangeurs dans les deux cas.

Finalement, le flux de chaleur transmis à la géothermie est déterminé, dans le cas de l'évacuation de chaud et de froid, de la manière suivante :

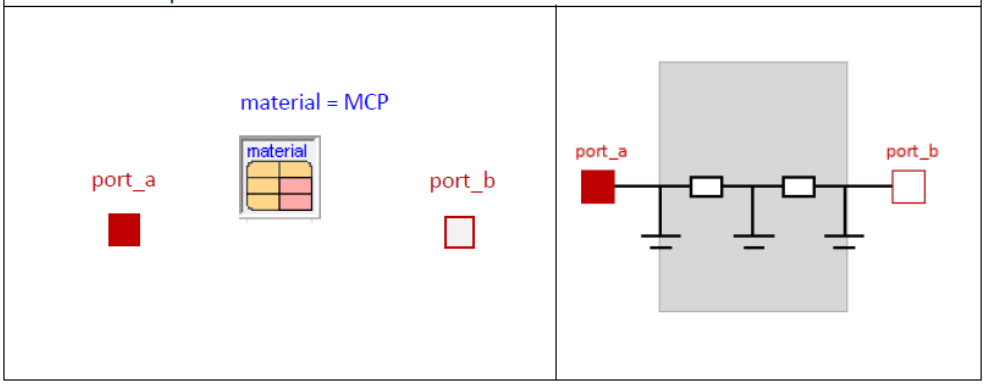
$$Q_{flow_{geo}} = -Q_{flow}$$

Tableau des révisions	
Création	AR, 11/05/2022
Ajout du schéma bloc	AR, 13/07/2022

B.8 PCM heat storage system

Objet : Stockage thermique par Matériaux à Changement de Phase
Phénomènes : Transfert thermique
Hypothèses : La conduction est transitoire et non stationnaire
Méthode : Le transfert de chaleur est calculé en se basant sur l'énergie spécifique interne du MCP. La relation entre T et u est donnée par interpolation de spline cubique, avec extrapolation linéaire.
Modèles de base: *Building.HeatTransfer.Conduction.SingleLayer / Building.HeatTransfer.Data.SolidsPCM.Generic*

Schéma du composant



Connecteurs :

Nom	Description
port_a	Fluide entrant
port_b	Fluide sortant

Paramètres du MCP :

Nom	Description	Unité	Valeur par défaut
x	Epaisseur du matériau	m	
k	Conductivité Thermique	W/(m.K)	
c	Capacité de Chaleur Spécifique	J/(kg.K)	
R	Résistance Thermique	M^2.K/W	x/k
TSol	Température de Solidification	K	
TLiq	Température de Fusion	K	
LHea	Chaleur latente de changement de phase	J/kg	
nSta	Nombres de variables d'états dans un matériau de référence de 0.2m de béton		3

Paramètres de SingleLayer :

Nom	Description	Unité	Valeur par défaut
A	Aire de transfert thermique	m^2	
T_a_start	Température initiale au port a	K	293.15
T_b_start	Température initiale au port b	K	293.15

Equations

Initalisation :

Calcul des points de support (ud,Td) et des dérivées aux points de support (dTdu)

Scale = 0.999, Tm1=TSol+(1-scale)*(TLiq-TSol)et Tm2=TSol+scale*(TLiq-TSol)

$$u_d = (c * scale * TSol, c * Tsol, c * Tm1 + \frac{LHea(Tm1-Tsol)}{TLiq-TSol}, c * Tm2 + \frac{LHea(Tm2-Tsol)}{TLiq-TSol}, c * TLiq + LHea,\ c * [TLiq + TSol(1 - scale)] + LHea)$$

$$T_d = (scale * TSol, TSol, Tm1, Tm2, TLiq, TLiq + TSol(1 - scale))$$

$$dT_{du} = (d1, d2, \ldots., dn), \quad \text{et} \quad \partial_i = \frac{Td_{i+1} - Td_i}{ud_{i+1} - ud_i}$$

Avec $d1 = \partial_1$, $dn = \partial_{n-1}$, $et\ di = \frac{\partial_{i-1} + \partial_i}{2}$ pour i =2..n-1 , n=taille(ud)

Calcul de Qflow (puissance en W), T (température en K) et u (énergie interne) :

$port_a.Q_{flow}$ = +Q_flow$_1$

$port_b.Q_{flow}$ = −Q_flow$_{end}$, avec end =nSta+1

$port_a.T = T_1\ et\ port_b.T = T_{nSta}$

Pour i = 1..nSta -1 (nR = nSta +1)

$$T_i - T_{i+1} = Qflow_{i+1} * RNod_{i+1} \text{ avec } RNod_i = \begin{cases} 0\ si\ i = 1\ ou\ i = nR \\ \frac{R}{2(nSta-2)}\ si\ i = 2\ ou\ i = nR - 1 \\ \frac{R}{nSta-2}\ sinon \end{cases}$$

$$\frac{du_i}{dt} = (Qflow_i - Qflow_{i+1}) * mInv_i, \quad et \quad mInv_i = \frac{1}{m_i}$$

Avec $m_i = A * x * d * \begin{cases} \frac{1}{2(nSta-1)}\ si\ i = 1\ ou\ i = nSta \\ \frac{1}{nSta-1}\ sinon \end{cases}$

Et $T_i = temp_u\,(u_d, Td, dTu, u_i)$

$\equiv\ LinearExtrapolation(u_i, ud_i, ud_{i+1}, Td_i, Td_{i+1}, dT_{du_i}, dT_{du_{i+1}})$

Cas 1 : $u_i > ud_i et\ u_i < ud_{i+1}$

$T_i = Td_i(2t^3 - 3t^2 + 1) + h * dTdu_i(t^3 - 2t^2 + t) + Td_{i+1}(-2t^3 + 3t^2) + h * dTdu_{i+1}(t^3 - t^2)$

Avec $h = ud_{i+1} - ud_i$ et $t = \frac{u_i - ud_i}{ud_{i+1} - ud_i}$
Cas 2 : $u_i \leq ud_i$
$T_i = Td_i + (u_i - ud_i)dTdu_i$
Cas 3 : $u_i \geq ud_{i+1}$
$T_i = Td_{i+1} + (u_i - ud_{i+1})dTdu_{i+1}$

Calcul de l'énergie : $\frac{dEnergie}{dt} = port_b.Qflow$
Conversion en kWh par division de 3600*1000
Calcul du SOC : SOC = Energie * 100 / Energie_totale, avec Energie_totale fixée à 1010kWh

Tableau des révisions de Building.HeatTransfer.Conduction.SingleLayer	
Ajout calcul énergie et SOC	KP, 6/08/2021

B.9 PV panels

Due to the lack of information regarding the specific characteristics of the PV systems that will be installed within the MSE eco-district, an equivalent system was considered with the following specifications for the solar panels:

* Unitary power of $400W_p$ (Watt peak),
* Mono-crystalline type
* Fixed mounting configuration
* Inclination of $39°$ for an optimal annual generation
* Azimuth of $0°$ for an optimal annual generation

The total power of the equivalent PV system is defined as follows:

* $500kW_p$ installed capacity
* system losses of 20%
* A footprint area of $2m^2$ per panel, resulting in a total area of $2500m^2$ for the 1250 panels.

The power generation data for this systems were extracted from PVGIS between the years 2005 and 2020 at hourly time steps. These data were then averaged to construct the time series used in the comprehensive simulation model. An overview of this PV generation time series is illustrated in figure B.4.

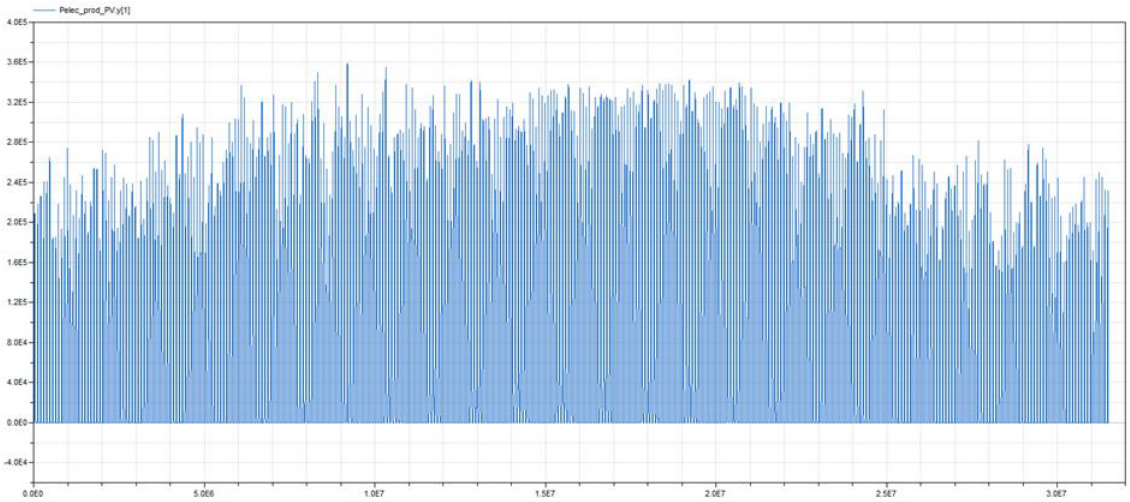


Figure B.4: An overview of the equivalent PV system's power generation (denoted $Pelec\_prod\_PV.y[t]$) over a complete year for the base scenario.

B.10 Battery energy storage system

Objet : Batterie
Phénomènes : Stockage d'énergie électrique
Hypothèses :
Méthode :
Modèle de base: *BuildingSystems.Technologies.ElectricalStorages.BatterySimple*

Schéma du composant

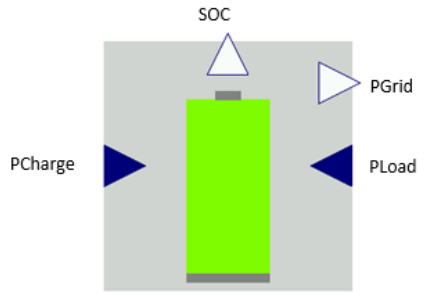
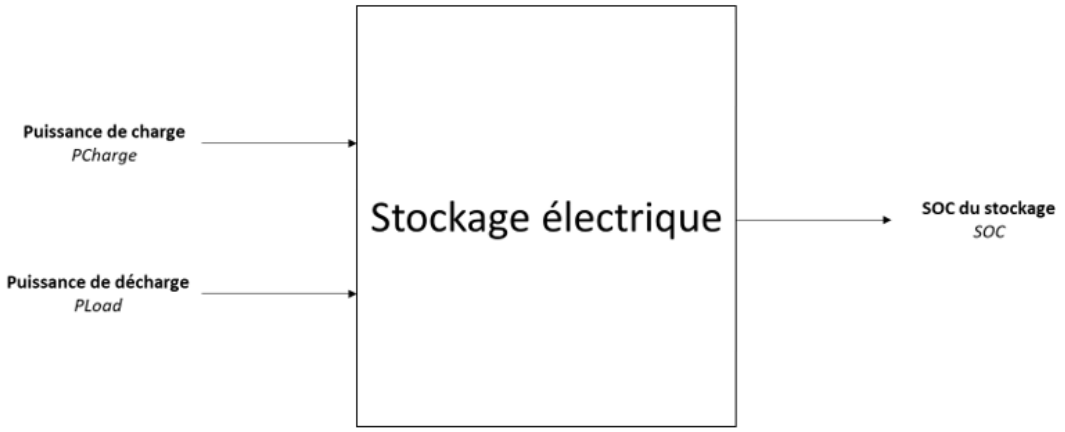


Schéma bloc



Connecteurs :

Nom	Description
PCharge	Puissance de la source (W)
PLoad	Puissance de la charge (W)
PGrid	Puissance électrique provenant du réseau (W)
SOC	État de charge du stockage

Paramètres :

Nom	Description	Unité	Valeur par défaut
nBat	Nombre de batteries	-	1
SOC_start	État de charge initial du stockage	-	1
E_nominal	Capacité nominale de la batterie	Wh	730*844

U_nominal	Tension nominale	V	730
SOC_min	Soc minimum lançant la charge automatique via PGrid	-	0.01
c	Ratio entre énergie disponible et capacité de stockage	-	0.4 (défaut)
k	Taux de batterie	s^{-1}	8/3600 (défaut)
etaCharge	Efficacité de charge	-	0.92 (défaut)
etaLoad	Efficacité de décharge	-	0.92 (défaut)
fDis	Facteur de perte	s^{-1}	0.1 (30*24*3600) (défaut)
PCharge_max	Puissance maximum de charge	W	730*1600
PLoad_max	Puissance maximum de décharge	W	730*1600
P	Coefficient de Peukert	-	1.05
a_mcr	Paramètre de taux de charge maximum	W/J	0.96/3600 (défaut)

http://www.ibpsa.org/proceedings/BS2019/BS2019_210463.pdf

Equations
Les équations décrivant le modèle de batterie utilisé ne sont pas détaillées.

Tableau des révisions	
Création	YBA, 14/12/2020
Adaptation des paramètres du modèle	AR, 13/05/2022
Ajout du schéma bloc	AR, 13/07/2022

B.11 Simplified model of the heat and cold storage systems

Objet : Stockage de chaud et de froid
Phénomènes : Échange de chaleur entre les stockages et les réseaux de chaud et de froid
Hypothèses : Pas de perte thermique, non prise en compte des pertes de charge
Méthode :
Modèle : Stockage_ch_fr_provisoire_V2

Schéma du composant

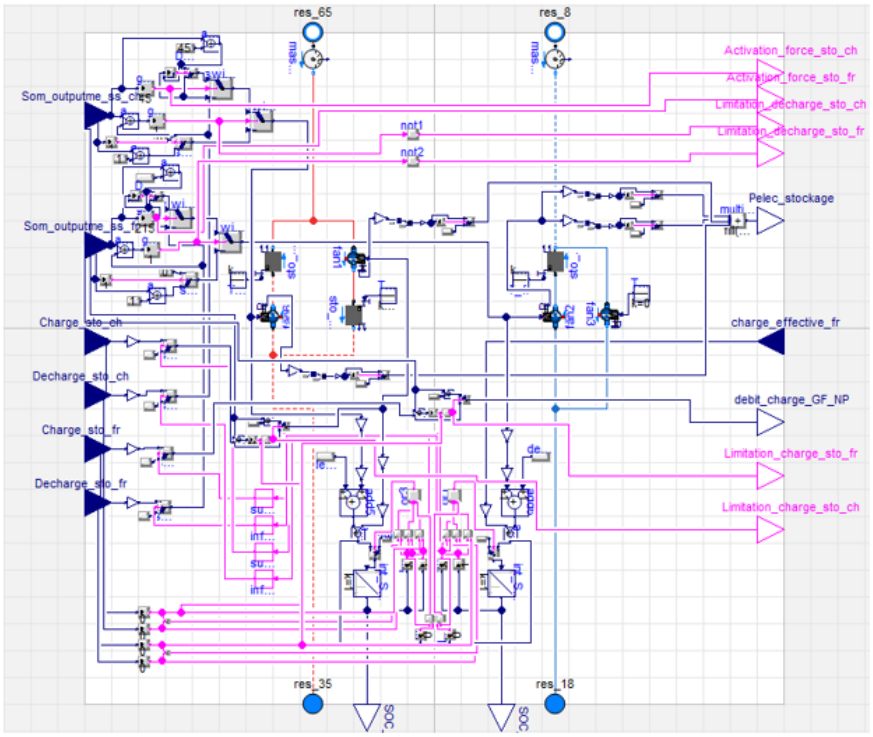


Schéma bloc

Connecteurs :

Nom	Description	Unité	Valeur
res_65	Port connecté à la partie 65°C du réseau de chaud	-	-
res_35	Port connecté à la partie 35°C du réseau de chaud	-	-
res_8	Port connecté à la partie 8°C du réseau de froid	-	-
res_18	Port connecté à la partie 18°C du réseau de froid	-	-
Som_outputme_ss_ch	Débit total appelé par les sous-stations de chaud	kg/s	-
Som_outputme_ss_fr	Débit total appelé par les sous-stations de froid	kg/s	-
Charge_sto_ch	Commande de charge du stockage de chaud	W	-
Decharge_sto_ch	Commande de décharge du stockage de chaud	W	-
Charge_sto_fr	Commande de charge du stockage de froid	W	-
Decharge_sto_fr	Commande de décharge du stockage de froid	W	-
Debit_charge_GF_NP	Consigne de débit de charge à transmettre au GF_NP	kg/s	-
Charge_effective_fr	Valeur de charge réelle du stockage retournée par le GF_NP	W	-
Activation_force_sto_ch	Booléen rendant compte de l'activation de la décharge forcée du stockage de chaud	-	-
Activation_force_sto_fr	Booléen rendant compte de l'activation de la décharge forcée du stockage de froid	-	-
Limitation_decharge_sto_ch	Booléen rendant compte de l'activation de la limitation de la décharge du stockage de chaud	-	-
Limitation_decharge_sto_fr	Booléen rendant compte de l'activation de la limitation de la décharge du stockage de froid	-	-
Limitation_charge_sto_ch	Booléen rendant compte de l'activation de la limitation de la charge du stockage de chaud	-	-
Limitation_charge_sto_fr	Booléen rendant compte de l'activation de la limitation de la charge du stockage de froid	-	-
SOC_sto_ch	SOC du stockage de chaud	-	-
SOC_sto_fr	SOC du stockage de froid	-	-
Pelec_stockage	Puissance électrique consommée par les auxiliaires des stockages	W	-

Paramètres :

Nom	Description	Unité	Valeur par défaut
debit_limite_chaud	Débit chaud maximum pouvant être traité par le système de production	kg/s	45
debit_limite_froid	Débit froid maximum pouvant être traité par le système de production	kg/s	215
T_ch	Température de sortie de l'échangeur à température paramétrable permettant la décharge du stockage de chaud	K	273.15 + 65
T_ch1	Température de sortie de l'échangeur à température paramétrable permettant la charge du stockage de chaud	K	273.15 + 35
T_fr	Température de sortie de l'échangeur à température paramétrable permettant la décharge du stockage de froid	K	273.15 + 8
T_fr1	Température de sortie de l'échangeur à température paramétrable permettant la charge du stockage de froid	K	273.15 + 18
Deperditions_sto_ch	Déperditions d'énergie journalières du stockage de chaud en % de E_max_ch	-	2%
Deperditions_sto_fr	Déperditions d'énergie journalières du stockage de froid en % de E_max_fr	-	2% (copié du stockage de chaud)
E_max_ch	Énergie maximale stockable dans le stockage de chaud	kWh	1200
E_max_fr	Énergie maximale stockable dans le stockage de froid	kWh	7590

Équations

Pour la partie stockage de chaud

La charge du stockage de chaud s'opère via l'utilisation d'une pompe qui prélève du fluide à 65°C avec un certain débit en sortie de système de production. Ce débit est ensuite traité par un échangeur à température paramétrable, dont la température est fixée à *T_ch1*. Cet échangeur permet au débit prélevé d'être réintégré en entrée de système de production à 35°C. Le débit à prélever en fonction de la puissance de charge du stockage de chaud est calculé de la manière suivante :

$$debit_{charge_{ch}} = \frac{Charge_{sto_{ch}}}{Cp_{eau} * dT}$$

Avec dT = 30 et $Cpeau$ = 4184 J/(kg.K)

La décharge du stockage de chaud est réalisée sur une branche en parallèle de la branche de charge. Elle a été modélisée en suivant le même principe que pour la charge mais de manière inversée en termes de températures et de prélèvement/réinjection de débit. On a donc également :

$$debit_{decharge_{ch}} = \frac{Decharge_{sto_{ch}}}{Cp_{eau} * dT}$$

Avec dT = 30 et $Cpeau$ = 4184 J/(kg.K)

Côté décharge du stockage de chaud, une boucle de régulation a été ajoutée pour pouvoir imposer la décharge du stockage de chaud lorsque le système de production ne peut subvenir seul aux besoins de chaud du réseau. Si *Som_outputme_ss_ch* est supérieur à *debit_limite_chaud* alors la régulation impose une décharge égale à la différence entre ces deux valeurs. Une autre boucle de régulation a été ajoutée pour permettre de limiter le débit de décharge du stockage dans le cas où ce dernier induirait un débit inférieur à 1 kg/s au niveau des TFP. Côté charge, une boucle de régulation a été ajoutée pour limiter cette dernière lorsque *Charge_sto_ch* impose un débit qui, s'il est sommé avec *Som_outputme_ss_ch,* ne peut être traité par le système de production. Les booléens suivants rendent compte de l'état de ces boucles de régulation :

- *Activation_force_sto_ch*
- *Limitation_decharge_sto_ch*
- *Limitation_charge_sto_ch*

Autrement, aucune limite de puissance maximale de charge ou de décharge n'a été imposée.

Le SOC du stockage est calculé comme étant l'intégration sur la durée de la simulation de la somme de *Deperditions_sto_ch* et des puissances de charge et de décharge, ramenées à *E_max_ch*. Ce SOC est contraint par une boucle de régulation pour être compris entre 0 et 1. À t = 0, il est fixé à 1.

$$SOC_{sto_{ch}} = 1 + \int_0^T (-Deperditions_{sto_{ch}} + \frac{Charge_{sto_{ch}}}{E_{max_{ch}}} + \frac{Decharge_{sto_{ch}}}{E_{max_{ch}}})dt$$

Avec un pas de simulation horaire et une contrainte imposant $SOC_{sto_{ch}} \in [0,1]$

Les puissances électriques consommées par les différents auxiliaires du stockage de chaud sont sommées avec celles du stockage de froid dans *Pelec_stockage*.

Pour la partie stockage de froid

La charge du stockage de froid s'opère uniquement via le GF_NP. Le débit à imposer côté évaporateur du GF_NP est calculé de la manière suivante :

$$debit_{charge_{GF\_NP}} = \frac{Charge_{sto_{fr}}}{Cp_{eau} * dT}$$

Avec *dT* = 10 et *Cpeau* = 4164 J/(kg.K)

La puissance effectivement traitée par le GF_NP est retournée au stockage via *Charge_effective_fr*. Cette valeur de puissance de charge effective est ensuite utilisée pour calculer le SOC.

La décharge du stockage de froid suit le fonctionnement de la décharge du stockage de chaud. On retrouve donc pour le débit à imposer :

$$debit_{decharge_{fr}} = \frac{Decharge_{sto_{fr}}}{Cp_{eau} * dT}$$

Avec *dT* = 10 et *Cpeau* = 4164 J/(kg.K)

Ici, encore, une boucle de régulation permet d'imposer la décharge du stockage de froid si le système de production ne peut répondre seul à la demande de froid du réseau. Si *Som_outputme_ss_fr* est supérieur à *debit_limite_froid* alors la régulation impose une décharge égale à la différence entre ces deux valeurs, comme pour le stockage de chaud. Une autre boucle de régulation a également été ajoutée pour permettre de limiter le débit de décharge du stockage dans le cas où ce dernier induirait un débit inférieur à 1 kg/s au niveau des TFP. Côté charge, une

boucle de régulation a été ajoutée pour limiter cette dernière lorsque *Charge_sto_fr* impose un débit qui, s'il est sommé avec *Som_outputme_ss_fr,* ne peut être traité par le système de production. Les booléens suivants rendent compte de l'état de ces boucles de régulation :

- *Activation_force_sto_fr*
- *Limitation_decharge_sto_fr*
- *Limitation_charge_sto_fr*

Autrement, aucune limite de puissance maximale de charge ou de décharge n'a été imposée.
Le SOC du stockage de froid est calculé comme pour le stockage de chaud via l'intégration sur la durée de la simulation de la somme de *Deperditions_sto_fr* et des puissances de charge et de décharge, ramenées à *E_max_fr*. Ce SOC est contraint par une boucle de régulation pour être compris entre 0 et 1. À t = 0, il est fixé à 1.

$$SOC_{sto_{fr}} = 1 + \int_0^T (Deperditions_{sto_{fr}} + \frac{Charge_{sto_{fr}}}{E_{max_{fr}}} + \frac{Decharge_{sto_{fr}}}{E_{max_{fr}}})dt$$

Avec un pas de simulation horaire et une contrainte imposant $SOC_{sto_{fr}} \in [0,1]$

Les puissances électriques consommées par les différents auxiliaires du stockage de froid sont sommées avec celles du stockage de chaud dans *Pelec_stockage*.

Tableau des révisions	
Création	AR, 12/05/2022
Ajout des calculs de SOC, d'éléments de régulation supplémentaires et de la prise en compte des synergies avec le GF_NP	AR, 03/06/2022
Ajout du schéma bloc	AR, 13/07/2022
Ajout d'éléments de régulation	AR, 29/07/2022

RÉSUMÉ

Cette thèse propose une approche de gestion de l'énergie basée sur l'Apprentissage par Renforcement Profond (DRL) pour les Systèmes Multi-Énergies Intelligents (SMEI). Le Système de Gestion Multi-Énergies Intelligente (SGMEI) est conçu pour optimiser la gestion des systèmes d'énergie flexibles, y compris le stockage de chaleur, de froid et d'électricité, ainsi que les systèmes de production dans les réseaux de chaleur et de froid, comm les Thermo-Frigo Pompes (TFPs). On propose ainsi l'application de cette approche sur l'étude de cas du projet Meridia Smart Energie (MSE), un projet réel de SMEI en cours de construction dans l'écoquartier de Nice Meridia, en France. L'agent DRL développé est comparé à un Contrôleur Prédictif (MPC) sur un premier cas d'étude simulé de SMEI simplifié, montrant que le DRL est capable d'approcher l'optimum théorique du MPC (à hauteur de 98%) en termes de réduction des coûts énergétiques. Cette étude suggère que le DRL est une approche prometteuse pour la gestion énergétique optimisée des SMEI. L'approche DRL est également appliquée sur un second cas d'étude à un jumeau numérique plus détaillé de MSE, développé sous Dymola, pour valider ces résultats sur un second cas d'étude plus complexe. Les futurs travaux porteront sur le transfert de l'apprentissage de la simulation à la réalité sur MSE et étendront l'application de cette approche à de nouveaux cas d'usage de SMEI.

MOTS CLÉS

Systèmes Multi-Energies Intelligents, Eco-quartiers, Smart Grids, Réseaux de Chaleur et de Froid, Systèmes de Gestion de l'Energie, Contrôle Optimal, Apprentissage par Renforcement Profond, Contrôle Prédictif.

ABSTRACT

This research introduces a Deep Reinforcement Learning (DRL)-based approach for the optimized energy management in Smart Multi-Energy Systems (SMES). A Smart Energy Management System (SEMS) is proposed to efficiently manage flexible energy systems, including heating, cooling, electricity storage, and District Heating and Cooling Systems. The Meridia Smart Energy (MSE) eco-district, a real-world SMES project in southern France, serves as the case-study. The DRL framework uses actor-critic architecture and is compared to Model Predictive Control (MPC). Results from a first simplified MSE simulation model show that DRL closely approximates MPC's theoretical optimum (within 98%) and even outperforms some realistic MPC variants. A more complex digital twin-based case-study further validates DRL's promise for SMES energy management. Future work includes real-world integration, exploring additional objectives, and expanding to other SMES use-cases.

KEYWORDS

Smart Multi-Energy Systems, Eco-districts, District Heating and Cooling Systems, Energy Management Systems, Optimal Control, Deep Reinforcement Learning, Model Predictive Control.