Claudia
Domínguez-
Barbero

Institute for Research in Technology
Higher Technical School of Engineering (ICAI)

Modeling and optimizing isolated microgrids using Reinforcement Learning techniques

Modeling and optimizing isolated microgrids using Reinforcement Learning techniques

Author: Ms. Claudia Domínguez-Barbero
Supervisor: Prof. Dr. Javier García-González
Co-supervisor: Prof. Dr. Miguel-Ángel Sanz-Bobi

Madrid | November 2025

A mis padres.

Acknowledgements

Summary

Microgrids are gaining increasing importance in the energy transition due to several key advantages. They can enhance the resilience of the power grid, enable access to electricity in developing countries, and reduce greenhouse gas emissions by facilitating higher penetration of renewables. The concept of a microgrid arises from grouping various distributed energy resources close to the loads, effectively forming a smaller-scale power network. These microgrids can supply energy to loads without relying on a complex interconnection system, making them more resilient to natural disasters and other grid-related issues. Moreover, surplus energy from one microgrid can be shared with neighboring microgrids. Another benefit of microgrids is their ability to operate independently from the main grid, making them viable for supplying power to areas where extending existing power lines is not feasible. However, they also present certain challenges. Because microgrids frequently include renewable energy sources, which by nature are intermittent, their reliability in power generation can be uncertain. For this reason, microgrids often include other non-renewable generation components to support the renewables when they cannot fully meet demand. Furthermore, energy storage systems are key to maximizing the potential of renewable energy by storing surplus power and then releasing it when renewable generation is scarce. Coordinating these heterogeneous components to achieve efficient management represents a significant challenge.

Traditional energy management methods in power grids typically rely on complex optimization models that require precise forecasts of demand and renewable generation, as well as frequent calibration and adjustments when operating conditions change. When these same techniques are applied to microgrids—due to their smaller scale—it becomes questionable whether the resource-intensive approach is really worthwhile. Smaller companies looking to install distributed generation components often lack the knowledge and resources for efficient energy management, leading them to simpler methods such as rule-based systems. These management solutions are commonly referred to as Energy Management System (EMS).

With the rise of Artificial Intelligence (AI), problems once considered highly complex or unsolvable are now being tackled with notable success. In the microgrid management literature, various AI techniques can be found that address these challenges (and others that are even more specific and technical). Among these techniques, Deep Reinforcement Learning (DRL) has gained significant traction across different industries, and in some cases stands out as a promising alternative to conventional optimization methods. This thesis examines the impact of these DRL techniques when applied to microgrids for energy management using two particular algorithms: the Deep Q-Network (DQN) and the Twin-Delayed Deep Deterministic Policy Gradient (TD3).

By framing the EMS design problem as a sequential decision-making task, the DRL agent—rep-

resenting the EMS—can learn control policies through direct interaction with the microgrid environment, thereby reducing the need for demand and renewable generation forecasting. The core strength of DRL lies in its capacity to refine control strategies based on trial-and-error experiences, enabling the system to learn from the environment's dynamics and to discover increasingly optimal policies. These policies can autonomously adapt to fluctuations in renewable output and demand.

Over the course of three fundamental studies, the thesis investigates different DRL algorithms and configurations, described in Chapters 3, 4, and 5: Chapter 3 demonstrates the viability of an EMS based on the DQN algorithm—one of the most commonly used in DRL—to manage an isolated microgrid and a historical dataset of solar irradiation and demand. By exploring various actions, such as turning a diesel generator on or off, or charging and discharging a battery, the DRL agent learns from the outcomes of those actions to develop an optimal policy aimed at minimizing operational costs and penalties for unserved power. This chapter also addresses an important technical detail: state representation in partially observed environments, such as microgrids. This aspect involves striking a balance between providing sufficient information for learning without overwhelming the agent with unnecessary noisy data. Additionally, to evaluate the quality of the DRL algorithm, its performance is compared against an optimization model with a perfect forecast, providing a benchmark for how close the DRL solution is to an ideal standard. The results show that the DQN-based EMS can effectively operate the microgrid when using a Convolutional Neural Network (CNN) architecture to deal with the temporal dimension inherent in the time-series of the renewable generation.

Chapter 4 builds on the idea that certain microgrid control variables (e.g., generator set-points or battery charge level) are more naturally represented in continuous terms. Therefore, this study transitions from DQN—which requires discretizing actions—to TD3, an algorithm capable of handling continuous action spaces. The results demonstrate more precise control and better performance in both a small-scale microgrid and a standard benchmark microgrid, such as Low-Voltage Microgrid Benchmark (CIGRE). Although TD3 can exhibit instabilities during training—common to Deep Deterministic Policy Gradient (DDPG)-based methods—proper hyperparameter tuning significantly improves stability, allowing the agent to fully leverage fine-grained continuous control actions. Additionally, a methodology for efficiently optimizing hyperparameters is proposed, which positively impacts the algorithm's performance.

Chapter 5 recognizes the limitations of simplified battery representations by incorporating a nonlinear battery loss model into the previous TD3-based approach. This chapter shows that TD3 not only adapts to the more complex dynamics but also exploits them to enhance battery and overall microgrid operation. Moreover, a more realistic interaction is introduced between the EMS and a real-time control system, demonstrating the agent's capacity to coordinate effectively with other systems within the microgrid. The results reveal a notable reduction in battery energy losses and operational costs when employing the nonlinear model, without significantly increasing computational overhead.

Taken together, this dissertation underscores the adaptability, scalability, and effectiveness of DRL-based approaches for microgrid energy management. By reducing dependence on model forecasting and heavy optimization models, DRL offers a more agile and robust solution suitable for real-world applications. Future work will focus on integrating demand-response mechanisms, exploring multi-agent approaches for larger systems, and examining market participation for grid-connected microgrids. These avenues hold promise for advancing clean energy objectives and building more resilient, cost-effective, and autonomous power systems through DRL-based EMS.

Resumen

Las microrredes están adquiriendo cada vez más importancia en la transición energética ya que poseen muchas ventajas clave, como por ejemplo la de incrementar la resiliencia de la red eléctrica, facilitar el acceso remoto de la energía en países en desarrollo, y reducir las emisiones de gases de efecto invernadero al facilitar la penetración de las renovables. El concepto de microrred nace al agrupar diversos recursos de energía distribuidos cerca de las cargas, formando, en esencia, una red eléctrica, pero mucho más pequeña. Estas microrredes permiten alimentar la demanda eléctrica sin tener que depender de un complejo sistema de interconexión, protegiendo el suministro de energía ante desastres naturales, u otros problemas que pudiesen darse en una red convencional. Además, estas microrredes pueden compartir la energía sobrante con otras microrredes cercanas. Otra ventaja viene dada por su posibilidad de operar de forma aislada, por lo que pueden ser desplegadas como alternativa de electrificación de zonas rurales, donde extender la red eléctrica principal sea inviable. Por otro lado, las microrredes presentan ciertos desafíos, como por ejemplo, tener que depender de fuentes de energía renovable que en muchos casos son variables a lo largo del tiempo y están sujetas a incertidumbre, lo cual afecta a la fiabilidad del suministro. Por este motivo, las microrredes suelen incluir otros sistemas de generación que, aunque no provengan de fuentes renovables, apoyan a la generación renovable en caso de que por ellas mismas no puedan suplir la demanda. Además, los sistemas de almacenamiento de energía también son clave a la hora de aprovechar al máximo la generación renovable, almacenando el exceso de energía, y posteriormente utilizando la energía ya almacenada en momentos de escasez. Toda esta coordinación entre los componentes heterogéneos de una microrred representa un desafío en si misma, por lo que hacerlo de manera eficiente para, por ejemplo, reducir el coste de explotación en el que se incurra, añade un nivel de dificultad adicional. Para ello, se suelen utilizar sistemas de gestión de la energía, en inglés, Energy Management System (EMS).

Los métodos tradicionales de gestión de la energía en los sistemas eléctricos de potencia suelen basarse en modelos de optimización complejos que exigen predicciones precisas de la demanda y la generación renovable, así como una frecuente calibración en caso de que las condiciones operativas varíen a lo largo del tiempo. La aplicación de esos modelos tradicionales al caso más pequeño de una microred, podría no ser posible por razones tanto técnicas (como por ejemplo no disponer de un optimizador o un modelo de predicción para la localización específica de la microred), como económicas (el coste de implantar esos modelos podría ser mayor que el ahorro conseguido. Por ello, cuando hay necesidad de diseñar un EMS en una microrred, se suele optar por sistemas más sencillos y baratos. Existen microrredes instaladas, que por lo general tienen una configuración de componentes muy simple de gestionar, o que utilizan un sistema basado en reglas para controlar la misma microrred.

La llegada de la Inteligencia Artificial, en inglés, Artificial Intelligence (AI), permite abordar satisfactoriamente problemas complejos, o incluso imposibles de resolver. En la literatura de microrredes se pueden encontrar técnicas de AI que son aplicadas para resolver la gestión de la energía en microrredes, además de muchos otros problemas de índole similar. Unas de estas técnicas que están teniendo mucho impacto en la industria en general son las técnicas de Aprendizaje por Refuerzo Profundo, en inglés Deep Reinforcement Learning (DRL), y que, en algunos casos, sirven como una alternativa prometedora frente a los métodos de optimización convencionales. En esta tesis se estudia el impacto de estas técnicas en su aplicación de microrredes para la gestión de la energía a través de dos algoritmos de DRL: la Deep Q-Network (DQN) y el Twin-Delayed Deep Deterministic Policy Gradient (TD3).

Al formular el problema de diseñar un EMS como un problema de toma de decisiones secuencial, el EMS se modela como el "agente" dentro del contexto del DRL. El agente entonces aprenderá políticas de control óptimas mediante de la interacción con la microrred (llamado de forma genérica "entorno" en el contexto del DRL), y eliminando la dependencia en modelos de predicción de demanda y generación intermitente. La esencia del DRL radica en su capacidad para ajustar su estrategia de control a partir de experiencias obtenidas por ensayo y error, lo que permite al agente de DRL aprender las dinámicas del entorno y encontrar políticas cada vez más óptimas. Estas políticas sirven para responder de manera autónoma a las fluctuaciones de la generación renovable y la demanda.

A lo largo de esta tesis se investigan diferentes algoritmos y configuraciones de DRL:

En el capítulo 3 se demuestra la viabilidad de un EMS basado en una DQN—uno de los algoritmos más comúnmente utilizados en el DRL—para gestionar una microrred aislada, utilizando un conjunto de datos históricos de irradiación solar y demanda con un perfil típico residencial. A través de la exploración de diferentes acciones, como por ejemplo el encender o apagar el generador diésel, o la de cargar o descargar la batería, el agente de DRL aprenderá a partir de observar el resultado de tomar estas acciones, y diseñará su política óptima con el objetivo de minimizar los costes de la operación, teniendo en cuenta las penalizaciones por incurrir en energía no suministrada. Además, con el objetivo de medir la bondad del algoritmo de DRL, éste se compara con un modelo de optimización clásica, que aunque tenga conocimiento perfecto del futuro, es decir, de la generación renovable y de la demanda, algo que no es realista en la realidad, puede proporcionar una visión aproximada de la bondad del algoritmo de DRL. Aunque en la tesis se discute más en detalle, los casos de estudio muestran que el EMS diseñado con la DQN es capaz de operar la microrred con muy buenos resultados. Además, en este capítulo se analiza el diseño del estado en entornos parcialmente observables, caso que ocurre en el problema en cuestión, y que es un detalle técnico muy importante en el modelado. Esta cuestión técnica busca establecer un equilibrio entre informar al agente lo suficiente como para que pueda tener un correcto aprendizaje, pero sin sobredimensionar la cantidad de información posible recibida, lo que supondría que el agente tendría que generalizar sobre un espacio hiperdimensional bastante lejos de la realidad, decrementando considerablemente la eficacia del algoritmo.

El capítulo 4 viene motivado por la idea de que ciertas variables de control en la microrred, por ejemplo, el punto de operación del generador o de la batería, tienen una naturaleza de rango continuo. Por ello, en este capítulo se pasa de aplicar la DQN—que requiere discretizar las acciones—a utilizar el TD3, capaz de manejar directamente espacios de acción continuos. Esta mejora ofrece un control más preciso de la microrred, obteniendo mejores resultados, probados esta vez en dos microrredes de diferentes tamaños: un tamaño residencial, muy similar a la utilizada en el capítulo

3, y otra microrred utilizada de referencia en la academia, la Low-Voltage Microgrid Benchmark (CIGRE). Por otro lado, el TD3 puede presentar inestabilidades durante el entrenamiento típicas de métodos basados en el algoritmo de Deep Deterministic Policy Gradient (DDPG). Por ello, es necesario un ajuste adecuado de los hiperparámetros del algoritmo, y así encontrar aquellos que mejoren esa estabilidad y aprovechar al máximo el control sobre todo el rango de acciones continuas. Además, este capítulo propone una metodología para optimizar los hiperparámetros de forma eficiente, ya que la misma búsqueda de estos hiperparámetros es otro problema complejo de por sí.

El DRL se apoya en el uso de redes neuronales, capaces de capturar patrones no lineales en sus aplicaciones. En el modelado de las baterías se tiende a simplificar al utilizar comportamientos linales, como por ejemplo, en las pérdidas, cuando la realidad presenta un comportamiento no lineal. El capítulo 5 adopta un modelo no lineal de pérdidas de energía en las baterías con el objetivo de analizar el potencial que alberga el DRL en este aspecto. En este capítulo se observa que el TD3 es capaz de capturar estas dinámicas no lineales, resultando en una reducción de pérdidas de energía al operar la batería, así como la reducción de los costes de la operación de la microrred con respecto a la alternativa de modelado utilizada anteriormente. Además, se adopta una interacción más fiel entre el EMS y un sistema de control secundario y primario, que actuaría en tiempo real, demostrando la capacidad de tomar decisiones eficientes en coordinación con el resto de sistemas participantes en la microrred. El estudio muestra una notable reducción en las pérdidas de energía de la batería, y en los costes de operación al utilizar el modelo no lineal, sin que ello suponga un aumento significativo en la carga computacional.

En su conjunto, entre los resultados de esta tesis destacan la adaptabilidad, escalabilidad y eficacia de las aproximaciones basadas en el DRL para la gestión de la energía en microrredes. Al eliminar la dependencia de métodos de predicción complejos necesarios por los modelos de optimización clásica, el DRL aporta una solución más ágil y robusta, adecuada para aplicaciones reales cuando los recursos escaseen.

En trabajos futuros se propone incorporar mecanismos de respuesta de la demanda, explorar enfoques multiagente para sistemas de mayor escala, y el profundizar en la participación de las microrredes en mercados eléctricos cuando estén conectadas a la red. Estas iniciativas abrirán nuevas posibilidades para que los sistemas de gestión de la energía basados en DRL contribuyan al avance de los objetivos de energía limpia y a la construcción de sistemas eléctricos más resilientes, rentables y autónomos.

Contents

List of Figures

List of Tables

Glossary

A3C Asynchronous Advantage Actor-Critic 34

AC Alternating Current 34

ACER Actor-Critic with Experience Replay 34, 35

ADP Approximated Dynamic Programming 31, 34, 38

AI Artificial Intelligence iii, vi, 8, 12, 83, 106

ANN Artificial Neural Network 12, 60

ARIMA AutoRegressive Integrated Moving Average 37

BDN Branching Dueling Q-Network 35, 38

C51 Categorical DQN 36, 38

CAISO California Independent System Operator 33

Cap&OpEx Capital & Operational Expenditure 38

Cap&OpEx/B Capital & Operational Expenditure with Battery utilization costs 38

CapEx Capital Expenditure 30

CIGRE Low-Voltage Microgrid Benchmark iv, vii, 33, 41, 64, 76, 77, 82, 89, 90, 92–94, 99, 104, 107

CNN Convolutional Neural Network iv, 12, 13, 18, 43, 44, 55, 60, 104, 106

CO2 Carbon Dioxide 30

CV Computer Vision 18

DDPG Deep Deterministic Policy Gradient iv, vii, 19, 24, 34, 38, 63–65, 74, 75, 82

DDQN Double Deep Q-Network 19

DER Distributed Energy Resources 8–10, 20, 36, 103

DG Distributed Generator 7, 20, 21, 28–30, 35, 36, 50

Di-Gen Diesel Generator 29, 32, 34, 35, 43–46, 49, 52–54, 58, 64, 67, 75–77, 83

DL Deep Learning 7, 12, 13, 17–19, 69, 108

DP Dynamic Programming 18, 31, 32

DQN Deep Q-Network iii, iv, vi, xv, 1, 4, 18, 19, 24, 31, 33–36, 38, 43, 44, 49, 55–57, 59–61, 63, 64, 67, 69–77, 82, 104–107

DRL Deep Reinforcement Learning iii, iv, vi, vii, 7, 8, 12, 13, 17–19, 21–24, 30, 35, 36, 40, 41, 43, 44, 47, 49, 50, 52–54, 58, 60, 63–65, 67–69, 73–75, 78, 82–84, 88, 89, 99, 102–109

ECM Equivalent Circuit Model 85

ED Economic Dispatch 30, 31, 37

EM Electrochemical Model 85

EMA Exponential Moving Average 16

EMS Energy Management System iii–vii, 9, 11–13, 15, 18, 20, 21, 23, 24, 31, 35–37, 40, 44, 46, 47, 49, 50, 55, 58, 60, 63, 64, 66, 68, 72, 74, 82–85, 87–89, 93, 102, 104, 106, 107

ENS energy not supplied 21, 28, 31, 37, 45, 50, 54, 77, 78, 84, 107

ER Experience Replay 18, 19, 34, 36, 55, 70, 99

ESS Energy Storage System 8, 9, 20, 21, 31–36, 44, 45, 48–51, 64, 65, 83, 85

ExtRa-Trees Extremely Randomized Trees 33

FC Fuel Cell 20, 21, 29, 33, 35, 36, 43, 45, 53, 58, 77, 78

FQF Fitted Q-Factorization 36

FQI Fitted Q-Iteration 33, 38

GMM Gaussian Mixture Model 64, 71, 106

GRU Gated Recurrent Units 13

HILP High Impact Low Probability 30

HP Heat Pump 33

IQN Implicit Quantile Network 36

KDE Kernel Density Estimation 92, 93

LCOE Levelized Cost of Electricity 28, 31, 38

Li-ion Lithium-Ion 20, 31, 44–46, 49, 51, 53, 58, 65, 75, 84, 85, 87, 89, 99, 102, 105, 107

LP Linear Programming 38

LSTM Long Short-Term Memory 12, 35

MC Markov Chain 18, 37

MCTS Monte Carlo Tree Search 35

MDP Markov Decision Process 13–15, 17, 20, 21, 30–34, 43, 47, 65, 67, 73, 84

MILP Mixed Integer Linear Programming 12, 29, 30, 38

MIP Mixed Integer Programming 35

MIQP Mixed Integer Quadratic Programming 54, 56, 57, 60, 73, 90, 91, 104

MISOCP Mixed Integer Second-Order Cone Programming 35

ML Machine Learning 12, 13

MLP Multi-Layer Perceptron xv, 12, 13, 69, 70

MPC Model Predictive Control 12

MSE Mean Squared Error 72

MT Microturbine 20, 29, 33, 77, 78

NADAM Nesterov-accelerated ADaptive Moment Estimation 56

NG Natural Gas 36

NN Neural Network 12, 18, 19, 30, 31, 34, 35, 55, 64, 69, 70, 72, 77, 89

NSGA-II Non-dominated Sorting Genetic Algorithm II 30, 38

OpEx Operational Expenditure 38

OpEx/B Operational Expenditure with Battery utilization costs 38

OpEx/E Operational Expenditure with Emissions 38

OpEx/H Operational Expenditure with Human thermal discomfort costs 38

OpEx/L Operational Expenditure with Energy loss costs 38

OpEx/T Operational Expenditure with Technical violation costs 38

OU Ornstein-Uhlenbeck 37

PCC Point of Common Coupling 34, 44, 45

PEM Point Estimate Method 37

PER Prioritized Experience Replay 36

POMDP Partially-Observable Markov Decision Process 31, 32, 40, 47, 84, 88, 89, 107

PPO Proximal Policy Optimization 34, 38

PSO Particle Swarm Optimization 29, 38

PV Photovoltaic Panel 8, 9, 11, 20, 21, 30–37, 46, 52, 54, 58, 73, 75–78, 85

PXM Proton Exchange Membrane 53

QR-DQN Quantile Regression DQN 36

RDPG Recurrent Deterministic Policy Gradient 34

RE Relative Error 57, 91

ReLU Rectified Linear Unit 34, 69

RES Renewable Energy Sources 7, 8, 32, 34, 35, 45, 46, 48, 50

RL Reinforcement Learning xiii, 12–23, 30–33, 43, 45, 46, 57, 60, 65, 67, 68, 72, 87, 88, 105, 106, 108, 109

RNN Recurrent Neural Network 12, 35

RO Robust Optimization 38

SDN Stochastic Dual Network 35

SGD Stochastic Gradient Descent 70

SMPC Stochastic Model Predictive Control 38

SoC State of Charge xiii, 20, 33, 34, 52, 60, 68, 75, 76, 85, 93

SOM Second-Order Model 37

TD Temporal-Difference 16, 19

TD3 Twin-Delayed Deep Deterministic Policy Gradient iii, iv, vi, vii, xv, 19, 24, 63–65, 67, 69, 70, 72–78, 82, 84, 87, 89, 91–94, 102, 104–107

ToU Time of Use 36, 37

TPE Tree-structured Parzen Estimator 64, 71, 106

TRPO Trust Region Policy Optimization 35

UC Unit Commitment 31, 37

VPP Virtual Power Plant 10, 11

WT Wind Turbine 20, 21, 30–35, 37, 76, 77, 85

Acronyms

GAMS General Algebraic Modeling Language 54, 56, 57

HOMER Hybrid Optimization of Multiple Energy Resources 35

IEA International Energy Agency 10

IEEE Institute of Electrical and Electronics Engineers 53

LINEAR Local Intelligent Networks and Energy Active Regions 33

SB3 Stable-Baselines3 69, 70

Chapter 1

Introduction

1.1 Motivation

The electric power system is undergoing a profound technological transformation driven by the increasing integration of Renewable Energy Sources (RES) and Distributed Generator (DG), the shift from passive consumers to active prosumers, and the digitalization of grid infrastructures. These changes are largely motivated by the broader energy transition currently underway, which aims to decarbonize and modernize the energy sector (Hirsch et al., 2018).

In the context of microgrids (defined in Section 1.2), this transformation makes their consideration and development increasingly relevant. Microgrids enable localized energy management and greater resilience, but they also introduce new challenges, such as their optimal design, operation, and stability, as well as their deployment and integration into larger energy networks. A more holistic challenge lies in their scalability to multi-agent systems, where coordination and autonomous decision-making become critical (Dev et al., 2025).

Within the scope of isolated microgrids, these challenges are even more pronounced. Limited resources and higher variability in supply and demand make their management and optimization more complex. Traditional optimization techniques can effectively solve certain aspects of the management problem, but they often fall short when it comes to the autonomous, plug-and-play deployment of such systems. Moreover, the smaller scale of microgrids often reduces the economic interest of large stakeholders, thereby increasing the need for cost-effective solutions that are affordable for small-scale operators and communities (Mariam et al., 2016).

To address these limitations, more flexible and adaptive methods are required—methods that can operate with minimal supervision, adapt to evolving conditions, and scale effectively. This is where Deep Reinforcement Learning (DRL) offers a promising alternative. Unlike classical optimization, which often requires precise models and centralized computation, DRL can learn directly from data and interaction with the environment, enabling decentralized control strategies and autonomous decision-making (Hasan et al., 2025).

However, DRL techniques are still relatively new, and research on their application in power systems remains limited. Because they rely on Deep Learning (DL), ensuring their stability and robustness in practical settings can be a complex and challenging task, often requiring extensive experimentation and fine-tuning (Henderson et al., 2018; Islam et al., 2017; Shakya et al., 2023).

A key motivation driving this work is the belief in the potential of DRL to address a wide array of challenges in power systems, not only in microgrids, but focusing on the latter (Dinata et al., 2024). The field offers fertile ground for innovative approaches, with opportunities to enhance efficiency, resilience, and scalability of these systems (Y. Huang et al., 2025; Sandeep et al., 2025). Furthermore, the growing interest among leading researchers in integrating DRL with classical optimization techniques highlights the relevance and timeliness of this work.

From a technical perspective, DRL's promise lies in its deployment flexibility and its ability to provide greater autonomy, potentially reducing infrastructure and maintenance costs. It also offers the capacity to assist in areas requiring rapid responses in decision-making, thereby streamlining the work of operators and technicians (Hadi et al., 2025).

The distributed nature of DRL is another compelling motivation. As computing systems have transitioned to distributed architectures for improved scalability and efficiency, power systems are undergoing a similar transformation. Distributed generation and active demand management play increasingly major roles in modern grids. By applying DRL, there is a significant opportunity to enhance the efficiency, resilience, and overall performance of these distributed systems (Y. Zhang et al., 2025).

In summary, this thesis reflects a convergence of motivations: an initial curiosity about the application of DRL in power systems, a commitment to addressing its practical challenges, and a broader vision to advance its role in that field. These motivations underline the importance of continuing this line of research, bridging the gap between Artificial Intelligence (AI) and classical optimization, and paving the way for more accessible, scalable, and intelligent solutions in the energy sector.

1.2 Microgrids and Energy Management Systems (EMS)

Microgrids are localized energy systems composed of Distributed Energy Resources (DER)—such as Photovoltaic Panels (PVs), Energy Storage Systems (ESSs), conventional generators—and controllable loads, integrated via power electronics and advanced control technologies. These systems can operate either connected to the main grid or in islanded mode, offering flexibility and autonomy in energy management (R. Lasseter & Paigi, 2004; R. H. Lasseter et al., 2003; Ton & Smith, 2012).

Microgrids are considered crucial enablers of the energy transition due to their ability to enhance resilience, support high shares of RES, and improve energy access and reliability (R. Lasseter & Paigi, 2004). Their deployment contributes to decarbonization objectives while fostering the development of smarter and more efficient energy infrastructures (Dev et al., 2025; Venkataramanan & Marnay, 2008).

The integration of RES, particularly solar and wind power, has emerged as a pivotal strategy to mitigate the environmental impacts associated with fossil fuel consumption. This increasing penetration of RES, combined with the evolving roles of consumers and generators within the power grid, is transforming the structure and operation of the electric power industry. In parallel, other energy-intensive sectors are undergoing profound changes, driven by the imperative to reduce carbon emissions and transition towards decentralized, efficient, and sustainable energy systems (A. Q. Huang et al., 2011; (IEA), 2024)

For instance, in 2024, the European Union reached a record high reliance on renewable energy for electricity generation, with renewables producing 1,313 TWh—47.3% of total electricity—surpassing fossil fuels, which fell to a historic low of 810 TWh (29.2%), and continuing to outpace natural gas for the second consecutive year. PV power saw the largest annual growth (+48.3 TWh or +19.6%), followed by hydropower (+41.9 TWh or +11.6%) and wind power (+9.5 TWh or +2.0%), with five-year increases of 150.2% and 32.7% for solar and wind, respectively. This rapid expansion of variable renewable sources, coupled with the steady decline of fossil fuel generation, underscores the need for flexible, decentralized energy systems capable of integrating distributed generation, enhancing grid resilience, and managing local supply—demand dynamics—making microgrids an increasingly critical component of Europe's clean energy transition (Eurostat, 2025).

In a typical microgrid—as a combination of DER, ESS, and loads within a controlled network—power electronic interfaces interconnect these components and an Energy Management System (EMS) coordinates operations. Furthermore, the inclusion of fast-acting power electronics (e.g., inverters, converters) is noted as critical for integrating various DERs and maintaining power quality (Asian Development Bank, 2020). Despite their advantages, microgrids face several challenges in deployment and operation:

- Microgrids introduce new stability and control challenges. With high penetration of inverter-based renewables and distributed generators, traditional protection and control schemes need adaptation. Key issues include bi-directional power flows (no longer a simple one-way grid flow), which can confuse protection devices, and limited inertia in inverter-dominated systems, which makes maintaining frequency and voltage stability more complex. Transitioning between grid-connected and islanded modes changes the short-circuit characteristics and can cause transient instability if not managed properly. Researchers underscore that low-inertia microgrids require advanced control (e.g., fast control or synthetic inertia) to ensure stable operation during disturbances (Dev et al., 2025; Srivastava et al., 2021).

- The intermittency of renewable DER and the variability of loads make real-time balancing in microgrids challenging, especially for small or islanded systems. Maintaining power supply reliability despite rapid fluctuations is difficult without sufficient reserves or storage. For example, Eyimaya and Altin (2024) discusses the "unpredictability and erratic nature" of renewable generation and demand, and notes that EMS strategies must handle this uncertainty to keep the microgrid stable and cost-effective. Ensuring an adequate power balance at all times may require sophisticated forecasting, fast-responding storage, and controls to prevent outages or quality issues when generation dips or spikes unexpectedly.

- Microgrid protection must deal with variable fault levels and network reconfigurations. Conventional protection schemes can fail due to protection "blinding" or false tripping in the presence of DERs and multi-directional flows. Adaptive and intelligent protection strategies (using advanced relays or even machine learning) are being developed to cope with these issues (Srivastava et al., 2021). Furthermore, many microgrids rely on communication networks (for centralized or distributed control and for coordination with the main grid). Communication failures or cyber-attacks pose a serious operational risk. A review by Akinwale et al. (2021) finds that delays, packet loss, or outages in the communication links can "degrade optimal operations of islanded microgrids." The authors advocate for robust control solutions that can maintain stable operation even if communications falter, rather than relying solely on high-speed networking. In practice, this means designing controllers with local autonomy or fallback modes (e.g., droop control inverters that can keep sharing load without communications). Robust, failsafe control and fallback mechanisms are essential to ensure safety and stability under faulted communication or islanded conditions (Altin & Eyimaya, 2021).

- Despite technical viability, microgrids face non-technical hurdles. High initial capital costs and uncertain return on investment can deter projects. Regulatory and business model challenges also loom large. Dev et al. (2025) surveys economic challenges, noting the "financial burdens associated with costly components, infrastructure, and maintenance," as well as market and regulatory hurdles that make it hard to monetize microgrid services. Because microgrids do not always fit neatly into existing utility regulations, developers can encounter permitting obstacles or a lack of compensation mechanisms for grid services. Organizationally, there may be a lack of expertise in designing, operating, and managing microgrids, especially for community-led projects—requiring capacity building and new partnerships. Even in developed markets, the business case can be complex: an industry brief by NCEL (2024) points out that microgrids often face "large political and legal constraints" that Virtual Power Plants (VPPs) do not[1]. Policy support (grants, favorable regulations) and standardized solutions (to reduce costs) are often needed to overcome these barriers (Tumilowicz, 2024).

Microgrids serve different needs across diverse contexts:

- In areas with limited or no access to a central grid, microgrids provide a crucial solution for electrification. They can be deployed as modular "micro-utilities" that bring power to villages or clusters of customers. For instance, in the World Economic Forum, Elliott (2024) describes a start-up deploying solar microgrid units in Nigerian communities, connecting small businesses to "clean, affordable electricity through solar-powered micro-utilities", which the International Energy Agency (IEA) sees as vital for expanding energy access (IEA, 2019). These remote microgrids typically use local renewable resources (solar, small hydro, etc.) plus storage and are often built to be scalable. They offer a technology that enables infrastructural bypass in developing regions, improving quality of life and economic opportunities by providing reliable electricity where traditional grid expansion is impractical or costly.

- Microgrids are increasingly used by communities and critical facilities in developed power systems to enhance resilience against grid outages. They can island during emergencies (such as natural disasters or grid failures) and keep key loads powered (NCEL, 2024). More generally, hospitals, emergency response centers, data centers, and campuses are deploying microgrids to ensure uninterruptible power for critical services (Tumilowicz, 2024). By disconnecting from a failing grid and using on-site generation and storage, these resilient microgrids provide a lifeline in disasters. This capability to "maintain power during outages" is a chief driver for microgrid adoption in storm-prone or mission-critical applications.

[1] Although the concepts of microgrid and VPP are often related, they differ in scope and purpose. A microgrid is a local, physical energy system that integrates distributed generation, storage, and loads within a confined electrical network, capable of operating either connected to or isolated from the main grid. In contrast, a VPP is a virtual, software-based aggregation of geographically dispersed DERs that are coordinated and optimized as a single entity, primarily to participate in electricity markets and provide grid services. In short, microgrids ensure local reliability and autonomy, while VPPs enables large-scale coordination and market integration.

- In modern smart grids, microgrids also serve as flexible building blocks that support broader grid operations. When grid-connected, they can function as VPP or active grid participants. They help with peak load management by either shaving local peaks or even exporting power/support to the grid during high demand periods (NCEL, 2024). They also enable advanced functionalities like demand response: for example, a utility can signal a microgrid to adjust its load or generation to assist the grid, and the microgrid's controller can respond by dispatching its batteries or shedding non-critical loads. Programs now exist (e.g., in Colorado and California) where networks of home and commercial microgrids/batteries are aggregated to provide grid services for compensation (NCEL, 2024; Tumilowicz, 2024). Additionally, microgrids are integral to the vision of smart cities and decarbonized future grids—they can manage local renewable generation (like solar PV on buildings), integrate electric vehicle charging, and improve overall efficiency by optimizing energy locally. As one industry analysis put it, microgrids operating in concert with advanced software can "offload and use their own energy during periods of peak demand," easing stress on the main grid. In summary, beyond serving their own customers, microgrids in aggregate help stabilize the larger grid and facilitate the integration of high shares of distributed renewable energy. Each microgrid can be a controllable cell in a smart grid, contributing to a more flexible and reliable energy infrastructure.

The central focus of this thesis is the EMS, which serves as the decision-making core for managing and coordinating the components of a microgrid. An EMS—in the context of a hierarchical control architecture—oversees the real-time balance between supply and demand. It monitors generation and loads, forecasts system behavior, optimizes resource dispatch, and ensures reliability, power quality, and economic efficiency. Although the term EMS is sometimes used to describe the entire control system, including the underlying electronic infrastructure, in this thesis, it specifically refers to the tertiary control loop—the highest level of control in a power system (Gao et al., 2019). In a typical microgrid, three control levels are distinguished:

- Primary control, which provides fast, local regulation of voltage and frequency through decentralized strategies such as droop control and reactive power control.

- Secondary control, which restores frequency and voltage deviations after primary control actions, using centralized or distributed communication-based strategies to maintain energy balance and optimize power flow.

- Tertiary control, or EMS in this context, which operates over longer time horizons—typically from minutes to hours—focusing on economic dispatch, resource scheduling, and coordination with the main grid.

In this thesis, the EMS problem is defined as addressing the challenges of this top-level control layer under uncertainty, while accounting for the dynamic behavior of distributed energy resources and their interactions with lower-level controls. This requires the ability to simulate microgrid operation in realistic conditions to evaluate different control strategies.

A wide range of techniques have been proposed in the literature to solve the EMS problem (Ahmad et al., 2023; Aladesanmi & Ogudo, 2023). These include:

- Rule-based approaches, which are simple to implement but limited in adaptability.

- Expert systems using fuzzy logic, which can incorporate heuristic knowledge and handle uncertainty in measurements.
- Classical optimization methods, such as Mixed Integer Linear Programming (MILP) combined with forecasting techniques—often embedded within a Model Predictive Control (MPC) framework—to perform economic dispatch and scheduling.
- Data-driven AI methods, including DRL, which can learn control policies directly from interaction with the environment. These methods remain experimental but have shown strong potential for handling complex, nonlinear, and stochastic systems.

In this work, DRL is explored as a potential approach to the EMS problem. While classical optimization methods remain powerful and widely used—especially when accurate models and forecasts are available—DRL offers complementary strengths. In particular, it can adapt to changing system conditions, operate without a complete model, and learn control strategies through interaction with the environment. These characteristics make it an interesting candidate for scenarios with high uncertainty and variability. However, DRL also presents challenges, including stability concerns, high computational demands, and the need for extensive training data, which must be considered when assessing its suitability for practical deployment.

1.3 Deep Reinforcement Learning in Microgrids

1.3.1 Deep learning (DL)

DL (LeCun et al., 2015) is a subset of Machine Learning (ML), which in turn is a branch of AI that focuses on algorithms and statistical models enabling computers to perform specific tasks without explicit instructions. At its core, DL is inspired by the structure and function of the human brain; in DL, these structures are called Artificial Neural Networks (ANNs) or simply Neural Networks (NNs). This approach leverages multiple layers of interconnected nodes, or neurons, to model complex patterns in data.

The concept of NNs dates back to the 1940s with the introduction of the first mathematical model of a neuron by McCulloch and Pitts (1943), and later with the well-known perceptron (Rosenblatt, 1958). However, the field of DL has seen significant advancements only in the last two decades, primarily due to increases in computational power, availability of large datasets, improvements in the algorithms for efficient parameter fitting, frameworks for rapid development and training of NNs such as TensorFlow (Martín Abadi et al., 2015) and PyTorch (Paszke et al., 2019), and many other tools developed by an active research and open-source community.

In this thesis, DL is employed primarily because the chosen methodology is DRL, which combines the principles of DL and Reinforcement Learning (RL). While the integration of DL within DRL will be discussed in detail later, it is worth noting here that DL provides the capacity to approximate complex, high-dimensional functions required in RL tasks, enabling the processing of rich, unstructured inputs and the generalization across diverse states.

A wide variety of neural network architectures exist, each suited to different types of input data and learning objectives. Among these are Multi-Layer Perceptron (MLP) (Rosenblatt, 1958), Convolutional Neural Network (CNN) (LeCun et al., 1989), (Lecun et al., 1998), Recurrent Neural Network (RNN) (Long Short-Term Memorys (LSTMs) Hochreiter & Schmidhuber, 1997 and

Gated Recurrent Unitss (GRUs) Chung et al., 2014), and transformer-based models (Vaswani et al., 2017). In this thesis, only convolutional architectures are used, as they are well-suited for processing structured data with local correlations, whether spatial or temporal in nature. Specifically, the structured state representations derived from the EMS simulation in this work are organized along a temporal dimension, making a one-dimensional CNN a natural choice for extracting local temporal features before the decision-making stage.

A convolutional neural network can be regarded as a specialized form of an MLP in which some of the connections and parameters are constrained to exploit locality in the input domain. Instead of fully connecting every neuron in one layer to every neuron in the next, convolutional layers apply learnable filters that are convolved with the input to detect local patterns. This structure enables parameter sharing and translation invariance, which significantly reduces the number of parameters and improves generalization when working with correlated data, whether in space or time. In the context of this thesis, these properties make 1D CNNs particularly effective for feature extraction from temporally structured state representations before feeding them into the decision-making layers of the DRL algorithm.

In summary, DL in this work serves as the function approximation backbone of the DRL framework, with one-dimensional convolutional networks chosen for their efficiency and inductive biases toward capturing local temporal dependencies. The subsequent sections introduce the principles of RL, setting the stage for the combined DRL approach adopted in this thesis.

1.3.2 Reinforcement learning (RL)

Reinforcement Learning (RL) techniques are optimal control techniques based on data and are used in various types of problems that, in general, are modeled as Markov Decision Processs (MDPs) (R. S. Sutton & Barto, 2018). This field is often classified as a subfield inside ML, since there is a learning process, which in this case is commonly referred to as learning "by trial and error" (Kaelbling et al., 1998). RL's core concept begins from a *tabula rasa*—although some expert components may also be added (C. Tang et al., 2025), or a pre-trained model (Black et al., 2024)—representation of knowledge—meaning it has no prior information about the environment—and must interact with it to gather data in order to learn (Silver et al., 2018). This learning paradigm, so-called "trial and error", involves an agent that explores the environment with some goal (the "trial"), collects experience, and then uses that experience to improve its policy (the "error" correction). The internal knowledge of the agent is updated iteratively through this process, which continues until a predefined learning objective is achieved. This process is similar to how a baby would learn to achieve different tasks. In conclusion, "trial and error" learning is the process of exploring the world and evaluating the outcomes of the actions taken to understand which sequence of actions should be chosen and thus carry out the task successfully. Figure 1.1 depicts a general diagram in RL, showing the interactions between the agent and the environment.

As a side note, the RL notation is complex and differs depending on the context. In economics, the notation is different from engineering, and also, many times, there is an abuse of notation in order to increase the clarity of the equations. One resource that can help to grow the insight of how RL works is the famous book of R. S. Sutton and Barto (2018). Additionally, the website of spinning-up from OpenAI is an excellent resource to recap these concepts with an extensive and more detailed description (Achiam, 2018). Along this thesis, the preferred notation follows R. S. Sutton and Barto (2018).

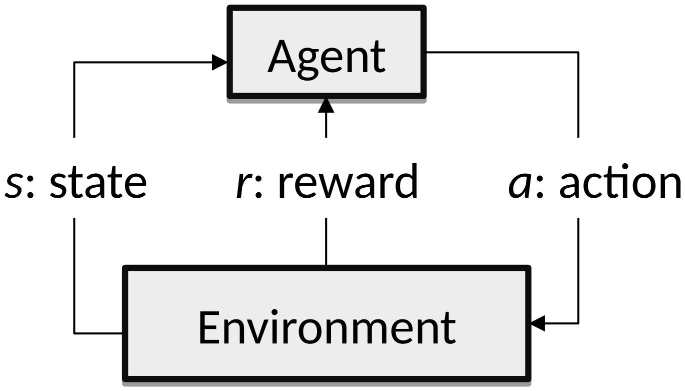


Figure 1.1: General interactions in RL.

1.3.2.1 Agent and environment

An MDP can be generally defined as a 4-tuple ($\mathcal{S}$, $\mathcal{A}$, $\mathcal{R}$, p) where $\mathcal{S}$ is the set of states s of the problem, $\mathcal{A}$ the set of possible actions to carry out during the process, $\mathcal{R} \subset \mathbb{R}$ the reward set and p the probability function to transition between states (Kaelbling et al., 1998). MDPs model systems where an agent interacts with its environment. In this system, the agent interacts with the environment through actions, and the agent observes the state of the system. Every time the agent takes an action, it gets a reward associated with that action and the current system state. A pedagogical example is the lunar lander game, included in the Gymnasium suit (Towers et al., 2023), which has some videos on its website and an explanation of how it works[2]. A spaceship must land on an uneven surface on the moon. In this problem, the agent represents the spaceship controls, i.e., the thrusters' power and direction, and the environment represents the surface delimitation of the moon and the position of the ship, where the latter is altered by the agent. The goal is to land the spaceship in the designated landing zone, marked by two flags, without crashing into the moon, while minimizing the total amount of fuel consumed.

In an MDP, the set of states $s \in \mathcal{S}$ contains each of the states of the system that hosts the problem. In the lunar lander example, the set of states is an 8-element tuple: the position of the ship in the x and y axis of the ship relative to the surface of the moon, the linear velocity in the x and y axis of the ship, the angle of the ship relative to the vertical, the angular velocity of the ship, and two boolean values indicating if the legs of the ship are in contact with the surface of the moon.

The set of actions $a \in \mathcal{A}$ includes all the agent's possible actions. In the lunar lander example, an action belongs to a set of 4 elements: do nothing, turn on the main engine, turn on the left engine, and turn on the right engine. These actions allow the control and guidance of the ship toward the landing zone. It is important to note that actions are distinct from states, although they are related. Additionally, actions must be elements that can be controlled within the environment. For example, one cannot control gravity or the moon's position in the lunar lander. Similarly, direct control of the ship's position is not allowed; only the force applied to it is.

The reward function $r : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow \mathcal{R}$ models the expected signal that the agent perceives after each action a and will depend on the current state of the system s and the reached state s'.

[2] https://gymnasium.farama.org/environments/box2d/lunar_lander/

This function is essential for the learning process, as different reward structures can significantly impact the algorithm's training dynamics and overall performance. A positive value will mean a positive reinforcement for the agent; on the contrary, a negative one will mean a penalty. In the lunar lander, the reward function rewards the ship's proximity to the landing zone, so if the ship gets closer, it will get a reward, and if it moves away, it will receive a penalty. In addition, fuel use is penalized, and if the ship lands in the safe zone, it will receive a hefty reward, but if it crashes, it will receive a large penalty.

The probability transition function $p(s'|s,a)$ is defined as a probability function $p : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow [0,1]$ where $\sum_{s' \in S} p(s'|s,a) = 1 \forall s \in \mathcal{S}, a \in \mathcal{A}$, which models the system's behavior according to its internal state and the decision made by the agent involved. In RL, this function is assumed unknown to the agent, and it is extracted explicitly (model-based) or implicitly (model-free) during the interaction between the agent and its environment. Suppose the chosen RL method explicitly extracts the transition function, i.e., the method explicitly incorporates a method to approximate the transition function to use it later to learn the optimal policy. In that case, the method is classified as model-based. In contrast, the method is classified as model-free when the policy function is directly learned (or the value function) using, for instance, bootstrapping from the Bellman equation (Howard, 2018; R. S. Sutton & Barto, 2018). Learning the transition function offers the advantage of enabling more precise and efficient planning. Moreover, if the approximation of the transition function is inaccurate, the resulting planning process will also be suboptimal (R. S. Sutton, 1990).

In the episodic case, a set of terminal states, denoted as Ω, is defined to represent the states at which an episode ends. When this set is empty, the problem has no designated terminal states and is therefore classified as a continuing problem. Conversely, problems with a non-empty Ω are referred to as episodic. For example, in the lunar lander environment, the set Ω comprises three distinct subsets: states in which the ship successfully lands in the safe zone, states in which it crashes, and states in which it moves off-screen. It is worth noting in advance that the EMS problem addressed later in this thesis is formulated as a continuing problem.

There is a huge variety of problems in the literature already implemented as MDPs. Perhaps the most popular source of RL problems is the Gymnasium suit (Towers et al., 2023), which contains a series of RL problems for students to practice and learn about RL. It is worth mentioning that Gymnasium was previously known as OpenAI Gym but was acquired by the Farama Foundation. Gymnasium is an open-source platform that allows researchers and developers to test and compare different RL algorithms in a variety of test environments. This framework is a common standard for modeling.

1.3.2.2 Agent's policy learning

In RL, the agent coexists with the environment and interacts with it in order to maximize the expected return. The policy maps states and actions, and it represents the agent's decision based on the observation from the environment state. In order to improve the policy, the agent can experience a learning process, where the policy changes (R. S. Sutton & Barto, 2018).

In RL, each change in the policy is called a learning step, where π_{t+1} is the new policy when that learning step is applied to π_t. In this sense, there is at least one optimal policy π^* according to a reward function in a particular MDP. The optimal policy is the one that maximizes the expected return (Eq. (1.1)) in the long term.

Mathematically, the policy is defined as $\pi : \mathcal{S} \to \mathcal{A}$, where $\pi(s)$ is the agent's action (denoted with the letter a) in state s. The return function is defined as the cumulative reward and formulated as

$$G_t \doteq \sum_{k=t+1}^{T} \gamma^{k-(t+1)} r_k \tag{1.1}$$

where γ is the discount factor. The undiscounted return is when $\gamma = 1$.

Using the Bellman equation (Howard, 2018), the value function of a particular state can be defined as

$$v_\pi(s) \doteq \mathbb{E}\left[G|S_t = s\right] = \mathbb{E}_\pi\left[r(s,a,s') + \gamma v_\pi(S_{t+1} = s')\right] \tag{1.2}$$

v is the state-value function for a given policy π, and s, a, and s' are the current state, the agent action, and the next state, respectively, and G is previously defined in (1.1).

Another version of the value function is the state-action pair value function $Q(s,a)$

$$Q_\pi(s,a) \doteq \mathbb{E}\left[r(s,a,s') + \gamma \max_{a'} Q_\pi(s',a')\right] \tag{1.3}$$

that can be derived from the Bellman equation (1.2), using the definition

$$q_\pi(s,a) \doteq \mathbb{E}\left[G|S_t = s, A_t = a\right] = \mathbb{E}_\pi\left[r(s,a,s') + \gamma v^\pi(S_{t+1} = s')\right] \tag{1.4}$$

where the single-value function $v()$ in each state corresponds to the best of the values in the state-action pair Q-function $Q()$ for that state (Watkins & Dayan, 1992). Moreover, the policy π can be defined as

$$\pi(s) \doteq \arg\max_a \left[Q_\pi(s,a)\right] \tag{1.5}$$

The learning process between RL algorithms can slightly differ among them, but generally they follow the same structure. The agent gathers experience from the environment and uses this experience to update the value function and improve the policy. There are also other algorithms that directly work with the policy function, i.e., called policy-gradient, but they will take less importance in this thesis given the need to simplify (R. S. Sutton et al., 2000), and the Actor-Critic algorithms, which combine both strategies (also used in this thesis). In the value-based algorithms, these experiences are generally defined as a tuple (s,a,r,s'), being r the reward obtained after taking action a in state s and reaching state s'. The information gathered from experience can be incorporated into the Bellman equation to compute the error of the value function or the Q-function, as given by (1.6).

$$\delta \doteq r + \gamma \max_{a'} Q_\pi(s',a') - Q_\pi(s,a) \tag{1.6}$$

This error, called the Temporal-Difference (TD) error (Seijen & Sutton, 2014), is then used to update the Q-function approximation function, thereby refining the policy to minimize the discrepancy. The usual technique to update this function is to apply the Exponential Moving Average (EMA) formula shown in (1.7), where α is the learning rate, usually constant.

$$Q(s,a) \leftarrow Q(s,a) + \alpha\delta \tag{1.7}$$

Another key concept in the RL framework is the trajectory. A trajectory, denoted as τ, is a sequence of experiences generated during a single run. It can be expressed as $\tau = (s_0, a_0, r_0, s_1, a_1, r_1, s_2, a_2, r_2, \ldots)$. In essence, a policy generates a continuous sequence of experiences—referred to as a trajectory—which represents all the information collected by the agent while interacting with the environment until reaching a terminal state.

1.3.2.3 Tabular and parameterized RL

One kind of RL algorithms, also quite basic to learn RL theory, are called tabular algorithms, as for instance, the Q-learning (Watkins & Dayan, 1992). This means that for each state or state-action pair, depending on whether the V or Q function is used, there is a value associated with it, and it can be stored in a table. These algorithms do not make any assumptions about other states or actions when updating the value function. This configuration has some nice properties, like the convergence to the optimal policy. However, they have some drawbacks, like the curse of dimensionality, where the number of states grows exponentially with respect to the number of state dimensions (R. S. Sutton & Barto, 2018).

In the tabular setting, RL algorithms are memory-intensive because each state—or each state–action pair—must store a separate value. Moreover, states are updated independently, which prevents the reuse of information gained from similar states and therefore limits generalization. A common solution to this limitation is to employ function approximation, a technique that enables the value function to generalize to previously unseen states (R. S. Sutton et al., 2000).

Function approximation, however, introduces its own drawbacks, such as the loss of guaranteed convergence to the optimal policy in most cases (R. S. Sutton, 1995), although recent research about this topic has been made (P. Xu & Gu, 2020; S. Zhang et al., 2023). Nonetheless, the success of DL has spurred the development of DRL algorithms, which have proven effective in tackling problems with large state and action spaces in practice (Mnih et al., 2015). In addition, the powerful representational capabilities of DL allow these methods to capture complex patterns in data, making them particularly well-suited for real-world problems that often involve intricate, non-convex dynamics.

DRL algorithms are much more suitable for the challenges addressed in this thesis, rather than using RL. This justification is better clarified in the following sections.

1.3.3 Deep reinforcement learning (DRL)

As noted in Section 1.3.2, RL methods are often applied to problems under uncertainty, frequently in settings where prior knowledge of the underlying system dynamics is limited or unavailable. While many approaches are model-free, others explicitly learn or use a model of the environment, as in model-based RL (Kaelbling et al., 1996). In most formulations, structural assumptions are still made—most commonly that the problem can be described within the MDP framework, which offers a well-defined mathematical formalism for sequential decision-making. However, this assumption does not always hold in real-world applications, where environments may be partially observable, nonstationary, or exhibit non-Markovian dependencies (Kaelbling et al., 1996).

Domain knowledge, when available, can be incorporated to guide the learning process, for example, by designing reward functions or constraining the action space. While such interventions can accelerate learning, they also introduce the risk of biasing the policy or inducing unintended behaviors if the design is flawed (Ibrahim et al., 2024).

At its core, RL combines aspects of Dynamic Programming (DP)—deriving value functions from a model of the environment—and Markov Chain (MC) methods—estimating returns from sampled experience. Although these connections help motivate the theoretical underpinnings of RL, they also highlight shared challenges: DP suffers from the curse of dimensionality in large state spaces, while MC can be sample-inefficient. RL inherits some of these difficulties, though advances such as function approximation and DRL have extended its applicability to problems with large or continuous state and action spaces (R. S. Sutton & Barto, 2018).

While RL offers a powerful framework for sequential decision-making, its deployment in real-world scenarios remains challenging due to computational demands, slow convergence, and scalability constraints. Moreover, evaluating learned policies and estimating computational complexity can be difficult, as training typically involves exploring a vast solution space without immediate feedback on performance. These factors make the practical development of RL methods a nontrivial task.

DRL addresses some of these limitations by integrating RL with DL, using NNs to approximate value functions or learn policies directly. This combination enables the handling of high-dimensional state and action spaces, the processing of inputs such as images or sensor data, and improved generalization across similar tasks (Mnih et al., 2015). Nevertheless, these benefits come with trade-offs: models often become less interpretable, training requires significant computational resources, and the learning dynamics can be highly unstable. Out of curiosity, in the literature, some recent works about model explainability can be found (Banker & Mesbah, 2025; Zolman et al., 2024).

In this thesis, DRL techniques are explored within the specific context of EMS in microgrids. The problem is examined within a controlled, reduced scope to systematically evaluate the effectiveness of the proposed approach.

1.3.4 DRL techniques used in this thesis

The Deep Q-Network (DQN) is one of the first DRL techniques that was successfully applied to a large set of problems, in particular, 49 ATARI video games, and was compared with the human performance in these video games (Mnih et al., 2015). The proposed DQN in that work combines the Q-learning algorithm with a CNN to approximate the value function from the screen pixels. The idea of adding convolutional layers arises particularly from the field of Computer Vision (CV), where there is a spatial correlation in the input image to the NN. In addition, to achieve convergence and improve the stability of learning, techniques such as Experience Replay (ER) and a *target network* are added. From this success, and together with the popularity increase of NNs in many areas, the field of DRL has grown rapidly, and in particular, in the area of robotics for its ability to process images and generalize in high-dimensional learning environments.

Two important details in the DQN are:

- ER is a technique designed to facilitate generalized learning of the Q-function across the state space. During online interaction, data gathered from the experience is stored in a large buffer known as replay memory. At each learning step, a batch is randomly sampled from this buffer. Retaining past experience improves data efficiency, as it can be reused multiple times for learning. Moreover, training on data consecutive in time tends to be inefficient due to their strong temporal correlations. By allowing the selection of non-consecutive data, ER mitigates this issue, further enhancing data efficiency. The main hyperparameters associated with this technique are the size of the replay buffer and the batch size used in each learning step (Mnih et al., 2015).

- *Target network* is a method whereby the NN used as a target of the Q-function is frozen every certain number of steps when calculating the TD error. This technique improves the stability of learning (Mnih et al., 2015).

When configuring the hyperparameters of DQN—and of DRL algorithms in general—a different experience is required compared to that used in DL settings. This stems from the instability of the online learning process in DRL, unlike the more stable nature of supervised or regression-based training. A notable example is the choice of minibatch size: a minibatch refers to a small, randomly sampled subset of experiences used to compute a gradient update during training. While DL applications typically use a default size of 32, with larger values often discouraged due to slower training without significant improvements, DRL can benefit from significantly larger batches. Some effective configurations use sizes up to 1024 (Raffin, 2020).

The DQN algorithm has been successfully applied to various problems, largely due to its relatively simple implementation compared to other RL techniques. However, it also presents several challenges, including the need to discretize the action space and its inherent instability. The instability in DQN arises from issues such as overestimation bias in the Q-function, sensitivity to hyperparameter selection, and difficulty in achieving stable convergence. These challenges can lead to learning oscillations and poor generalization, making it difficult to scale DQN to more complex environments. While DQN is well-suited for discrete action spaces, many real-world problems, such as microgrid energy management, involve continuous control. To address this limitation, algorithms designed for continuous action spaces have been developed, such as Twin-Delayed Deep Deterministic Policy Gradient (TD3), which is used in this thesis (Fujimoto et al., 2018). Despite this, other works demonstrated that a DRL family can use a discrete action space while maintaining a good performance over continuous action spaces (Tavakoli et al., 2018).

TD3 belongs to the actor–critic family (Konda & Tsitsiklis, 1999) and can be viewed as an extension of Deep Deterministic Policy Gradient (DDPG) (Lillicrap et al., 2015) incorporating several improvements. One of these is Clipped Double Q-learning, a variant of the double-network approach from the Double Deep Q-Network (DDQN) algorithm (van Hasselt et al., 2016), designed to reduce overestimation bias. To mitigate variance, TD3 also employs delayed policy updates, which limit the influence of inaccurate Q-value estimates on policy updates, and target policy smoothing, which adds clipped noise to target actions during critic updates to make value estimates less sensitive to small action perturbations.

While it may initially seem beneficial for the model to learn the environment thoroughly, in practice, direct access to the real environment is often unavailable, necessitating the use of a simulator. In the absence of access to a real microgrid, this thesis developed a dedicated simulation environment to enable the training and evaluation of DRL models, ensuring controlled experimentation and reproducible results.

1.3.5 Microgrid Energy Management System as an MDP

This section presents the modeling of a microgrid within the MDP framework, aligning it with the RL problem. The objective is to introduce the actual model used in this thesis while providing a broader overview of the microgrid modeling process as an MDP. To achieve this, each component of the model is first introduced from a general perspective, followed by an explanation of the specific simplifications applied. This approach allows the reader to understand the overall modeling process without overburdening the notation of the final model.

1.3.5.1 EMS model overview

To recall, a microgrid is a system that consists of a set of DER and loads. DERs can be renewable energy sources, such as PV and Wind Turbine (WT), ESS, such as Lithium-Ion (Li-ion) batteries, or fuel-based generators, such as a diesel-based generator, a Fuel Cell (FC) or a Microturbine (MT), among others. Loads can be from a single residential profile to large industrial or aggregated ones. The goal of the EMS is to minimize the cost of the energy consumption while maintaining the power balance in the microgrid. Power balance is maintained by controlling the power output of DERs. Additionally, the power consumption of the loads can also be controlled to maintain the power balance in the microgrid, called flexible demand, although this feature is not considered in this thesis.

In this context, the EMSs of a microgrid can be modeled as an RL problem by defining the state space, action space, and a reward function. In the practice for this particular problem, when a simulated environment is used, the transition function is also required.

1.3.5.2 State space

Following the MDP notation in section 1.3.2, the state space is defined by the state of the whole system, which, in this case, is composed of every state of every component and the state of the elements it interacts with. For instance, the state of a PV system can be represented by its output power. The average production over each time interval (e.g., 1 hour or 15 minutes) can be modeled as a time series. In this thesis, we assume that solar irradiance is the primary factor influencing PV power output and that it has a direct proportional relationship. Similarly, the state of a WT system is represented by its power output.

Unlike PV and WT systems, components such as DGs and ESSs can be represented with simpler state definitions. For ESSs, the state is commonly defined by the State of Charge (SoC), which provides sufficient information for decision-making without requiring a time series representation. In the case of DGs, no explicit state representation is necessary, as their operation can be directly controlled based on demand. Unlike batteries, which also respond to demand but are constrained by their internal charge levels, DGs can dispatch power independently of any internal energy storage limitations.

1.3.5.3 Action space

The action space is characterized by the power output of the DERs and the power flow through the ESS. Furthermore, the PV and WT can be curtailed to reduce their power output relative to the power available at the moment. The ESS can be charged or discharged, which also affects the power balance. The DG can be turned on or off, and, when active, it can dispatch a controllable amount of power. Depending on the specific objective, the action space may incorporate further technical controls, such as those related to the voltage regulation.

Additionally, the action space can be reduced using physical restrictions or using auxiliary systems, such as the balance equation and a control system. For example, in an isolated microgrid setup, considering that the objective function is to minimize energy costs incurred during the operation, the power output of the PV and WT can be straightforwardly ignored as a controllable action since its energy cost is zero, i.e., using less power than the available does not improve the cost in any possible future state of the system.

1.3.5.4 Reward

The objective function for optimal operation is defined as the minimization of costs over time. At each timestep, a cost is incurred based on the power produced by some specific components, such as the DG and FC. Additionally, the virtual cost of energy not supplied (ENS) needs to be considered as well in order to penalize the agent for not supplying the correct amount of power for a particular level of demand.

1.3.5.5 Transition function

In DRL algorithms, the transition function is not explicitly used, as learning is based on data gathered through interactions. However, access to a real microgrid is often unavailable, and a simulation system is required to generate synthetic data. Moreover, direct interaction with a real environment can be costly, slow, and potentially hazardous, making simulation a crucial step in pre-training the agent before deployment. In this context, defining the transition function is essential for the simulations used in this thesis.

To simulate a microgrid as an MDP, it is necessary to model both the interactions between components and the behavior of each component individually. The level of detail in this modeling can vary depending on the specific assumptions made. A microgrid's internal interactions primarily involve the exchange of electrical energy between components and the management of stored energy in the ESS. Additionally, external factors, such as weather conditions, energy prices, and non-flexible load demand, influence the system's dynamics. For simplicity, some aspects, such as the impact of temperature on component efficiency or mechanical constraints like generator start-up times, may be omitted in certain models. For instance, if the EMS decides to discharge the battery for a fixed duration, the energy depleted will depend on the power output and the available stored energy within the battery. By defining these interactions appropriately, the microgrid can be represented within an MDP framework, allowing RL techniques to optimize its operation.

1.3.5.6 Final states

The microgrid has no final states; therefore, the MDP is continuing. However, in the implementation for the training process, the MDP is truncated after a given number of timesteps, usually when the limited data available is exhausted, similar to an episodic MDP setup, but being a truncated MDP, not a terminated MDP, i.e., ending without reaching a final state.

1.3.5.7 Discount factor

The discount factor γ in the microgrid setting introduces a slight modification to the classical objective function. This parameter determines the balance between valuing future rewards and prioritizing immediate ones, mirroring the way humans naturally weigh present rewards against long-term rewards. A high γ encourages the agent to prioritize long-term objectives; however, it also presents certain disadvantages, such as slower convergence and increased sensitivity to approximation errors, which make its selection a nontrivial task. While this parameter has traditionally been set based on empirical observation, more recent approaches leverage automated methods to facilitate its tuning—such as the one presented in Chapter 4.

1.4 Challenges and Research Questions

The initial research question of this thesis emerges from the ongoing digital transformation of contemporary systems alongside a profound shift in power systems driven by the integration of renewable energy sources. These energy resources, inherently distributed, suggest a future where multiple clusters of energy resources might emerge. In this scenario, individual operation and collective cooperation could become the new standard for coordinating these systems, thereby radically altering the existing paradigm of power systems currently relying on a centralized approach. The question arises: To what extent is such a scenario feasible from a technological standpoint? Can we leverage recent technological advancements, such as the exponential growth in computational power and the burgeoning field of artificial intelligence, which are profoundly reshaping industry dynamics, automating complex processes, and creating new technological opportunities?

A preliminary analysis identifies two promising technologies that could effectively complement each other to achieve the described scenario:

1. Microgrids, defined as distributed electrical systems, can accelerate the adoption of renewable energy systems due to their inherent distributed nature.

2. RL techniques within the booming field of machine learning are ideally suited for modeling autonomous agents capable of making complex decisions effectively.

Building on this foundation, three fundamental research questions emerge, addressing key challenges in the application of RL for microgrid energy management. These questions delve deeper into the modeling, scalability, and exploitation of nonlinear system dynamics, shaping the core focus of this thesis.

1. How should a microgrid be modeled for RL applications? What are the most critical and relevant aspects to consider in the modeling process, and which elements have less impact on performance? Identifying the key trade-offs in microgrid modeling is crucial for ensuring that RL algorithms can learn effectively while maintaining computational feasibility and practical applicability.

2. Which other DRL algorithms can be effectively applied to microgrid energy management, and what challenges arise when using more advanced techniques or scaling up to larger microgrids? As more sophisticated DRL algorithms are explored, new challenges emerge, such as stability issues, training complexity, and computational demands. How should these

challenges be addressed when applying advanced DRL techniques or expanding the scope from small-scale microgrids to larger, more complex systems? Furthermore, is it still feasible to transition from theoretical and simulation-based research to real-world implementations at a practical scale?

3. To what extent can neural networks exploit nonlinear dynamics in DRL-based microgrid management? In theory, neural networks are capable of capturing complex nonlinear behaviors, which could be particularly useful in microgrid systems with intricate dynamics, such as nonlinear battery losses and fluctuating renewable generation. How much can this characteristic be leveraged to improve decision-making in DRL applications for microgrids? What are the benefits and limitations of using more detailed nonlinear modeling compared to conventional simplifications?

1.5 Objectives

The primary objective of this thesis is to explore the application of DRL in microgrid energy management, addressing key challenges related to modeling, scalability, and nonlinear system dynamics. To achieve this, the following specific objectives are defined:

1. Achieving an RL-based EMS application as a baseline, designing the experimental analysis:
   - To implement a software to simulate a microgrid operating in isolated mode.
   - To implement the interaction between DRL algorithms and the microgrid model, ensuring compliance with the general RL framework.
   - To incorporate third-party tools for implementing and evaluating DRL algorithms, or to develop custom tools when no suitable third-party options are available.
2. Enhancing the scalability of experimental components:
   - To analyze more advanced DRL algorithms that improve the efficiency and robustness of microgrid energy management strategies.
   - To analyze different microgrid sizes and DRL performance in each one.
3. To draw conclusions about the impact of incorporating nonlinear dynamics into the model when applying DRL techniques.
   - To investigate the nonlinear properties of microgrid components that can be integrated into the simulation framework, enhancing the fidelity of the models.
   - To implement selected nonlinearities and systematically analyze their impact on system performance and DRL-based decision-making.

To achieve the above objectives, an essential step—common to any research work—is the detailed analysis of the state of the art. This review serves to identify key gaps in current microgrid modeling approaches, to understand the fundamental principles and recent trends in RL and DRL, as its applications to energy systems, and to examine the main operational and optimization challenges of microgrid management, with a particular focus on RL and DRL-based methodologies.

1.6 Publications

This thesis has resulted in the publication of three journal articles, two of which are ranked in the first quartile (Q1).

1. "Optimising a Microgrid by Deep Reinforcement Learning Techniques", published in Energies, MDPI, Q3, (Domínguez-Barbero et al., 2020). Described in Chapter 3.
2. "Twin-Delayed Deep Deterministic Policy Gradient Algorithm for the Energy Management of Microgrids", published in Engineering Applications of Artificial Intelligence, Elsevier, Q1, (Domínguez-Barbero et al., 2022). Described in Chapter 4.
3. "Energy Management of a Microgrid Considering Nonlinear Losses in Batteries through Deep Reinforcement Learning", published in Applied Energy, Elsevier, Q1, (Domínguez-Barbero et al., 2024). Described in Chapter 5.

1.7 Dissertation Outline

This section indicates the outline of this thesis from this point in the text.

The second chapter presents an overview of the current literature on microgrid modeling, particularly emphasizing the optimization of the EMS responsible for microgrid operations. The reviewed techniques mainly focus on DRL, although classical optimization methods have also been examined due to their longstanding history of use. Each study has been carefully analyzed, with detailed discussions highlighting the most nuanced aspects of the research. A discussion regarding the identified gaps will follow.

The third chapter considers a basic microgrid to be solved by using the first popular algorithm in the DRL field, i.e., the DQN. From several François-Lavet works as a starting point, we delve into the bias-overfitting problem and perform an exhaustive search of the parameter that controls this trade-off. Experiments show counterintuitive outcomes that can give better insights into implementing DRL techniques to this problem.

The fourth chapter extends the example from the latter. In particular, it addresses the limitation of the DQN approach in selecting decisions on a continuous range. The actor-critic DDPG algorithm, together with the popular improvements in the state of the art, makes the chosen algorithm, the TD3, a better candidate for the EMS of our microgrid. This chapter discusses its correct implementation in this field, where a previous process of using a hyperparameter optimization tool is recommended.

The fifth chapter exploits one of the distinguished capabilities of DRL techniques that differentiates them from the classical optimization ones. In particular, the inclusion of nonlinear dynamics of the batteries in the simulated environment is studied from the DRL perspective. Altogether, the environment also includes a new control process model underneath that harmonizes with the DRL learning process and facilitates other techniques to be evaluated using the simulator.

The sixth chapter, which is the final chapter of the thesis, summarizes the work presented throughout the document. It begins by discussing the conclusions drawn from the research and concludes by highlighting potential future work involving DRL in microgrids and power systems in general.

An outline scheme about the contributions in this thesis can be found in Figure 1.2.

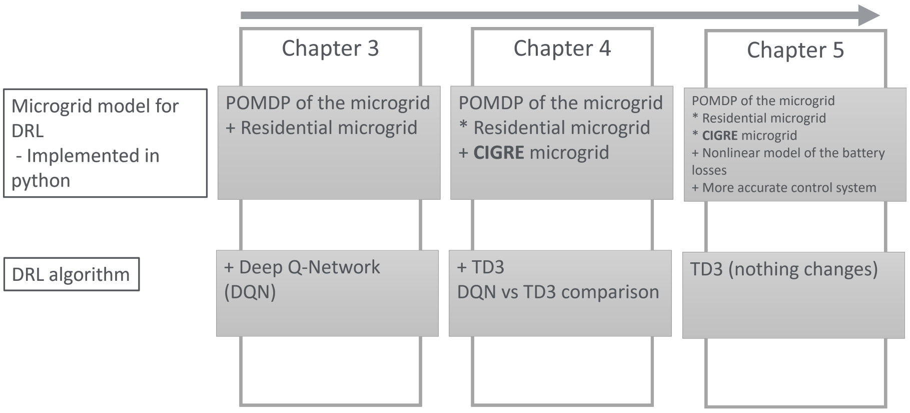


Figure 1.2: Contributions scheme.

Chapter 2

Background and Related Work

2.1 Introduction to Microgrid Modeling

Mathematical modeling in power systems is common in both academia and industry. Power systems contain many parameters, variables, and constraints, so computer models and algorithms are used on a daily basis in order to make optimal planning and operating decisions.

Microgrid modeling shares many characteristics with large electrical network modeling, such as the power balance equation. In addition, each component that makes up the microgrid has properties similar to those used in large electrical systems. For example, batteries have a charge and discharge efficiency, loads have a consumption profile, generators have an operating cost, and so on. In summary, microgrid modeling inherits some of the characteristics of modeling power systems in general.

However, the conventional modeling of power systems needs to improve its scalability and often relies on simplifications, whereas microgrid modeling can focus on the details of the components. Microgrids cannot leverage the benefits of aggregation properties, such as the law of large numbers. For instance, the typical consumption profile aligns with the aggregated values of observed demand in larger systems. These larger systems can be accurately modeled using this typical curve, leading to high precision in real-world applications. In contrast, the demand profile in microgrids can fluctuate significantly, and obtaining large datasets of fine-grained information can be expensive and often difficult to achieve.

To effectively model a microgrid, one must closely examine the variability of the elements that affect it, particularly regarding demand fluctuations and the intermittent nature of generation components like solar panels and wind turbines. Because of their distributed and renewable nature, these components are highly sought after and play a crucial role in the deployment of microgrids amid the current push for decarbonization.

This chapter summarizes the state of the art of microgrid modeling and the techniques applied to solve the optimization problems that arise from these models. In particular, we discuss the similarities and differences in modeling, advantages and disadvantages, and the work done in recent years.

2.2 Mathematical Modeling of Microgrids in the Literature

Microgrid modeling involves defining an objective function, or utility function, along with a set of constraints. The objective function varies across studies; while minimizing operating costs is the most common goal, other objectives include reducing emissions, enhancing resilience, maximizing profit, or minimizing investment costs.

In power system optimization, a key distinction exists between planning (investment) and operational problems. This analysis primarily focuses on modeling operational problems, as they are the focus of this thesis within a microgrid system. However, studies addressing planning problems also require an operational model, and in some cases, analyzing them can provide valuable insights.

The objective of the agent is to maximize its utility function $u(t)$ extended to the whole time horizon.

$$\max U = \left\{ \int_t^T u(t) \cdot dt \right\} \tag{2.1}$$

The instantaneous utility function $u(t)$ in (2.1) represents the negative operation costs and depends on time t. In the case of an isolated microgrid with only a fossil fuel generator, the cost function is expressed as in (2.2), where C^{g} is the cost of the power dispatched P^{g} by the generator $\mathrm{g} \in \mathrm{G}$, and c^{ens} is the cost of the ENS. Note that although the cost function is expressed in terms of continuous time, later in the rest of the thesis, it is discretized in time steps evenly separated, usually one hour, but sometimes less. Thus, the utility in time step t is denoted as u_t Eq. (2.2) can therefore be expressed as follows:

$$\max U \equiv \min -U = \sum_t^T -u_t = \left\{ \sum_t^T \left[\sum_{\mathrm{g}}^{\mathrm{G}} C^{\mathrm{g}}(P_t^{\mathrm{g}}) \Delta t \right] + c^{\mathrm{ens}} \cdot \mathrm{ens}_t \right\} \tag{2.2}$$

Eq. (2.2) represents the operating costs over time, with a penalty for the ENS, where c^{ens} is normally greater than $C^{\mathrm{g}}(P)$ for all $\mathrm{g} \in \mathrm{G}$ over the feasible domain of P, i.e., generating with any distributed generator is generally cheaper than not supplying the demand. Note that the optimal decision may be to incur the cost of not supplying energy while the distributed generator is shut down. For instance, this can happen due to component constraints like setting a minimum power rating. In general, such cases are expected to occur rarely.

For instance, the formulation used in (Gupta & Gupta, 2015) considers both an isolated and connected microgrid. The formulation for connected microgrid involves other dynamics related to the ENS and curtailment, and dealing with time-series for prices. In addition, Sections 2.1 and 2.2 include power reserves, which can be canceled using the equalities to fit the equations from above.

Another example can be seen in (François-Lavet, Gemine, et al., 2016), where authors use the Levelized Cost of Electricity (LCOE) and add an income for storing H_2. The LCOE in the objective function is an extension of Eq. (2.2), where it includes inversion costs and discount rates on income and energy production over the years.

The formulation used in (Alavi et al., 2015) uses a multi-objective cost function that includes emission, reliability, and operation costs. The model computes the emission cost of the DGs, in this case, an MT and a Diesel Generator (Di-Gen). Both DGs use a formula for each emission, i.e., CO2, SO2, and NOx. This formula includes the active power generated, the fuel/energy efficiency (lb/kWh), and the fuel price per weight unit ($/lb).

In (Sukumar et al., 2017), authors apply a strategy of solving three incremental optimization problems in order to make the best decision. The first objective function includes the cost of the FC and a marginal cost for the battery, and they do not prevent the battery from charging and discharging in the same period. The authors poorly justify the marginal battery cost in obtaining the battery schedule. The battery operation can have issues when charging and discharging are allowed at the same time. In order to manage the efficiency of the battery, it is necessary to split the battery operation to distinguish between charging and discharging. A typical example is to use two continuous non-negative variables ($P^{\mathrm{b}\leftarrow}$ and $P^{\mathrm{b}\rightarrow}$ for charging and discharging the battery b) and a binary variable (B^{b}) to force only one variable to be activated at a time.

$$\Delta S^{\mathrm{b}} = (B^{\mathrm{b}}) * P^{\mathrm{b}\leftarrow} - (1 - B^{\mathrm{b}}) * P^{\mathrm{b}\rightarrow} \tag{2.3}$$

Another option is to avoid using the binary variable and performing a post-processing of the battery operation, i.e., canceling the redundant charge/discharge flow in the final battery operation. This formulation adds some overhead to the solver due to the binary variables, but it avoids the extra development of the post-processing algorithm. A drawback in this work is that solving three separate optimization problems introduces a significant computing overhead.

In (Akinyele et al., 2018), the authors explore the microgrid from the point of view of STEEP factors, which are Social, Technological, Economic, Environmental, and Political factors. They focus on a use case in Nigeria and comprehensively analyze every detail. Although they do not focus on the modeling and optimizing part, they present several formulas to estimate the present value of the microgrid, the wind turbine, the biomass, and the hydro, which can help to simulate and optimize the microgrid operation using available data read from simple sensors. Furthermore, they discuss other humanitarian factors to include in microgrid models that can enable their successful deployment in developing countries like Nigeria.

In (R.-K. Kim et al., 2020), the authors propose a MILP model to optimize the operation of a microgrid, combined with a Particle Swarm Optimization (PSO) algorithm to deal with the non-linear behavior of the battery. They compare three approaches: the MILP, the PSO, and a combination of both of them. Moreover, they use two models, i.e., a linearized model suitable for the MILP and a non-linear model that is more precise and available when using the PSO. The MILP model needs to discretize the objective function that contains the quadratic cost of the diesel, the efficiency of the inverter, and the efficiency of the battery. The linearization of the quadratic cost of the diesel used includes binary variables to select the active linear segment. Furthermore, an equation is needed to force that, at most, one binary variable can be active simultaneously, i.e., only one slope can be used. The PSO model uses the MILP model because the authors argue that finding an initial feasible point for the PSO algorithm is hard. Unfortunately, the extra dimensions added to the particles after computing the 11 different solutions when solving the MILP model are poorly detailed. They should have discussed the computational efforts, which seem very complicated since, after each PSO iteration, they need to execute the MILP to make its solution feasible again. This infeasibility after the PSO algorithm occurs because the battery includes an intertemporal constraint that is not considered in the PSO model.

In (Garcia-Torres et al., 2021), the authors optimize the operation of a microgrid using a stochastic approach with two scenarios: an optimistic and a pessimistic one, which is adding or subtracting the mean deviation (check (1) in the same paper) to each value in the time series of the energy remainder, i.e., after subtracting the loads to the total renewable generation. The authors make some assumptions based on the Spanish market, limiting the scope of the model to a similar context. The model includes two markets, the day-ahead and the flexibility market, which are solved separately. The first one optimizes the power dispatch of the microgrid components, solving an economic dispatch with unit commitment, including start-up and shut-down constraints, as well as extra costs from turning on the generators. The second is not analyzed in detail as it is beyond the scope of this thesis. The microgrid includes a battery, a hydrogen tank, renewables such as PVs and WTs, and a demand. They also add a cost for the battery usage over time to model its degradation. To solve the model, the authors used Tomlab, a Matlab tool that includes several solvers.

In (Vilaisarn et al., 2022), the authors solve a multi-objective planning and operational problem in a microgrid taking into account the resilience of the system, and they use a NN and the Non-dominated Sorting Genetic Algorithm II (NSGA-II) algorithm. Their authors split the problem into several layers. First of all, they use a NN to speed up the two-stage optimization process, learning a surrogate NN model to retrieve the cost of the optimal operation, trained with data gathered from a previous stage where they used the MILP model instead. Afterward, they analyze a Pareto frontier of the multi-objective space of the planning problem, obtained using the NSGA-II algorithm. Moreover, they simulate High Impact Low Probability (HILP) events, using the methodology in (Amirioun et al., 2018; Panteli et al., 2017) in order to analyze the Capital Expenditure (CapEx) considering the resilience of the system. In addition to the resilience index and the operating costs, the environmental impact through the Carbon Dioxide (CO2) emissions produced by DGs and the electricity from the grid are also considered. The microgrid is the IEEE-33 bus system with PVs and WTs.

These models provide a foundation for understanding the complexity of microgrid modeling and how classical optimization algorithms can be applied to optimize microgrid operation. However, traditional optimization methods often require accurate system models, predefined constraints, and deterministic problem formulations, which may limit their applicability in dynamic and uncertain environments.

To leverage DRL techniques for microgrid operation, the problem must be formulated as an MDP and implemented within a simulation framework. This approach allows an agent to iteratively interact with the environment, learn optimal decision-making strategies through experience, and adapt to uncertainties such as fluctuating renewable generation, variable demand, and evolving grid conditions. By structuring the problem in this way, DRL enables data-driven optimization, eliminating the need for explicit system modeling and allowing for more flexible and scalable control strategies.

Some papers already model the microgrid as a MDP. For example, in (Jasmin et al., 2011), the authors model a MDP to solve the economic dispatch problem using RL for a single time period. Each generator represents a stage, and the agent must select the power that that generator will dispatch. After each stage, a different generator is selected as the next candidate, and the process repeats until all generators have a power setpoint assigned. The selection order of the generators must remain the same at all times. This kind of modeling differs from the one used in this thesis in the microgrid case. Nevertheless, this work can bring new ideas to address the problem when a large number of components have to be dispatched simultaneously, like in an Economic Dispatch

(ED) or Unit Commitment (UC), explained in the same article. The key challenge in this problem is that the cost function is often modeled in the academy as quadratic, making it unsuitable for classical optimization techniques such as the simplex method. Instead, the paper employs this cost function to model the fuel-based generators within the microgrid. Additionally, two RL algorithms are explored in that paper. In particular, two exploration methodologies of the Q-learning: the ε-greedy and the pursuit exploration.

In (Střelec & Berka, 2013), the authors model the microgrid as a MDP to solve the ED and UC problems. The model of the microgrid considers two energy vectors: heat and electricity. The chillers of the microgrid are powered by electricity to satisfy the cooling demand. The model considers a WT and the grid as external energy sources and a thermal ESS to store the thermal energy. To compute the optimal schedule, the authors use three Approximated Dynamic Programming (ADP) methods, which are compared with DP (the optimal) and a myopic approach (the reference).

In (François-Lavet, Taralla, et al., 2016), the authors use a DQN to solve the microgrid energy management, where they consider a PV, a hybrid ESS of a lithium battery and a combined hydrogen system with an electrolyzer and a hydrogen fuel cell. They consider an isolated setup, using data for solar and the residential load gathered in the Belgium location. They analyze six different PV sizes and three different state configurations: (1) no extra information beyond the readings from sensors, (2) extra information about the day of the year, (3) additionally including future predictions of solar production over 24h and 48h. The objective function utilizes the LCOE metric. The model considers linear efficiencies for the ESSs, i.e., the Li-ion battery and the hydrogen system. The operational costs include the ENS and the hydrogen energy charging and discharging at a constant cost per kWh. They solve the microgrid energy management (the EMS problem) with a NN trained using a DQN (Mnih et al., 2015). The NN architecture includes a convolutional layer set to process time-series data, specifically residential consumption and solar generation. This work is the most similar to the one in Chapter 3: the microgrid is quite similar, except for the hydrogen modeling. The study analyzed is also different, where Chapter 3 analyzes the number of samples considered in the state needed for the Partially-Observable Markov Decision Process (POMDP) (Kaelbling et al., 1996).

In (François-Lavet, Taralla, et al., 2016), the authors apply a DQN to address the microgrid EMS problem, considering a configuration that includes a PV system and a hybrid ESS composed of a lithium-ion battery and a combined hydrogen system with an electrolyzer and a hydrogen fuel cell. The study focuses on an isolated microgrid, using solar irradiation and residential load data collected in Belgium. Six different PV capacities are evaluated, along with three state representations: (1) sensor readings only; (2) sensor readings plus information about the day of the year; and (3) the previous configuration augmented with 24-hour and 48-hour forecasts of accumulated solar generation (the accumulated single value). The optimization objective is based on the LCOE metric. The model assumes linear efficiencies for both the Li-ion battery and the H_2 system. Operational costs include the ENS as well as hydrogen charging and discharging, each at a constant cost per kWh. The EMS problem is solved using a NN trained via a DQN (Mnih et al., 2015), where the NN architecture incorporates a convolutional layer designed to process time-series data of residential consumption and solar generation. This study is the most similar to the one presented in Chapter 3, as the microgrid configuration is comparable, with the exception of the hydrogen system modeling. The scope of analysis also differs: whereas (François-Lavet, Taralla, et al., 2016) focuses on varying PV sizes and state information, Chapter 3 investigates the number of samples included in the state representation required to address the POMDP (Kaelbling et al., 1996).

In (B. Kim et al., 2016), the authors model an electricity exchange between the service provider and the consumers as a MDP to use the Q-learning algorithm. They use bilevel optimization, where the outer problem is to maximize the service provider's profit, and the inner problem is to minimize the cost of the consumers. The service provider is modeled as an RL agent, whose decision in each timestep is to set the electricity price. On the other hand, consumers decide on the quantity of electricity they want to buy once the price is set. The service provider cannot know the customers' strategy in this setup. However, the authors design a strategy to estimate the aggregated demand of the consumers to improve the learning process. The authors assume the service provider knows some information about the transition probabilities. With this information, they simulate new experiences for the RL agent without explicitly interacting with the environment. However, some questions regarding the model are noted. For example, the Q-learning algorithm needs a discretized state space that must be defined. Besides, the state space uses a continuous variable, i.e., the demand of the customers, that is only discretized in the proposed algorithm and not in the Q-learning used for the comparison. Since the authors claim this discretization is a contribution, it cannot be applied to the compared algorithm. Furthermore, the authors include in the Q-learning state space the demand (d in that paper that differs from the energy consumed e) from each consumer, and they do not explain how the service provider can gather that information. One notable implementation decision in this work is the inclusion of the hour of the day in the state space, a choice also explored in several other studies. This approach provides a way to incorporate time-dependent patterns into the model, particularly for handling time series such as electricity demand. However, it also introduces a constraint on the richness of temporal information captured. Alternative implementations often use a fixed-size window of the most recent observations instead. Given the high autocorrelation in time series data, this method increases the likelihood that the observed state accurately represents the true state of the environment. In this work, as in many others, the optimization horizon does not exceed a single day, limiting the impact of this decision. However, this approach may not generalize well to unseen observations.

In (Nguyen & Crow, 2016), the authors use DP to optimize the energy management of a microgrid with non-linear efficiencies. The microgrid includes a PV array, two WT, an ESS that combines two technologies, and a fuel-based generator. Besides considering the energy not supplied costs and the costs of the fuel-based generator, the authors include the battery usage. They consider the energy stored as a fuel that can be bought and sold. The authors also include a constraint to disallow the battery from being charged using the fuel-based generator. This modeling strategy can sometimes give suboptimal solutions. For instance, in Chapters 3 and 4, it will be shown that the optimal operation charges the battery using the Di-Gen in order to avoid energy not being supplied later on when the Di-Gen cannot supply the demand alone due to its small size. The uncertainty of the RES and the load is modeled as a single realization with the addition of the errors as a random variable. The authors compute the final error directly and use a normal distribution to capture the dynamics of the load remainder, which the controllable components of the microgrid will later supply. Additionally, the authors force the use of the grid with a random probability, which is computed after setting the power of the controllable components, and the energy remainder is computed after the grid is used. A caveat of this work is that the authors do not include the details of the DP algorithm.

In (Qiu et al., 2016), the authors use an ESS with mixed technologies, which increases the efficiency of the whole ESS. The authors use the Q-learning to optimally operate the mixed ESS.

The microgrid includes a PV and two battery technologies: lead-acid and vanadium redox. The lead-acid outperforms the other when the power dispatched is low, and vice versa. The state space is modeled to include both batteries' SoC, discretized in three evenly distributed ranges. The model only allows one ESS technology to operate simultaneously. Therefore, the action space includes a binary decision for the ESS. The action involves selecting the battery with the highest priority in the operation. Later, the power needed to satisfy the balance between the PV generation and the demand is computed and relieved by the ESS, following the indicated priority. The reward function penalizes the losses of the batteries and the energy not supplied.

In (Mbuwir et al., 2017), the authors use a Fitted Q-Iteration (FQI) algorithm to optimize the operation of a residential microgrid. The microgrid includes a PV and a ESS, both through inverters, and also includes a connection to the grid. The authors model the inverter with a non-linear efficiency following the work in (Driesse et al., 2008), and the battery with a constant efficiency of 0.9. This work performs three experiments for each of the two scenarios. These two scenarios assume constant and sinusoidal prices, respectively. The authors design three experiments by varying the RL's agent observation. The first experiment does not consider the exogenous variables, the second considers all, and the third only considers the load plus Gaussian noise. Within the exogenous variables, the authors include the quarter-hour of the day and the day of the week, as in (B. Kim et al., 2016). Note that, in this thesis, and as in (François-Lavet, Taralla, et al., 2016), the history rather than the information of the period to operate is considered. The FQI algorithm is classified as a batch RL algorithm that, in this case, uses a Extremely Randomized Trees (ExtRa-Trees) to approximate the Q-function (Geurts et al., 2006). To simplify the model, the authors discretize the agent's state and action spaces. The authors gathered the data from the Local Intelligent Networks and Energy Active Regions (LINEAR) project.

In (Ji et al., 2019), the authors use a DQN algorithm to optimize the operation of a microgrid. This work uses the Low-Voltage Microgrid Benchmark (CIGRE) microgrid, with a few differences from the one used in this thesis. The model considers a quadratic cost in both the MT and FC, and the efficiency of the ESS is constant. The microgrid can buy and sell energy to the grid, but the price when selling is 10% lower than buying, which deters the possibility of using the ESS for arbitrage. The data used comes from California Independent System Operator (CAISO), where they use the time series of the year 2016 for PVs, WTs, the demand, and the local marginal prices. The dataset is split into the training set, containing the first 21 days of each month (252 days in total), and the test set, containing the rest of the days (114 days). The DQN used has 101 output neurons, one for each possible action of the ESS, as the authors claim. However, the MT and the FC decisions need to be more detailed.

In (Mbuwir et al., 2019), the authors use a FQI algorithm to optimize the operation of a microgrid with flexibility. The microgrid extends the one used in (Mbuwir et al., 2017) by adding a Heat Pump (HP), where its thermodynamics are based on a second-order thermal parameter model. The problem is modeled as a multi-agent system. Hence, the authors introduce two MDP that interact with each other, where both agents must compete for the same resource, i.e., the power needed to satisfy the demand. The agent used for controlling the HP should minimize the energy usage to keep the inner temperature of the building in a comfortable range. The agent is implicitly penalized when the temperature goes outside this range because a backup system is activated, forcing a suboptimal decision. On the other hand, the agent used to control the battery should minimize the energy cost of interacting with the utility grid to satisfy the local demand.

In (Shuai et al., 2019), the authors extend the work of (Nguyen & Crow, 2016) by using the same equations for the battery efficiency and using an ADP algorithm. In this work, the authors extend the objective function by adding a battery usage cost. This cost penalizes both charging and discharging, which differs from the modeling approach used in (Nguyen & Crow, 2016). In addition, the model includes capacity restrictions for the Alternating Current (AC) network. Unfortunately, the transition function of the MDP is hardly reproducible.

In (Bi et al., 2020), the authors use a DQN algorithm to optimize the operation of a microgrid. The microgrid includes a PV, a WT, an ESS, and a Point of Common Coupling (PCC) that only allows buying energy. The MDP includes the ESS linear efficiency and the balance equation. The flow between the utility network and the microgrid is computed after the balance equation when the ESS dispatch is already set. The objective function considers the cost of the energy bought from the utility network. The observation includes all the exogenous variables, i.e., the PV and WT generation, the demand, and the price of the utility network, as well as the hour of the day and the SoC of the ESS. The action space is discretized into three values, each one representing each battery behavior: idle, charging, or discharging the battery. Authors assume that the energy in the battery is always cheaper than the energy bought from the grid, forcing the model to recharge the ESS only when having RES surplus. The NN used has two hidden layers with 500 and 200 neurons, respectively, with a Rectified Linear Unit (ReLU) activation function.

In (Lei et al., 2021), the authors use a DDPG algorithm to optimize the operation of a microgrid. The microgrid includes a PV, an ESS, a Di-Gen, and a load bank. The latter serves to curtail the generation surplus. The considered microgrid runs isolated from the utility network. This work's novelty resides in using modified versions of both the DDPG and the Recurrent Deterministic Policy Gradient (RDPG) algorithms. A humble critique of this work is that this modification considers as many actor networks as hours in the day, overcomplicating the solution. They test the generalization of the method on the next day.

In (Nakabi & Toivanen, 2021), the authors propose two variations of the Asynchronous Advantage Actor-Critic (A3C) and Proximal Policy Optimization (PPO) algorithms to optimize the operation of a microgrid. The microgrid model includes a thermal controllable load and a price-responsive load. Additionally, the microgrid includes a ESS and RES generation. The optimization horizon is one day, but ten days are used for training, and they are selected randomly for each episode. The data used for the thermal dynamics are generated from a normal distribution. It is worth discussing the comparison between the methods used in this work. The proposed algorithm modifications consist of considering the ER, transforming the on-policy methods into off-policy methods. In general, on-policy methods are more sample efficient than off-policy because the experience used to update the gradients is extracted from the current policy rather than other similar policies, and therefore, the gradients computed are less biased with respect to the gradients of the current policy. One can refer to (R. S. Sutton et al., 2000) for more information. Therefore, the implementation modifications in this work may reduce the learning efficiency per gradient update. On the other hand, the on-policy methods are more budget-greedy, i.e., the former needs more time steps than the latter, but the former is also more stable. The on-policy methods are faster in computing time, given the best learning efficiency, when the time to gather new experience is not too high. The authors only compare the algorithms using the time steps metric. In summary, the best algorithm to apply strongly depends on the particular problem and the metric to optimize. As a side note, in the literature, there is already an off-policy version of the A3C algorithm, called Actor-Critic with Experience Replay (ACER) (Z. Wang et al., 2017), that indeed includes an ER. This paper describes the mechanism of the ACER algorithm to achieve sample efficiency and stability. ACER adopts the Retrace algorithm from (Munos et al., 2016) to estimate the Q-function, proposes a version of the truncated importance weight technique, and an efficient version of the Trust Region Policy Optimization (TRPO) for the policy updates. Additionally, ACER's authors propose a novel network architecture called Stochastic Dual Networks (SDNs), inspired by the Dueling network from (Z. Wang et al., 2016), in order to efficiently estimate the value function through the importance sampling technique (Levine & Koltun, 2013; Meuleau et al., 2000; J. Tang & Abbeel, 2010).

Regarding the DRL algorithm properties to exploit, in (Shuai & He, 2021), the authors propose to solve the EMS problem of a microgrid using the MuZero algorithm, which is an extension of the Monte Carlo Tree Search (MCTS) algorithm but using a learned model for the value estimation and another learned model to estimate the transition probabilities. This algorithm belongs to the model-based family in DRL. The authors use an RNN with LSTM layers. The studied microgrid considers a load, a DG with quadratic cost, a connection to the utility grid allowing both buying and selling energy, an ESS with linear efficiency, a WT, and a PV. The authors consider the reactive power in the model and line losses. The results include a Mixed Integer Second-Order Cone Programming (MISOCP) model that uses perfect information as an upper bound, and show that the MuZero algorithm outperforms the Mixed Integer Programming (MIP) model. The training data is from the year 2016, and the test data is from the years 2015 and 2016, where the authors use the data from 2015 to test the model trained with the data from 2016. While the results are promising, the reproducibility of the approach is constrained, as the paper does not report the specific hyperparameter values used during training. Similar to the previous work, in (Shuai et al., 2021), authors address the same problem but now use the Branching Dueling Q-Network (BDN) algorithm.

In (Phan et al., 2022), the authors use a DQN algorithm to optimize the operation of a microgrid. The microgrid includes RES (a PV and a WT), ESS, a Di-Gen, a hydrogen FC, an electrolyzer, and a hydrogen tank. The PV and a WT compute the production using the irradiation and the wind speed time series. The ESS is modeled as a lead-acid battery, with a constant efficiency of 0.9, and the FC together with the electrolyzer and hydrogen tank serves as another ESS, also with a constant efficiency that includes several parameters, such as the low heating value of the hydrogen, the thermodynamic efficiency, the fuel utilization coefficient, the mass of fuel reacting with the FC, the FC efficiency and the ratio between the nominal hydrogen mass flow and the actual generated hydrogen mass flow in the electrolyzer. The reward function is meticulously designed to penalize several energy usages. The diesel usage is linearly penalized, as seen in other works. The FC and the lead-acid battery are penalized when discharged in undesired situations, like discharging when there is a surplus in the energy balance, considering the load, RES, and the Di-Gen. Similar behavior in the reward function is applied when using the FC and the electrolyzer, i.e., when using the Di-Gen, the battery, or the FC, and rewarding the energy stored in the battery as well as in the hydrogen tank. The EMS agent is penalized when the chosen action is out of the technical bounds of the controllable components. This work claims to use the DQN algorithm, but the NN structure is unusual since the action is considered as an input of the neural network, which makes this algorithm impossible to implement following (Mnih et al., 2015). It seems that the authors of this paper wrongly depicted the network architecture, as no other special details are described. The comparison also includes a human-expert rule-based algorithm to evaluate the DQN. The authors use a Matlab tool called RL-Toolbox for the optimal operation. The optimal size is calculated using Hybrid Optimization of Multiple Energy Resources

(HOMER) (Lambert et al., 2006).

In (Panda et al., 2024), the authors compare several DRL algorithms on the operation of a microgrid. The microgrid uses an ESS to arbitrage using the utility network. The microgrid also includes a PV and a load to satisfy. The prices used the Time of Use (ToU) mechanism with three periods: peak, off-peak, and shoulder, where each period has a different fixed price. The algorithms compared are the DQN, the Categorical DQN (C51) (Bellemare et al., 2017), the Implicit Quantile Network (IQN) (Dabney, Ostrovski, et al., 2018), the Fitted Q-Factorization (FQF) (Yang et al., 2019), the Quantile Regression DQN (QR-DQN) (Dabney, Rowland, et al., 2018), and the Rainbow (Hessel et al., 2018). Additionally, they compare different strategies used in the chosen algorithms. The proposed strategy is to use a Prioritized Experience Replay (PER) (Schaul et al., 2015) instead of the common ER. The authors also analyze the consideration of using a noisy network (Fortunato et al., 2017; Hessel et al., 2018) and apply reward shaping to the reward function, i.e., to include custom signals in the reward function. In this case, the optimization horizon is one day long, and the training and evaluation datasets are 7 and 3 days, respectively. The evaluation set is also used to train the model after several training steps to perform better in the test set, but this methodology is uncommon. Usually, the evaluation set is exclusively used to choose the best-performing model to avoid biasing the model to the training set.

In conclusion, various DRL implementations are suitable for addressing the EMS problem. However, comparing these approaches requires analyzing multiple factors simultaneously, where even minor variations can lead to different conclusions. To facilitate a clearer comparison of the reviewed works, Table 2.1 provides an overview of their key characteristics. This table categorizes the studies based on essential aspects of microgrid modeling and control. For clarity, abbreviations used in the table are detailed below.

The table is divided into two subtables. The first subtable highlights the microgrid components and the specific problems each study addresses, revealing that most include at least one DER. The second subtable focuses on data sources and algorithmic choices, showcasing the diversity of approaches employed in the literature.

- Photovoltaic Panel (PV): Considered or not in the microgrid.
- Wind Turbine (WT): Considered or not in the microgrid.
- Energy Storage System (ESS): When considered, the type of ESS used: Generic if not specified, Lithium-Ion (Li-ion), hydrogen (H_2) when a FC and an electrolyzed are combined together with an H_2 storage, Lead-Acid Battery (LA), Vanadium-Redox Battery (VRB), Sodium-Sulfur (NaS), and thermal when thermal storage and a cooling and/or heating system is considered.
- ESS eff.: The efficiency dynamic model of the ESS when considered.
- Distributed Generator (DG): When considered, the type of DG used: Generic if not specified, Microturbine (MT), Diesel Generator (Di-Gen), Fuel Cell (FC) with H_2 or Natural Gas (NG) or a chiller/heater that consumes energy in the form of electrical power or fuel to control the temperature.
- DG eff.: The efficiency dynamic model of the particular fuel-based DG is used when considered.

- DG cost: If the cost function is linear, quadratic, or with another shape like a piecewise function of quadratic functions (PwQ)
- Load: The behavior of the loads to satisfy in the microgrid: static if the load only changes with respect to the time, dynamic if the load changes with respect to pricing signals given by another agent, Demand Response (DR) when the load is controllable, and flexible if there is a flexible demand.
- Utility Grid (UG): Considered or not in the microgrid, and if so, specify if it can be used to buy energy, sell, or both.
- Energy not supplied (ENS): Considered or not in the microgrid. Usually, if the utility grid is considered, the ENS is not. There are some special exceptions.
- Lines: If the model considers the network (for instance, power losses and line capacity).
- Problem: The optimization problem to solve. ED and UC are denoted explicitly in these works. EMS usually considers both, but the complexity of the problem mainly resides in managing the microgrid over time.

The first five columns of the second subtable denote how the dataset D is chosen. These datasets are time series, whether extracted from a specific location or synthetically generated using a statistical model. The columns of the second subtable are:

- D^{pv}: P^{pv} or irr means the dataset is formed using input data from the power generated by the PV array or from irradiation. If the inputs are irradiation, the power is calculated using a mathematical model of the PV, but it is independent of the mathematical model of the problem to solve. If the dataset is generated using a statistical model, the models are: Point Estimate Method (PEM), AutoRegressive Integrated Moving Average (ARIMA), or MC.
- D^{wt}: P^{wt} or V^{wt} means the dataset is formed using data from the power generated by the WT or from the velocity of the wind. If the data inputs are the velocity of the wind, the P^{wt} values are calculated using a mathematical model of the WT. This model is independent of the mathematical model of the problem to solve. If the dataset is generated using a statistical model, the models are: PEM, ARIMA, or MC.
- D^{load}: P^{load} means the dataset is formed using data from the load. If the dataset is generated using a statistical model, the models are: Ornstein-Uhlenbeck (OU) or Poisson.
- D^{prices}: C^{net} means the data are the costs of consuming from the utility network. ToU(·) means the data comprises a fixed number of values, and each value can span several timesteps. The number of different values used is denoted in parentheses. Unknown means the authors claim to use a utility network but do not specify where the data came from; fixed means the price stays constant all the time; sin means the price follows a sinusoidal pattern.
- D^{temp}: If the model used is OU or Second-Order Model (SOM), it substitutes the use of real data.
- Length: Length of the optimization horizon in time units (1 hour, 1 day, 1 year, etc).

- Δt: The time between each timestep (15 minutes, 1 hour, etc).
- Timesteps: The number of timesteps in the optimization horizon: $\frac{length}{\Delta t}$.
- Scenarios: The number of scenarios used in the experiments. By default, the number of scenarios is one. Several works use more scenarios to increase the generalization capacity of the model. Others use a single large scenario instead.
- Algorithm: The algorithm used to solve the problem. Sometimes, a combination of the two is used, for instance, Robust Optimization (RO) and Linear Programming (LP). In that case, their names are separated by a comma. In some cases, several algorithms are compared, and only the most relevant is mentioned. Nevertheless, the text mentions all the relevant algorithms. The different algorithms mentioned in the table are: RO, LP, PSO, MILP, Stochastic Model Predictive Control (SMPC), NSGA-II, Q-learning, ADP, DQN, FQI, DDPG, PPO, MuZero, BDN and C51.
- Objective: The objective function that the algorithm tries to minimize. The most common is the Operational Expenditure (OpEx), which refers to the operation costs of the microgrid. Other works extend this objective function with more terms, such as Operational Expenditure with Emissions (OpEx/E), Operational Expenditure with Battery utilization costs (OpEx/B), Operational Expenditure with Technical violation costs (OpEx/T), Operational Expenditure with Human thermal discomfort costs (OpEx/H), Operational Expenditure with Energy loss costs (OpEx/L), LCOE, Capital & Operational Expenditure (Cap&OpEx), and Capital & Operational Expenditure with Battery utilization costs (Cap&OpEx/B).

Paper	PV	WT	ESS	ESS eff.	DG	DG eff.	DG cost	Load	UG	ens	Lines	Problem
(Gupta & Gupta, 2015)	Yes	Yes	-	-	Generic	No	Linear	Static	Buy	No	No	EMS
(Alavi et al., 2015)	Yes	Yes	Generic	Linear	Di-Gen & MT	No	Quadratic	Static	Both	No	No	EMS
(François-Lavet, Gemine, et al., 2016)	Yes	No	Li-ion & H_2	Linear	H_2-FC	No	Linear	Static	No	Yes	No	EMS
(Sukumar et al., 2017)	Yes	No	Generic	Linear	NG-FC	Linear	Linear	Static	Both	No	No	EMS
(Amrollahi & Bathaee, 2017)	Yes	Yes	Generic	Linear	-	-	-	DR	No	No	No	EMS
(R.-K. Kim et al., 2020)	Yes	No	Generic	Nonlinear	Di-Gen	No	Quadratic	Static	No	No	No	EMS
(Garcia-Torres et al., 2021)	Yes	Yes	Generic & H_2	Linear	-	-	-	Static	Both	No	No	EMS
(Vilaisarn et al., 2022)	Yes	Yes	NaS	Linear	Generic	No	Linear	Static	Both	No	Yes	EMS
(Jasmin et al., 2011)	No	No	-	-	Generic	No	PwQ	Static	No	Yes	No	ED
(Střelec & Berka, 2013)	No	Yes	Thermal	Linear	Chiller	Nonlinear	Linear	Static	Both	No	No	EMS
(François-Lavet, Taralla, et al., 2016)	Yes	No	Li-ion & H_2	Linear	H_2-FC	No	Linear	Static	No	Yes	No	EMS
(B. Kim et al., 2016)	No	No	-	-	-	-	-	Dynamic	Both	Yes	No	DMP
(Nguyen & Crow, 2016)	Yes	Yes	VRB & LA	Nonlinear	Di-Gen	No	Quadratic	Static	Both	No	No	EMS
(Qiu et al., 2016)	Yes	No	VRB & LA	Nonlinear	-	-	-	Static	No	Yes	No	EMS
(Mbuwir et al., 2017)	Yes	No	Generic	Linear	-	-	-	Static	Both	No	No	EMS
(Ji et al., 2019)	Yes	Yes	Generic	Linear	MT & FC	No	Quadratic	Static	Both	No	No	EMS
(Mbuwir et al., 2019)	Yes	No	Generic	Nonlinear	-	-	-	DR	Both	Yes	No	EMS
(Shuai et al., 2019)	Yes	Yes	Generic	Nonlinear	Di-Gen & MT	No	Quadratic	Static	Both	No	Yes	EMS
(Bi et al., 2020)	Yes	Yes	Generic	Linear	-	-	-	Static	Buy	No	No	EMS
(Lei et al., 2021)	Yes	No	Generic	Linear	-	-	-	Static	No	Yes	No	EMS
(Nakabi & Toivanen, 2021)	No	Yes	Generic	Linear	-	-	-	Flexible	Both	Yes	No	EMS
(Shuai & He, 2021)	Yes	Yes	Generic	Linear	Di-Gen	No	Quadratic	Static	Both	No	Yes	EMS
(Shuai et al., 2021)	Yes	Yes	Generic	Linear	Di-Gen	No	Quadratic	Static	Both	No	Yes	EMS
(Phan et al., 2022)	Yes	Yes	Generic & H_2	Linear	Di-Gen	No	Linear	Static	No	No	No	EMS
(Panda et al., 2024)	Yes	No	Generic	Linear	-	-	-	Static	Both	No	No	EMS

Paper	D^{PV}	D^{WT}	D^{load}	D^{prices}	D^{temp}	Length [h]	Δt	Timesteps	Scenarios	Algorithm	Objective
(Gupta & Gupta, 2015)	P^{pv}	ARIMA	P^l	C^{net}	-	24	1h	24	-	RO, LP	OpEx
(Alavi et al., 2015)	PEM	PEM	P^l	ToU(3)	-	24	1h	24	-	RO, PSO	OpEx/E
(François-Lavet, Gemine, et al., 2016)	P^{pv}	-	P^l	-	-	26280	1h	26280	-	MILP	LCOE
(Sukumar et al., 2017)	P^{pv}	-	P^l	ToU(2)	-	24	1h	24	-	MILP	OpEx/B
(Amrollahi & Bathaee, 2017)	irr	V^{wt}	P^l	-	-	24	15m	96	-	MILP	Cap&OpEx
(R.-K. Kim et al., 2020)	P^{pv}	-	P^l	-	-	24	15m	96	-	PSO, MILP	OpEx/T
(Garcia-Torres et al., 2021)	P^{pv}	P^{wt}	P^l	C^{net}	-	24	1h	24	-	SMPC	OpEx/B
(Vilaisarn et al., 2022)	irr	V^{wt}	P^l	C^{net}	-	24	1h	24	4	NSGA-II, MILP	Cap&OpEx
(Jasmin et al., 2011)	-	-	-	-	-	-	-	-	-	Q-learning	OpEx
(Střelec & Berka, 2013)	-	MC	OU	Unknown	OU	24	1h	24	>1	ADP	OpEx
(François-Lavet, Taralla, et al., 2016)	P^{pv}	P^{wt}	P^l	-	-	26280	1h	26280	-	DQN	LCOE
(B. Kim et al., 2016)	-	-	Poisson	-	-	24	1h	24	-	Q-learning	OpEx
(Nguyen & Crow, 2016)	P^{pv}	P^{wt}	P^l	Fixed	-	24	1h	24	2	ADP	OpEx/B
(Qiu et al., 2016)	P^{pv}	-	P^l	-	-	24	10m	1440	-	Q-learning	OpEx/B
(Mbuwir et al., 2017)	P^{pv}	-	P^l	sin	-	24	15m	96	3	FQI	OpEx
(Ji et al., 2019)	P^{pv}	P^{wt}	P^l	C^{net}	-	24	1h	24	365	DQN	OpEx
(Mbuwir et al., 2019)	P^{pv}	-	P^l	sin	SOM	24	15m	24	1	FQI	OpEx/H
(Shuai et al., 2019)	P^{pv}	P^{wt}	P^l	C^{net}	-	24	1h	24	40	ADP	OpEx/L
(Bi et al., 2020)	P^{pv}	P^{wt}	P^l	C^{net}	-	24	30m	48	1	DQN	OpEx
(Lei et al., 2021)	P^{pv}	-	P^l	-	-	24	1h	24	4	DDPG	OpEx
(Nakabi & Toivanen, 2021)	-	P^{wt}	P^l	C^{net}	SOM	24	1h	24	10	PPO	OpEx/H
(Shuai & He, 2021)	P^{pv}	P^{wt}	P^l	ToU(4)	-	24	1h	24	372	MuZero	OpEx
(Shuai et al., 2021)	P^{pv}	P^{wt}	P^l	ToU(4)	-	24	1h	24	372	BDN	OpEx
(Phan et al., 2022)	irr	V^{wt}	P^l	-	-	48	1h	48	Undeter.	DQN	Cap&OpEx/B
(Panda et al., 2024)	P^{pv}	-	P^l	ToU(3)	-	24	1h	24	10	C51	OpEx/T

Table 2.1: Comparative table of works in the literature. Upper table shows the microgrid configuration and problem to solve. Lower table shows the data, algorithm, and objective function.

2.3 Gaps in the Literature

Despite significant advancements in applying DRL to microgrid EMS, several key gaps remain in the literature. These gaps arise from methodological, technical, component selection, and scalability perspectives, posing challenges that hinder the comparability, robustness, and real-world applicability of existing approaches. This thesis systematically addresses several of these gaps. These gaps are divided by topics and outlined below. Additionally, Table 2.2 compares these gaps with other works.

1. Methodological Gaps:

   (a) A first limitation is the *short decision horizons* commonly considered in the literature. Many studies restrict the optimization horizon to one day, limiting the ability of the trained models to generalize long-term operational strategies. In reality, microgrid operation is an infinite-horizon problem, requiring policies that remain effective over extended time spans. This thesis addresses this gap by formulating the EMS problem as a continuing task and by training models on datasets spanning at least one year of operation, detailed in Chapter 3.

   (b) A second methodological gap is the *lack of systematic analysis of observation representations.* While some studies enrich the state space with contextual or forecast information, few investigate how the choice of observation stack size affects performance. This thesis explicitly studies the impact of the observation history length on DRL training under a POMDP formulation, identifying configurations that improve learning stability and performance, as described in Chapter 3.

2. Technical Gaps:

   (a) Many existing studies rely on *discrete action spaces* for microgrid decision-making, which can simplify training but restrict control precision. Continuous action spaces, although more challenging to learn, are closer to real-world control signals and provide finer decision granularity. In Chapter 4, this thesis addresses this gap by employing a continuous action space formulation for the EMS problem.

   (b) Another technical gap—addressed in Chapter 4 as well—lies in the *lack of attention to stability and reproducibility.* Reported DRL results often exhibit sensitivity to hyperparameters and random initialization, making them difficult to replicate or deploy in practice. This thesis incorporates stability analysis into the evaluation of the proposed methods, ensuring that the learning process and outcomes can be reproduced reliably.

3. Component-Related Gaps:

   (a) Microgrid studies often model system components with simplified assumptions, neglecting nonlinear behaviors such as battery degradation, temperature effects, or efficiency variations. This simplification can underestimate the challenges of real-world deployment. This thesis addresses this gap, in Chapter 5, by modeling nonlinear losses in the battery, thereby providing a more realistic setting for evaluating the effectiveness of DRL-based control.

Table 2.2: Gaps in the literature.

Paper	Ji et al. (2019)	Shuai et al. (2019)	Nakabi and Toivanen (2021)	Panda et al. (2024)	François-Lavet, Gemine, et al. (2016)	Chapter 3 (2020)	Chapter 4 (2023)	Chapter 5 (2024)
Algorithm	DQN	ADP	PPO	C51	DQN	DQN	TD3	TD3
(1a) Length $\geq$ 1Y	✗	✗	✗	✗	✓	✓	✓	✓
(1b) Obs stack size	✓	✓	✗	✗	✓	✓	✓	✓
(2a) Continuous action space	✗	✗	✗	✗	✗	✗	✓	✓
(3b) Training stability analysis	✗	-	✗	✓	✗	✗	✓	✓
(3a) Nonlinear losses	✗	✓	✗	✗	✗	✗	✗	✓
(3b) Adv. control system	✗	✗	✗	✗	✗	✗	✗	✓
(4a) CIGRE (scalability)	✓	✗	✗	✗	✗	✗	✓	✓

(b) In addition, existing approaches often lack control flexibility in their simulated environment. This thesis considers a flexible microgrid control system using more components to participate actively in the energy balance.

4. Scalability Gaps:

(a) Most existing applications of DRL to microgrids focus on small-scale residential systems, with limited discussion of scalability to larger and more complex networks. This thesis evaluates the proposed methodology on two scales: a small residential microgrid and the CIGRE benchmark microgrid, thus explicitly assessing scalability. Chapters 4 and 5 consider both.

By addressing these gaps, this thesis contributes to making DRL-based microgrid optimization more robust, scalable, and applicable to real-world deployments.

Chapter 3

Optimizing a Microgrid System using Deep Reinforcement Learning Techniques

3.1 Introduction

The microgrid, conceptualized as an MDP, requires a robust framework for defining the state space in RL applications, particularly when utilizing DRL. This foundational step ensures that the agent identifies and executes optimal actions depending on the current system's state. However, the determination of what constitutes the state is intricate and non-trivial. Unlike a game of chess, where the state is unambiguously represented by the positions of pieces on a board, the state definition in a microgrid involves selecting from potentially numerous variables influenced by uncertainty over time.

Expanding on the work by François-Lavet et al. (2019), which explores the theoretical implications of state configuration on DRL's bias-overfitting trade-offs, this chapter further investigates how varying definitions of the state impact the efficacy of microgrid management using a CNN-based DQN. In particular, this chapter mainly focuses on the span of the temporal window defining the state observation, a critical factor in the performance of the DRL model.

Our study develops a microgrid system similar to the one examined by François-Lavet, Taralla, et al. (2016), incorporating a Di-Gen as a backup power source in addition to the hydrogen FC, thereby increasing the complexity of the model.

The experiments carried out in our study, discussed in this chapter, elucidate the effects when configuring the observed information on model performance, providing a nuanced understanding of the energy management problem in microgrids under varying observation window sizes.

The contributions of the work described in this chapter are the following:

- Firstly, different window-size configurations for the state model are explored. Performance differences with the CNN-based DQN are discussed, highlighting conclusions about the optimal window size.

- Secondly, the microgrid system considered is more complex than the previously solved in the state-of-the-art using DRL techniques, particularly a DQN with CNNs. Two controllable devices are considered, but only one is directly decided; the other is derived from the power balance constraint.

- Thirdly, based on the opportunity cost concept, the implicit modeling approach effectively assumes the batteries in the optimal solution. Other solutions in the literature explicitly include battery usage as a synthetic cost that is unnecessary.
- Fourthly, the implicit modeling of the batteries does not require the usage of synthetic costs as done in other works within the literature.

The chapter is organized as follows: Section 3.2 introduces the microgrid elements and their modeling fundamentals. Section 3.4 presents its application to a case study where the microgrid structure is described in detail. Section 3.5 analyzes the DRL EMS behavior, comparing the results across different modeling configurations for the observation window size. Finally, Section 3.6 concludes with a discussion of the findings from this work.

3.2 The Microgrid Framework

3.2.1 General overview of a microgrid

As previously discussed in Section 1.2, a microgrid is an energy generation and storage system capable of operating either independently or connected with the main electrical grid. The general concept of a microgrid encompasses the interaction among various components, which can be categorized into three major groups: loads, generation (renewable and non-renewable), and ESSs.

Within these categories exist a wide array of technologies, each offering distinct properties. Certain technologies may be more desirable than others, depending on the specific context and requirements.

Furthermore, combining different technologies can provide more flexible and efficient solutions. For instance, integrating Li-ion battery storage with hydrogen storage provides a combination of short-term and long-term energy storage capabilities, respectively. Li-ion batteries are highly efficient for energy storage, but scaling up their capacity is expensive. In contrast, hydrogen storage systems have lower efficiency but offer a lower cost per unit of storage capacity. Figure 3.1 illustrates several components that can be found in a microgrid. The PCC illustrates the switch at the point of connection between the microgrid and the main grid.

This section introduces the proposed model, which investigates an isolated microgrid with solar arrays, employing this specific combination of short-term and long-term ESS solutions. The primary objective is to leverage the efficiency of Li-ion batteries for daily operations while preventing the wastage of surplus energy by utilizing the cost-effective hydrogen storage system. This configuration enables the hydrogen storage to supplement the lithium batteries during periods of reduced solar irradiance and to support the Di-Gen during the winter season.

3.2.2 Description of the studied microgrid

The studied microgrid in this work is depicted in Figure 3.1, where the included components are squared using a continuous line, while the dashed line indicates the component is not considered

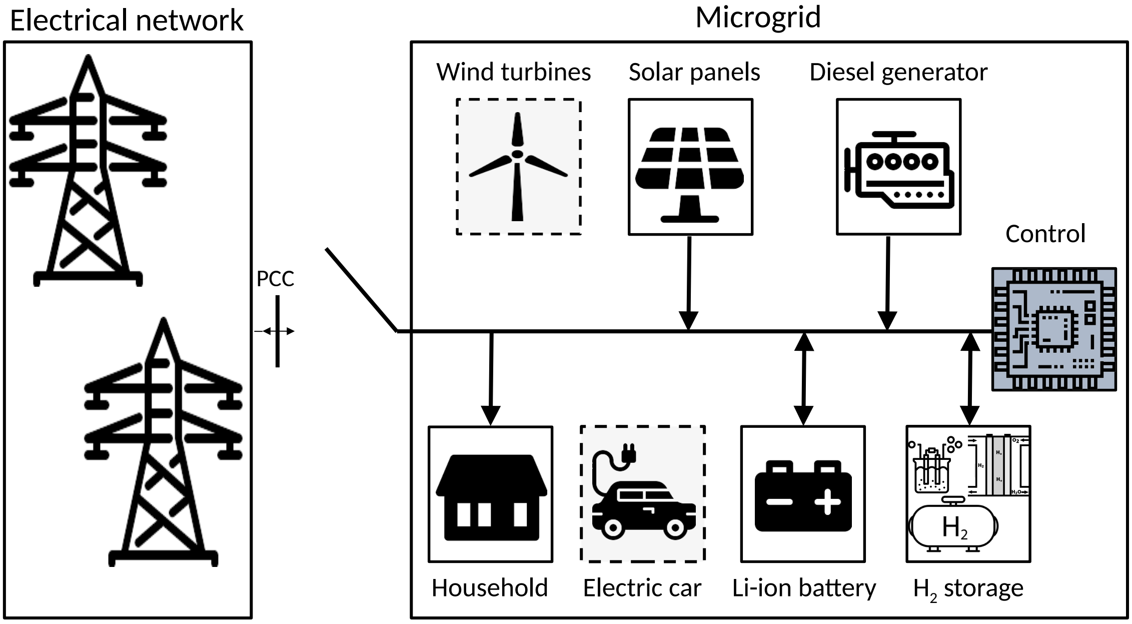


Figure 3.1: Residential microgrid scheme that includes hydrogen storage. Elements in a dashed box are elements that are not considered in the microgrid setup in this chapter but that are valid for a different setup.

in this study. The microgrid in question is isolated as it can be seen by the depicted open switch in the PCC point. The arrows indicate the power flow directions between the microgrid bus and the components.

The main elements to be defined are the residential energy demand, the solar panel, a DiGen, and the ESS consisting of a Li-ion battery and a hydrogen-based system where the latter is composed of an an hydrogen tank that stores hydrogen, a FC to generate electricity from the hydrogen stored, and an electrolyzer, able to generate hydrogen using electricity. Both the physical characteristics of these elements in the studied microgrid, as well as the data used for fitting and testing the models, are discussed later in Section 3.4.1. Finally, the control system is in charge of determining the operation of all the controllable elements. This system is referred to *the agent* in the RL context.

3.2.2.1 Demand

The load is modeled as a time series with hourly intervals that represent the average consumption of each hour, and it is assumed that historical data is available. Furthermore, this demand is not dependent on the behavior of all the other elements of the microgrid. In case there is not enough generation to supply the microgrid demand, the ENS will take the required positive value to ensure the demand balance equation is satisfied. This ENS can be understood as a virtual generator able to supply any demand, making the mathematical solution always feasible, but severely penalized in order to discourage the optimization algorithm from using it in place of other kinds of energy sources. On the other hand, the excess of RES generation is assumed to be curtailed without any cost (RES spillages).

3.2.2.2 Generation

The microgrid generation system consists of a PV and a Di-Gen. The PV generation profile can be represented by an hourly time series, where each hourly value corresponds to the accumulated generation within that period. On the other hand, the Di-Gen can provide backup energy in case of a lack of RES production. The Di-Gen power dispatched follows the controller (the EMS) set-point, producing power in a deterministic manner, i.e., possible failures of the equipment are out of the scope of this study. Moreover, the Di-Gen can be turned on and off. The controller will send the corresponding signal for this purpose. For simplicity, the generation cost curve is modeled as a quadratic function where the independent term, i.e., the no-load cost, is only incurred when the generator is committed. Additionally, startup and shutdown costs are neglected. Further details can be seen in section 3.4.

3.2.2.3 Storage

The microgrid in this study combines two energy storage technologies, as mentioned before. A Li-ion battery is used as a short-term storage supporting the microgrid in periods when RES cannot supply the demand. It is characterized by the maximum capacity $S^{\mathrm{b}}_{\mathrm{max}}$, the maximum power $P^{\mathrm{b}}_{\mathrm{max}}$ for both charge and discharge, and its efficiencies, respectively, η^{b} for charge and ζ^{b} for discharge. A representation of the battery and the energy flux inside, with their losses during charge and discharge processes, is depicted in 3.2. It is important to highlight that, under the proposed RL approach, it would be possible to take into account a more detailed model of the battery to capture the non-linear relationship among the energy stored, the maximum power, the efficiencies, and other characteristics. This more detailed model is studied later in Chapter 5. However, this chapter relies on a simplified model, since the full model falls outside its intended scope. The hydrogen storage device is used as a long-term storage, supporting large periods of high demand and low RES. The characterization of the hydrogen storage is analogous to the Li-ion battery: the maximum capacity $S^{\mathrm{h_2}}_{\mathrm{max}}$, maximum power $P^{\mathrm{h_2}}_{\mathrm{max}}$ for charge and discharge, and the efficiencies $\eta^{\mathrm{h_2}}$ for charge and $\zeta^{\mathrm{h_2}}$ for discharge.

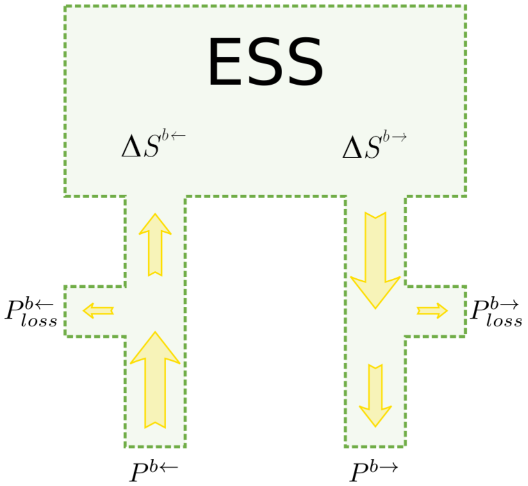


Figure 3.2: Energy flow in a battery using the nomenclature defined in this thesis.

3.3 Application of the DQN in an isolated microgrid as a POMDP

3.3.1 State definition

To effectively design the EMS using DRL techniques, it is essential to model the demand of the microgrid with a high degree of specificity. Since demand behaves as a time-dependent variable, fully characterizing its underlying process is impractical. Achieving a perfect representation would require accounting for every conceivable influence on demand, a goal that remains unattainable in real-world scenarios. As a practical alternative, previous demand measurements are often used to approximate the inherent temporal structure. However, this approach prompts an essential question: How much historical data is sufficient to accurately capture the system's time-series characteristics? The resolution of this question lies at the core of the present research. In DRL literature, the volume and structure of the information retained at any decision point is referred to as the "belief" (Kaelbling et al., 1998). This belief consolidates a collection of past observations, enabling it to function as a Markovian state within the broader framework of a MDP. Defining and refining this belief representation is a central focus of this work. From this point onward, the terms "belief" and "state" may be used interchangeably. This choice does not compromise the conceptual distinction previously established. Instead, it serves to clarify the structure of the microgrid information required for the DRL method, while leaving the broader modeling framework unaffected by this terminological decision.

Following the adequate notation for these kind of models, denoted as a POMDP, the belief representation is defined in place of the state typically defined when applying the more simplified approach (the MDP). Therefore, a window-based belief h_t is modeled for each timestep t as in (3.1) (François-Lavet et al., 2019).

Let h_t be the belief of the state s_t, which is a k-tuple of consecutive observations $o \in \Omega$ of past information (see (3.5)), and o_t is the directly observed information in a particular timestep t.

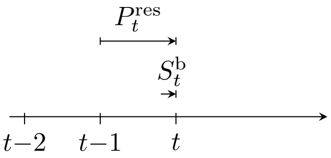


Figure 3.3: Representation of the observed elements over time.

$$h_t = (o_{t-k+1}, ..., o_{t-1}, o_t) \tag{3.1}$$

The element o_t comprises the power output of each RES, the total microgrid demand, and the energy stored in each ESS; these components are presented collectively in (3.2).

$$o_t = (P_{t-1}^{\mathrm{res}_1}, ..., P_{t-1}^{\mathrm{res}_{|\mathrm{Res}|}}, D_{t-1}, S^{\mathrm{b}_1}, ..., S^{\mathrm{b}_{|\mathrm{B}|}}) \tag{3.2}$$

Formally, the observation o_t at time period t is defined as an n-tuple, where $n = |\mathrm{Res}| + 1 + |\mathrm{B}|$. In this representation, P_{t-1}^{res} denotes the average power generation of each res $\in$ Res during period $t-1$, D_{t-1} represents the average demand in the same period $t-1$, and S_t^{b} corresponds to the stored energy of each b $\in$ B at the beginning of period t.

- P_{t-1}^{res}: The average power output over period $t-1$ for a particular RES res $\in$ Res.
- D_{t-1}: The total microgrid demand over period $t-1$.
- S_t^{b}: The energy stored in ESS b $\in$ B at the beginning of period t.

The domain of each variable is:

$$\begin{aligned} P_t^{\mathrm{res}} &\in \mathbb{R}^+ && \forall \mathrm{res} \in \mathrm{Res}, \forall t \in T \\ D_t &\in \mathbb{R}^+ && \forall t \in T \\ S_t^{\mathrm{b}} &\in [S_{\mathrm{min}}^{\mathrm{b}}, S_{\mathrm{max}}^{\mathrm{b}}] && \forall \mathrm{b} \in \mathrm{B}, \forall t \in T \end{aligned} \tag{3.3}$$

Figure 3.3 illustrates the distinction between the average power at $t-1$ as defined in (3.4), being $P(t)$ the continuous value, and the instantaneous stored energy, i.e., the energy available at the beginning of period t.

$$\frac{\int_{t-\Delta t}^{t} P(t) \cdot dt}{\Delta t} \tag{3.4}$$

These discrete-time equations are more convenient than the continuous-time ones in the context of this thesis while being accurate enough. Therefore, from now on, magnitudes that express power as P will represent the average power in the discrete-time interval between t and $t + \Delta t$.

In this chapter, the considered residential microgrid configuration matches with the observation o_t detailed in (3.5).

$$o_t = (P_{t-1}^{\mathrm{pv}}, D_{t-1}, S_t^{\mathrm{b}}, S_t^{\mathrm{h}_2}) \tag{3.5}$$

3.3.2 Action definition

At each time period t, the DRL-based EMS agent must determine the control actions for every controllable component of the microgrid. In this chapter, these components are the Di-Gen and two ESSs: the Li-ion battery and the hydrogen system. Accordingly, the agent's decisions consist of specifying the amount of power to be dispatched from the Di-Gen and the amount of power to be charged or discharged from each ESS.

Additionally, specific considerations in the EMS problem must be taken into account. First, one controllable component can be omitted from the action space, since the control stability system may later use that component, modifying its operating set-point (further details are provided in Section 3.3.4). In this case, where the microgrid has three controllable components, the DRL-based EMS agent directly selects the set-points for only two of them.

Second, the actions selected by the EMS agent represent reference set-points for power output, guiding long-term operational optimization. However, the secondary control system—responsible for maintaining real-time power balance—may adjust these references as needed to ensure stability and reliability. This dynamic interaction highlights the division of responsibilities: while the EMS provides strategic decision-making, lower-level controls perform immediate adjustments to preserve operational balance.

In summary, the action space of the DRL-based agent at each time t is defined as in (3.6), where j denotes the controllable component omitted, belonging to the set $\mathrm{G} \cup \mathrm{B}$.

$$a_t = \{P^i \mid i \in \mathrm{G} \cup \mathrm{B}, i \neq j, j \in \mathrm{G} \cup \mathrm{B}\} \tag{3.6}$$

The action domain, defined by its parts, is as in (3.7), where $P^{\mathrm{g}}_{\mathrm{min}}$ and $P^{\mathrm{g}}_{\mathrm{max}}$ are the minimum and maximum power values that the power generator can dispatch when it is on. Additionally, the generators can be turned off; thereby, the 0 value is considered in the action domain. $P^{\mathrm{b}\leftarrow}_{\mathrm{min}}$ and $P^{\mathrm{b}\rightarrow}_{\mathrm{min}}$, and $P^{\mathrm{b}\leftarrow}_{\mathrm{max}}$ and $P^{\mathrm{b}\rightarrow}_{\mathrm{max}}$ are the minimum and maximum power values during the charge and discharge processes of the battery, respectively.

$$\begin{aligned} P^{\mathrm{g}}_t &\in \{0\} \cup [P^{\mathrm{g}}_{\mathrm{min}}, P^{\mathrm{g}}_{\mathrm{max}}] && \forall \mathrm{g} \in \mathrm{G}, \forall t \in T \\ P^{\mathrm{b}}_t &\in [-P^{\mathrm{b}\leftarrow}_{\mathrm{max}}, -P^{\mathrm{b}\leftarrow}_{\mathrm{min}}] \cup [P^{\mathrm{b}\rightarrow}_{\mathrm{min}}, P^{\mathrm{b}\rightarrow}_{\mathrm{max}}] && \forall \mathrm{b} \in \mathrm{B}, \forall t \in T \end{aligned} \tag{3.7}$$

In the considered microgrid, the action space is defined by (3.8).

$$a_t = (P^{\mathrm{d}}, P^{\mathrm{h_2}}) \tag{3.8}$$

Since the DQN operates only in discrete action spaces, the continuous set-points must be discretized. For the diesel generator, the output power is restricted to (3.9), corresponding to the states *off*, half load, and full load, respectively.

$$P^{\mathrm{d}}_t \in \{0, \tfrac{1}{2}P^{\mathrm{d}}_{\mathrm{max}}, P^{\mathrm{d}}_{\mathrm{max}}\} \tag{3.9}$$

Similarly, the hydrogen system is discretized as in (3.10), representing maximum charging (electrolyzer), idle, and maximum discharging (fuel cell).

$$P^{\mathrm{h_2}}_t \in \{-P^{\mathrm{h_2}}_{\mathrm{max}},\, 0,\, P^{\mathrm{h_2}}_{\mathrm{max}}\} \tag{3.10}$$

Thus, the joint action space contains 3 × 3 = 9 possible configurations at each time step.

3.3.3 Reward definition

The reward function (3.11) is directly related to the generation costs of the fuel-based DG, plus the penalty of the ENS in each timestep t. In order to minimze costs, due the agent maximizes rewards, the reward function needs to represent the negative cost of the microgrid.

$$r_t = r(s_t, a_t, s_{t+1}) = -\sum_{\mathrm{g}}^{\mathrm{G}} C^{\mathrm{g}}(P_t^{\mathrm{g}}) \cdot \Delta t - c^{\mathrm{ens}} \cdot \mathrm{ens}_t \tag{3.11}$$

In (3.11), $C^{\mathrm{g}}(\cdot)$ is the quadratic cost function defined by coefficients δ_2, δ_1 and δ_0 in (3.12), and Δt is the duration of the time period (1 hour in the cases presented in this study). To use a linear cost, δ_2 is set to 0.

$$C^{\mathrm{g}}(P) = \begin{cases} \delta_2^{\mathrm{g}}(P)^2 + \delta_1^{\mathrm{g}} P + \delta_0^{\mathrm{g}} & \text{if } P > 0 \\ 0 & \text{if } P = 0 \end{cases} \tag{3.12}$$

3.3.4 Transition definition

The transition function determines the next state of the microgrid from a given state–action pair. In the microgrid context, however, accurately modeling the uncertainty inherent in RESs and demand patterns remains challenging. To address this, the present work relies on historical data (François-Lavet, Taralla, et al., 2016).

When historical data of RESs and loads are used, the transition function must also compute the next storage levels. This requires, first, determining the power flows between each component and the microgrid—based on the RES generation, load demand, and the set-points provided by the EMS. Then, the system power balance equation (3.13) is solved for the remaining variables: the component without a set-point specified by the DRL-based EMS agent, the ENS, and the curtailed RES.

For notation, D_t, P_t^{G}, P_t^{Res}, and P_t^{B} denote the total energy contributions of demand, generators, renewable sources, and storage devices, respectively, during period t, as defined in (3.16).

$$D_t \cdot \Delta t + \mathrm{curt}_t = (P_t^{\mathrm{Res}} + P_t^{\mathrm{G}} + P_t^{\mathrm{B}}) \cdot \Delta t + \mathrm{ens}_t \quad \forall t \in T \tag{3.13}$$

$$\text{(total fuel-based gen.)} \qquad P_t^{\mathrm{G}} = \sum_{\mathrm{g} \in \mathrm{G}} P_t^{\mathrm{g}} \; \forall t \in T \tag{3.14}$$

$$\text{(total RES gen.)} \qquad P_t^{\mathrm{Res}} = \sum_{\mathrm{res} \in \mathrm{Res}} P_t^{\mathrm{res}} \; \forall t \in T \tag{3.15}$$

$$\text{(total ESS gen./load)} \qquad P_t^{\mathrm{B}} = \sum_{\mathrm{b} \in \mathrm{B}} P_t^{\mathrm{b}} \; \forall t \in T \tag{3.16}$$

The set-points assigned to the controllable components are enforced by the lower control loops while maintaining system balance. To achieve this, the control loops rely on the component without an assigned set-point. Specifically, its actual power flow is determined from the balance equation, minimizing ens_t and curt_t subject to the component's technical constraints (e.g., (A.10)–(A.14) in Appendix A). In this case, the LI-ION battery is computed using (3.17). Finally, ens_t and curt_t are obtained as the residual terms required to satisfy the balance equality.

$$D_t \cdot \Delta t + \text{curt}_t = (P_t^{\text{pv}} + P_t^{\text{b}} + P_t^{\text{h}_2} + P_t^{\text{d}}) \cdot \Delta t + \text{ens}_t \quad \forall t \in T \tag{3.17}$$

The detailed steps are as follows: First, compute the remainder power P_t^{R} in the microgrid using the known variables at one side of the balance equation as in (3.18). Additionally, compute the maximum available power that the battery can allow given the current energy inside it. Let $\overline{P_t^{\text{b}\leftarrow}}$ and $\overline{P_t^{\text{b}\rightarrow}}$ be the maximum charge and discharge power values between the battery b and the microgrid, defined in (3.20) at time t.

$$P_t^{\text{R}} = D_t - P_t^{\text{pv}} - P_t^{\text{d}} - P_t^{\text{h}_2} \qquad \forall t \in T \tag{3.18}$$

$$\overline{P_t^{\text{b}\leftarrow}} = \min \begin{cases} (\eta^{\text{b}})^{-1}(S_{\max}^{\text{b}} - S_t^{\text{b}}) & \forall t \in T \\ P_{\max}^{\text{b}\leftarrow} \end{cases} \tag{3.19}$$

$$\overline{P_t^{\text{b}\rightarrow}} = \min \begin{cases} \zeta^{\text{b}}(S_t^{\text{b}} - S_{\min}^{\text{b}}) & \forall t \in T \\ P_{\max}^{\text{b}\rightarrow} \end{cases} \tag{3.20}$$

The battery charge and discharge powers, $P_t^{\text{b}\leftarrow}$ and $P_t^{\text{b}\rightarrow}$, are defined in (3.21) and (3.22), respectively. The net battery power is then obtained from (3.23), which already incorporates the technical limits of the battery. These equations ensure that any violation of the limits results in the power being projected onto the closest feasible value. Moreover, both $P_t^{\text{b}\leftarrow}$ and $P_t^{\text{b}\rightarrow}$ are non-negative, and at least one of them must be zero.

$$P_t^{\text{b}\leftarrow} = \begin{cases} 0 & \text{if } -P_t^{\text{R}} < 0 \\ -P_t^{\text{R}} & \text{if } 0 \le -P_t^{\text{R}} \le \overline{P_t^{\text{b}\leftarrow}} \\ \overline{P_t^{\text{b}\leftarrow}} & \text{if } -P_t^{\text{R}} > \overline{P_t^{\text{b}\leftarrow}} \end{cases} \qquad \forall t \in T \tag{3.21}$$

$$P_t^{\text{b}\rightarrow} = \begin{cases} 0 & \text{if } P_t^{\text{R}} < 0 \\ P_t^{\text{R}} & \text{if } 0 \le P_t^{\text{R}} \le \overline{P_t^{\text{b}\rightarrow}} \\ \overline{P_t^{\text{b}\rightarrow}} & \text{if } P_t^{\text{R}} > \overline{P_t^{\text{b}\rightarrow}} \end{cases} \qquad \forall t \in T \tag{3.22}$$

$$P_t^{\text{b}} = -P_t^{\text{b}\leftarrow} + P_t^{\text{b}\rightarrow} \qquad \forall t \in T \tag{3.23}$$

Later, ens_t and curt_t variables are set to satisfy the system balance equation.

Once the balance equation is solved, the state of charge S_t^{b} of all storage devices is computed using the corresponding ESS energy balance equations—for example, (3.24) and (3.25) for the LI-ION battery and the hydrogen system, respectively (here, b refers to the LI-ION technology). In these equations, η^{b} and ζ^{b} denote the charge and discharge efficiency constants, respectively.

$$S_t^{\text{b}} = \begin{cases} S_{t-1}^{\text{b}} - \mid P_t^{\text{b}} \mid / \zeta^{\text{b}} & \text{if} \quad P_t^{\text{b}} \ge 0 \\ S_{t-1}^{\text{b}} + \mid P_t^{\text{b}} \mid \eta^{\text{b}} & \text{if} \quad P_t^{\text{b}} < 0 \end{cases} \qquad \forall t \in T \tag{3.24}$$

$$S_t^{\mathrm{h_2}} = \begin{cases} S_{t-1}^{\mathrm{h_2}} - \mid P_t^{\mathrm{h_2}} \mid /\zeta^{\mathrm{h_2}} & \text{if} P_t^{\mathrm{h_2}} \geq 0 \\ S_{t-1}^{\mathrm{h_2}} + \mid P_t^{\mathrm{h_2}} \mid \eta^{\mathrm{h_2}} & \text{if} P_t^{\mathrm{h_2}} < 0 \end{cases} \qquad \forall t \in T \tag{3.25}$$

Both storage devices must operate within their feasible limits, which are enforced through continuous monitoring at each decision step. In practice, the SoC of a real battery cannot be directly measured and is instead estimated from observable variables such as voltage and current.

3.4 Case study

This section describes the case study used in this chapter, as well as the different configurations of the proposed DRL that have been used to perform the comparative analysis.

3.4.1 Description of the data

The time series data used for the experiments can be found in François-Lavet et al. (2016). In particular, for both the PV generation and demand, with an hourly profile of three consecutive years. The input data includes the maximum generation profile of the solar panels, the hourly energy load, and the technical specifications of all other components. Most of the parameters related to the characteristics of the components were obtained from François-Lavet, Gemine, et al. (2016) and Jasmin et al. (2011). The remaining parameters were obtained from the manufacturer's specifications of specific components available in the market, as well as commonly assumed values. All these parameters are explained in detail hereafter and summarized in Table 3.1.

3.4.1.1 Photovoltaic Panels

The dataset described in 3.4.1 contains a PV generation profile that was gathered from a Belgium location. This profile demonstrates a factor of 1 to 5 among different months based on accumulated values, which represents a challenge where the introduction of more components is necessary to supply the demand. The PV system uses a standard panel size of 6kW, which is discussed in EnergySage (2024). This size also aligns with a reasonable heuristic approach of dividing the annual demand by the total irradiation, taking into account a standard efficiency for the PV panels. Consequently, the PV system designed for this case study encompasses an area of $30m^2$, with a maximum installed capacity of 6kW. This capacity is half of what is proposed in François-Lavet, Gemine, et al. (2016), complicating the problem due to the limited availability of renewable energy and necessitating more precise control over the other components, specifically of the Di-Gen that has been included to face the limitation mentioned.

3.4.1.2 Load

The dataset described in 3.4.1 contains a demand profile that represents a typical shape of residential consumption. In this case, it has an average daily demand of 18.33 kWh, but this particular value is pure academic, with the sole intent to develop a case study where all the elements are crucial in the operation. Moreover, each non-served kWh has a 1€ cost, which is a high-enough penalization to encourage the algorithm to avoid it at all costs. This value is commonly used in the

Table 3.1: Microgrid parameters.

Component	Parameter	Value	Unit
PV	$P^{pv}_{\max}$	6	[kW]
Load	$D_{\max}$	2.1	[kW]
Di-Gen	$P^{\mathrm{d}}_{\max}$	1.0	[kW]
	δ_2	0.31	[€/kW^2]
	δ_1	0.108	[€/kW]
	δ_0	0.0157	[€]
Li-ion battery	$P^{\mathrm{b}}_{\max}$	3.0	[kW]
	S^{b}_0	0	[kWh]
	$S^{\mathrm{b}}_{\max}$	2.9	[kWh]
	η^{b}	0.95	[-]
	ζ^{b}	0.95	[-]
H_2 storage	$P^{\mathrm{h_2}}_{\max}$	1.0	[kW]
	$S^{\mathrm{h_2}}_0$	100	[kWh]
	$S^{\mathrm{h_2}}_{\max}$	200	[kWh]
	$\eta^{\mathrm{h_2}}$	0.65	[-]
	$\zeta^{\mathrm{h_2}}$	0.65	[-]
ENS	c^{ens}	1	[€/kWh]

literature and supposes a huge penalization for the DRL algorithm, considering that the Di-Gen cost does not surpass 0.5€/kWh generated in the worst case.

3.4.1.3 Energy storage elements

The Li-ion battery charge and discharge efficiency rates are 0.95 for both processes. The hydrogen electrolyzer (charge) and FC (discharge) have efficiency rates of 0.65 for both cases (François-Lavet, Gemine, et al., 2016). The battery capacity and hydrogen storage capacity are 2.9 kWh and 200 kWh. The battery model imitates an LG Chem RESU3.3 designed for a household with a maximum discharge power rate of 3.0 kW The hydrogen FC is based on a Horizon 1000 W Proton Exchange Membrane (PXM), which has a maximum power ratio of 1.0 kW. By symmetry, the electrolyzer also has a maximum power ratio of 1.0 kW.

3.4.1.4 Non-renewable generation

The proposed microgrid has a Di-Gen with a nominal rate of 1.0 kW. The cost curve of the Di-Gen generator has been adjusted to a quadratic curve, fitted by (3.12). These coefficients were adapted from the Institute of Electrical and Electronics Engineers (IEEE) 30 bus system generators used in Jasmin et al. (2011), and shown in Table 3.1.

3.4.2 Optimization-based model MIQP used as a reference model and benchmark

In the hypothetical case where the microgrid's hourly demand and solar power for the entire time horizon were known, it would be possible to formulate a deterministic optimization problem in order to obtain the most favorable scheduling of all the elements of the microgrid. The value of the objective function in this setting would represent the optimal solution in the best-case scenario. This value could serve as a benchmark value to compare the results obtained by alternative methods. The decision variables of this optimization problem are the charge and discharge power of the batteries, the possible generation of the Di-Gen, the ENS, and the possible spillages. This benchmark model was implemented in General Algebraic Modeling Language (GAMS) (GAMS Development Corporation, 2013), where the detailed equations were omitted here for the sake of simplicity, but added in Appendix A.

As the independent term of the Di-Gen cost is only incurred when the generator is on, it requires the use of unit-commitment binary variables. The constraints include the energy balance at the microgrid and the energy in the storage devices, taking into account their charge and discharge efficiencies, their maximum storage levels, and the maximum rated power. As the objective is a quadratic function and some binary variables are needed, the resulting model is a Mixed Integer Quadratic Programming (MIQP).

3.4.3 Naive strategy

A naive algorithm has also been implemented to mimic the results that a simple strategy could achieve. The insight of this method is to charge the batteries when there is a surplus of energy, and discharge them otherwise. The pseudocode of this strategy is shown in Algorithm 1, where *pv_gen* is the PV power [kW] and *load* is the household consumed power [kW].

Algorithm 1 Naive algorithm.

```
$R^+ = R^- = 0$
if pv_gen > load then                              ▷ If energy surplus then charge the battery
    $R^+ = \text{pv\_gen} - \text{load}$
    $S^{b\leftarrow} = S_{max} - S^{b}$
    $P^{b\leftarrow} = \min\{R^+, S^{b\leftarrow}(\eta^{b})^{-1}, P^{b\leftarrow}_{max}\}$
    $R^+ = R^+ - P^{b\leftarrow}$
    $P^{h_2\leftarrow} = \min\{R^+, S^{h_2}, P^{h_2\leftarrow}_{max}\}$
    $R^+ = R^+ - P^{h_2\leftarrow}$
else                                               ▷ If not then discharge the battery
    $R^- = \text{load} - \text{pv\_gen}$
    $P^{b\rightarrow} = \min\{R^-, S^{b}, P^{b\rightarrow}_{max}\}$
    $R^- = R^- - P^{b\rightarrow}$
    $P^{h_2\rightarrow} = \min\{R^-, S^{h_2}, P^{h_2\rightarrow}_{max}\}$
    $R^- = R^- - P^{h_2\rightarrow}$
    $P^{d} = \min\{P^{d}_{max}, R^-\}$
end if
curt = $R^+$                  ▷ Energy surplus after using the battery is curtailed or not supplied
ens = $R^-$
```

It is assumed that the naive algorithm could be implemented in a microcontroller, making decisions in a continuous manner based on the instantaneous status of the microgrid. In this sense, this differs from the DRL approach of this chapter that would require at least the information of the last hour in case the size of the time window was 1 h.

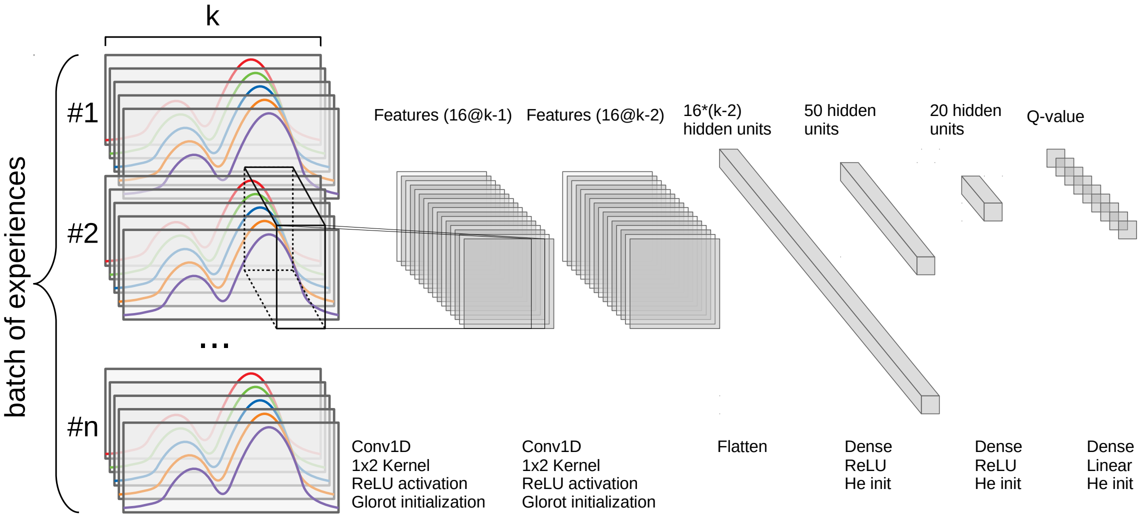


Figure 3.4: CNN architecture used in the proposed DQN.

3.4.4 DQN configuration

This chapter uses a DQN due to its implementation simplicity and powerful performance (Mnih et al., 2015). Additionally, three main modules have been implemented to address the EMS problem.

1) The neural network includes a CNN to deal with the observation window. Its particular architecture is depicted in Figure 3.4.

2) An ER memory (Lin, 1993) to store the experiences gathered during the interactions and later train the NN using mini-batches randomly selected from the ER.

3) A method to process the agent observations and combine them with internal agent data (the last observations), to make the next state and store it in the memory.

The CNN architecture proposed is similar to the one in François-Lavet, Taralla, et al. (2016). This CNN takes a whole tensor as the input of the first layer instead of scatter time series input in 1D convolutional layers, and is merged with the remaining inputs after the convolution computation. This configuration brings more simplicity and scalability to the model without losing performance in the results despite the information redundancy. Our CNN includes two 1D convolutional layers. The initialization procedures of these layers are the ones from Glorot and Bengio (2010), called Glorot initialization, for the convolutional layers and from He et al. (2015), called He initialization, for dense layers. All the parameter values used are shown in Table 3.2.

In order for the model to operate over unknown data, a regularization technique must be applied during its training phase. In this work, the technique applied is called *early stopping*. This technique consists of splitting the training data into a smaller training set and a validation set. The validation set helps to choose the best snapshot of the model before the model starts to overfit. The model is continuously monitored to observe its performance over the validation set while its parameters are adjusted using the training set. In particular, the NN policy $\pi_\theta(s) = \arg\max_a Q(s, a|\theta)$ is evaluated over the microgrid with the validation set after each θ update. If

Table 3.2: DQN parameters.

Parameter	Value
Batch size	20
Memory size	10000
Optimizer	NADAM (Dozat, 2016)
Error measure	MSE
Exploration function	$f(t) = 0.1 + 0.9e^{-t*10^{-6}}$
Early stopping scope	200
Discount factor (γ)	0.99

the obtained $G(\rho|\theta)$ improves the best of the ones already evaluated so far, that particular θ is saved as the best policy found so far. Also, the implemented *early stopping* method uses a termination condition. If after a fixed number of steps—the *early stopping scope*—is reached under a particular condition, the learning process is stopped. This condition works as follows: when the best policy has not been improved anymore, the training process is stopped. This is a particular implementation of the *early stopping* technique carried out, but others can be used as well.

This work carries out the analysis of different window sizes defined by the parameter k, as stated in (3.1). Specifically, it examines a range from 3 hours to 24 hours, using increments of 3 hours between values. Note that when $t < k$, some of the indices of the observations in (3.1) yield negative values, for instance, when the index is $t - k$. In such cases, the corresponding values for those observations are set to zero.

3.5 Results

This section presents the results obtained with the DQN method applied to the microgrid described previously. The DQN has been implemented in Python 3.7.7 using TensorFlow 2.1.0. The MIQP model has been coded in GAMS using the solver CPLEX 12.9.0.0. The computer system was an Intel Core i7-8550U (1.80GHz - 4.00GHz) with 16GB RAM running under Ubuntu 18.04 LTS x64.

The results gather several options for the definition of the considered information of the microgrid observation in each timestep, varying the window size parameter k in (3.1). As stated before, the motivation of this analysis is to find how much recent information should be used in each timestep in order to obtain the best performance of the microgrid. To put the results obtained with the DQN in perspective, other models have been included in this analysis: the perfect-information MIQP model, the naive strategy, and a random policy. The MIQP model represents an optimistic (or upper) bound of the objective function, i.e., a lower bound of the operation cost that is minimized with such a microgrid. On the other hand, the random policy achieves a lower bound reference value, where the average value of ten realizations (runs of the model) is included in the assessment. Let the undiscounted return $G(\tau|\pi)$ be the sum of the rewards from the trajectory τ resulting from policy π. Based on the three-year dataset, the return G is used to measure the goodness of each proposed policy, including the proposed algorithm in this work, i.e., the DQN. The undiscounted return G is the cost of the microgrid in euros, and the trajectory τ is the operation of the microgrid. Note that, in the case of the DQN, the first two years of available data are used to train that model, and that these years are separated in

Table 3.3: Accumulated cost of each algorithm [€].

	Obj. F. ↔ Cost [€]				
Algorithm	1$^{\text{st}}$ year	2$^{\text{nd}}$ year	3$^{\text{th}}$ year	Total	$\frac{\lvert \text{X}-\text{Reference}\rvert}{\text{Reference}} \cdot 100\%$
MIQP (gap 6%)	967.34	864.05	846.04	2677.43	0.00% (Reference)
RL k=3	1305.94	1126.49	1239.35	3671.77	37.14%
RL k=6	1355.80	1140.92	1248.61	3745.33	39.89%
RL k=9	**1299.56**	**1123.53**	**1230.50**	**3653.59**	**36.46%**
RL k=12	1389.60	1198.63	1308.11	3896.33	45.52%
RL k=15	1348.61	1174.02	1268.02	3790.64	41.58%
RL k=18	1503.69	1307.67	1443.58	4254.94	58.92%
RL k=21	1634.33	1441.21	1551.68	4627.22	72.82%
RL k=24	1515.46	1320.44	1439.12	4275.02	59.67%
Naive	3778.74	3681.04	3678.82	11138.60	316.02%
Random	4816.64	4554.78	4695.17	14066.59	425.38%

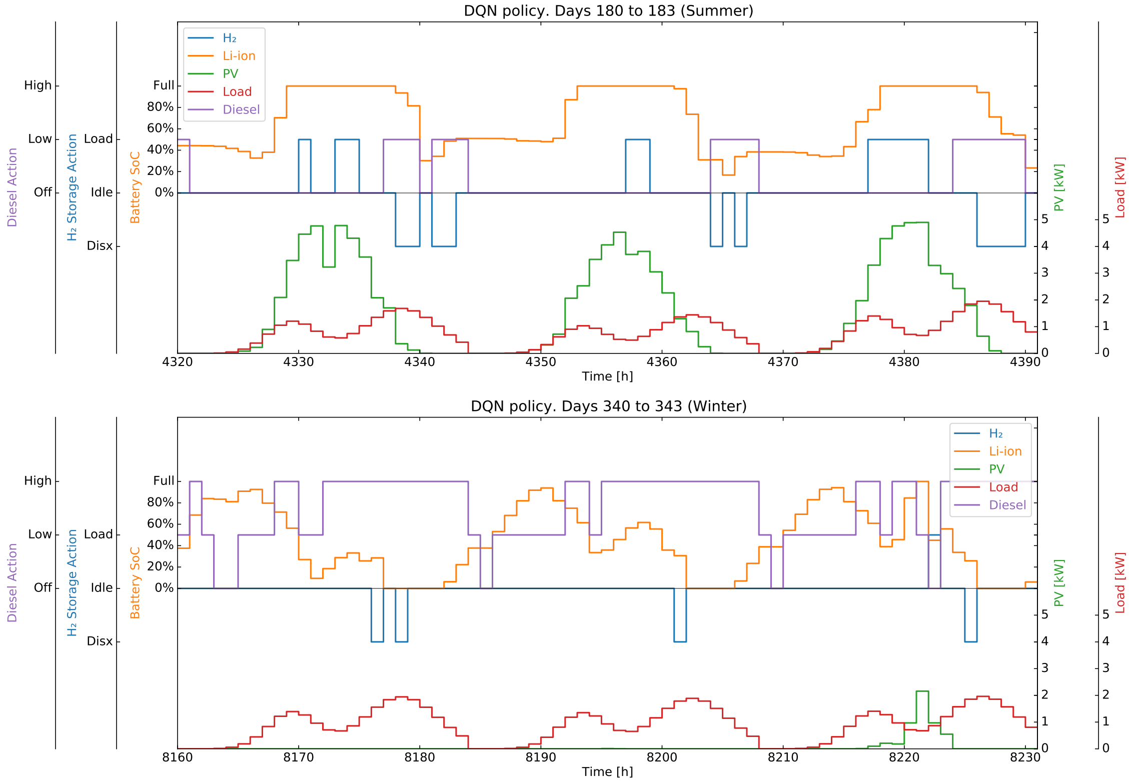
the training set (the first year) and the development set (the second year), whereas the last year is used for testing. Concerning the reference solution provided by the MIQP, the model operates the microgrid without any discretization of the decision variables. The same applies to the naive and the random strategies.

The MIQP model generates the upper bound given by a three-year accumulated cost of 2677.43€, with a relative gap of 6.06% after 24 hours of computing. Gap is calculated following the formula provided by the GAMS software, displayed in (3.26).

$$gap = \frac{|\text{Best feasible} - \text{Current bound}|}{|\max\{\text{Best feasible}, \text{Current bound}\}|} \tag{3.26}$$

Table 3.3 shows the total cost incurred after three years of operating the microgrid based on the policy learned through different configurations of the proposed DQN. Using the same idea when computing the gap, the Relative Error (RE) between the method analyzed and its upper bound is shown in the last column of this table. The results obtained with the DQN are significantly superior to those obtained using other strategies, with costs ranging from 3653.59€ for the best result with a window size of 9 to 4627.22€ for the worst result with a window size of 21. Notably, even the higher cost associated with the least effective window size is far better than that resulting from the naive or random strategies. Moreover, when compared to the reference—the MIQP model—the performance associated with the least effective window size is twice as high as that of the best-chosen window size. These findings highlight the significant impact of the window size parameter on the DQN's performance; a larger window size can lead to increased challenges in learning the policy for the DQN. Another observation is the comparison between the best-case DQN and the MIQP results. The DQN demonstrates strong performance while remaining relatively easy to implement. When compared to the MIQP approach, the worst-case performance of DQN deviates by at most 36% from the ideal optimum (because that value is an upper bound). Given that MIQP benefits from perfect foresight, it is reasonable to infer that, even in the worst case, the actual deviation is likely lower.

Figure 3.5 shows an example of the hourly microgrid operation carried out by the proposed


DQN policy. Days 180 to 183 (Summer)

DQN policy. Days 340 to 343 (Winter)

Figure 3.5: DQN operation of the microgrid, for summer (top) and winter (bottom). Hourly steps in a 3-day window.

DRL method. This snapshot of the EMS operation covers six days from the first year, including three consecutive winter days and three summer days. The upper part of this figure shows the operation of the Di-Gen and the hydrogen system, and also the energy storage of the Li-ion battery.

Figure 3.5 shows a clear daily pattern operation of the Li-ion battery, but it differs with the season. During summer, the daily energy surplus provided by the solar panels is used to fill this battery during daylight hours, being systematically discharged during the rest of the hours. However, the lack of solar irradiation in winter is compensated by the continuous generation of the Di-Gen to cover the demand in the main hours, using the Di-Gen surplus of the off-peak hours to charge the battery for its usage during non-sunlight hours. On the contrary, during summer, the use of the Di-Gen is limited to some hours where there is not enough solar irradiation. Concerning the hydrogen FC, it is used in a similar Di-Gen pattern, to cover the demand when there PV generation is not enough.

A numerical example is presented hereafter in order to illustrate the behavior of the model. Assuming that the state s_t considers a tensor of shape $(k, 4)$ with $k = 9$, from the output results

Table 3.4: Hydrogen usage over three years.

	Hydrogen $\Delta S^{\mathrm{h_2}}$ [kWh]		
Operation mode	1st year	2nd year	3rd year
Charging	546.00	587.50	578.50
Discharging	594.50	586.00	578.00

shown in Figure 3.5 (top), the detailed values that correspond to hour $t = 4380$ are the next ones:

$$h_t = \left(o_{t-k+1}, \quad ..., \quad o_{t-1}, \quad o_t\right) = \\ = \begin{pmatrix} P^{\mathrm{pv}}_{t-k}, & ..., & P^{\mathrm{pv}}_{t-2}, & P^{\mathrm{pv}}_{t-1} \\ D_{t-k}, & ..., & D_{t-2}, & D_{t-1} \\ S^{\mathrm{b}}_{t-k+1}, & ..., & S^{\mathrm{b}}_{t-1}, & S^{\mathrm{b}}_{t} \\ S^{\mathrm{h_2}}_{t-k+1}, & ..., & S^{\mathrm{h_2}}_{t-1}, & S^{\mathrm{h_2}}_{t} \end{pmatrix} = \\ = \begin{pmatrix} 0.002, & 0.151, & 0.461, & 1.122, & 1.973, & 3.301, & 4.295, & 4.767, & 4.891 \\ 0.061, & 0.186, & 0.447, & 0.837, & 1.222, & 1.398, & 1.270, & 0.963, & 0.711 \\ 1.028, & 0.989, & 1.001, & 1.257, & 1.932, & 2.260, & 2.900, & 2.900, & 2.900 \\ 36.000, & 36.000, & 36.000, & 36.000, & 36.000, & 36.650, & 37.300, & 37.950, & 38.600 \end{pmatrix}.$$

Then, the action taken by the DQN, using its policy function—which is the argmax function previously defined in (1.5)—is:

$$a_t = \left(P^{\mathrm{d}}_t, \quad P^{\mathrm{h_2}}_t\right) = \\ \left(0.000 \quad -1.000\right)$$

As a consequence of this action, the new state in hour $t+1 = 4381$ is reached, with its corresponding new observation stack:

$$h_{t+1} = \left(P^{\mathrm{pv}}_t, \quad D_t, \quad S^{\mathrm{b}}_{t+1}, \quad S^{\mathrm{h_2}}_{t+1}\right) = \\ = \left(4.899, \quad 0.672, \quad 2.900, \quad 39.250\right),$$

in which P^{b}_t, that is calculated using the equation (A.3), takes the value 0.0 kW because it is full, and the curtailment P^{curt}_t, that is calculated using the balance equation (3.17), takes the value of 3.217 kW.

Another interesting result that can be highlighted is the management of the hydrogen storage along the year. One could expect that since there is less solar generation in winter, it would be better to store energy during summer in order to have it available when necessary. Figure 3.6 shows the hydrogen energy storage for the whole three-year period since this profile is barely appreciable in the 3-day window of Figure 3.5. It can be seen that the proposed DQN is capable of finding an optimal yearly pattern, charging the hydrogen tank in summer and discharging it in winter, and this optimal behavior is obtained just with the most recent information at each step. A summary of the usage of the hydrogen storage is shown in Table 3.4.

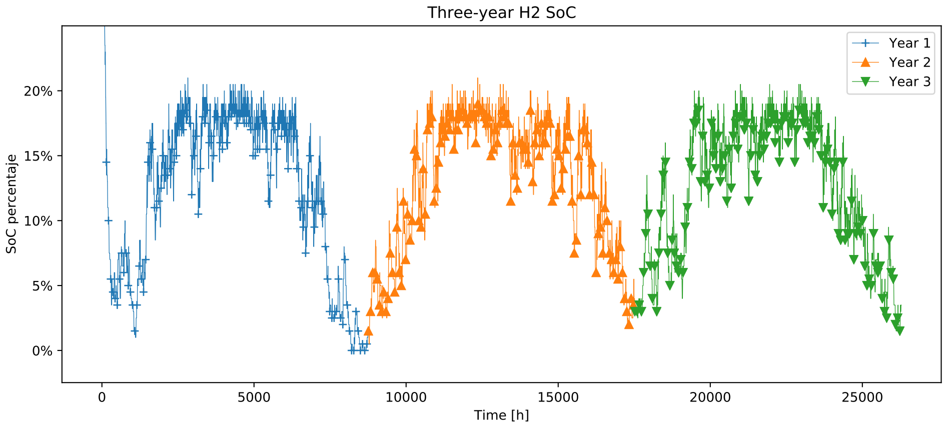


Figure 3.6: Hydrogen SoC percentage over three years.

3.6 Final remarks

This chapter presents the application of DRL techniques to manage the elements of an isolated microgrid. The advantage of the proposed approach lies in the algorithm's ability to learn from its own experience, allowing it to adapt to the specific characteristics of the microgrid without requiring an explicitly designed optimization model. This self-learning capability enables the algorithm to refine its decision-making over time based on observed operational patterns, making it a promising solution for scaling up microgrid management without the need for tailor-made optimization models for each case. However, as the application of RL, and specifically DRL to power systems is an emerging research field, the selection of the most appropriate ANN architecture, or the definition of what information should be contemplated for the EMS, are questions that will be tackled in the following chapters. Concerning the first one, the proposed CNN architecture provides simplicity and satisfactory performance. Moreover, this study provides a sensible set of variables to configure the EMS input regarding the second question. In addition, the effect of increasing the window size defined has been analyzed, and the numerical results show that considering a size of stacking 9 contiguous hours is optimal for the studied microgrid. In order to measure the quality of the DRL results, this work uses a MIQP optimization model to obtain an upper bound solution as a reference in terms of the obtained reward. With such a reference model as a benchmark, a naive algorithm similar to the ones used in practical implementation results in an operation 316.02% worse, whereas the developed DRL model reaches a value of 36.46 % (the lower, the better). As a final remark, the results indicate that while the naive strategy performs slightly better than a purely random approach (316.02% vs. 425.38%), its performance remains far from optimal. This highlights the critical need for more sophisticated control strategies, as relying on a quickly developed heuristic can lead to highly inefficient microgrid operation. The substantial gap between these approaches further emphasizes the importance of employing advanced learning-based methods to achieve more reliable and cost-effective energy management. To conclude, the EMS problem addressed in this chapter exhibits a continuous nature, which is further discussed in the following chapter. Although the DQN algorithm does not natively support continuous action spaces—requiring specific adaptations such as the discretization applied here—it remains a simple, stable, and effective approach that yields satisfactory results. The next chapter explores an alternative method designed to overcome this limitation, albeit at the expense of increased computational complexity and potential challenges in training stability, which are less prominent when using DQN.

Chapter 4

TD3 Algorithm for the EMS of Microgrids

4.1 Introduction

In the literature, the DQN method, used in Chapter 3, has proven to be an effective alternative to classical optimization approaches, offering several advantages in microgrid management. One such advantage is their ability to operate without reliance on external predictors, significantly simplifying the overall EMS design. Nevertheless, most studies on energy management optimization using DRL focus on short-term horizons—often limited to a single day, and few to a couple of weeks—leaving the long-term dynamics of microgrid operation largely unaddressed.

This chapter addresses this gap by considering a multi-year time horizon and incorporating several modeling "tricks" learned from Chapter 3 to ensure near-optimal solutions. In particular, the previous chapter demonstrated the efficacy of applying a DQN-based strategy to microgrid control, yielding satisfactory outcomes. However, one notable shortcoming of that approach was the need to discretize the action space, which can limit its control granularity. To overcome this, this chapter analyzes the performance of the TD3, briefly described in Section 1.3.4, and its application is one of the contributions of this thesis. The TD3 is particularly well-suited to address the limitations of discrete actions when using the DQN, and its control over continuous action spaces has the potential to produce more precise—and thus more optimal—decisions. Despite these advantages, TD3 inherits stability challenges from DDPG, although the improvements introduced in TD3 partially mitigate these issues. To further improve learning stability and enhance performance, this chapter conducts a methodical hyperparameter optimization, a practice increasingly adopted in the literature of DRL. Additionally, the stability of the learning process is monitored to ensure the performance of the algorithm, since an unstable algorithm can result in significant performance issues. This expanded scenario enables us to validate the proposed method's adaptability to a wider range of microgrid architectures and confirm its suitability for real-world applications requiring flexible, robust, and long-term energy management strategies.

Moreover, hyperparameter selection in many works is done arbitrarily, whereas a broader and more systematic search has been shown to improve performance. Efficiently conducting such searches requires specialized tools designed for hyperparameter optimization. Finally, while existing case studies are often well-constructed, they primarily focus on residential-scale microgrids. This chapter aims to extend the scope by exploring both a smaller-scale microgrid and a larger low-voltage distribution network, ensuring broader applicability.

In summary, the main contributions of this chapter with respect to previous works are:

- The application of the TD3 (Fujimoto et al., 2018) to the microgrid energy management problem with a residential demand, in isolated mode, using three years of hourly data, overcoming the action discretization seen in Chapter 3. This approach is appropriate for considering continuous actions over the components of the microgrid.
- The performance of the TD3 technique is enhanced using the Tree-structured Parzen Estimator (TPE), a Gaussian Mixture Model (GMM)-based method (Bergstra et al., 2011) for hyperparameter optimization (Bergstra et al., 2011), implemented via the Optuna framework (Akiba et al., 2019). Hyperparameter optimization not only enhances performance but also facilitates the discovery of stable configurations, leading to more reproducible and consistent learning outcomes.
- A performance comparison is carried out between two different DRL algorithms. The DQN (Mnih et al., 2015), a more common approach, and the TD3. Furthermore, this chapter discusses the advantages and disadvantages of both methods and why the TD3 obtains better results than the DQN on the energy management problem.
- In addition to a small-scale microgrid system, the TD3 algorithm is implemented as the EMS of the CIGRE microgrid benchmark (Kariniotakis et al., 2005; Papathanassiou et al., 2005). The results from applying the proposed algorithm implementation, with the chosen hyperparameters, show that it is a candidate algorithm to manage a real-world microgrid efficiently.

This chapter is organized as follows. The second section describes the algorithm proposed. The third section describes in detail the mathematical formulation of the microgrid model and its implementation. The fourth section explains the strategies used and the particularities of the case study and comments on the comparison between both DRL techniques. Finally, the last section presents the conclusions.

4.2 An Actor-Critic method for continuous control of the EMS

In the previous chapter, the DQN was used for the EMS of an isolated microgrid. However, that algorithm can only make decisions over a discrete set of actions. This caveat conflicts with a real scenario in power systems, where components such as a Di-Gen or a ESS can dispatch any power level in a continuous range. Given that DQN can only make decisions over a discrete set of actions, alternative techniques such as DDPG have been developed to address this limitation (Lillicrap et al., 2015). The DDPG modifies the NN architecture to additionally use the action as an input in order to compute the value of the Q-function. At the same time, another NN is used to compute the value of the action, becoming the policy function. This architecture is denoted as Actor-Critic. Figure 4.1 shows an NN architecture using such an actor-critic approach where the actor network

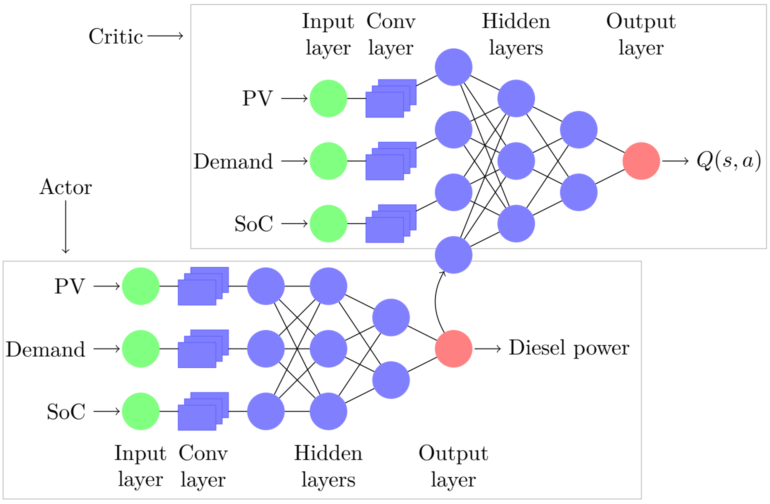


Figure 4.1: TD3 architecture example.

estimates the $\pi(s)$, and the critic network estimates $Q(s,a)$ (Watkins & Dayan, 1992). However, due to the instability of DDPG learning, achieving convergence is challenging. This increases the experimental burden, as longer training and repeated runs are needed to account for variability and ensure reproducible results.

The TD3 algorithm extends the fundamental DRL equations introduced in Chapter 1 (Sections 1.3.2 and 1.3.4). The exact equations for the TD3 algorithm are omitted for the sake of brevity, but they can be found in Fujimoto et al. (2018), as well as in online resources such as Achiam (2018).

4.3 Microgrid framework using an MDP

As seen in Chapter 3, applying an RL algorithm requires defining an MDP. Specifically, this involves specifying the set of states, actions, state transitions, reward function, observations, and conditional observation probabilities (Kaelbling et al., 1998).

For the optimal microgrid management problem considered, the following subsections outline the relationship between the mathematical formulation and the microgrid configuration depicted in Figure 4.2.

In this chapter, some of the equations defining the MDP are reintroduced due to differences in the microgrid configuration compared to the previous one. The main distinction is the exclusion of the hydrogen component, a choice made to establish a more standardized microgrid benchmark. In practice, microgrids are typically composed of renewable sources—primarily solar energy—together with an ESS, such as the Li-ion battery used here, and an auxiliary fuel-based generator that compensates for renewable shortages when necessary. Accordingly, several equations from Section 3.3 have been adapted to reflect this simplified configuration. The specific changes are:

- The hydrogen tank's stored energy is no longer included in the observation vector.

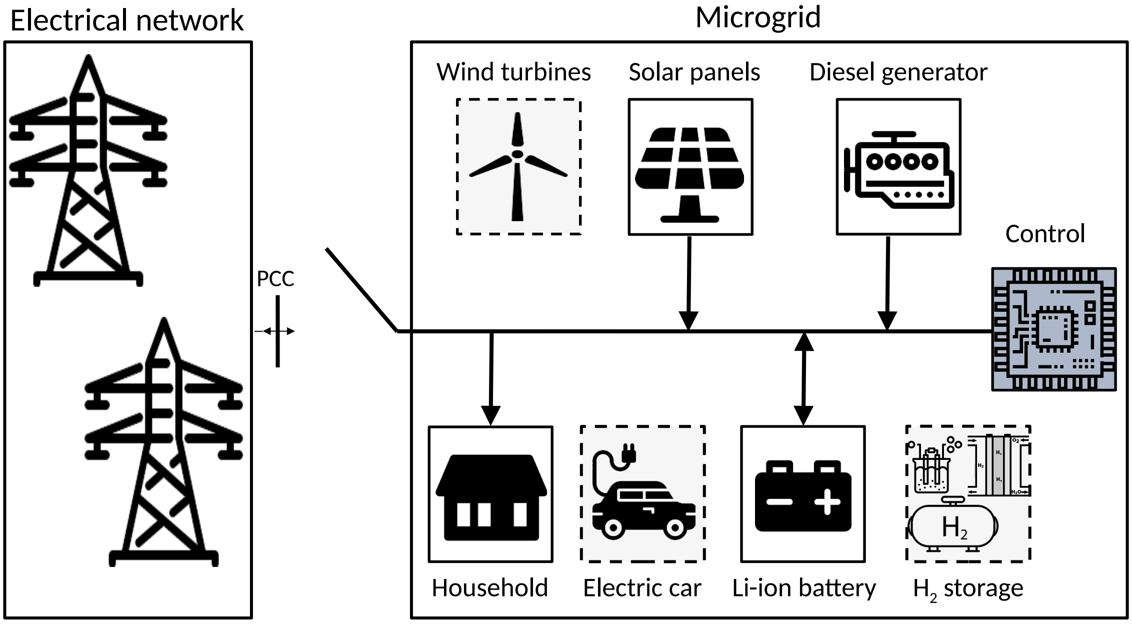


Figure 4.2: Residential microgrid scheme using a PV, a Li-ion battery and a Di-Gen.

- The power flow between the hydrogen storage system and the microgrid is no longer part of the EMS action space.
- The balance equation no longer contains variables associated with the hydrogen storage system.

4.3.1 State definition

The observation notation follows the formulation introduced in Chapter 3, with the main difference that the hydrogen component is not included in this case. Apart from this modification, the remaining details of the observation design remain unchanged.

$$o_t = (P^{\mathrm{pv}}_{t-1}, D_{t-1}, S^{\mathrm{b}}_t) \tag{4.1}$$

Additionally, the size k of the window-based belief h_t is fixed to three consecutive observations, a choice motivated by the experimental results presented in the previous chapter. Although the optimal window size was found to be nine, using three observations achieved nearly identical performance (37.14% vs. 36.46%) while substantially reducing the dimensionality of the state space. By requiring only one-third of the representation size, a window length of three is expected to facilitate faster learning for the algorithm.

4.3.2 Action definition

Building on the general definition of the action space introduced in Section 3.3.2, the specific formulation employed in this chapter is given in (4.2), where the battery is the omitted element.

$$a_t = (P_t^{\mathrm{d}}) \tag{4.2}$$

The discrete action space remains limited to three values as described in (3.9) (this time hydrogen is discarded). In contrast, in the continuous action space, the decisions span a range defined as $P^{\mathrm{d}} \in \{0\} \cup [P_{\mathrm{min}}^{\mathrm{d}}, P_{\mathrm{max}}^{\mathrm{d}}]$.

In this chapter, the TD3 algorithm is introduced, which operates in continuous action spaces. By default, the action space is normalized to $(-1, 1)$ through the *tanh* activation function (Fujimoto et al., 2018; Raffin et al., 2021) (ref to Section 4.4.2.2 for more details), which does not directly match the definition previously established for P^{d}. To address this, a custom mapping function is defined as $P^{\mathrm{d}} \colon (-1, 1) \to \{0\} \cup [P_{\mathrm{min}}^{\mathrm{d}}, P_{\mathrm{max}}^{\mathrm{d}})$.

Let $z \colon (-1, 1) \to (0, Z)$ be a linear function given in (4.3), where y denotes the output of the *tanh* activation function. Setting $Z = P_{\mathrm{max}}^{\mathrm{d}}$ maps the normalized output space to one that aligns with the feasible operating range of the Di-Gen. The final definition of P^{d} is then provided as a piecewise function in (4.4).

$$z = P_{\mathrm{max}}^{\mathrm{d}} \cdot \frac{y+1}{2} \tag{4.3}$$

$$P_t^{\mathrm{d}} = \begin{cases} 0 & \text{if } z < P_{\mathrm{min}}^{\mathrm{d}} \\ z & \text{if } P_{\mathrm{min}}^{\mathrm{d}} \leq z \leq P_{\mathrm{max}}^{\mathrm{d}} \\ P_{\mathrm{max}}^{\mathrm{d}} & \text{if } z > P_{\mathrm{max}}^{\mathrm{d}} \end{cases} \tag{4.4}$$

4.3.3 Reward definition

The reward function remains as in Section 3.3.3.

4.3.4 Transition definition

All equations related to the transition function of the MDP in this chapter follow those presented in Section 3.3.4, with the sole modification that the hydrogen component is excluded. For the sake of brevity, the explicit equations are not repeated here.

4.4 Implementation and results

This section presents the implementation and results obtained by applying two types of RL algorithms to the energy management case of a microgrid. The algorithms applied are a value-based type named DQN (Mnih et al., 2015) and an actor-critic type named TD3 (Fujimoto et al., 2018). The reason for using both algorithms is to analyze the potential of addressing the continuous action spaces of the microgrid components, like the Di-Gen and the battery that the TD3 can exploit.

The implementation of the microgrid simulator and the DRL algorithms applied are described through the metrics monitored during the training process. All simulations have been run on a server with Ubuntu 20.04.1, an Intel i9-10900X CPU, two Nvidia GeForce RTX 3090 GPUs, and CUDA version 11.2.

Table 4.1: Microgrid parameters.

Component	Parameter	Value	Unit
Load	D_{max}	2.1	[kW]
PV	$P^{\text{pv}}_{\text{max}}$	6	[kW]
Di-Gen	$P^{\text{d}}_{\text{min}}$	0.1	[kW]
	$P^{\text{d}}_{\text{max}}$	1.0	[kW]
	δ_0	0.0157	[€]
	δ_1	0.1080	[€/kW]
	δ_2	0.3100	[€/kW^2]
Li-ion	S^{b}_0	0.0	[kWh]
	$S^{\text{b}}_{\text{min}}$	0.0	[kWh]
	$S^{\text{b}}_{\text{max}}$	3.3	[kWh]
	$P^{\text{b}\leftarrow}_{\text{max}}$	2.97	[kW]
	$P^{\text{b}\rightarrow}_{\text{max}}$	2.97	[kW]
	η^{b}	0.9	
	ζ^{b}	0.9	
ENS	c^{ens}	1	[€/kWh]

4.4.1 Microgrid implementation

The microgrid behavior is implemented using the *Gymnasium* software framework (Towers et al., 2023) described in Section 1.3.2.1.

The first process initializes the microgrid state, setting up the environment before interaction begins and providing an initial set of observations. The second process updates the environment based on the agent's chosen action, computing the resulting state, determining the associated operational costs, and returning the corresponding reward along with the updated observations.

This framework was not used in the previous chapter because it was not known at the time of its implementation; however, it is highly recommended as it streamlines the integration of other RL frameworks, promoting modularity and compatibility with various RL algorithms.

As in Chapter 3, the initial SoC S^{b}_0 is set to zero, and both solar irradiation and power demand are assumed to be zero for $t < 0$. Consequently, the first observation and the first state correspond to vectors of zeros. The next-state dynamics follow the methodology described in Section 3.3.4, and the reward function is defined as in Section 3.3.3, with the sole modification that all hydrogen-related variables are omitted.

The dataset used for P^{pv}_t and D_t variables of the observation vector are the same than as in Chapter 3. Additionally, Table 4.1 shows the parameters of the microgrid used.

4.4.2 Agent implementation

In the introductory chapter, specifically in Section 1.3.2, we describe the concept of an agent in RL. In the case of a microgrid, this agent corresponds to the EMS, as highlighted in the previous chapter. However, implementing this DRL agent in practice involves several important decisions, such as selecting the appropriate algorithm, setting its parameters, and structuring the code effectively. This section takes a closer look at these issues, aiming to guide readers in reproducing the methods used in this research. It identifies the most critical elements and offers

Table 4.2: Feature-extractor configuration for DQN and TD3.

Layer	Features	Kernel size	Stride	Padding
Conv #1	32	3	1	2
Conv #2	64	3	1	2

insights on how to evaluate them properly. A contribution of this research is demonstrating how these techniques can be effectively applied, appropriately combining open-source tools that are easy to access. More specifically, this section outlines the software utilized, the structure of the neural network, and the chosen hyperparameters, along with the process of determining them. By sharing these details, we hope to help others in understanding, implementing, and enhancing the approaches presented here.

4.4.2.1 Software

The DRL algorithms in this work have been implemented using Stable-Baselines3 (SB3) (Raffin et al., 2021), an open-source framework with reliable high-quality algorithms. The SB3 framework requires a defined action space and a state space model. *Gymnasium*, as described in Section 4.4.1, provides these elements, allowing seamless integration with SB3. Also, the NN architecture is implemented using PyTorch (Paszke et al., 2019), and explained in Section 4.4.2.2 with more detail.

The framework already has the DRL algorithms implemented, making the integration and replication more accessible, and serves as a baseline against custom implementations (Raffin et al., 2019), (Henderson et al., 2018).

4.4.2.2 Neural network architecture

The NN architecture used in both DQN and TD3 algorithms is composed of a feature-extractor made of 1D-convolutional layers, followed by a flatten layer and a MLP of fully-connected layers. This general schema is the same as that used in Chapter 3 for the DQN, but in this chapter, the hyperparameters used are different. For the TD3, Figure 4.1 depicts the characteristic actor-critic schema of the NN that it uses. In the TD3, the weights of the feature-extractor are shared between the actor and the critic. Moreover, the specific number of layers and neurons, for both TD3 and DQN, are detailed in Tables 4.2 and 4.3 for the feature-extractor module and the MLP module, respectively. The non-linear activation functions used in both DQN and TD3 are ReLU (Nair & Hinton, 2010) except for the output of the last layer being the *tanh* for the TD3 actor network and the identity function for the other networks. The initialization in DQN and TD3 is the default used in PyTorch.

In general, this is the common approach implemented in SB3. Regarding the hyperparameters, Section 4.4.2.3 details in depth the selection of them and the methodology.

4.4.2.3 Hyperparameters

Hyperparameter optimization is an aspect of DL that must be addressed to train models effectively. This optimization is critical in DRL algorithms, as maintaining stability during the learning phase

Table 4.3: Three size configurations for the MLP in both DQN and TD3.

Size config.	Layer #1	Layer #2	Layer #3
Small	128	64	64
Medium	256	256	256
Big	500	400	300

can be challenging at times (Henderson et al., 2018). In the previous work described in Chapter 3, hyperparameter optimization was done manually; however, at the time the described work was being carried out, tools for automating that process became quite popular. Several of them are Ray Tune (included in the Ray tool), Optuna, Hyperopt, or Scikit-Optimize. These tools can quickly explore a broader hyperparameter space compared to the traditional methods, which are still commonly used in academia. This capability leads to significant improvements in the quality of the algorithms. For this work, we chose to use Optuna (Akiba et al., 2019), following the recommendations from the authors of SB3, who also utilize it in their examples.

The hyperparameter candidates during the fine-tuning are shown in Table 4.4, and are described below:

- The NN architecture selection can impact in the performance (Islam et al., 2017). The network size parameter represents the number of neurons in each layer. Three different configurations have been considered, inspired by the ones analyzed in (Henderson et al., 2018). The number of neurons for each configuration is shown in Table 4.3.

- Learning rate denotes the update step in the Stochastic Gradient Descent (SGD) algorithm. This parameter affects the evolution of the learning for the neural network, in which a large value is more unstable, whereas a small one needs more time to converge. This parameter is sampled using a log uniform distribution, in contrast with the linear uniform distribution used in the others.

- Buffer size specifies the maximum number of experiences in the ER. This parameter fixes the dataset's maximum size and affects the probability of sampling new experiences over old ones.

- Batch size is the number of experiences used to compute the loss for a gradient step.

- Rho (ρ) is the soft update coefficient, also called Polyak update: $\theta_{targ} \leftarrow \rho\theta_{targ} + (1 - \rho)\theta$. This parameter controls the target network changing rate, smoothing the policy error.

- Gamma (γ) is the discount rate factor applied in the Bellman equation.

- Train frequency means the frequency at which the model parameters are updated over timesteps. More frequency implies faster learning but higher instability.

- Gradient steps refers to the number of times the NN parameters are updated by the SGD per training timestep (Raffin et al., 2019).

- Noise (the distribution type) and noise standard error (σ), the kind of exploration noise and its variance used in TD3, respectively.

Table 4.4: Space search for hyperparameters.

Hyperparameter	Range
Network size	{(S)mall, (M)edium, (B)ig}
Learning rate	$[10^{-5}, 1.0]$
Buffer size	$\{10^4, 10^5, 10^6\}$
Batch size	$\{16, 32, 64, 128, 256, 512, 1024, 2048\}$
Rho (ρ)	$\{0.001, 0.005, 0.01, 0.02\}$
Gamma	$\{0.9, 0.95, 0.98, 0.99, 0.995, 0.999\}$
Train frequency	$\{1, 2, 4, 8, 16, 32, 64, 128\}$
Gradient steps	$\{-1, 1, 2, 4\}$
Noise type	{None, Normal, Ornstein-Uhlenbeck}
Noise std. err.(σ)	$[0, 1]$

Table 4.5: Optimal hyperparameters.

Category	Hyperparameter	DQN	TD3
Memory	Learning start	10^2	10^2
	Buffer size	10^5	10^5
Exploration[1]	Start	1.0	-
	Stop	0.1	-
	Fraction	0.1	-
	Noise type	-	Normal
	Noise std. err. (σ)	-	0.5
Optimizer	Learning rate	10^{-5}	10^{-5}
	Batch size	512	512
	Poliak (ρ)	0.005	0.005
	Gradient steps	2	2
	Train freq.	2	2
MDP	Discount (γ)	0.95	0.95
NN	Network size	M	M

The optimization technique used is called TPE (Bergstra et al., 2011), a GMM method, with 200 iterations. Additionally, TPE is combined with pruning, which involves the early end of a training process based on the evolution when a bad result is foreseen. This technique avoids wasting resources in a poor training process, releasing resources for new training, and speeding up the fine-tuning process. The evolution is measured with the finite-horizon undiscounted return obtained from the validation dataset. The pruning logic is activated when the return obtained in the validation is lower than the median of the finished training processes in the same timestep.

In order to save more time in the hyperparameter optimization process, only two weeks were used because it was observed that adding more samples did not add any significant improvement. The chosen weeks belong to the summer season, specifically, the next two weeks after day 180 of the first year. Also, the use of two other weeks has been explored, one belonging to the winter period and another to spring, but without improvement.

[1] The DQN exploration rate decays linearly during the beginning of the training process. It starts at *start* value,

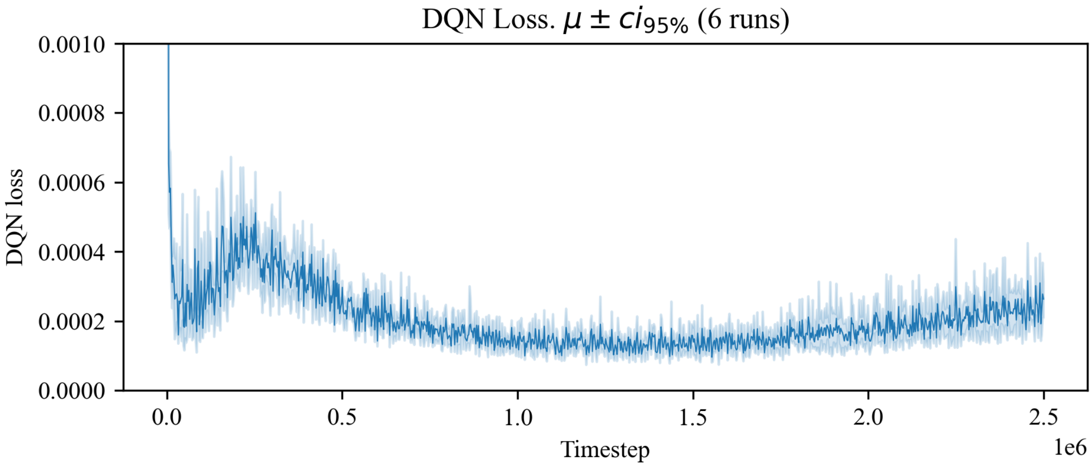


Figure 4.3: DQN MSE with 0.95 confidence interval over 6 random seeds.

The values obtained with the optimization of the hyperparameters are rounded to handle them better, but without losing the original performance. The final values are displayed in Table 4.5.

In addition, the state-space and the action-space are normalized by dividing all these values by the $S^{\mathrm{b}}_{\mathrm{max}}$ constant. The insight is to keep the same proportions in all values since all units are in kW, in contrast with other kinds of normalization processes that could break this proportion.

4.4.3 Training process analysis

The training process is based on $2.5 \cdot 10^6$ training steps, i.e., gradients applied to NN parameters, and the evaluation is performed every 200 training steps.

Figures 4.3 and 4.4 present the Mean Squared Error (MSE) loss for DQN and TD3, respectively. While both losses minimize the same metric (MSE), the TD3 critic loss is computed using continuous actions, and includes added noise as discussed in Section 4.4.2.3. This additional noise increases the total error but does not compromise stability. On the contrary, it aids TD3 in achieving better convergence (Fujimoto et al., 2018). To indicate the spread of the loss values over different runs, a shadow bounded by a confidence interval of 95% is computed using the bootstrap method. Figure 4.3 presents the usual loss pattern for a normal DQN learning process. At the very beginning of the learning process, the error sharply decreases. After that, as long as the exploration rate changes, the loss gradually increases. Just as the exploration rate hits its minimum and stabilizes, the error starts decreasing again, describing a hump and stabilizing at the end.

Figure 4.4 shows two plots with the loss of the critic and actor networks. On the one hand, the critic loss is computed similarly to the DQN loss, i.e., minimized. Therefore, it has the same pattern as the DQN loss figure. On the other hand, the actor loss function computes the Q-value estimation and is maximized. Both plots stabilize at the end of the training process, assuring that the training process is completed and fulfilling the RL goal, which, within the context of the EMS, means that active power set-points provided are optimal.

and reaches *stop* value when a *fraction* proportion of the training process is passed.

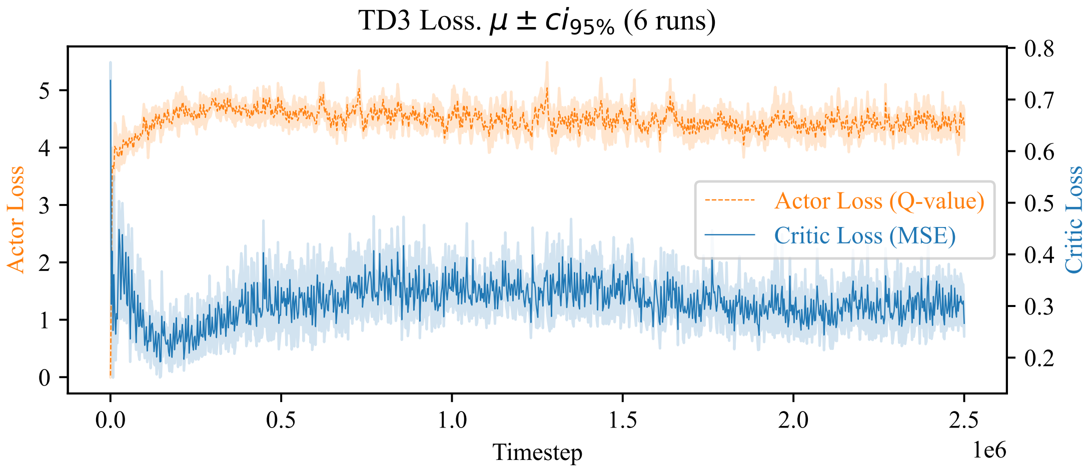


Figure 4.4: TD3 actor and critic MSE with 0.95 confidence interval over 6 random seeds. Actor loss is the upper plot with the left axis. Critic loss is the lower plot with the right axis.

4.4.3.1 Generalization over limited data

In theory, the DRL could learn by trial-and-error, starting from scratch and improving its learning during the real operation of the microgrid throughout its lifespan. However, as the amount of experience needed to learn properly could be very high, the learning phase can be accelerated by simulating the operation of the microgrid using real or synthetic data of the demand and the PV available generation. In this context, the available dataset used in the microgrid has been split into a training set, a validation set, and a test set, following the same idea as in Chapter 3. Each one of these sets contains hourly data of the demand and the available solar generation for a whole year's time scope (first, second, and third year, respectively). Both the training and validation sets are used during the learning stage. In particular, the validation dataset is used to avoid overfitting and to ensure that the model is able to generalize, using the *early stopping method* (seen in Chapter 3). Once the learning process has finished, the algorithm is faced against unexplored data (the test set) in order to assess the extent to which the model is able to generalize over unknown profiles of solar irradiation and demand.

Figure 4.5 shows the average return of the average model (of 6 seeds) operating on the validation dataset over the whole training process. Additionally, the shadow represents a confidence interval of 95% computed using the bootstrap method. Following the TD3 plot, the best model is near the timestep $2 \cdot 10^6$. The DQN performance keeps increasing but with larger variance after the timestep $2 \cdot 10^6$. This observation suggests that the best model will be near timestep $2 \cdot 10^6$ for both algorithms.

Table 4.6 shows the best result obtained from the best model over the training, validation, and test datasets, for both DQN and TD3 algorithms. Furthermore, the gap between the results and the optimal bound is displayed. The optimal bound is calculated using a MIQP from the deterministic version of the MDP described in Section 4.2, and solved using the Gurobi solver (Gurobi Optimization, LLC, 2024).

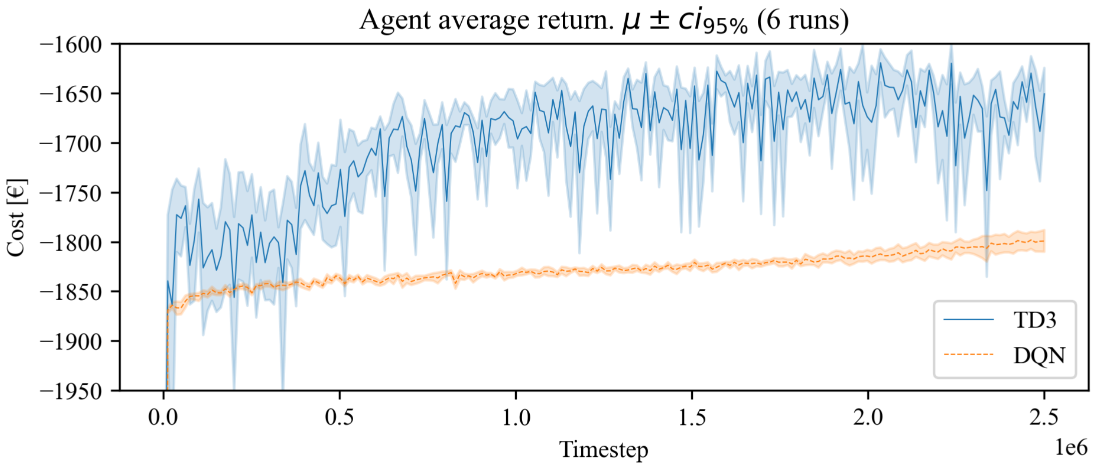


Figure 4.5: DQN and TD3 average returns with 0.95 confidence interval evaluated over the second year data with 6 different random seeds.

Table 4.6: Operation cost (and the gap with the optimal bound) using DQN and TD3.

	Obj. F. ↔ Cost [€]		
Algorithm	1^{st} year (gap%)	2^{nd} year (gap%)	3^{rd} year (gap%)
DQN	1808.87 (38.1%)	1607.82 (42.5%)	1716.26 (42.4%)
TD3	1561.68 (19.2%)	1363.80 (20.8%)	1452.21 (20.5%)

In Table 4.6, the performance of TD3 is significantly better than that of the DQN, keeping a similar gap through the training, validation, and test datasets. From the results, one concludes that TD3 improves the quality of the obtained policies with respect to the ones obtained with DQN. Besides, the DQN with a higher number of discrete decisions may improve its performance by using larger action spaces, as later studies appear to use up to 5000 actions in a discrete action space, although this can lead to computational performance penalties when compared to DDPG-based algorithms (H. Wang et al., 2023). For example, in (H. Xu et al., 2021), authors claim that using 200 discrete action spaces worsens the computational performance significantly, although it can reach the performance of the TD3 algorithm in their particular problem. Nevertheless, there is no work, as far as our knowledge of this thesis, that demonstrates DQN in general is suitable for large action spaces, nor for the EMS of the microgrid control over years.

4.4.3.2 Robustness of the training process

The DRL techniques used are susceptible to intrinsic random processes of the algorithms, such as network parameter initialization or gathering experiences (Mnih et al., 2015). In (Henderson et al., 2018), there are some recommendations when using DRL techniques. For instance, repeating the training process using different seeds and analyzing the dispersion of the average return can show the robustness of the algorithm's learning process, which, as discussed before, is a critical issue.

The training process of the TD3 algorithm, which applies the improvements proposed in (Fujimoto et al., 2018) to reduce the instability of the learning process, is analyzed and compared with the DQN. In addition, six training processes have been performed for the DQN and TD3 algorithms using six different seeds.

Figures 4.3 and 4.4 show the training process with a 95% confidence interval. In these figures, despite the applied improvements for robust learning, TD3 has a higher variance than DQN. This fact is because DDPG-based algorithms generally have a higher variance than other DRL algorithms (Henderson et al., 2018), and it became more critical to apply an advanced hyperparameter optimization technique with this kind of algorithm. In this work, the TD3 method has a better return than DQN, even comparing the lower bound of the TD3 with the upper bound of the DQN, concluding that the stability of the TD3 is enough to robustly outperform the DQN with the chosen configuration. However, different microgrid configurations may need further analysis in this matter. A study of the DRL training process with different microgrid configurations may help analyze the microgrid problem, observing the critical parameters that have a larger impact on the algorithm training process and spotting differences in the microgrid configurations from the different results obtained. Nevertheless, this analysis could help to choose to train a new model or, possibly, to transfer a trained one when applying DRL to a new microgrid.

4.4.4 Microgrid operation results

Beyond the learning process, it is interesting to analyze the operation obtained by the DQN and TD3 algorithms. The behavior of the TD3 algorithm can be seen in Figure 4.6, which shows on the horizontal axis the 24 hours of the day. Each dot corresponds to an hourly value, and it has been colored according to the month it belongs to, distinguishing between summer (a) and winter (b) during the first year of operation. For each month, the average value is marked in solid lines. In both seasons, the patterns of the Di-Gen schedule during the night do not vary since there is no solar generation, as shown in Figure 4.6 (a) and (b). During the morning hours, the controller dispatches a little bit of Di-Gen power to store some energy in the battery for the coming day. This strategy deals with the possibility of having a lack of PV power since the Di-Gen cannot satisfy the morning demand by itself, needing power support from the Li-ion battery. This strategy seems optimal because extreme strategies of leaving the battery depleted or completely full could lead to insufficient power or substantial waste of solar energy in facing scenarios with a lack or plenty of PV power on the present day. On the one hand, the Di-Gen is not frequently used during the daytime hours of summer (Figure 4.6 (a)). However, a few particular cases have high Di-Gen power values. These events occur more frequently in August and are related to days where the PV has less output than a typical summer day. On the other hand, there is often a lack of PV power in winter (Figure 4.6 (b)). To solve this, the controller turns on the Di-Gen at maximum power. Then, if some PV generation occurs, the controller reduces the Di-Gen power to avoid incurring the corresponding cost.

Figure 4.7 (a-d) shows the Di-Gen schedule, the demand and solar power (right vertical axis), and the SoC of the battery (left vertical axis) of the implemented DRL algorithms. The strategies that the controller performs in the different hours of the day are learned by both TD3 and DQN algorithms, as can be seen in Figure 4.7 (a) and (c) and Figure 4.7 (b) and (d), respectively. The difference between the algorithm configurations is that the TD3 has better resolution, dispatching the Di-Gen power. This advantage improves the intra-day operation, as for example, on day 344

Table 4.7: Microgrid CIGRE parameters.

Component	Parameter	Value	Unit
Load (total)	$D_{\max}$	40.00	[kW]
PV (total)	$P^{\text{pv}}_{\max}$	13.0	[kW]
WT	$P^{\text{wt}}_{\max}$	10.0	[kW]
MT	$P^{\text{mt}}_{\min}$	3.0	[kW]
	$P^{\text{mt}}_{\max}$	30.0	[kW]
	δ^{mt}_0	0.4710	[€]
	δ^{mt}_1	0.1080	[€/kW]
	δ^{mt}_2	0.0103	[€/kW^2]
FC	$P^{\text{fc}}_{\min}$	0.0	[kW]
	$P^{\text{fc}}_{\max}$	10.0	[kW]
	δ^{fc}_0	0.0	[€]
	δ^{fc}_1	0.2	[€/kW]
	δ^{fc}_2	0.0	[€/kW^2]
Li-ion	S^{b}_0	0.0	[kWh]
	$S^{\text{b}}_{\min}$	4.0	[kWh]
	$S^{\text{b}}_{\max}$	30.0	[kWh]
	$P^{\text{b}\leftarrow}_{\max}$	15.00	[kW]
	$P^{\text{b}\rightarrow}_{\max}$	30.00	[kW]
	η^{b}	0.9	[-]
	ζ^{b}	0.9	[-]
ENS	c^{ens}	1	[€/kWh]

(hours 8256-8280), in the last hours of the day, the battery is depleted, but the controller can use the Di-Gen to satisfy the demand. In this case, the TD3 determines with high accuracy how much Di-Gen power must be dispatched, whereas DQN cannot do it with such precision. In addition, the battery SoC at the end of the day during summer is lower in the TD3 operation due to the improvement in Di-Gen power control, and this results in a high-quality operation since there is more room for the solar surplus, reducing costs. To sum up, for the considered case study, the TD3 leads to a cost reduction when compared with the alternative of the DQN studied; therefore, the TD3 should be a more preferable algorithm.

4.4.5 Case study with the Low-Voltage Microgrid Benchmark (CIGRE)

To illustrate that the proposed method can be scaled-up, this subsection shows the obtained results with the CIGRE microgrid (Kariniotakis et al., 2005; Papathanassiou et al., 2005) depicted in Figure 4.8. All the input-data parameters are presented in Table 4.7, where some of them have been adapted to allow an off-grid operation.

A new dataset of 3 years (26280 hours) from Renewables.ninja (Staffell & Pfenninger, 2016; Stefan Pfenninger, 2016) was included in the model for the WT power generation. Thus, instead of (4.1), the observations at timestep t are the ones shown in (4.5). Notice that for the sake of simplicity, PV production has been aggregated into a single parameter given that a single-node approach has been considered, and similarly for the loads.

Table 4.8: Operation costs of the TD3 algorithm over CIGRE microgrid using three years of data.

	Cost [€]			
Period	Di-Gen	FC	ENS	Total
1st year	21288	12046	3066	36400
2nd year	19385	11570	2455	33411
3rd year	21162	12107	2690	35959

$$o_t = (P^{\text{pv}}_{t-1}, P^{\text{wt}}_{t-1}, D_{t-1}, S^{\text{b}}_t) \tag{4.5}$$

Additionally, the decision variables are increased to three continuous actions (two plus the omitted). Thus, instead of (4.2), the actions at timestep t are the ones defined in (4.6), where P^{mt}_t and P^{fc}_t are the dispatched power of the MT and FC respectively, and P^{b}_t is the generated (or consumed if negative) power of the battery—which is omitted from the action space.

$$a_t = (P^{\text{mt}}_t, P^{\text{fc}}_t) \tag{4.6}$$

The reward function and the balance equations follow the same formulation as in the residential case, but here they incorporate the additional components: the MT, replacing the Di-Gen, and the FC.

In this case, only the TD3 algorithm has been used, given that it outperforms the selected DQN for the residential case.

The TD3 algorithm applied to the CIGRE's microgrid was trained as in the previous case study described in Section 4.4.4. The set of hyperparameters used is the same as in the previous case. Although a new hyperparameter optimization process could have been carried out, the previously used hyperparameters were sufficiently effective for operating the CIGRE. Moreover, given that the goal of this chapter is not to identify the optimal configuration for the CIGRE case, but rather to propose a general strategy for determining suitable hyperparameters—as was done successfully for the residential microgrid—reusing the existing set is justified. The training process has been carried out similarly by splitting the dataset into the training/validation/test datasets, one year each, with the only difference of using $5 \cdot 10^6$ training steps, evaluating the NN model with a frequency of 100 training steps.

Figure 4.9 (a-d) shows some time windows of the obtained operation from our trained model. In particular, snapshots of the operation during days 180-187 (a) and 340-347 (b), corresponding to the training set, and days 910-917 (c) and 1070-1077 (d), corresponding to the test set. In these figures, the lines at the top part represent the input time series of the model, such as the aggregated load and generation available of non-controllable devices: PVs and the WT. On the other hand, the lines at the bottom show the power generated by the controllable devices, such as the MT, the FC, and the battery, both charge and discharge. In this operation, there is no surplus generation from the PV but from the WT during the night; therefore, the algorithm stores energy in the battery during these hours. Furthermore, the battery is discharged around the peak of demand, which is the best opportunity to save costs, avoiding the highest power peaks from the MT that are the most expensive ones.

Notice that the FC is preferred over the MT throughout the entire horizon due to its lower cost, and this is why the FC is used at full rate since the first hours of every day. By contrast, the MT behaves as a load-follower, and thus it decreases its output when the PV generation is increased. The capability of DRL to provide such clear patterns without more data than the trial-and-error experiences should be highlighted.

Additionally, Table 4.8 shows numerical results from executing the TD3 algorithm over three-year data. Note that the algorithm was trained using only the first two years, so the third year's whole operation was executed without any prediction or posterior training. Costs between years have similar values, meaning that TD3 still generalizes well over new data, i.e., no retraining is needed in similar scenarios.

Regarding the components and renewable generation utilization, Figure 4.10 shows the generation mix over the three years grouped in quarters. The MT is expected to generate more than other technologies because it has the highest power generation capacity. Nevertheless, renewable generation oscillates between 15%-30% of the load, and almost 6% is shifted through the battery, so between 21%-36% of the renewable generation is used over the total generation. The incurred ENS during the operation is not depicted in Figure 4.10 since doing so would be barely visible because it represents, at the most, 1.2% of the total generation.

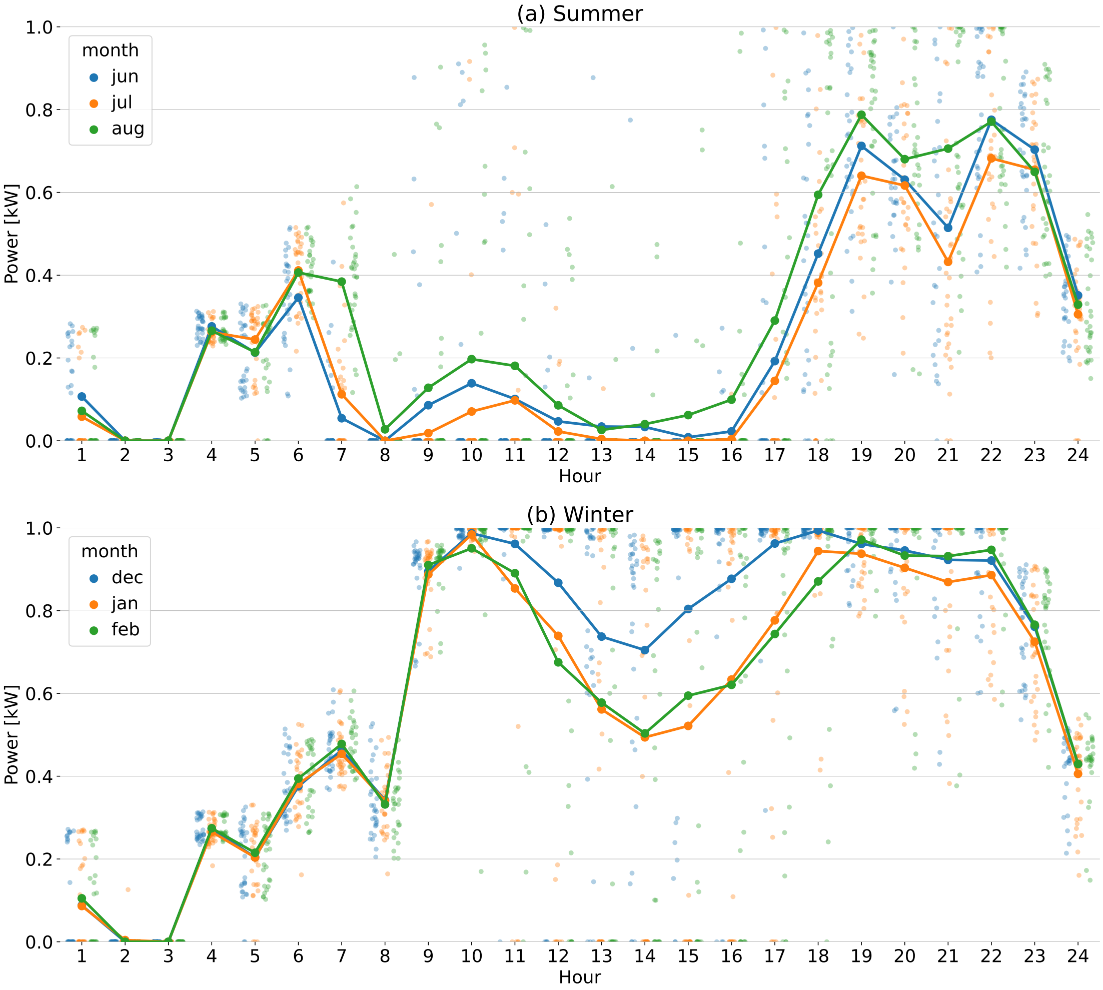


Figure 4.6: Scatter plot of Di-Gen power during a season (top: summer, down: winter), grouped in hours, with the average, using TD3.

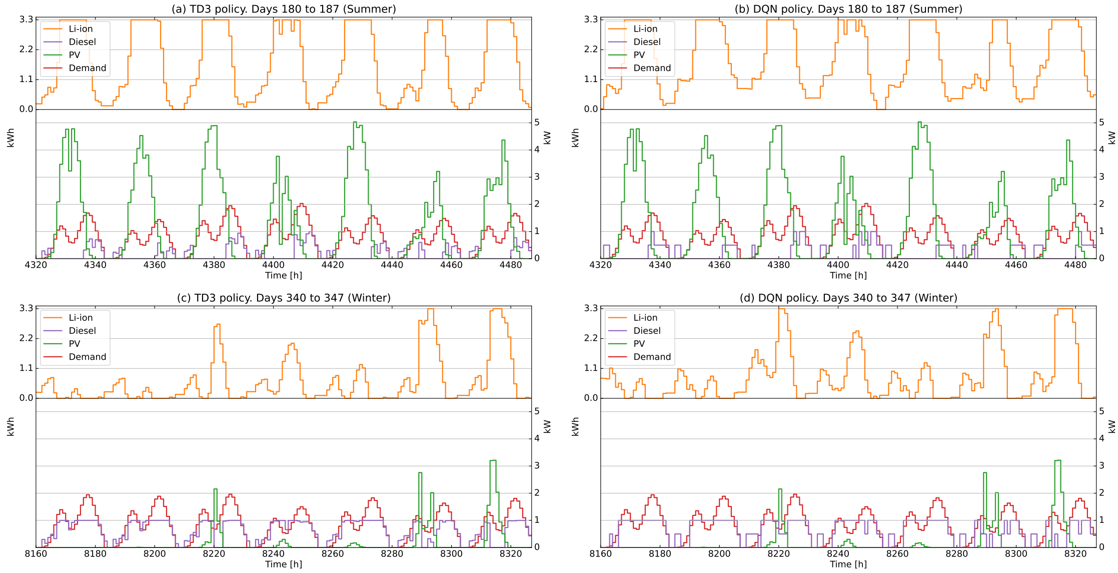


Figure 4.7: Operation of TD3 (a, c) and DQN (b, d) agents during 3 days during summer (a, b) and winter (c, d) seasons.

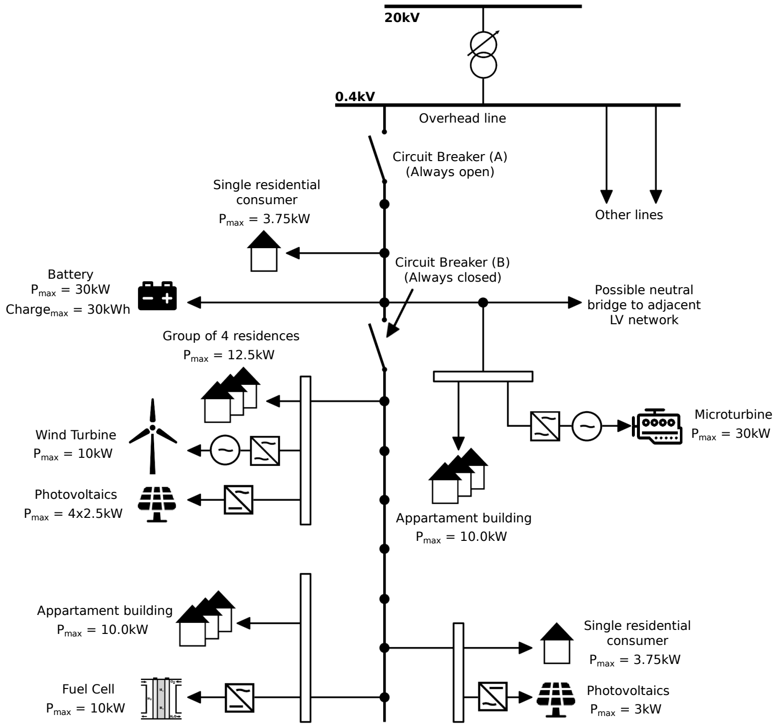
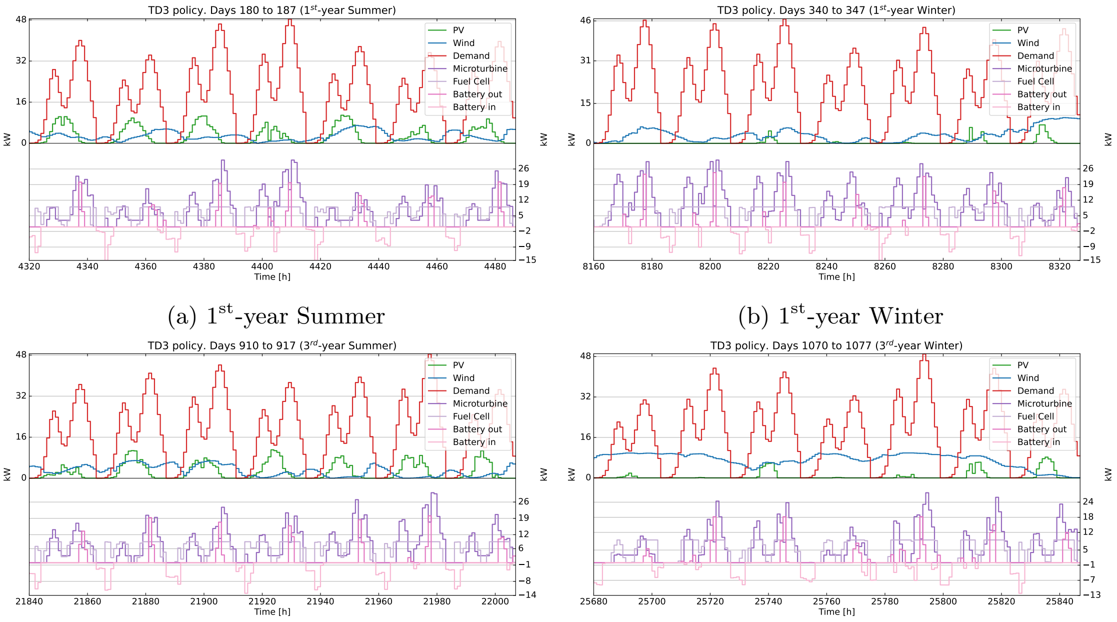


Figure 4.8: Microgrid CIGRE.



(a) 1st-year Summer

(b) 1st-year Winter

(c) 3rd-year Summer

(d) 3rd-year Winter

Figure 4.9: Operation of TD3 on CIGRE.

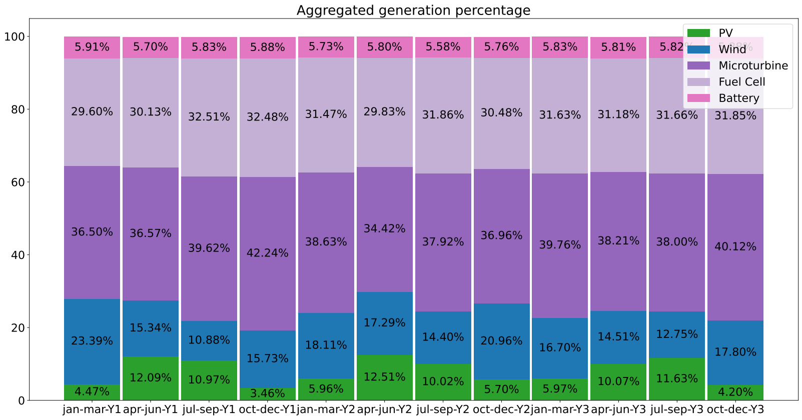


Figure 4.10: Generation mix of the TD3 operation for the CIGRE microgrid benchmark over 26280 hours (three years of hourly data).

4.5 Final remarks

Building upon previous work where applying DRL techniques to microgrid energy management, this study takes a significant step forward by employing more advanced algorithms that allow more performant solutions through more precise control. In particular, the TD3 algorithm can make decisions over a continuous range that is more precise than the previous algorithm, the DQN, which needs to define a discrete number of decisions where to choose. Two microgrid configurations, one similar to previous work and a CIGRE microgrid, are analyzed. Results obtained verify that, indeed, the TD3 outperforms the DQN due to this property. Additionally, the TD3, due to it being a DDPG-based algorithm, is generally less stable than the DQN. Hence, a complementary hyperparameter optimization strategy is carried out to mitigate that issue. In summary, this chapter shows that TD3 could be used for the EMS of a microgrid because its components operate in a continuous range. Future research directions can integrate additional features into the microgrid environment, such as demand response mechanisms and the inclusion of more realistic dynamics of the components. For instance, the non-linear behavior of the battery loss will be studied in the next chapter. Another promising avenue involves extending the problem to grid-connected microgrids, where agents can actively participate in energy markets by buying and selling electricity. By combining management strategies with market considerations, this line of investigation can lead to more sophisticated and economically relevant DRL-driven energy management solutions.

Chapter 5

EMS Considering Nonlinear Losses in Batteries through DRL

5.1 Introduction

A key component in microgrids is the ESS. As introduced in Chapter 1, the ESS is responsible for storing energy for later use, and it integrates seamlessly with renewable resources by helping to manage their inherent intermittency. At first glance, the operational strategy for an ESS may seem straightforward: charge when there is surplus renewable energy and discharge when there is a deficit. However, this operational strategy is only truly optimal when all energy generation components are renewable and incur no cost when operated. Once a fuel-based generator—such as a diesel generator with controllable dispatch and associated costs—is introduced, the optimal use of the battery becomes considerably more complex. Previous studies show that in scenarios where the Di-Gen alone cannot meet the full demand (e.g., the Di-Gen max power is 1 kW whereas the peak demand is 2.1 kW), it is sometimes necessary to rely on stored energy in the battery that is generated, at least in part, by the diesel source.

Building on this insight, the battery opens up a broader spectrum of operational strategies, including determining exactly how much energy needs to be stored to prepare for future scenarios and whether that energy should be sourced from a mix of technologies rather than from renewables alone. In this context, paying special attention to the battery is crucial to fully leveraging its capabilities and maximizing its benefits.

In general, battery modeling in the EMS literature considers constant efficiencies. Nevertheless, some studies explore the complexities associated with modeling nonlinear battery dynamics, as detailed in Section 2.2, whether they consider the inverter directly connected between the ESS and the rest of the network, the internal dynamics of the battery, or even both (R.-K. Kim et al., 2020; Mbuwir et al., 2017, 2019; Nguyen & Crow, 2016; Shuai et al., 2019). By leveraging DRL techniques, however, a more realistic—potentially nonlinear—representation of battery behavior can be incorporated without imposing significant additional computational costs. Such an advanced modeling approach promises a more accurate solution and can considerably improve the feasibility of real-world deployments. Moreover, contemporary AI techniques, such as DRL, are well-suited to address the challenges posed by these nonlinearities.

From the modeling point of view, EMSs for microgrids face a significant challenge when it comes to accurately representing battery losses, among others, as they are nonlinear in nature. Linear modeling for optimization struggles to adequately incorporate these losses, where a common approach is to assume fixed efficiencies for the charge and discharge processes. As a result, there may be significant deviations between the expected and actual behavior in microgrids, thereby giving suboptimal results. This underscores the need for more sophisticated modeling techniques, although increasing the computational burden of the solution would be undesirable.

As mentioned in Section 1.2, the microgrid control hierarchy typically consists of three levels, each addressing different operational aspects, (Gao et al., 2019). The EMS tertiary control optimizes the economic and efficient operation of the microgrid in a steady-state framework, considering only active power set-points (Střelec & Berka, 2013). Below it, the secondary and primary control levels operate to deliver the real-time active power, which may deviate from the prescribed set-points. These lower-level controls can be modeled in a simplified manner, which is sufficient to enhance the quality of the data used for DRL training and, in turn, improve the overall effectiveness of the learning process. This, in turn, may improve data quality for DRL, leading to more effective learning and decision-making.

This chapter introduces a novel approach that leverages DRL to model and optimize the EMS task in microgrids, explicitly accounting for the nonlinear dynamics associated with battery losses. The model employs a more precise—thought still approximated—nonlinear representation of real battery behavior, which may improve the cost efficiency of energy management systems in real-world scenarios. Building upon previous works—namely, Chapters 3 and 4—this study adopts the TD3 algorithm due to its stability and its ability to operate effectively in continuous action spaces (Fujimoto et al., 2018).

The main contributions of this chapter can be summarized as follows:

- First of all, a microgrid model that includes the nonlinear behavior of Li-ion batteries is proposed for the training of the TD3 algorithm. This model extends the POMDP of the microgrid, developed in Chapter 3 and 4, with the nonlinear equations of the battery losses.

- Secondly, a methodology is introduced to assess the performance improvements of the proposed algorithm compared to EMSs models with linear battery losses. Additionally, this chapter incorporates a novel approach to modeling the microgrid control system, enhancing the robustness of the proposed method, i.e., reducing the ENS.

- Thirdly, this method is extended to manage a microgrid of realistic size, ensuring that the proposed method is valid for both a residential microgrid and a low-voltage distribution network extended into a microgrid.

This chapter is organized as follows: Details about issues regarding battery modeling are summarized in Section 5.2. Section 5.3 describes the MDP model used for the EMS problem. Section 5.4 describes the control strategy underneath the EMS as part of the MDP model and its implications in the optimization process. The case study and results can be found in Section 5.5. Finally, conclusions are summarized in Section 5.7.

5.2 Battery operation model

Batteries play a pivotal role in electrical energy storage due to their ability to store and release energy in a highly controllable manner. The intermittent and fluctuating nature of PVs and WTs systems requires the deployment of efficient ESSs, and in this context, batteries, particularly Li-ion variants, have emerged as a promising solution to address these challenges. A battery is a highly complex system, and its modeling should be tailored to the specific application for which it is intended in order to ensure that the chosen model adequately represents battery behavior and performance characteristics relevant to the desired use case, balancing accuracy and computational efficiency. In Fotouhi et al. (2016), the authors present a review of the literature on different approaches to model Li-ion batteries. Broadly speaking, one can distinguish between Electrochemical Models (EMs) and electrical Equivalent Circuit Models (ECMs) (García-González & Guerrero, 2024), For an EMS in a microgrid, it is sufficient to use an ECM to represent the voltage, the SoC, and the power capabilities of the battery rather than using a detailed EM. Hence, an ECM has been used in the work reported here.

5.2.1 Equivalent Circuit Model

The Shepherd model (Shepherd, 1965) describes the output voltage of a battery in the discharge process as:

$$V_{\text{batt}}(t) = E_o - Ri(t) - K\frac{Q(t)}{Q(t) - it(t)}i(t) + Ae^{-B\frac{it(t)}{Q(t)}} \tag{5.1}$$

where V is the voltage of the battery, E_o is the constant potential of the cell, R is the internal resistance, i is the current withdrawn from the battery, K is the polarization coefficient, Q is the amount of available charge, it is the total electrical charge extracted from the battery at time t measured from the moment the discharge started ($it = \int idt$), and A and B are constants to model the initial exponential drop expressed in the last term of (5.1). The values of E_o, K, Q, R, A, and B must be determined empirically.

In Tremblay and Dessaint (2009), the authors start from an equation similar to (5.1) in which they introduce some improvements that consider not only a constant discharge current but also the case of a variable charging or discharging current. The authors of Nguyen and Crow (2016) take the equations from Tremblay and Dessaint (2009) and derive simplified expressions of the voltage drop in the battery that is used to estimate the discharge/charge losses. Shuai et al. (2019) refine previous expressions to obtain the following nonlinear equations of battery losses that will be used in this chapter:

$$P^{\text{b}\leftarrow}_{loss} = \frac{10^3(R_{in} + \frac{K}{1.1 - SoC})}{V_r^2}(P^{\text{b}\leftarrow})^2 + \frac{10^3 S^{\text{b}}_{\max}K(1 - SoC)}{SoC \cdot V_r^2}P^{\text{b}\leftarrow} \tag{5.2}$$

$$P^{\text{b}\rightarrow}_{loss} = \frac{10^3(R_{in} + \frac{K}{SoC})}{V_r^2}(P^{\text{b}\rightarrow})^2 + (\frac{10^3 S^{\text{b}}_{\max}K(1 - SoC)}{SoC \cdot V_r^2})P^{\text{b}\rightarrow} \tag{5.3}$$

where

- $P^{\text{b}\leftarrow}$: power consumed by the battery in [kW]

- $P_{loss}^{\text{b}\leftarrow}$: power losses while charging [kW]
- $P^{\text{b}\rightarrow}$: power generated by the battery [kW]
- $P_{loss}^{\text{b}\rightarrow}$: power losses while discharging [kW]
- R_{in}: internal resistance [Ohm]
- SoC^{b}: state of charge expressed in terms of the estimated stored energy [%]
- V_r: nominal voltage rate of the battery [V]
- $S_{\max}^{\text{b}}$: nominal capacity rate of the battery [kWh]

As explained in García-González (2023), the second terms of expressions (5.2) and (5.3) could be dismissed in order to obtain a more accurate model of the losses. This would result in the following expressions of the losses:

$$P_{loss}^{\text{b}\leftarrow} = 10^3(R_{in} + \frac{K}{1.1 - SoC^{\text{b}}})(\frac{P_t^{\text{b}\leftarrow}}{V_r})^2 \tag{5.4}$$

$$P_{loss}^{\text{b}\rightarrow} = 10^3(R_{in} + \frac{K}{SoC^{\text{b}}})(\frac{P_t^{\text{b}\rightarrow}}{V_r})^2 \tag{5.5}$$

Using the expressions of the losses, the energy stored in the battery S^{b} changes according to (5.6) and (5.7) for the charge and discharge, respectively.

$$\frac{dS^{\text{b}}}{dt} = P^{\text{b}\leftarrow} - P_{loss}^{\text{b}\leftarrow} \tag{5.6}$$

$$\frac{dS^{\text{b}}}{dt} = -P^{\text{b}\rightarrow} - P_{loss}^{\text{b}\rightarrow} \tag{5.7}$$

Discrete-time versions of (5.6) and (5.7) for the charge and discharge process, respectively, can be written as:

$$S_{t+1}^{\text{b}} - S_t^{\text{b}} = [P_t^{\text{b}\leftarrow} - P_{loss}^{\text{b}\leftarrow}(P_t^{\text{b}\leftarrow}, S_t^{\text{b}})]\Delta t \tag{5.8}$$

$$S_{t+1}^{\text{b}} - S_t^{\text{b}} = [-P_t^{\text{b}\rightarrow} - P_{loss}^{\text{b}\rightarrow}(P_t^{\text{b}\rightarrow}, S_t^{\text{b}})]\Delta t \tag{5.9}$$

5.3 EMS of a microgrid using an MDP

This section presents only the new contributions to the model previously defined in Chapters 3 and 4, specifically those related to the nonlinear losses of the battery.

5.3.1 State definition

Refer to Section 4.3.1.

5.3.2 Action definition

Unlike in the previous chapters, the action notation here represents the set-point as $\tilde{P}_t^i$ for a component $i \in \mathrm{G} \cup \mathrm{B}$ at time t. This notation distinguishes between the set-points determined by the EMS agent and the actual power flows after they are adjusted by the control system to maintain microgrid balance. Apart from this modification, the remaining definitions follow those in Sections 3.3.2 and 4.3.2, in particular, those related to the continuous action space.

5.3.3 Reward definition

The reward function remains the same as in Section 3.3.3.

5.3.4 Transition definition

In this chapter, the new dynamics of the system are defined. In particular, the dynamics of the Li-ion battery and the more advanced control system.

The energy inside battery b satisfies the energy balance equation (5.10), where $P_t^{\mathrm{b}\leftarrow}$ and $P_t^{\mathrm{b}\rightarrow}$ are the charge and discharge battery power, and η^{b} and ζ^{b} are the corresponding nonlinear efficiency values for the charge and discharge processes, respectively.

$$S_{t+1}^b = S_t^b + \left[P_t^{\mathrm{b}\leftarrow} \cdot \eta^b(P_t^{\mathrm{b}\leftarrow}, S_t^b) - P_t^{\mathrm{b}\rightarrow} \frac{1}{\zeta^b(P_t^{\mathrm{b}\rightarrow}, S_t^b)} \right] \Delta t \tag{5.10}$$

In particular, these efficiencies are defined in (5.11) and (5.12) for all $\mathrm{b} \in \mathrm{B}$ (only the Li-ion in this case), using equations (5.2) and (5.3).

$$\eta^{\mathrm{b}}(P^{\mathrm{b}\leftarrow}, S^{\mathrm{b}}) = \frac{P^{\mathrm{b}\leftarrow} - P_{loss}^{\mathrm{b}\leftarrow}(P^{\mathrm{b}\leftarrow}, S^{\mathrm{b}})}{P^{\mathrm{b}\leftarrow}} \tag{5.11}$$

$$\zeta^{\mathrm{b}}(P^{\mathrm{b}\rightarrow}, S^{\mathrm{b}}) = \frac{P^{\mathrm{b}\rightarrow}}{P^{\mathrm{b}\rightarrow} + P_{loss}^{\mathrm{b}\rightarrow}(P^{\mathrm{b}\rightarrow}, S^{\mathrm{b}})} \tag{5.12}$$

In batteries, the charge and discharge processes cannot happen at the same time, which can be formally modeled with the constraint defined in (5.13), which represents that these two variables are orthogonal with respect to each other, i.e., at least one variable has to be 0. Hence, P_t^{b} is defined from $P_t^{\mathrm{b}\leftarrow}$ and $P_t^{\mathrm{b}\rightarrow}$ by the equation in (5.14).

$$P_t^{\mathrm{b}\leftarrow} \perp P_t^{\mathrm{b}\rightarrow} \quad \forall \mathrm{b} \in \mathrm{B}, t \in T \tag{5.13}$$

$$P_t^{\mathrm{b}} = P_t^{\mathrm{b}\rightarrow} - P_t^{\mathrm{b}\leftarrow} \quad \forall \mathrm{b} \in B, t \in T \tag{5.14}$$

5.3.5 DRL using TD3

The results obtained in the previous chapter using TD3 serve as the motivation for applying the same base algorithm in this chapter. Since the focus is on modifying how the environment behaves—specifically, how the microgrid battery accounts for losses and how set-point actions influence real-time operation—there is no need to alter the algorithm itself. These modifications do not affect the TD3 implementation due to the well-defined interface between the RL framework and the agent-environment interaction. Furthermore, since DRL can inherently capture nonlinear relationships, the same DRL algorithm remains sufficient to adapt to these nonlinearities.

5.4 Including the control system in the model

Practical approaches in the field for implementing a controller include, beyond the EMS, an underlying real-time control system that will eventually guarantee the balance between generation and loads. Due to the battery operation flexibility, this is traditionally done by letting the control system directly manage the battery and excluding the battery action as a decision for the RL-based EMS (François-Lavet, Taralla, et al., 2016), i.e., the battery is not given any reference by the EMS. This approach can be generalized by selecting any other component of the microgrid, instead of the battery, to take care of the balance in real-time. To avoid complications when the balancing device saturates, a better option is to use more than one component to take care of the balance.

5.4.1 Modeling the control system by using a priority list

Any EMS set-point may be ignored for the benefit of safety. When a component saturates, i.e., it cannot keep the balance without violating its physical constraints, another component must replace the role of the saturated one. For this purpose, a priority list of components is defined to reinforce balance stability as much as possible. For example, let us consider the isolated microgrid of Figure 4.2 using the components that are connected. When the battery is full and the diesel unit has reached its power limit, if the demand is suddenly reduced, the microgrid will have an energy surplus that the battery cannot store. In this situation, the classical approach would spill the energy surplus. On the contrary, if there were a priority list of components, the diesel unit could be chosen to take care of the balance in this example, i.e., the control system would decrease the diesel power, thereby minimizing the operation costs. In general, this methodology tries to simulate the real-time interactions between the EMS third-level control with the second and first level control, but still in a simplified manner. There are many open questions and possible extensions about this methodology, such as how this methodology should be implemented with other constraints like start-ups, shut-downs, ramps, etc. However, a detailed model of this lower-level control is beyond the scope of this work.

This behavior has been added in the RL environment-agent loop of the proposed POMDP. Although this approach protects the EMS and reduces the operational costs, it may also damage the trial-and-error learning process of the DRL agent since the error perceived after making a bad decision is reduced. Technically, the agent will perceive a more sparse reward, which does not benefit the learning process (Rengarajan et al., 2022). The strategy used in this chapter during the experiments is to combine both approaches, i.e., to apply the classical approach during the agent's training and validate its performance with the second approach using the priority list. This combination takes the best of both worlds: it avoids slowing down the learning process while correctly assessing the agent's performance on the more realistic microgrid operation.

A side note: the priority list needs a chosen order of the elements. For reaching optimality, the order is not important. Let $f^{u}_{\text{priority}} : \mathcal{A} \to \mathcal{A}$ be a function that follows the methodology described above using the combination u from the set of all possible combinations for the priority list order U_{priority}. For all combination $u \in U_{\text{priority}}$, and for all action $a \in \mathcal{A}$, the methodology satisfies that $r(f^u_{\text{priority}}(a)) \geq r(a)$, being r the reward function, and for all $s \in \mathcal{S}$, it also satisfies that $Q(s, f^u_{\text{priority}}(a)) \geq Q(s, a)$, as it is deduced from the methodology explanation from above. Moreover, for all $s \in \mathcal{S}$, the methodology satisfies that $\pi^*(s) = f^u_{\text{priority}}(\pi^*(s))$; therefore, the methodology does not comprise the optimality of the algorithm for any combination u. This means that the combination is not important when reaching the optimal policy, but different combinations can lead to different learning processes.

5.5 Performance comparison between linear vs. nonlinear Li-Ion battery models

This section highlights the advantages of using DRL methods for an EMS when considering Li-ion batteries. These methods employ NNs at their core, which proficiently approximate nonlinear dynamics, as the universal approximation theorem states. This ensures an enhanced adaptation to the complex behavior of the batteries, optimizing the EMS performance and reliability in real-world applications. The simulation experiments consider the same two isolated microgrids used in Chapter 4: a residential and the CIGRE microgrids.

The microgrid configuration used in this chapter remains unchanged with respect to the previous, except for the battery, which requires additional parameters to account for nonlinear equations. The parameters for all components in both case studies, including those related to the nonlinear battery model, are presented in Table 5.1.

The additional battery parameters include the number of cells, the internal resistance constant (R_{in}), the nominal voltage (V_r), and the polarization constant (K). The number of cells is essential for scaling the battery, as other battery parameters depend on the maximum capacity ($S_{\max}$). Adjusting $S_{\max}$ requires modifying multiple parameters, making scaling more complex. To address this, it is more practical to define a fixed number of cells and scale only the input and output power, assuming an even distribution among them. This approach simplifies parameter adjustments while maintaining model consistency. Further details on these parameters are provided in Section 5.2.1.

The strategy to deal with limited data is the same as in Chapter 4, but the methodology to evaluate the methods is new, since now the comparison is between two different microgrid models.

The following subsections analyze the results obtained from applying the TD3 method to two different battery models: a linear and a nonlinear model. First of all, the methodology to compare the approach with both battery models is detailed in Section 5.5.1. Secondly, the operation costs comparing both battery-loss models are discussed in Section 5.5.2 for each case study. Finally, in Section 5.5.3, the battery efficiency and energy losses are analyzed and compared between models.

5.5.1 Comparison methodology

The comparison made in this chapter involves two POMDP approaches for the microgrid system, each characterized by a different energy loss model of the battery, i.e., using equations (5.11) and (5.12) for the efficiencies in the nonlinear model, and constant values for the linear one. Consequently, these variations lead to differential behaviors in the microgrid system. In this sense, the TD3 trained using the linear model of the battery (TD3-L) and the same but using the nonlinear model of the battery (TD3-NL) are both evaluated in the microgrid using the nonlinear

Table 5.1: Component specifications of the microgrid.

Component	Parameter	Resi.	CIGRE	Unit
Load (total)	D_{max}	2.1	40.0	[kW]
PV (total)	$P^{\text{pv}}_{\text{max}}$	6.0	13.0	[kW]
WT	$P^{\text{pv}}_{\text{max}}$	-	10.0	[kW]
Di-Gen/MT	$P^{\text{d/mt}}_{\text{max}}$	1.0	30.0	[kW]
	$P^{\text{d/mt}}_{\text{min}}$	0.1	3.0	[kW]
	$\delta^{\text{d/mt}}_{0}$	0.0157	0.4710	[€]
	$\delta^{\text{d/mt}}_{1}$	0.1080	0.1080	[€/kW]
	$\delta^{\text{d/mt}}_{2}$	0.3100	0.0103	[€/kW^2]
FC	$P^{\text{fc}}_{\text{max}}$	-	10.0	[kW]
	$P^{\text{fc}}_{\text{min}}$	-	0.0	[kW]
	δ^{fc}_{0}	-	0.0	[€]
	δ^{fc}_{1}	-	0.2	[€/kW]
	δ^{fc}_{2}	-	0.0	[€/kW^2]
Li-ion	S_{max}	3.3	33.0	[kWESSh]
	S_{min}	0.4	4.0	[kWESSh]
	S_0	0.4	4.0	[kWESSh]
	$P^{\text{b}\leftarrow}_{\text{max}}$	3.0	30.0	[kW]
	$P^{\text{b}\leftarrow}_{\text{min}}$	0.0	0.0	[kW]
	$P^{\text{b}\rightarrow}_{\text{max}}$	3.3	33.0	[kW]
	$P^{\text{b}\rightarrow}_{\text{min}}$	0.0	0.0	[kW]
	η-linear	0.9	0.9	[kWESSh/kWh]
	ζ-linear	0.9	0.9	[kWh/kWESSh]
	number of cells	1	10	[p.u.]
Li-ion individual cell	internal resistance cons. (R_{in})	0.01	0.01	[Ω]
(nonlinear)	nominal voltage (V_r)	51.8	51.8	[V]
	polarization constant (K)	0.06	0.06	[V/Ah] [Ω]
ENS	c^{ens}	1	10	[€/kWh]

model of the battery, allowing a fair and reliable comparison since the same microgrid is used and that microgrid model is the closer to a real one. Figure 5.1 depicts the comparison methodology.

Regarding the control system detailed in Section 5.4, the priority list to select the component responsible for taking care of the balance should be predefined for the evaluation stage. In the case of the residential microgrid, the battery comes first, and the diesel group second, while for the CIGRE microgrid case, the battery comes first, the microturbine second, and the fuel cell comes third. This order has been chosen with the aim of prioritizing the most flexible—the one with the greatest dispatchable power range—, the cheapest, and the quickest in response terms.

5.5.2 Operating costs

5.5.2.1 Residential microgrid

Table 5.2 shows the total operational cost of operating the microgrid over three years. Additionally, the table includes the results from using a MIQP model, solved with the Gurobi solver (Gurobi

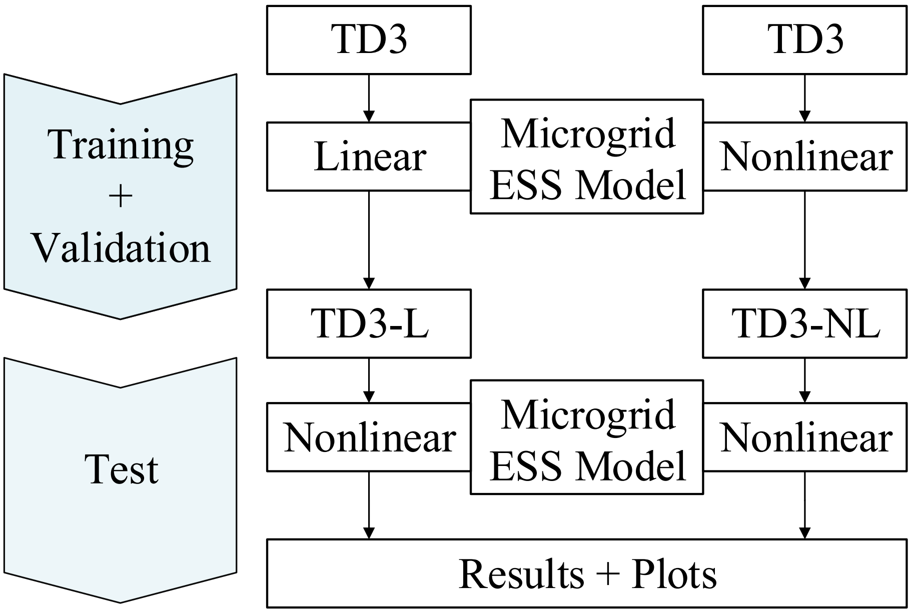


Figure 5.1: Train-eval methodology.

Table 5.2: Yearly cost of each algorithm (residential).

		Obj. F. ↔ Cost [€]		
Algorithm	Training	1st year	2nd year	3rd year
Upper Bound	-	1569.67	1367.50	1427.66
TD3	Nonlinear	1628.42	1431.44	1518.12
TD3	Linear	1665.46	1442.65	1545.39

Optimization, LLC, 2024) in a fully informed deterministic scenario. This last method is combined with a rolling horizon of 24 hours (Le & Day, 1982), (Sethi & Sorger, 1991), and serves as an approximated upper bound to elucidate the goodness of the TD3. Notice that the MIQP cannot handle the nonlinear equation (5.10), and the linear equation in (5.15) is used instead, where η and ζ are constants, which both take the value of 0.9.

$$S^{\mathrm{b}}_{t+1} = S^{\mathrm{b}}_{t} + \eta^{\mathrm{b}} P^{\mathrm{b}\leftarrow} - P^{\mathrm{b}\rightarrow}_{t} \frac{1}{\zeta^{\mathrm{b}}} \; \forall \mathrm{b} \in \mathrm{B}, t \in T \tag{5.15}$$

The results in Table 5.2 show that the TD3-NL outperforms the TD3-L when both are tested with a nonlinear battery model. Compared with the TD3-L, the TD3-NL costs are reduced by 37.04€, 11.21€ and 27.27€ in each consecutive year. In addition, both TD3 configurations perform quite efficiently in the third year (the test dataset) when compared with the first and second years (the training and evaluation datasets). For instance, as the percentage RE (see formula in (5.16), that is the same used in Table 3.3 to compute the last column) with respect to the upper bound (the reference), TD3-L has an RE of 8.25% in the third year versus 6.10% and 5.50% in the first

Table 5.3: Yearly cost of each algorithm (CIGRE).

		Obj. F. ↔ Cost [€]		
Algorithm	Training	1^{st} year	2^{nd} year	3^{rd} year
TD3	Linear	34044.10	31293.51	33793.13
TD3	Nonlinear	33279.29	30633.29	33158.29

and second year respectively. Similarly, TD3-NL has 6.34% in the third year versus the 3.74% and 4.68% in the first two. These results imply savings of 2.31% in the first year, 0.82% in the second, and 1.91% in the third when using nonlinear battery dynamics in the model.

$$\mathrm{RE} = \frac{|\mathrm{X} - \mathrm{Reference}|}{\mathrm{Reference}} \cdot 100\% \tag{5.16}$$

5.5.2.2 CIGRE microgrid

The CIGRE microgrid's results mirror the residential case on a proportional scale. Table 5.3 shows savings of 764.81€, 660.22€ and 634.84€ in each consecutive year when using the TD3-NL. These savings correspond to the 2.25%, 2.11%, and 1.88% with respect to the TD3-L costs using the formula in (5.16).

The TD3-NL performs better when operating the larger microgrid, indicating that the algorithm can handle different-sized problems seamlessly. An extended analysis can be found in 5.7.

5.5.3 Battery efficiency and energy losses

Beyond the total costs, this chapter analyzes additional metrics related to battery management, such as battery efficiency during charge and discharge processes, and the energy loss in the battery after each charge/discharge operation. Efficiency and energy loss metrics are strongly related to energy utilization (by equations (5.6) and (5.7)) and can be used to analyze the battery management performance. Efficiency helps visualize the operation patterns, whereas energy loss helps to quantify these patterns.

5.5.3.1 Residential microgrid

Regarding the residential case study, Figures 5.2 and 5.3 show histograms of the battery efficiency during the discharge and charge processes, where each bar of the histogram corresponds to the number of hours the battery was operated with a particular efficiency (see formulas (5.11) and (5.12)). These histograms are also combined with the Kernel Density Estimation (KDE) curve and its average value (the vertical dashed line), and include both TD3-NL and TD3-L results.

For the discharge process, the TD3-NL average efficiency is 0.9055, whereas that of the TD3-L is 0.8508 (an improvement of 6.4%); meanwhile, during the charge process, the TD3-NL achieves an average efficiency of 0.8092 whereas that of the TD3-L is 0.6822 (an improvement of 18.6%). These experiments indicate that the model approach not only reduces operational costs but also increases battery efficiency as a consequence.

Figure 5.4 shows a 3D scatter plot with the discharge efficiencies for the TD3-L(a) and TD3-NL(b). Similarly, Figure 5.5 shows the same for the charge process. In both figures, the axes in

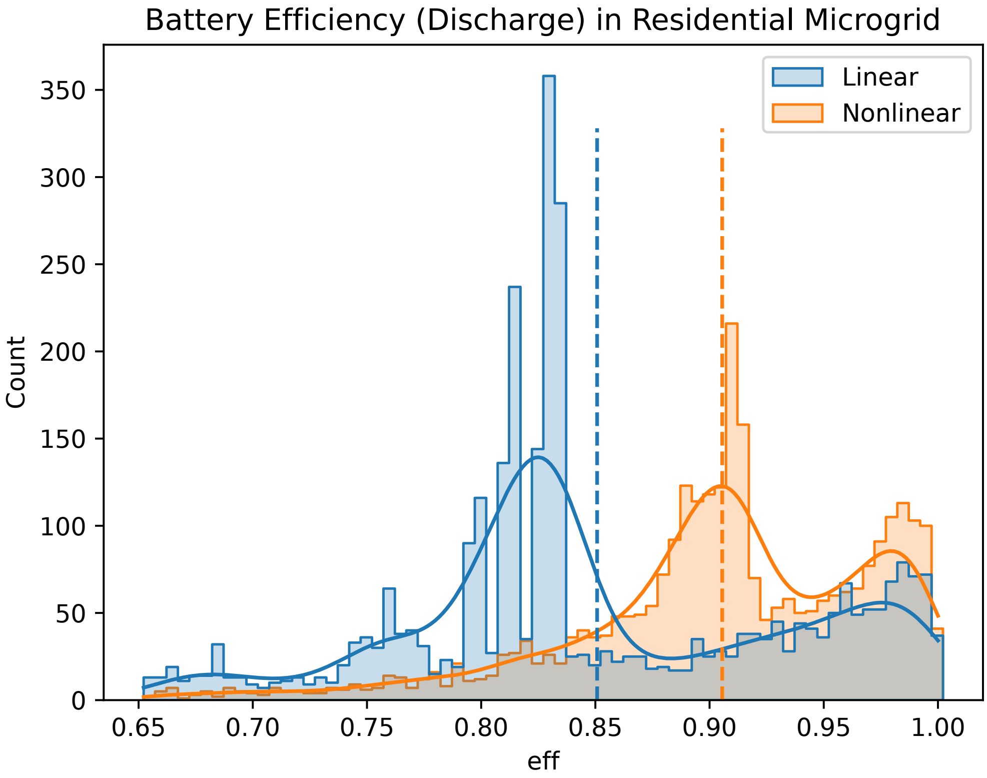


Figure 5.2: Discharge efficiency comparison of TD3 trained using the linear and nonlinear battery model in a residential microgrid for the 3rd year.

the base represent the power and the SoC, and the vertical axis the efficiency, which also uses a color gradient to help visualize them (lighter means higher efficiency).

In Figure 5.4(a), dots are sparse in the center of the area and more populated in the edges, whereas in (b) they are clustered in the high-efficiency area and in the low-power situations. Figure 5.5 shows similar differences but more prominently since the charge process can reach very low efficiencies. These figures make visible the large change in the operation patterns of the battery.

Regarding battery energy losses, Table 5.4 shows that the consideration of the nonlinear battery model can drive the EMS to reduce losses substantially. In particular, the TD3-L energy loss percentage over the total energy stored in the battery is 34.59%, 34.30%, 34.87% in each one of the three years, whereas the TD3-NL reduces it down to 24.17%, 22.96%, and 23.36%.

5.5.3.2 CIGRE microgrid

Figures 5.6 and 5.7 show the histograms corresponding to the CIGRE case study. KDE and the average value are depicted as in Figures 5.2 and 5.3. During discharge, TD3-L achieves an average efficiency value of 0.9170, whereas TD3-NL achieves an average of 0.9421 (i.e., a 2.7% improvement). During charge, TD3-L achieves an average of 0.8246, whereas TD3-NL achieves

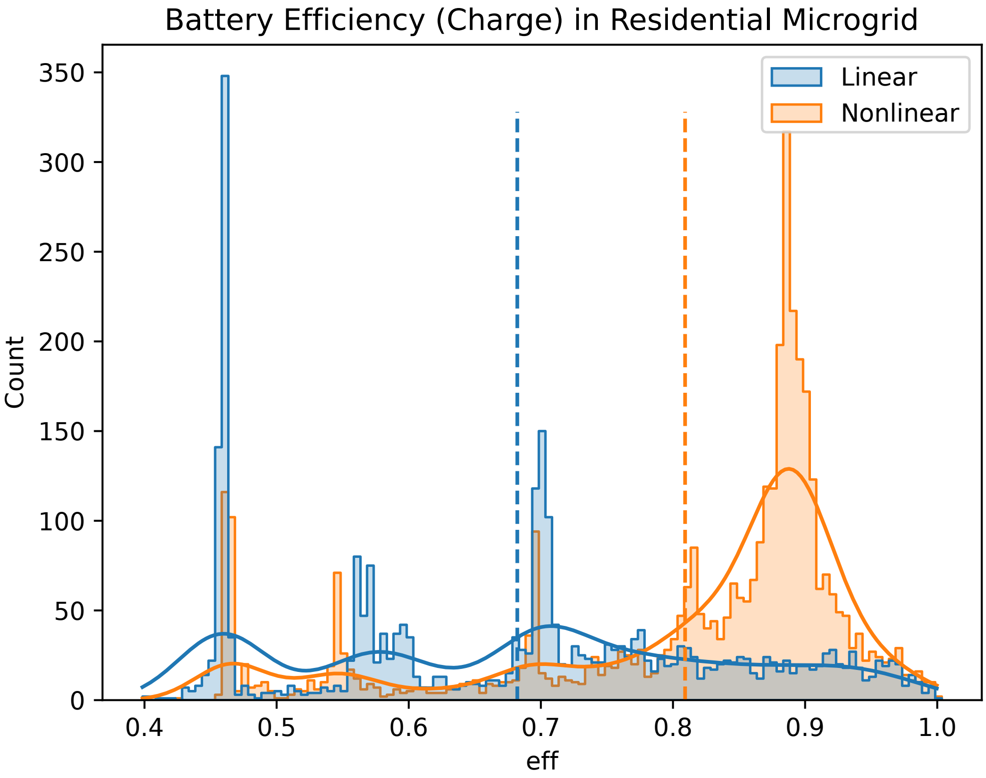
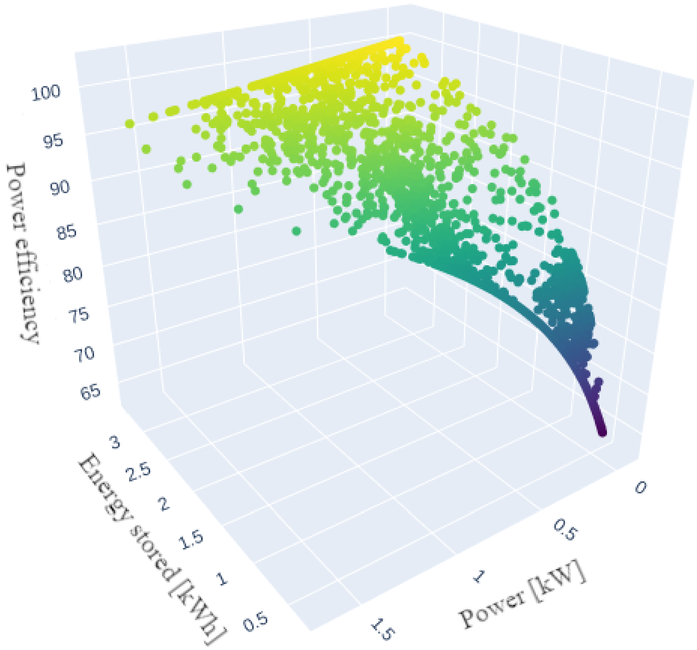


Figure 5.3: Charge efficiency comparison of TD3 trained using the linear and nonlinear battery model in a residential microgrid for the 3rd year.

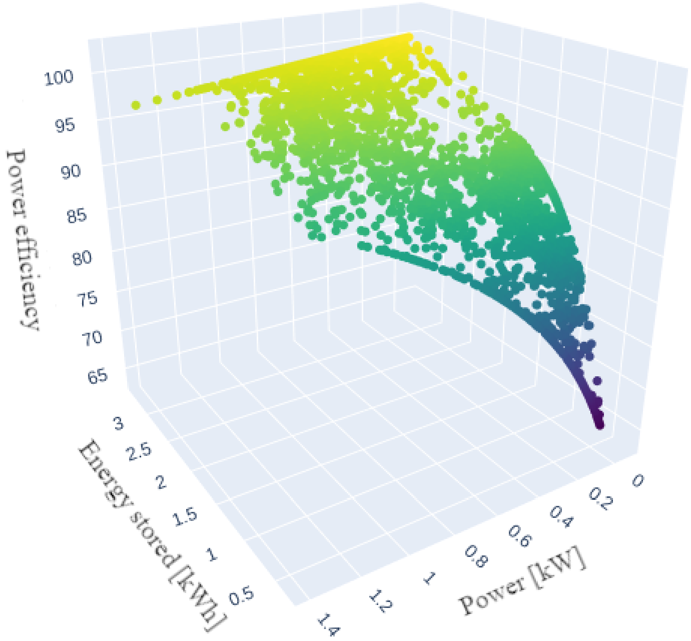
an average of 0.9054 (i.e., a 9.8% improvement).

Regarding the energy losses of the battery in the CIGRE case, TD3-L obtains 26.67%, 26.46%, and 26.65% of energy losses over the total energy stored in the battery, for each one of the three years, whereas TD3-NL reduces it to 15.72%, 15.57%, and 15.60%. The total energy losses by year are displayed in Table 5.5.

As a final observation, the results from the CIGRE case study are proportionally similar to the residential case study, but with an absolutely huge difference given the scale between both. This indicates that the algorithm can deal very similarly with different microgrid sizes.

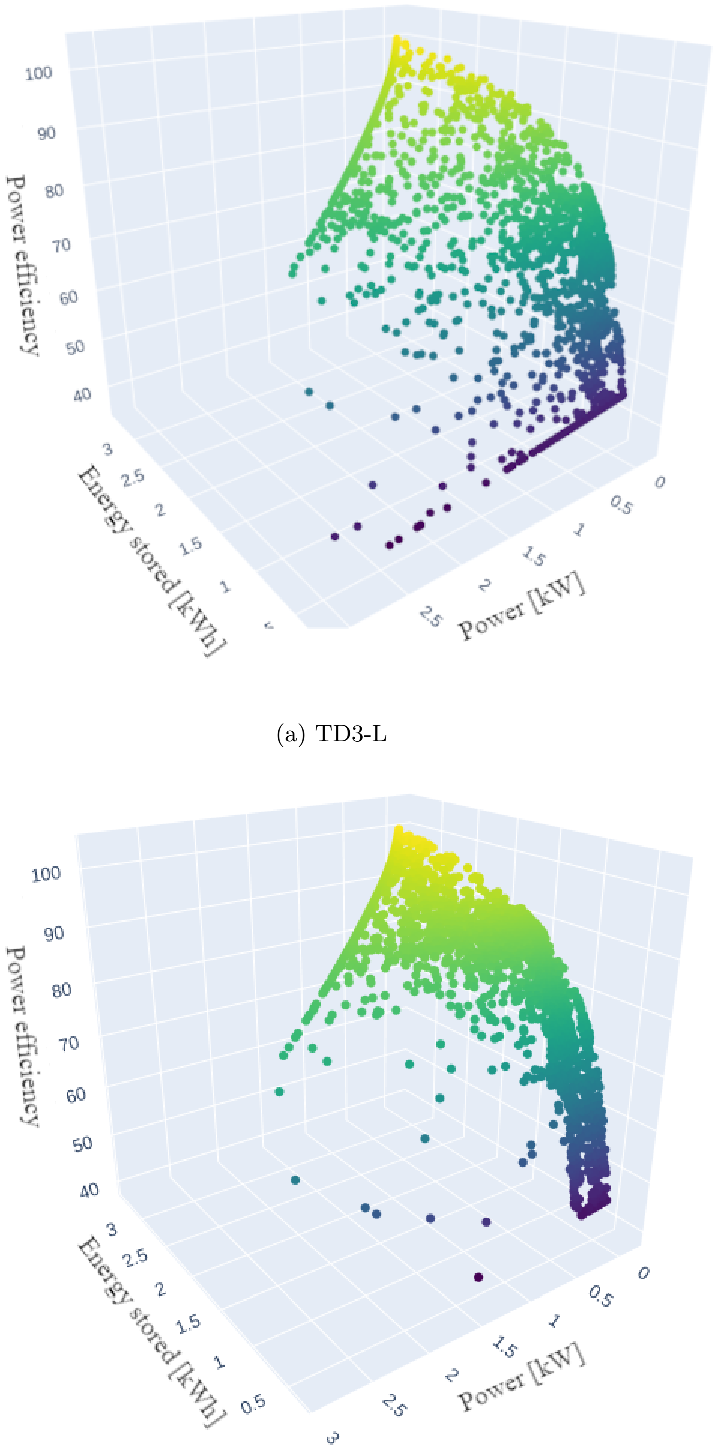


(a) TD3-L



(b) TD3-NL

Figure 5.4: 3D discharge efficiency for the TD3 in residential microgrid.



(a) TD3-L

(b) TD3-NL

Figure 5.5: 3D charge efficiency for the TD3 in residential microgrid.

Table 5.4: Energy losses of each algorithm and the difference between them (Residential).

	Energy Losses [kWh]		
Model	1$^{\text{st}}$ year	2$^{\text{nd}}$ year	3$^{\text{rd}}$ year
Linear	473.1695	502.5786	510.7831
Nonlinear	298.8784	285.2664	300.2103
(Difference)	174.2911	217.3122	210.5728

Table 5.5: Energy losses of each algorithm and the difference between both (CIGRE).

	Energy Losses [kWh]		
Model	1$^{\text{st}}$ year	2$^{\text{nd}}$ year	3$^{\text{rd}}$ year
Linear	6043.8106	5917.9887	6050.7169
Nonlinear	2585.0424	2524.1663	2561.6490
(Difference)	3458.7681	3393.8224	3489.0678

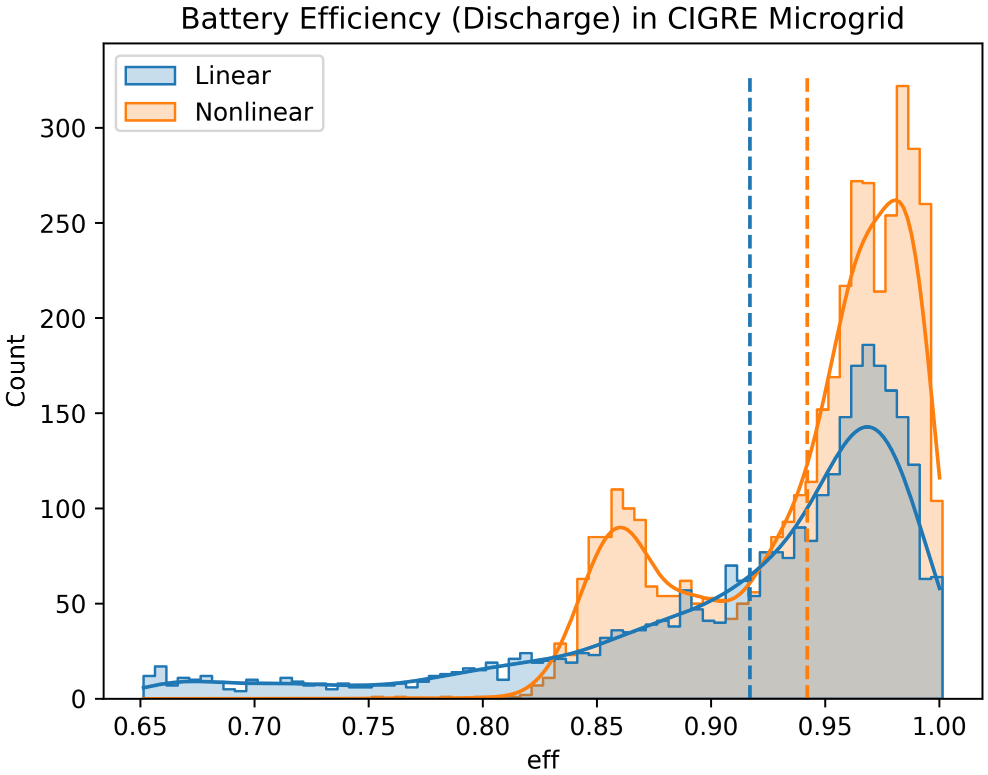


Figure 5.6: Discharge efficiency comparison of TD3 trained using the linear and nonlinear battery model in CIGRE microgrid (3$^{\text{rd}}$ year).

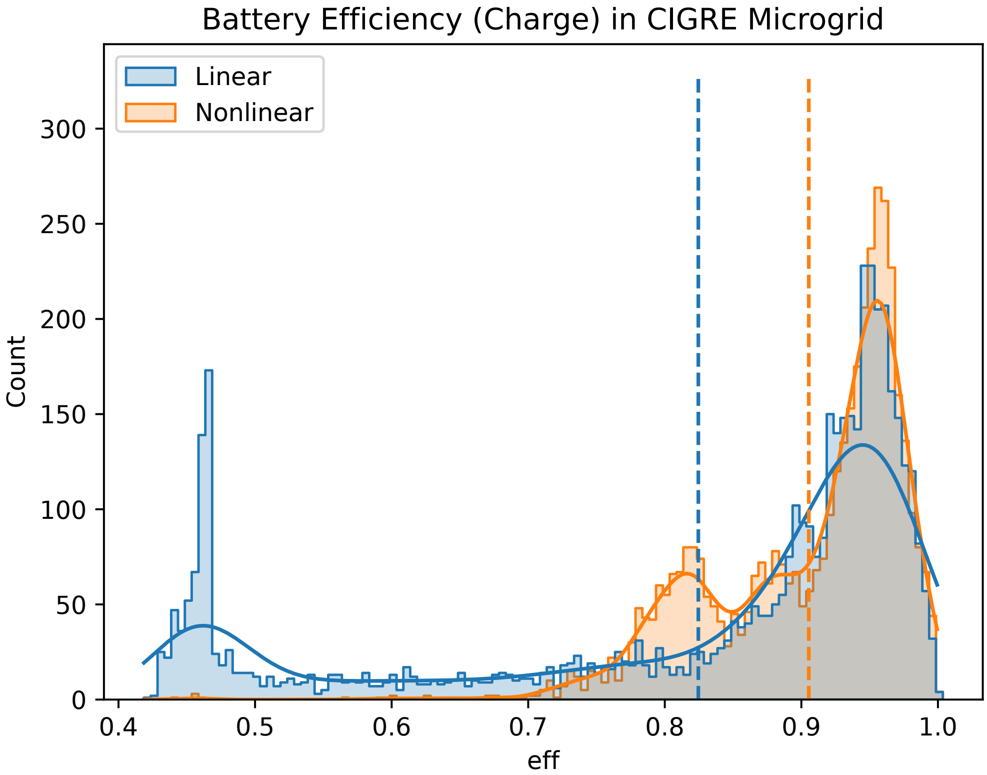


Figure 5.7: Charge efficiency comparison of TD3 trained using the linear and nonlinear battery model in CIGRE microgrid (3^{rd} year).

5.6 Extended experimental results

This section analyzes several experiments on the CIGRE case in order to study the variability of the training and evaluation process using the DRL approach proposed. In particular, six learning trials were conducted with the linear model and another six with the nonlinear model of the battery. During this process, for each trial, the data from year 1 is used for training, and data from year 2 is used for validation (this is the same setup used for the results in Section 5.5). Once the models have been adjusted, they can be applied to any data series. Thus, they have been applied to the 1st, 2nd, and 3rd year of the data described before. It is worth mentioning that the time series data from the last year were not considered at any stage of the model adjustment, and therefore, provide a good indicator of the model performance. The simulation results are depicted in Tables 5.6 and 5.7. Table 5.6 shows the accumulated cost (in euros) for each of the six adjusted models with the linear losses applied for each one of the available years. Table 5.7 shows the same results for the nonlinear case.

In each table, the variability of the results is due to the random initialization of the neural network and other random processes during the training, such as the exploration of the algorithm and the sampling from the ER memory. However, while this variability is a well-known characteristic that occurs every time a DRL model is adjusted, it is interesting to note that the obtained standard deviation is small. The absolute values of the coefficients of variation (i.e., the ratio between the standard deviation and the mean) are 1.12%, 1.27%, and 0.96% for the linear losses, and 0.72%, 0.79%, and 0.56% for the nonlinear losses.

Therefore, we can state that the results are quite similar between different trials, concluding that the approach is stable, i.e., there is high confidence in a single trial to obtain an acceptable performance. Additionally, the table comparison shows that the nonlinear approach leads to better performance on average. Comparing the mean values for each year, the nonlinear model outperforms the linear one by 1.60%, 1.48%, and 1.52% for years 1, 2, and 3, respectively.

Besides, these executions show the overhead in training the proposed approach. Figure 5.8 shows the computational burden, in timesteps, required to train the model and the performance of the best-chosen model.

This figure shows that considering a nonlinear model of the Li-ion battery losses instead of a linear model does not imply an extra burden in the learning process. From the data observed, the average number of timesteps needed with the linear model is 29.38 million, and with the nonlinear model is 13.80 million. Besides, the same figure corroborates the outperforming of using the nonlinear model to train the proposed DRL approach.

Table 5.6: Results from 6 trials using the linear model. Performance of the trained model (in euros) for each year.

	Obj. F. ↔ Cost [€]		
Linear	1st year	2nd year	3rd year
Trial 1	33138.86	30462.88	32962.70
Trial 2	33113.54	30388.39	32988.47
Trial 3	33242.64	30526.44	33107.89
Trial 4	34115.57	31449.56	33831.69
Trial 5	33306.55	30584.21	33219.58
Trial 6	33402.46	30614.49	33235.40
Average	33386.30	30671.00	33224.29
Std	372.75	390.12	318.33

Table 5.7: Results from 6 trials using the nonlinear model. Performance of the trained model (in euros) for each year.

	Obj. F. ↔ Cost [€]		
Nonlinear	1rd year	2rd year	3rrd year
Trial 1	32748.90	30096.43	32620.46
Trial 2	33326.87	30700.60	33090.92
Trial 3	32688.01	30119.19	32628.19
Trial 4	32749.10	30086.15	32617.68
Trial 5	32782.68	30125.02	32656.95
Trial 6	32816.07	30170.34	32705.12
Average	32851.94	30216.29	32719.89
Std	236.52	239.05	184.71

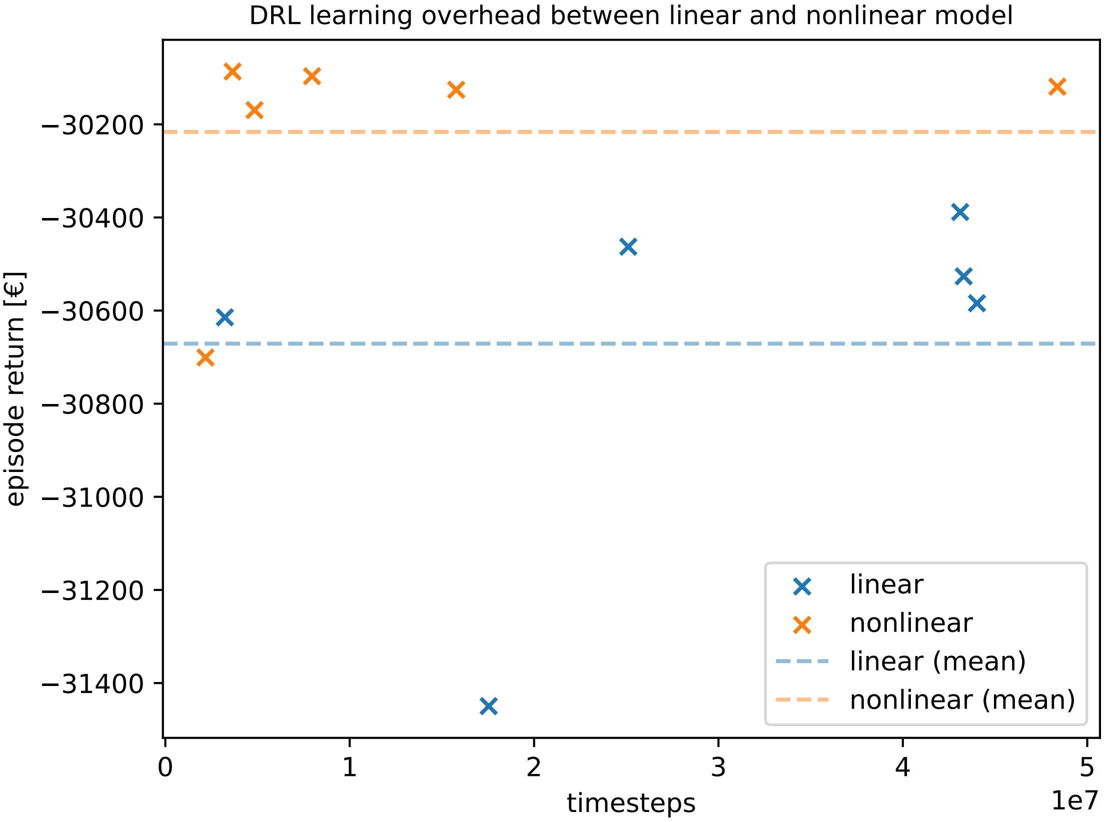


Figure 5.8: Computational burden required to train the model, measured in timesteps

5.7 Final remarks

This chapter extends the application of the TD3 algorithm to an EMS in a microgrid, seen in the previous chapter, and compares its performance using a linear (TD3-L) and a nonlinear (TD3-NL) Li-ion battery-loss model. The research addresses several critical questions:

1. To what extent can the DRL-based algorithm comprehend and adapt to the inclusion of nonlinear battery loss dynamics, especially considering its limited information?
2. What is the behavior of the TD3 when integrated with a real-time control system that adjusts EMS decisions?
3. How effective is the battery managed by the algorithm?

The experiments provide satisfactory answers to these queries. The TD3 algorithm demonstrates a capacity to synergize with the control system using the methodology explained in 5.4. Moreover, the inclusion of nonlinear dynamics helps the algorithm to obtain even better results due to its ability to discern and leverage these nonlinear dynamics.

The findings indicate that incorporating nonlinear battery losses can achieve approximately a 2% reduction in total microgrid operational costs without compromising computational performance. Additionally, the TD3-NL model significantly improves battery efficiency, reducing battery energy losses by approximately 50% compared to the TD3-L model. This reduction in losses represents about 10% of the total energy stored in the battery, meaning that a greater proportion of the stored energy is effectively utilized.

An important observation is that the modeling efforts and computational resources required by the learning algorithm are similar in handling linear or nonlinear equations. This stands out as a major benefit over other optimization techniques.

Chapter 6

Conclusions

6.1 Conclusions

The seminal idea of this research was to address the energy management problem in isolated microgrids through the lens of Deep Reinforcement Learning (DRL). Across three core studies, two DRL techniques, and two microgrid architectural configurations have been investigated to highlight how learning-based systems can autonomously and adaptively manage energy flows in increasingly complex power environments. This chapter brings together the key findings in this thesis, draws overarching conclusions, and provides a final synthesis of the major contributions and implications for future work.

Microgrids are hailed as vital components in modern power systems, enabling localized control, integration of Distributed Energy Resourcess (DERs), and improved resiliency. However, their operational complexity arises from uncertain renewable generation, dynamic load profiles, and the continuous nature of many control variables (e.g., battery charging/discharging levels). Classical optimization-focused control strategies often require highly specialized and mathematically complex formulations—particularly when dealing with nonlinearities—making them cumbersome to apply. Conversely, rule-based strategies are straightforward to implement but often fail to provide high-quality solutions, as they lack adaptability and optimization capabilities.

DRL offers a powerful alternative by learning control policies directly from data and interactions with the environment. Through iterative interactions, the DRL agent refines its strategy to optimize operational objectives—such as minimizing costs, ensuring supply-demand balance, and extending the lifetime of critical assets (e.g., batteries). DRL offers a fundamentally different approach compared to classical optimization. Instead of relying on an optimization process at each decision point, DRL algorithms train a global policy in advance—although this training may take hours, days, or more. Once trained, the policy can produce decisions almost instantaneously (within a few milliseconds). In contrast, classical optimization typically solves a planning problem over a finite horizon, producing a sequence of decisions in a single run. These decisions can remain valid over time, even when based on an approximated model, as long as the deviation from the real system is small. However, when the error accumulates or the environment undergoes significant changes, the optimization must be re-executed to adapt to the new scenario. Moreover, DRL algorithms can handle nonlinearities, partial observability, and continuous action spaces, without making complex modeling changes, making them ideally suited to the multifaceted challenges of microgrid energy management. About the performance, in this thesis, there is no fair comparison between both approaches—the classical optimization approach needs a forecasting method that is not explored in this thesis because it is out of scope. However, classical optimization methods are used to better evaluate the DRL algorithms as an optimistic performance case.

Nevertheless, applying DRL to power systems is an emerging research field. Several open questions have been discussed and analyzed in this thesis, such as: how to optimally structure the DRL agent's neural architecture, how best to encode system states and external variables into the DRL framework, and how to address nonlinearities on components like batteries. As demonstrated in this dissertation, there is no universal approach to these questions; instead, careful consideration of microgrid characteristics, algorithm-specific hyperparameter tuning, and robust benchmarking against optimization baselines are essential steps in driving the field forward.

The specific conclusions gathered from this thesis are the following:

Chapter 3 introduced a Convolutional Neural Network (CNN) architecture to manage an isolated microgrid. The simplicity of this architecture, combined with its robust performance, illustrated that CNN-based DRL solutions can effectively handle the sequential decision-making nature of microgrid Energy Management System (EMS).

A highlight of this study was the demonstration that the DRL agent learns from experience rather than relying on a tailor-made optimization model. This feature underscores the scalability of DRL solutions to diverse and evolving microgrid configurations.

Through a thorough evaluation, stacking nine hours of contiguous data was shown to be the most effective input window size for the studied microgrid, emphasizing the importance of capturing temporal dependencies in DRL applications for EMS. Nevertheless, stacking three hours also gives very good results, making it the most appropriate to simplify its practical implementation.

By benchmarking the DRL strategy against a perfect-information Mixed Integer Quadratic Programming (MIQP) reference model, this work demonstrates significant improvements over a naive baseline. Specifically, while a controller following a random policy performed 425.38% worse than the reference solution, the naive controller improved upon this but still performed 316.02% worse. In contrast, the DRL model achieved a much lower deviation of 36.46%, highlighting its effectiveness in optimizing microgrid operation.

This demonstrates that DRL techniques could improve over simpler systems. However, several considerations must be addressed. First, DRL methods require historical data for training, whereas naive approaches rely solely on expert-defined rules. Ideally, the training data should be collected from the same environment where the microgrid will be deployed, or alternatively, be based on sufficiently accurate synthetic data. Additionally, DRL methods do not require a forecasting module, which can simplify system design.

Building on prior work with Deep Q-Network (DQN), Chapter 4 adopted the Twin-Delayed Deep Deterministic Policy Gradient (TD3) algorithm. Due to its ability to handle continuous actions more precisely, TD3 outperformed DQN in both a microgrid resembling the previous setup and a standard Low-Voltage Microgrid Benchmark (CIGRE) microgrid.

While TD3 offered finer control granularity, it displayed less training stability than DQN—an inherent trait of policy-gradient-based methods. A complementary hyperparameter optimization strategy was undertaken to mitigate this instability, thereby unlocking the superior performance potential of TD3.

This evidence can be seen from the results of Chapter 4, when the DQN got 42.4% worse performance compared to the reference, and in contrast, the TD3 got 20.5% the same value. In absolute numbers, TD3 expenses were of 1452.21€ whereas DQN were from 1715.26€, in a residential microgrid in the test dataset over a time scope of 1 year.

Chapter 5 introduced a deeper look at battery modeling by comparing a linear and a nonlinear Lithium-Ion (Li-ion) battery loss model for the DRL setup from the previous work (when using TD3). This exploration demonstrates that DRL agents can exploit the behavior of nonlinear relationships among decision variables such as battery losses, leading to improved energy scheduling and cost savings.

A novel contribution was the integration of the DRL policy in a system closer to how a real-time control system behaves compared to the approaches currently used in the literature. The TD3 algorithm proved capable of synergizing with real-time updates, delivering near-optimal decisions even in uncertain environments.

Empirical results indicated that integrating a nonlinear loss model can achieve around 2% total cost savings with respect to using a linear one. Furthermore, battery energy losses were reduced by 50% compared to the linear model—a roughly 10% reduction in total battery energy usage—highlighting the importance of more accurate physical modeling in DRL frameworks.

Crucially, the more accurate nonlinear loss model did not substantially increase the computational burden. This finding underscores a major advantage: DRL agents can handle more intricate system dynamics with minimal extra modeling or runtime costs, a benefit that sets DRL apart from traditional optimization-based methods that might struggle with higher complexity models.

Taken collectively, the studies in this dissertation underscore the transformative potential of DRL for microgrid energy management. From discrete-action DQN-based approaches to continuous-action TD3 methods, the results reveal a clear trajectory: more sophisticated DRL algorithms, supported by careful model design and hyperparameter tuning, can achieve higher performance, better adapt to system nonlinearities, and more flexibly accommodate evolving grid complexities. The incorporation of nonlinear system models—especially for battery storage—further cements DRL's position as a technique that can handle intricate operational conditions without incurring prohibitive computational overhead.

On the other hand, evaluating the effectiveness of DRL techniques is a complex process. When new contexts are to be explored, successful research often depends on having sufficient DRL expertise, intuition, and computational resources. This body of work is only a stepping stone. The field of DRL for microgrid energy management remains vastly unexplored, with an overwhelming number of potential research directions. The vast array of DRL algorithms, the combinatorial explosion of neural network architectures, and the intricate design choices available in Reinforcement Learning (RL) create an almost limitless landscape of possibilities. Understanding which aspects of energy management problems align best with these advancements is a key challenge. Techniques such as Imitation Learning could help uncover more effective reward functions, particularly in capturing social behaviors like the real cost of unmet energy demand or modeling demand response in a way that realistically reflects human decision-making. Transfer Learning could enable scalable DRL applications across different microgrids with minimal computational effort, facilitating broader deployment. Additionally, safe RL must be explored to ensure that trained policies can be deployed without risking grid stability, avoiding potential system failures or irreversible operational issues.

Beyond algorithmic advancements, microgrids themselves have the potential to revolutionize the energy landscape. As renewable penetration increases, the transition from traditional meshed grids to more decentralized, distributed energy networks requires new coordination strategies. Additionally, microgrids could serve as a key enabler for bringing modern energy solutions to remote locations, from underserved communities in Africa to lunar settlements envisioned in programs like NASA's Artemis mission and in other works (Walth et al., 2024).

Scalability and the costly continuous monitoring of these systems further highlight the necessity of reducing supervision through advanced automation. Techniques like DRL can pave the way for autonomous or semi-autonomous microgrid operation with minimal computational resources (edge Artificial Intelligence (AI)).

Ultimately, by pairing advanced DRL architectures with realistic microgrid simulations and robust benchmarking methodologies, the field stands to make significant strides toward autonomy and efficiency in energy management. The findings presented here not only validate the feasibility of DRL in critical energy applications but also pave the way for more dynamic, scalable, and intelligent microgrid control strategies in the near future.

In conclusion, this dissertation demonstrates that DRL-based approaches are well-positioned to address the multifaceted challenges of microgrid energy management.

6.2 Main Contributions

The contributions in this thesis can be summarized as follows, and matched by the objective:

Regarding the first objective of achieving an RL-based EMS application as a baseline:

- Firstly, the development of a set of DRL-based models that enable the EMS of a microgrid to operate with satisfactory results in a simulation environment.
- Secondly, the results presented in Chapter 3 comprise practical window-size configurations that enable DRL technology to take a more pragmatic approach in real scenarios.
- Thirdly, the microgrid system considered is more complex than the previously solved in the state-of-the-art using DRL techniques, particularly a DQN with CNNs. Indeed, Chapter 3 includes not one but two controllable devices in the action space. Note that the number of decisions grows exponentially based on the number of components.
- Forthly, based on the opportunity cost concept, the implicit modeling approach effectively assumes the batteries in the optimal solution. Other solutions in the literature explicitly include battery usage as a synthetic cost that is unnecessary.

Regarding the second objective of enhancing the scalability of experimental components:

- The application of the TD3 (Fujimoto et al., 2018) to the microgrid energy management problem with a residential demand, in isolated mode, using three years of hourly data. This overcomes the limitations found in Chapter 3, i.e., the limited number of actions used. This approach is appropriate for considering continuous actions over the components of the microgrid.
- The TD3 technique performance is improved using the Tree-structured Parzen Estimator (TPE), a Gaussian Mixture Model (GMM)-based method (Bergstra et al., 2011) for hyperparameter optimization using Optuna software (Akiba et al., 2019). The application of the combined techniques also improves the stability of the learning process.

- A performance comparison is carried out between two different DRL algorithms. The DQN (Mnih et al., 2015), a more common approach, and the TD3. Furthermore, that chapter discusses the advantages and disadvantages of both methods and why the TD3 obtains better results than the DQN on the energy management problem.
- In addition to a small-scale microgrid system, the TD3 algorithm is implemented as the EMS of the CIGRE microgrid benchmark (Kariniotakis et al., 2005; Papathanassiou et al., 2005). The results from applying the proposed algorithm implementation, with the chosen hyperparameters, suggest that the TD3 is a candidate algorithm to manage a real-world microgrid efficiently.

Finally, regarding the third objective of drawing conclusions about the impact of incorporating nonlinear dynamics into the model when applying DRL techniques:

- First of all, a microgrid model that includes the nonlinear behavior of Li-ion batteries is proposed in Chapter 5 for the training of the TD3 algorithm. This model extends the Partially-Observable Markov Decision Process (POMDP) of the microgrid, developed in Chapters 3 and 4, with the nonlinear equations of the battery losses.
- Secondly, Chapter 5 introduces a methodology to assess the performance improvements of the proposed algorithm compared to EMSs models with linear battery losses. Additionally, this chapter incorporates a novel approach to modeling the microgrid control system, allowing a reduction in the energy not supplied (ENS) (and curtailment) since more elements take part in case the set points given by the DRL algorithm reach an unbalanced energy state between generation and demand.
- Thirdly, this method is extended to manage a microgrid of a larger size than a residential one, ensuring that the proposed method is valid for both a residential microgrid and a low-voltage distribution network extended into a microgrid.

6.3 Future research

Drawing upon the insights and limitations highlighted throughout this dissertation, several promising directions emerge for extending and enhancing DRL-based microgrid energy management systems. These prospective avenues not only aim to address current challenges but also seek to push the boundaries of what DRL can achieve in increasingly dynamic and multi-faceted power systems:

Related to power systems' modeling:

- A closely related research direction, sometimes discussed during the development of this thesis and even preceding its inception, is the expansion of the microgrid EMS to grid-connected scenarios. In such settings, microgrids participate in electricity trading within wholesale or local energy markets, engaging in both buying and selling to optimize operational efficiency and economic benefits. DRL algorithms could optimize trading strategies under uncertain prices, supporting both cost reduction and revenue generation. In this regard, preliminary tests have already been conducted, but significant work remains to be done in order to have some results to show. These initial tests involved a setup consisting of only a battery and the main grid, utilizing historical time series of real energy prices. The objective was to develop a bidding strategy for the battery to optimize energy trading.

- In scenarios with multiple microgrids or distributed energy resources, a multi-agent DRL system could coordinate decisions while balancing local and global objectives, leading to more robust and scalable solutions.

- Another promising avenue for future research is examining how DRL policies adapt to evolving regulatory and policy constraints. Investigating their response to mechanisms such as net metering or capacity markets could offer valuable insights into their real-world applicability, compliance, and overall viability within dynamic grid regulations.

- An additional research direction involves integrating Demand Response strategies into the DRL framework. By enabling the system to react to electricity price signals and user incentives dynamically, this approach could further optimize operational costs and enhance peak load management.

- Furthermore, extending DRL to allocate energy intelligently among loads with different priorities or elasticity levels can enhance reliability and user satisfaction.

- Regarding the battery modeling, another future research could integrate more sophisticated degradation models to inform charging and discharging decisions that extend battery life while minimizing operational costs. Moreover, more precise battery behavior simulators, such as pyBaMM, could be integrated into the microgrid model, allowing RL to optimize directly using these detailed simulations (Sulzer et al., 2021). This integration would provide significantly more accurate data for training policies, leading to better-informed decision-making and improved energy management strategies.

- In the same regard, exploring a combination of different storage technologies (e.g., supercapacitors or hydrogen storage) within a DRL framework could optimize the utilization of each storage's unique characteristics.

Regarding improvements on the algorithm side:

- Incorporating advanced forecasting methods (e.g., Deep Learning (DL)-based solar and load forecasts) into the state representation for the DRL agent—the feature extraction module of the DL architecture—could improve decision-making under uncertainty.

- Investigating hierarchical RL—where upper-level policies coordinate high-level objectives (e.g., cost minimization) and lower-level policies handle local tasks (e.g., battery control)—could substantially reduce complexity and improve stability.

- As mentioned in the conclusions, Safe RL could facilitate the deployment and autonomous operation of microgrids by preventing potential issues during operation, such as stability failures or other critical disruptions.

- Transfer Learning techniques could also facilitate the development of large-scale meshed networks of interconnected microgrids. By leveraging knowledge from previously trained models, these techniques would enable faster adaptation to new environments, reducing the need for extensive retraining.

- Inverse RL could be used to study reward functions based on human behavior when interacting with electrical loads. This approach would enable a more efficient management of energy and flexible demand by learning from real user preferences and decision-making patterns, ultimately leading to more adaptive and human-centered energy management strategies.
- As real-world systems deviate from idealized models, designing DRL agents that remain stable under system parameter variations or sensor/data inaccuracies is critical. Techniques like domain randomization or adversarial training can be explored.
- Future work could integrate explainable AI (XAI) approaches to interpret the decisions made by DRL policies, fostering stakeholders' trust and facilitating regulatory approval.
- Additionally, the investment problem could be explored by leveraging the ability to compute a value function that estimates the objective function more accurately than other approaches that rely solely on deep learning. This would provide a more robust framework for investment decision-making in microgrid planning.
- Conversely, Imitation Learning could be applied to accelerate DRL training by learning from classical optimization results or real operational data from expert decisions. This approach would enable the agent to quickly adopt effective strategies, reducing training time and improving initial policy performance. In contrast, it may introduce bias.
- Experimental validation of DRL policies in real-time simulation environments or pilot microgrid testbeds can bridge the gap between theoretical models and practical deployment.
- Strategies to reduce computation time and memory requirements—along with the capability for continuous learning in a live system—will be crucial for widespread adoption.

Exploring cross-domain synergies:

- By applying blockchain technology, energy trading between distributed microgrids could be fully automated. DRL algorithms could analyze environmental variables to identify the most optimal local cooperative solution and, in real time, execute smart contracts with other microgrids to facilitate energy transactions.
- As microgrids become more connected, cybersecurity threats grow in importance. Investigating DRL algorithms resilient to malicious data manipulation or communications interference represents a valuable extension.
- Future work could link microgrid energy management to broader urban infrastructures, such as electric vehicle fleets and district heating systems, creating integrated, multi-sector control strategies.

Appendix A

Mixed Integer Quadratic Programming Formulation

$$\min \quad \sum_t [\delta_0 U_t^{\mathrm{d}} + \delta_1 P_t^{\mathrm{d}} + \delta_2 (P_t^{\mathrm{d}})^2 + \mathrm{ens} \cdot c^{\mathrm{ens}}] \tag{A.1}$$

$$\text{subject to} \quad D_t - \mathrm{ens} = (P_t^{\mathrm{pv}} - \mathrm{curt}_t) + (P_t^{\mathrm{b}\rightarrow} - P_t^{\mathrm{b}\leftarrow}) + (P_t^{\mathrm{h_2}\rightarrow} - P_t^{\mathrm{h_2}\leftarrow}) + P_t^{\mathrm{d}} \quad \forall t \in T \tag{A.2}$$

$$S_t^{\mathrm{b}} = -\frac{P_t^{\mathrm{b}\rightarrow}}{\zeta^{\mathrm{b}}} + \eta^{\mathrm{b}} P_t^{\mathrm{b}\leftarrow} + S_{t-1}^{\mathrm{b}} \quad \forall t \in T \tag{A.3}$$

$$S_{\min}^{\mathrm{b}} \leq S_t^{\mathrm{b}} \leq S_{\max}^{\mathrm{b}} \quad \forall t \in T \tag{A.4}$$

$$S_t^{\mathrm{h_2}} = -\frac{P_t^{\mathrm{h_2}\rightarrow}}{\zeta^{\mathrm{h_2}}} + \eta^{\mathrm{h_2}} P_t^{\mathrm{h_2}\leftarrow} + S_{t-1}^{\mathrm{h_2}} \quad \forall t \in T \tag{A.5}$$

$$S_{\min}^{\mathrm{h_2}} \leq S_t^{\mathrm{h_2}} \leq S_{\max}^{\mathrm{h_2}} \quad \forall t \in T \tag{A.6}$$

$$S_0^{\mathrm{h_2}} \leq S_t^{\mathrm{h_2}} \quad \text{\# To force not leaving the battery completely depleted} \quad t = |T| \tag{A.7}$$

$$P_t^{\mathrm{d}} \leq U_t^{\mathrm{d}} * P_{\max}^{\mathrm{d}} \quad \forall t \in T \tag{A.8}$$

$$U_t^{\mathrm{d}} \in \{0, 1\} \quad \forall t \in T \tag{A.9}$$

$$\mathrm{ens}_t, P_t^{\mathrm{d}}, \mathrm{curt}_t, P_t^{\mathrm{b}\rightarrow}, P_t^{\mathrm{b}\leftarrow}, P_t^{\mathrm{h_2}\rightarrow}, P_t^{\mathrm{h_2}\leftarrow} \geq 0 \quad \forall t \in T \tag{A.10}$$

$$P_t^{\mathrm{d}} \leq P_{\max}^{\mathrm{d}} \quad \forall t \in T \tag{A.11}$$

$$\mathrm{curt}_t \leq P_t^{\mathrm{pv}} \quad \forall t \in T \tag{A.12}$$

$$P_t^{\mathrm{b}\rightarrow}, P_t^{\mathrm{b}\leftarrow} \leq P_{\max}^{\mathrm{b}} \quad \forall t \in T \tag{A.13}$$

$$P_t^{\mathrm{h_2}\rightarrow}, P_t^{\mathrm{h_2}\leftarrow} \leq P^{\mathrm{h_2}} \quad \forall t \in T \tag{A.14}$$

where (A.1) is the objective function, (A.2) is the demand balance constraint, (A.3) and (A.5) model the dynamics of the battery and the hydrogen storage, respectively, while (A.4) and (A.6) keeps their energy amount between technical limits, (A.7) establishes a minimum hydrogen storage level in the last hour, and (A.8) indicates that unit commitment of the diesel generator is a binary variable. The remaining equations impose the upper and lower bounds on the decision variables.

Bibliography

Achiam, J. (2018). Spinning Up in Deep Reinforcement Learning. Retrieved December 2, 2024, from https://spinningup.openai.com

Ahmad, S., Shafiullah, M., Ahmed, C. B., & Alowaifeer, M. (2023). A Review of Microgrid Energy Management and Control Strategies. *IEEE Access*, *11*, 21729–21757. https://doi.org/10.1109/ACCESS.2023.3248511

Akiba, T., Sano, S., Yanase, T., Ohta, T., & Koyama, M. (2019). Optuna: A next-generation hyperparameter optimization framework. *Proceedings of the 25rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.*

Akinwale, O. S., Mojisola, D. F., Adediran, P. A., Akinwale, O. S., Mojisola, D. F., & Adediran, P. A. (2021). Mitigation strategies for communication networks induced impairments in autonomous microgrids control: A review. *AIMS Electronics and Electrical Engineering*, *5*(4), 342–375. https://doi.org/10.3934/electreng.2021018

Akinyele, D., Belikov, J., & Levron, Y. (2018). Challenges of Microgrids in Remote Communities: A STEEP Model Application. *Energies*, *11*(2), 432. https://doi.org/10.3390/en11020432

Aladesanmi, E. J., & Ogudo, K. A. (2023). Microgrids Overview and Performance Evaluation on Low-voltage Distribution Network. *Clean Energy and Sustainability*, *2*(1), 10008. https://doi.org/10.35534/ces.2023.10008

Alavi, S. A., Ahmadian, A., & Aliakbar-Golkar, M. (2015). Optimal probabilistic energy management in a typical micro-grid based-on robust optimization and point estimate method. *Energy Conversion and Management*, *95*, 314–325. https://doi.org/10.1016/j.enconman.2015.02.042

Altin, N., & Eyimaya, S. E. (2021). A Review of Microgrid Control Strategies. *2021 10th International Conference on Renewable Energy Research and Application (ICRERA)*, 412–417. https://doi.org/10.1109/ICRERA52334.2021.9598699

Amirioun, M. H., Aminifar, F., & Lesani, H. (2018). Resilience-Oriented Proactive Management of Microgrids Against Windstorms [Conference Name: IEEE Transactions on Power Systems]. *IEEE Transactions on Power Systems*, *33*(4), 4275–4284. https://doi.org/10.1109/TPWRS.2017.2765600

Amrollahi, M. H., & Bathaee, S. M. T. (2017). Techno-economic optimization of hybrid photovoltaic/wind generation together with energy storage system in a stand-alone micro-grid subjected to demand response. *Applied Energy*, *202*, 66–77. https://doi.org/10.1016/j.apenergy.2017.05.116

Asian Development Bank. (2020, August). *Handbook on Microgrids for Power Quality and Connectivity:* (tech. rep.). Asian Development Bank. Manila, Philippines. https://doi.org/10.22617/TIM200182-2

Banker, T., & Mesbah, A. (2025). Model-free Reinforcement Learning for Model-based Control: Towards Safe, Interpretable and Sample-efficient Agents. https://doi.org/10.48550/arXiv.2507.13491

Bellemare, M. G., Dabney, W., & Munos, R. (2017). (C51) A Distributional Perspective on Reinforcement Learning. *Proceedings of the 34th International Conference on Machine Learning*, 449–458. https://proceedings.mlr.press/v70/bellemare17a.html

Bergstra, J., Bardenet, R., Bengio, Y., & Kégl, B. (2011). Algorithms for Hyper-Parameter Optimization. *Advances in Neural Information Processing Systems*, *24*.

Bi, W., Shu, Y., Dong, W., & Yang, Q. (2020). Real-time Energy Management of Microgrid Using Reinforcement Learning. *2020 19th International Symposium on Distributed Computing and Applications for Business Engineering and Science (DCABES)*, 38–41.

Black, K., Brown, N., Driess, D., Esmail, A., Equi, M., Finn, C., Fusai, N., Groom, L., Hausman, K., Ichter, B., Jakubczak, S., Jones, T., Ke, L., Levine, S., Li-Bell, A., Mothukuri, M., Nair, S., Pertsch, K., Shi, L. X., … Zhilinsky, U. (2024). $Π_0$: A Vision-Language-Action Flow Model for General Robot Control. https://doi.org/10.48550/arXiv.2410.24164

Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. (2014, December). Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. https://doi.org/10.48550/arXiv.1412.3555

Dabney, W., Ostrovski, G., Silver, D., & Munos, R. (2018). Implicit Quantile Networks for Distributional Reinforcement Learning [ISSN: 2640-3498]. *Proceedings of the 35th International Conference on Machine Learning*, 1096–1105. Retrieved October 31, 2024, from https://proceedings.mlr.press/v80/dabney18a.html

Dabney, W., Rowland, M., Bellemare, M., & Munos, R. (2018). Distributional Reinforcement Learning With Quantile Regression. *Proceedings of the AAAI Conference on Artificial Intelligence*, *32*(1). https://doi.org/10.1609/aaai.v32i1.11791

Dev, A., Kumar, V., Khare, G., Giri, J., Amir, M., Ahmad, F., Jain, P., & Anand, S. (2025). Advancements and Challenges in Microgrid Technology: A Comprehensive Review of Control Strategies, Emerging Technologies, and Future Directions. *Energy Science & Engineering*, *13*(4), 2112–2134. https://doi.org/10.1002/ese3.2095

Dinata, N. F. P., Ramli, M. A. M., Jambak, M. I., Sidik, M. A. B., & Alqahtani, M. M. (2024). Designing an optimal microgrid control system using deep reinforcement learning: A systematic review. *Engineering Science and Technology, an International Journal*, *51*, 101651. https://doi.org/10.1016/j.jestch.2024.101651

Domínguez-Barbero, C., García-González, J., Sanz-Bobi, M. A., & Sánchez-Úbeda, E. F. (2020). Optimising a microgrid system by deep reinforcement learning techniques. *Energies*, *13*(11). https://doi.org/10.3390/en13112830

Domínguez-Barbero, C., García-González, J., Sanz-Bobi, M. Á., & García-Cerrada, A. (2024). Energy management of a microgrid considering nonlinear losses in batteries through Deep Reinforcement Learning. *Applied Energy*, *368*, 123435. https://doi.org/10.1016/j.apenergy.2024.123435

Domínguez-Barbero, C., García-González, J., & Sanz-Bobi, M. Á. (2022). Twin-delayed deep deterministic policy gradient algorithm for the energy management of microgrids. *Engineering*

Applications of Artificial Intelligence, *13*(11). https://doi.org/10.1016/j.engappai.2023.106693

Dozat, T. (2016). Incorporating Nesterov Momentum into Adam. *ICLR 2016 workshop*. https://openreview.net/forum?id=OM0jvwB8jIp57ZJjtNEZ

Driesse, A., Jain, P., & Harrison, S. (2008). Beyond the curves: Modeling the electrical efficiency of photovoltaic inverters [ISSN: 0160-8371]. *2008 33rd IEEE Photovoltaic Specialists Conference*, 1–6. https://doi.org/10.1109/PVSC.2008.4922827

Elliott, D. (2024, September). This start-up is using microgrids to bring reliable electricity to Nigeria. Retrieved August 14, 2025, from https://www.weforum.org/stories/2024/09/startup-reliable-power-nigeria/

EnergySage. (2024). EnergySage: 6kW solar system. https://www.energysage.com/solar/6kw-solar-system-compare-prices-installers/

Eurostat. (2025). Energy production and imports. Retrieved August 13, 2025, from https://ec.europa.eu/eurostat/statistics-explained/index.php?title=Energy_production_and_imports

Eyimaya, S. E., & Altin, N. (2024). Review of Energy Management Systems in Microgrids. *Applied Sciences*, *14*(3), 1249. https://doi.org/10.3390/app14031249

Fortunato, M., Azar, M. G., Piot, B., Menick, J., Osband, I., Graves, A., Mnih, V., Munos, R., Hassabis, D., Pietquin, O., Blundell, C., & Legg, S. (2017). Noisy Networks for Exploration. *ICLR 2018*. http://arxiv.org/abs/1706.10295

Fotouhi, A., Auger, D. J., Propp, K., Longo, S., & Wild, M. (2016). A review on electric vehicle battery modelling: From Lithium-ion toward Lithium–Sulphur. *Renewable and Sustainable Energy Reviews*, *56*, 1008–1021. https://doi.org/10.1016/j.rser.2015.12.009

François-Lavet, V., et al. (2016). DeeR.

François-Lavet, V., Gemine, Q., Ernst, D., & Fonteneau, R. (2016, April). Towards the Minimization of the Levelized Energy Costs of Microgrids using both Long-term and Short-term Storage Devices. In *Smart Grid: Networking, Data Management, and Business Models* (pp. 295–319). CRC Press.

François-Lavet, V., Rabusseau, G., Pineau, J., Ernst, D., & Fonteneau, R. (2019). On Overfitting and Asymptotic Bias in Batch Reinforcement Learning with Partial Observability. *Journal of Artificial Intelligence Research*, *65*, 1–30.

François-Lavet, V., Taralla, D., Ernst, D., & Fonteneau, R. (2016). Deep Reinforcement Learning Solutions for Energy Microgrids Management [Permalink: https://hdl.handle.net/2268/203831]. *European Workshop on Reinforcement Learning'13*.

Fujimoto, S., van Hoof, H., & Meger, D. (2018). Addressing function approximation error in actor-critic methods. *Proceedings of the 35th International Conference on Machine Learning*, *80*, 1587–1596.

GAMS Development Corporation. (2013). General Algebraic Modeling System (GAMS) Release 24.2.1.

Gao, F., Kang, R., Cao, J., & Yang, T. (2019). Primary and secondary control in dc microgrids: A review. *Journal of Modern Power Systems and Clean Energy*, *7*(2), 227–242. https://doi.org/10.1007/s40565-018-0466-5

García-González, J. (2023). Expressions of Power Losses when Charging and Discharging Li-Ion Batteries. *techrxiv*. https://doi.org/10.36227/techrxiv.23253914.v1

García-González, J., & Guerrero, S. (2024). Optimal management of a microgrid li-ion battery considering non-linear losses using the integer zig-zag formulation. *Electric Power Systems Research*, *235*, 110776. https://doi.org/https://doi.org/10.1016/j.epsr.2024.110776

Garcia-Torres, F., Bordons, C., Tobajas, J., Real-Calvo, R., Santiago, I., & Grieu, S. (2021). Stochastic optimization of microgrids with hybrid energy storage systems for grid flexibility services considering energy forecast uncertainties. *IEEE Transactions on Power Systems*, *36*(6), 5537–5547. https://doi.org/10.1109/TPWRS.2021.3071867

Geurts, P., Ernst, D., & Wehenkel, L. (2006). Extremely randomized trees. *Machine Learning*, *63*(1), 3–42. https://doi.org/10.1007/s10994-006-6226-1

Glorot, X., & Bengio, Y. (2010, May). Understanding the difficulty of training deep feedforward neural networks. In Y. W. Teh & M. Titterington (Eds.), *Proceedings of the thirteenth international conference on artificial intelligence and statistics* (pp. 249–256, Vol. 9). PMLR. http://proceedings.mlr.press/v9/glorot10a.html

Gupta, R. A., & Gupta, N. K. (2015). A robust optimization based approach for microgrid operation in deregulated environment. *Energy Conversion and Management*, *93*, 121–131. https://doi.org/10.1016/j.enconman.2015.01.008

Gurobi Optimization, LLC. (2024). Gurobi Optimizer Reference Manual [Last access: 2024-03-11]. https://www.gurobi.com

Hadi, M., Elbouchikhi, E., Zhou, Z., Saim, A., Shafie-khah, M., Siano, P., Rahbarimagham, H., & Colom, P. M. (2025). Artificial intelligence for microgrids design, control, and maintenance: A comprehensive review and prospects. *Energy Conversion and Management: X*, *27*, 101056. https://doi.org/10.1016/j.ecmx.2025.101056

Hasan, M. A., Hossain, M. S., Roslan, M. A., Azmi, A., Hwai, L. J., Nazib, A. A., & Ahmad, N. S. (2025). A comprehensive review of control strategies and efficiency optimization for islanded AC microgrids. *IFAC Journal of Systems and Control*, *33*, 100326. https://doi.org/10.1016/j.ifacsc.2025.100326

He, K., Zhang, X., Ren, S., & Sun, J. (2015). Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. *The IEEE International Conference on Computer Vision (ICCV)*.

Henderson, P., Islam, R., Bachman, P., Pineau, J., Precup, D., & Meger, D. (2018). Deep reinforcement learning that matters. *Proceedings of the AAAI Conference on Artificial Intelligence*, *32*(1).

Hessel, M., Modayil, J., Hasselt, H. v., Schaul, T., Ostrovski, G., Dabney, W., Horgan, D., Piot, B., Azar, M., & Silver, D. (2018). Rainbow: Combining Improvements in Deep Reinforcement Learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, *32*(1). https://doi.org/10.1609/aaai.v32i1.11796

Hirsch, A., Parag, Y., & Guerrero, J. (2018). Microgrids: A review of technologies, key drivers, and outstanding issues. *Renewable and Sustainable Energy Reviews*, *90*, 402–411. https://doi.org/10.1016/j.rser.2018.03.040

Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, *9*(8), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

Howard, R. A. (2018). *Dynamic programming and markov processes* (Second). John Wiley.

Huang, A. Q., Crow, M. L., Heydt, G. T., Zheng, J. P., & Dale, S. J. (2011). The Future Renewable Electric Energy Delivery and Management (FREEDM) System: The Energy Internet. *Proceedings of the IEEE*, *99*(1), 133–148. https://doi.org/10.1109/JPROC.2010.2081330

Huang, Y., Rong, X., Bie, Z., Li, J., Huang, B., Zhao, T., & Li, G. (2025). Artificial intelligence for resilient power system: Motivations, advances, and challenges. *Smart Power & Energy Security*. https://doi.org/10.1016/j.spes.2025.06.001

Ibrahim, S., Mostafa, M., Jnadi, A., Salloum, H., & Osinenko, P. (2024). Comprehensive Overview of Reward Engineering and Shaping in Advancing Reinforcement Learning Applications. *IEEE Access*, *12*, 175473–175500. https://doi.org/10.1109/ACCESS.2024.3504735

IEA. (2019). *Nigeria Energy Outlook 2019* (tech. rep.). IEA. Retrieved August 14, 2025, from https://www.iea.org/articles/nigeria-energy-outlook

(IEA), I. E. A. (2024). Renewables 2024. Retrieved November 1, 2024, from https://www.iea.org/reports/renewables-2024

Islam, R., Henderson, P., Gomrokchi, M., & Precup, D. (2017). Reproducibility of Benchmarked Deep Reinforcement Learning Tasks for Continuous Control. *arXiv:1708.04133 [cs]*.

Jasmin, E. A., Imthias Ahamed, T. P., & Jagathy Raj, V. P. (2011). Reinforcement Learning approaches to Economic Dispatch problem. *International Journal of Electrical Power & Energy Systems*, *33*(4), 836–845. https://doi.org/10.1016/j.ijepes.2010.12.008

Ji, Y., Wang, J., Xu, J., Fang, X., & Zhang, H. (2019). Real-Time Energy Management of a Microgrid Using Deep Reinforcement Learning [Number: 12, Publisher: Multidisciplinary Digital Publishing Institute]. *Energies*, *12*(12), 2291.

Kaelbling, L. P., Littman, M. L., & Moore, A. W. (1996). Reinforcement Learning: A Survey. *Journal of Artificial Intelligence Research*, *4*, 237–285. https://doi.org/10.1613/jair.301

Kaelbling, L. P., Littman, M. L., & Cassandra, A. R. (1998). Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, *101*(1-2), 99–134.

Kariniotakis, G., Soultanis, N., Tsouchnikas, A., Papathanasiou, S., & Hatziargyriou, N. (2005). Dynamic modeling of microgrids. *2005 International Conference on Future Power Systems*, 7 pp.–7. https://doi.org/10.1109/FPS.2005.204227

Kim, B., Zhang, Y., van der Schaar, M., & Lee, J. (2016). Dynamic pricing and energy consumption scheduling with reinforcement learning. *IEEE Transactions on Smart Grid*, *7*(5), 2187–2198. https://doi.org/10.1109/TSG.2015.2495145

Kim, R.-K., Glick, M. B., Olson, K. R., & Kim, Y.-S. (2020). MILP-PSO Combined Optimization Algorithm for an Islanded Microgrid Scheduling with Detailed Battery ESS Efficiency Model and Policy Considerations. *Energies*, *13*(8), 1898. https://doi.org/10.3390/en13081898

Konda, V. R., & Tsitsiklis, J. N. (1999). Actor-Critic Algorithms. *NIPS*, 7.

Lambert, T., Gilman, P., & Lilienthal, P. (2006). Micropower system modeling with homer. *Integration of alternative sources of energy*, *1*(1), 379–385.

Lasseter, R., & Paigi, P. (2004). Microgrid: A conceptual solution [ISSN: 0275-9306]. *2004 IEEE 35th Annual Power Electronics Specialists Conference*, *6*, 4285–4290 Vol.6. https://doi.org/10.1109/PESC.2004.1354758

Lasseter, R. H., Akhil, A. A., Marnay, C., Stephens, J., Dagle, J. E., Guttromson, R. T., Meliopoulous, A. S., Yinger, R. J., & Eto, J. H. (2003, October). *Integration of Distributed Energy Resources: The CERTS MicroGrid Concept* (Report No. LBNL-50829). Consortium for Electric Reliability Technology Solutions. Berkeley, CA.

Le, K. D., & Day, J. T. (1982). Rolling Horizon Method: A New Optimization Technique for Generation Expansion Studies. *IEEE Transactions on Power Apparatus and Systems*, *PAS-101*(9), 3112–3116. https://doi.org/10.1109/TPAS.1982.317523

LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., & Jackel, L. D. (1989). Backpropagation Applied to Handwritten Zip Code Recognition. *Neural Computation*, *1*(4), 541–551. https://doi.org/10.1162/neco.1989.1.4.541

Lecun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition [Conference Name: Proceedings of the IEEE]. *Proceedings of the IEEE*, *86*(11), 2278–2324. https://doi.org/10.1109/5.726791

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning [Publisher: Nature Publishing Group]. *Nature*, *521*(7553), 436–444. https://doi.org/10.1038/nature14539

Lei, L., Tan, Y., Dahlenburg, G., Xiang, W., & Zheng, K. (2021). Dynamic energy dispatch based on deep reinforcement learning in iot-driven smart isolated microgrids. *IEEE Internet of Things Journal*, *8*(10), 7938–7953. https://doi.org/10.1109/JIOT.2020.3042007

Levine, S., & Koltun, V. (2013). Guided Policy Search [ISSN: 1938-7228]. *Proceedings of the 30th International Conference on Machine Learning*, 1–9. https://proceedings.mlr.press/v28/levine13.html

Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., & Wierstra, D. (2015). Continuous control with deep reinforcement learning. *arXiv:1509.02971 [cs, stat]*. https://doi.org/10.48550/arXiv.1509.02971

Lin, L.-J. (1993). *Reinforcement learning for robots using neural networks* [Doctoral dissertation]. Carnegie Mellon University.

Mariam, L., Basu, M., & Conlon, M. F. (2016). Microgrid: Architecture, policy and future trends. *Renewable and Sustainable Energy Reviews*, *64*, 477–489. https://doi.org/10.1016/j.rser.2016.06.037

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Jia, Y., Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, ... Xiaoqiang Zheng. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems [Software available from tensorflow.org]. https://doi.org/10.5281/zenodo.4724125

Mbuwir, B. V., Geysen, D., Spiessens, F., & Deconinck, G. (2019). Reinforcement learning for control of flexibility providers in a residential microgrid. *IET Smart Grid*, *3*(1), 98–107.

Mbuwir, B. V., Ruelens, F., Spiessens, F., & Deconinck, G. (2017). Battery Energy Management in a Microgrid Using Batch Reinforcement Learning. *Energies*, *10*(11), 1846. https://doi.org/10.3390/en10111846

McCulloch, W. S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, *5*(4), 115–133. https://doi.org/10.1007/BF02478259

Meuleau, N., Peshkin, L., Kaelbling, L. P., & Kim, K.-E. (2000). *Off-policy policy search* (tech. rep.). MIT Articical Intelligence Laboratory.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., & Hassabis, D. (2015). Human-level control through deep reinforcement learning. *Nature*, *518*(7540), 529–533. https://doi.org/10.1038/nature14236

Munos, R., Stepleton, T., Harutyunyan, A., & Bellemare, M. (2016). Safe and Efficient Off-Policy Reinforcement Learning. *Advances in Neural Information Processing Systems*, *29*. https://

proceedings.neurips.cc/paper/2016/hash/c3992e9a68c5ae12bd18488bc579b30d-Abstract.html

Nair, V., & Hinton, G. E. (2010). Rectified linear units improve restricted boltzmann machines. *Proceedings of the 27th International Conference on International Conference on Machine Learning*, 807–814.

Nakabi, T. A., & Toivanen, P. (2021). Deep reinforcement learning for energy management in a microgrid with flexible demand. *Sustainable Energy, Grids and Networks*, *25*, 100413. https://doi.org/10.1016/j.segan.2020.100413

NCEL. (2024, July). Microgrids and Virtual Power Plants Issue Brief. Retrieved August 14, 2025, from https://ncel.net/resources/microgrids-and-virtual-power-plants-issue-brief/

Nguyen, T. A., & Crow, M. L. (2016). Stochastic Optimization of Renewable-Based Microgrid Operation Incorporating Battery Operating Cost. *IEEE Transactions on Power Systems*, *31*(3), 2289–2296. https://doi.org/10.1109/TPWRS.2015.2455491

Panda, D. K., Turner, O., Das, S., & Abusara, M. (2024). Prioritized experience replay based deep distributional reinforcement learning for battery operation in microgrids. *Journal of Cleaner Production*, *434*, 139947. https://doi.org/10.1016/j.jclepro.2023.139947

Panteli, M., Trakas, D. N., Mancarella, P., & Hatziargyriou, N. D. (2017). Power Systems Resilience Assessment: Hardening and Smart Operational Enhancement Strategies [Conference Name: Proceedings of the IEEE]. *Proceedings of the IEEE*, *105*(7), 1202–1213. https://doi.org/10.1109/JPROC.2017.2691357

Papathanassiou, S., Hatziargyriou, N., Strunz, K., et al. (2005). A benchmark low voltage microgrid network. *Proceedings of the CIGRE symposium: power systems with dispersed generation*, 1–8.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., … Chintala, S. (2019). Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, & R. Garnett (Eds.), *Advances in neural information processing systems* (Vol. 32). Curran Associates, Inc.

Phan, B. C., Lee, M.-T., & Lai, Y.-C. (2022). Intelligent Deep-Q-Network-Based Energy Management for an Isolated Microgrid. *Applied Sciences*, *12*(17), 8721. https://doi.org/10.3390/app12178721

Qiu, X., Nguyen, T. A., & Crow, M. L. (2016). Heterogeneous energy storage optimization for microgrids. *IEEE Transactions on Smart Grid*, *7*(3), 1453–1461. https://doi.org/10.1109/TSG.2015.2461134

Raffin, A. (2020). Rl baselines3 zoo.

Raffin, A., Hill, A., Ernestus, M., Gleave, A., Kanervisto, A., & Dormann, N. (2019). Stable baselines3 [Last access: 2024-12-02]. https://github.com/DLR-RM/stable-baselines3

Raffin, A., Hill, A., Gleave, A., Kanervisto, A., Ernestus, M., & Dormann, N. (2021). Stable-Baselines3: Reliable Reinforcement Learning Implementations. *Journal of Machine Learning Research*, *22*(22), 8.

Rengarajan, D., Vaidya, G., Sarvesh, A., Kalathil, D., & Shakkottai, S. (2022). Reinforcement Learning with Sparse Rewards using Guidance from Offline Demonstration. *arXiv:2202.04628 [cs]*. https://doi.org/10.48550/arXiv.2202.04628

Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, *65*(6), 386–408. https://doi.org/10.1037/h0042519

Sandeep, S. D., Mohanty, S., Mohanty, S. B., & Puhan, P. S. (2025). A comprehensive review on DC microgrid control and energy management strategies. *Results in Engineering*, *26*, 105479. https://doi.org/10.1016/j.rineng.2025.105479

Schaul, T., Quan, J., Antonoglou, I., & Silver, D. (2015). Prioritized Experience Replay. *ICLR 2016*. http://arxiv.org/abs/1511.05952

Seijen, H., & Sutton, R. (2014). True Online TD(lambda) [ISSN: 1938-7228]. *Proceedings of the 31st International Conference on Machine Learning*, 692–700. https://proceedings.mlr.press/v32/seijen14.html

Sethi, S., & Sorger, G. (1991). A theory of rolling horizon decision making. *Annals of Operations Research*, *29*(1), 387–415. https://doi.org/10.1007/BF02283607

Shakya, A. K., Pillai, G., & Chakrabarty, S. (2023). Reinforcement learning algorithms: A brief survey. *Expert Systems with Applications*, *231*, 120495. https://doi.org/10.1016/j.eswa.2023.120495

Shepherd, C. M. (1965). Design of Primary and Secondary Cells: II . An Equation Describing Battery Discharge. *Journal of The Electrochemical Society*, *112*(7). https://doi.org/10.1149/1.2423659

Shuai, H., Fang, J., Ai, X., Wen, J., & He, H. (2019). Optimal Real-Time Operation Strategy for Microgrid: An ADP-Based Stochastic Nonlinear Optimization Approach. *IEEE Transactions on Sustainable Energy*, *10*(2), 931–942. https://doi.org/10.1109/TSTE.2018.2855039

Shuai, H., & He, H. (2021). Online Scheduling of a Residential Microgrid via Monte-Carlo Tree Search and a Learned Model. *IEEE Transactions on Smart Grid*, *12*(2), 1073–1087.

Shuai, H., Li, F., Pulgar-Painemal, H., & Xue, Y. (2021). Branching Dueling Q-Network-Based Online Scheduling of a Microgrid With Distributed Energy Storage Systems. *IEEE Transactions on Smart Grid*, *12*(6), 5479–5482. https://doi.org/10.1109/TSG.2021.3103405

Silver, D., Hubert, T., Schrittwieser, J., Antonoglou, I., Lai, M., Guez, A., Lanctot, M., Sifre, L., Kumaran, D., Graepel, T., Lillicrap, T., Simonyan, K., & Hassabis, D. (2018). A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play. *Science*, *362*(6419), 1140–1144. https://doi.org/10.1126/science.aar6404

Srivastava, A., Mohanty, R., Ghazvini, M. A. F., Tuan, L. A., Steen, D., & Carlson, O. (2021). A Review on Challenges and Solutions in Microgrid Protection. *2021 IEEE Madrid PowerTech*, 1–6. https://doi.org/10.1109/PowerTech46648.2021.9495090

Staffell, I., & Pfenninger, S. (2016). Using bias-corrected reanalysis to simulate current and future wind power output. *Energy*, *114*, 1224–1239. https://doi.org/10.1016/j.energy.2016.08.068

Stefan Pfenninger, I. S. (2016). Renewables ninja. version: 1.1, coord: Lat. 39.459 - lon. -2.173, dates: 2028-01-01 - 2020-12-31, dataset: Merra2, capacity: 10kw, height: 80m, turbine: Ge 1.5.sle [Last access: 2023-11-27]. https://www.renewables.ninja/

Střelec, M., & Berka, J. (2013). Microgrid energy management based on approximate dynamic programming. *IEEE PES ISGT Europe 2013*, 1–5. https://doi.org/10.1109/ISGTEurope.2013.6695439

Sukumar, S., Mokhlis, H., Mekhilef, S., Naidu, K., & Karimi, M. (2017). Mix-mode energy management strategy and battery sizing for economic operation of grid-tied microgrid. *Energy*, *118*, 1322–1333. https://doi.org/10.1016/j.energy.2016.11.018

Sulzer, V., Marquis, S. G., Timms, R., Robinson, M., & Chapman, S. J. (2021). Python Battery Mathematical Modelling (PyBaMM). *Journal of Open Research Software*, *9*(1). https://doi.org/10.5334/jors.309

Sutton, R. S. (1990). Integrated Architectures for Learning, Planning, and Reacting Based on Approximating Dynamic Programming. *In Proceedings of the Seventh International Conference on Machine Learning*, 216–224.

Sutton, R. S. (1995). Generalization in reinforcement learning: Successful examples using sparse coarse coding. *Proceedings of the 8th International Conference on Neural Information Processing Systems*, 1038–1044.

Sutton, R. S., McAllester, D. A., Singh, S. P., & Mansour, Y. (2000). Policy Gradient Methods for Reinforcement Learning with Function Approximation. In S. A. Solla, T. K. Leen, & K. Müller (Eds.), *Advances in Neural Information Processing Systems 12* (pp. 1057–1063). MIT Press. Retrieved March 16, 2020, from http://papers.nips.cc/paper/1713-policy-gradient-methods-for-reinforcement-learning-with-function-approximation.pdf

Sutton, R. S., & Barto, A. G. (2018). *Reinforcement learning: An introduction* (Second). The MIT Press. http://incompleteideas.net/book/the-book.html

Tang, C., Abbatematteo, B., Hu, J., Chandra, R., Martín-Martín, R., & Stone, P. (2025). Deep Reinforcement Learning for Robotics: A Survey of Real-World Successes. *Proceedings of the AAAI Conference on Artificial Intelligence*, *39*(27), 28694–28698. https://doi.org/10.1609/aaai.v39i27.35095

Tang, J., & Abbeel, P. (2010). On a Connection between Importance Sampling and the Likelihood Ratio Policy Gradient. *NIPS'10*, 1000–1008.

Tavakoli, A., Pardo, F., & Kormushev, P. (2018). Action Branching Architectures for Deep Reinforcement Learning [Number: 1]. *Proceedings of the AAAI Conference on Artificial Intelligence*, *32*(1). https://doi.org/10.1609/aaai.v32i1.11798

Ton, D. T., & Smith, M. A. (2012). The U.S. Department of Energy's Microgrid Initiative. *The Electricity Journal*, *25*(8), 84–94. https://doi.org/10.1016/j.tej.2012.09.013

Towers, M., Terry, J. K., Kwiatkowski, A., Balis, J. U., Cola, G. d., Deleu, T., Goulão, M., Kallinteris, A., KG, A., Krimmel, M., Perez-Vicente, R., Pierré, A., Schulhoff, S., Tai, J. J., Shen, A. T. J., & Younis, O. G. (2023, March). Gymnasium. https://doi.org/10.5281/zenodo.8127026

Tremblay, O., & Dessaint, L.-A. (2009). Experimental Validation of a Battery Dynamic Model for EV Applications. *World Electric Vehicle Journal*, *3*(2), 289–298. https://doi.org/10.3390/wevj3020289

Tumilowicz, N. (2024, December). Microgrids: Enhancing Grid Resilience and Shaping the Future of Energy Distribution. Retrieved August 13, 2025, from https://www.microgridknowledge.com/microgrids/utility/article/55243732/microgrids-enhancing-grid-resilience-and-shaping-the-future-of-energy-distribution

van Hasselt, H., Guez, A., & Silver, D. (2016). Deep Reinforcement Learning with Double Q-Learning [event-place: Phoenix, Arizona]. *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, 2094–2100.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017, June). Attention Is All You Need. https://doi.org/10.48550/arXiv.1706.03762

Venkataramanan, G., & Marnay, C. (2008). A larger role for microgrids. *IEEE Power and Energy Magazine*, *6*(3), 78–82. https://doi.org/10.1109/MPE.2008.918720

Vilaisarn, Y., Rodrigues, Y. R., Abdelaziz, M. M. A., & Cros, J. (2022). A Deep Learning Based Multiobjective Optimization for the Planning of Resilience Oriented Microgrids in Active Distribution System. *IEEE Access*, *10*, 84330–84364. https://doi.org/10.1109/ACCESS.2022.3197194

Walth, M., Sajadi, A., Carbone, M., & Hodge, B.-M. (2024). Lunar power grid: Network structure and spontaneous synchronization. https://arxiv.org/abs/2404.06374

Wang, H., Ye, Y., Zhang, J., & Xu, B. (2023). A comparative study of 13 deep reinforcement learning based energy management methods for a hybrid electric vehicle. *Energy*, *266*, 126497. https://doi.org/10.1016/j.energy.2022.126497

Wang, Z., Bapst, V., Heess, N., Mnih, V., Munos, R., Kavukcuoglu, K., & Freitas, N. d. (2017). Sample Efficient Actor-Critic with Experience Replay [arXiv: 1611.01224]. *ICLR 2017 Conference*. http://arxiv.org/abs/1611.01224

Wang, Z., Schaul, T., Hessel, M., Hasselt, H., Lanctot, M., & Freitas, N. (2016). Dueling Network Architectures for Deep Reinforcement Learning [ISSN: 1938-7228]. *Proceedings of The 33rd International Conference on Machine Learning*, 1995–2003. https://proceedings.mlr.press/v48/wangf16.html

Watkins, C. J. C. H., & Dayan, P. (1992). Q-learning. *Machine Learning*, *8*(3), 279–292. https://doi.org/10.1007/BF00992698

Xu, H., Cai, X., Shu, J., & Lu, J. (2021). Influence of Discrete and Continuous Action Spaces on Deep Reinforcement Learning-Based Pricing Strategy Optimization for Electricity Retailers. *2021 IEEE Sustainable Power and Energy Conference (iSPEC)*, 3843–3848. https://doi.org/10.1109/iSPEC53008.2021.9735962

Xu, P., & Gu, Q. (2020). A Finite-Time Analysis of Q-Learning with Neural Network Function Approximation [ISSN: 2640-3498]. *Proceedings of the 37th International Conference on Machine Learning*, 10555–10565. https://proceedings.mlr.press/v119/xu20c.html

Yang, D., Zhao, L., Lin, Z., Qin, T., Bian, J., & Liu, T.-Y. (2019). Fully Parameterized Quantile Function for Distributional Reinforcement Learning. *Advances in Neural Information Processing Systems*, *32*. Retrieved November 1, 2024, from https://proceedings.neurips.cc/paper_files/paper/2019/hash/f471223d1a1614b58a7dc45c9d01df19-Abstract.html

Zhang, S., Li, H., Wang, M., Liu, M., Chen, P.-Y., Lu, S., Liu, S., Murugesan, K., & Chaudhury, S. (2023). On the Convergence and Sample Complexity Analysis of Deep Q-Networks with $ $-Greedy Exploration. https://doi.org/10.48550/arXiv.2310.16173

Zhang, Y., Zhu, R., Song, X., & Yang, P. (2025). Distributed operational optimization of multi-energy microgrid clusters considering system robustness. *Electric Power Systems Research*, *248*, 111990. https://doi.org/10.1016/j.epsr.2025.111990

Zolman, N., Fasel, U., Kutz, J. N., & Brunton, S. L. (2024). SINDy-RL: Interpretable and Efficient Model-Based Reinforcement Learning. https://doi.org/10.48550/arXiv.2403.09110