



POLITECNICO DI MILANO
DIPARTIMENTO DI ELETTRONICA, INFORMAZIONE E BIOINGEGNERIA
DOCTORAL PROGRAMME IN INFORMATION TECHNOLOGY

EMPOWERING DIGITAL TWINS FOR BATTERY ENERGY
STORAGE SYSTEMS WITH MACHINE LEARNING: DESIGN,
DEVELOPMENT AND APPLICATIONS
OF THE FRAMEWORK

Doctoral Dissertation of:
Davide Salaorni

Supervisor:
Prof. Marcello Restelli

Co-Supervisor:
Prof. Francesco Trovò

Tutor:
Prof. Cinzia Cappiello

The Chair of the Doctoral Program:
Prof. Luigi Piroddi

2025 – Cycle XXXVIII

Acknowledgements

While the PhD path is conceived to be a solo journey, I would never have made it to the end without the support of several people who have, in their own personal ways, made the toughest achievement of my life so far a little lighter. Below, I'd like to say my "thank you" to them.

First and foremost, I want to thank my supervisors Marcello and Francesco, who guided and advised me with their great expertise and showed me the path to being a (hopefully good) researcher. Most importantly, they surprisingly believed in me even when I was not that sure of myself, and that meant a lot to me.

I'm particularly grateful to Federico, my exceptional mentor from RSE, together with Silvia and Andrea, who have always been very supportive and available to help me with meaningful suggestions and intense data collections. At RSE, I found very kind people and a great TGM team. Despite some pizza-related inconveniences, I had a great time with them (Andrea, take care of ErNESTO for me!).

A huge "thank you" goes to my colleagues of the RL3 group and the people at AIRLab whom I had the pleasure of meeting over these years. I genuinely think they are some of the smartest young researchers in Italy at the moment, and I never would have expected that at some point I'd be a part of this group (even as a *pirate*).

Throughout these years, Milano was my home, and I would like to mention the tons of people who made me love this city. For the sake of simplicity, I'll do it by clustering: thank you to my friends from PaoloVI, PoliMi, AGS, and soft skill PhD lectures. Some of them deserve a special mention, but I do not want to be too pedantic, and I bet they know how precious they have been. I also want to thank my historical friends from Verona (both "alte" and "basse") who, even after such a long time away from Veneto, still make me feel at home.

Finally, my thanks go to my wonderful family. While they are probably still wondering what the hell I am doing at 30 (without a *real* job), they have always been very supportive. In particular, I want to thank mum and dad for their open-mindedness and for demonstrating their willingness to question themselves for the sake of their loved ones; I'm very proud of them for this. A special "thank you" goes to my sister Giorgia, for her bravery and for cheering me on every time, even from far away; I know I'm not really good at showing it, but it means a lot to me. Last but certainly not least, my thanks

go to Anna, my anchor and my rock, the person who probably deserves a PhD just for standing by me, and who makes me feel loved every single day. To her, I promise: my academic studies end here.

While this may seem like a mere list of names, my thanks are sincere, and these people have probably done more for me than they realize. In this stream of consciousness that I am pasting into this acknowledgment section, I hope I have remembered everyone, but if I missed someone it was not intentional, and I thank them as well.

I am genuinely grateful for this experience which, while tough and heavy at times, taught me the power of grasping the true meaning of things. So again, thank you all, I really mean it!

Abstract

THE GLOBAL TRANSITION toward sustainable energy solutions heavily relies on the widespread deployment of energy storage systems. By mitigating the intermittency of renewable energy sources, batteries have become the cornerstone of modern, decentralized power grids. Essentially, batteries act as energy buffers, storing excess renewable generation and discharging to meet demand when generation falls short. However, optimizing their performance, safety, and longevity remains a critical challenge with profound implications for the entire energy ecosystem.

In this context, *digital twins* have emerged as a transformative paradigm. By bridging the physical and digital worlds, digital twins provide high-fidelity virtual replicas of real-world systems, enabling risk-free *what-if* simulations and extensive offline analysis. Furthermore, when coupled with recent advancements in machine learning, digital twins unlock unprecedented capabilities for real-time monitoring and autonomous control, leveraging a virtually unlimited supply of simulated data.

In this dissertation, we explore the intersection of digital twins and the energy storage domain, with a particular focus on integrating advanced machine learning techniques to enhance system monitoring and control. Specifically, we investigate the adoption of a digital twin for a battery storage system, spanning from its core design and implementation to its application across multiple replicas to simulate a renewable energy community. Within this framework, machine learning serves as a pivotal component applied in three distinct forms. First, we employ unsupervised learning and drift detection techniques to enable the digital twin to dynamically adapt to the evolving physical conditions of the battery. Second, we frame the energy management problem of a microgrid as a Markov Decision Process, adopting a reinforcement learning approach to optimize the battery's operational policy. Finally, we extend the analysis to a renewable energy community, proposing a novel multi-agent approach to drive the behavior of self-interested community members toward global cooperation and coordination.

The proposed methodologies are rigorously validated through an extensive experimental campaign using real-world datasets, facilitated by the collaboration with RSE S.p.A., a leading Italian research institution in the energy sector. Ultimately, this research aims to contribute to the advancement of the energy field by demonstrating the immense potential of integrating digital twins with learning systems to foster a more efficient, sustainable, and resilient energy future.

Sommario

LA TRANSIZIONE GLOBALE verso soluzioni energetiche sostenibili si sta dimostrando fortemente legata alla diffusione capillare di sistemi di accumulo. Riuscendo a mitigare la natura intermittente delle fonti energetiche rinnovabili, le batterie sono diventate un pilastro fondamentale delle moderne reti elettriche sempre più decentralizzate. Di fatto, questi dispositivi fungono da veri e propri buffer energetici, immagazzinando energia derivata dalla generazione rinnovabile in eccesso, per poi scaricarsi e soddisfare la domanda quando la produzione risulta insufficiente. Tuttavia, l'ottimizzazione delle prestazioni, della sicurezza e della longevità delle batterie rimane una sfida cruciale, la cui rilevanza impatta l'intero ecosistema energetico.

In questo contesto, i *digital twin* (gemelli digitali) si sono affermati come un paradigma fortemente trasformativo. Andando a connettere il mondo fisico con quello digitale, i digital twin si presentano come repliche virtuali altamente accurate dei sistemi reali, consentendo *simulation what-if* in situazioni di rischio che comprometterebbero il sistema fisico e approfondite analisi offline. Inoltre, la loro integrazione con i recenti progressi nel campo del Machine Learning li fornisce di capacità senza precedenti per il monitoraggio in tempo reale e il controllo autonomo, grazie anche alla disponibilità virtualmente illimitata di dati simulati su cui addestrare i modelli.

In questa tesi esploreremo l'intersezione tra i digital twin e i sistemi energetici, con un focus particolare sull'utilizzo di tecniche avanzate di Machine Learning per migliorare il monitoraggio e il controllo del sistema. Nello specifico, ci concentreremo su un digital twin di un sistema di accumulo, partendo dalla fase di progettazione e implementazione, fino all'utilizzo di sue molteplici repliche per simulare una Comunità Energetica Rinnovabile. In questo scenario, il Machine Learning funge da componente cardine, declinato in tre forme distinte. In primo luogo, impiegheremo tecniche di *unsupervised learning* e di *drift detection* per consentire al digital twin di adattarsi dinamicamente alle mutevoli condizioni fisiche della batteria. In secondo luogo, formuleremo il problema di gestione ottima di una microrete sottoforma di Processo Decisionale di Markov, adottando tecniche di *reinforcement learning* per ottimizzare la politica operativa della batteria. Infine, estenderemo l'analisi a una Comunità Energetica Rinnovabile, proponendo un innovativo approccio multi-agente per guidare il comportamento dei membri della comunità, naturalmente mossi da interessi individuali, verso una cooperazione e un coordinamento che benefici l'intera rete.

Nel corso della tesi, valideremo le metodologie proposte attraverso un'ampia campagna sperimentale basata su dataset reali, da è stata favorita dalla collaborazione con RSE S. p. A., un istituto di ricerca italiano leader nel settore energetico. In definitiva, questa ricerca mira a contribuire al progresso del settore energetico, dimostrando il potenziale derivante dall'integrazione tra digital twin e Machine Learning per promuovere un futuro più efficiente, sostenibile e resiliente.

Contents

List of Tables	IX
List of Figures	XI
List of Algorithms	XIII
List of Symbols and Acronyms	XV
1 Introduction	1
1.1 Definition of Digital Twin	2
1.2 Why Machine Learning for Digital Twins?	3
1.3 Digital Twins for Energy Systems	5
1.3.1 Design and Implementation of Digital Twins	5
1.3.2 Real-Time Monitoring and Adaptation to Physical Systems	6
1.3.3 Control and Optimization of Microgrids	7
1.3.4 Energy Management for Renewable Energy Communities	8
1.4 Research Objectives and Original Contributions	9
1.4.1 Research Objectives	9
1.4.2 Original Contributions	10
1.5 Overview	11
I Design and Validation of the Digital Twin Framework	15
2 Design of the Digital Twin Framework	17
2.1 Design Philosophy: Modularity and Integration	18
2.1.1 The ERNESTO Framework	19
2.1.2 Architectural Components	19
2.2 The Electrical Model	20
2.2.1 First-order Thevenin Model	20
2.2.2 Second-order Thevenin model	21
2.3 The Thermal Model	21

Contents

2.4	State of Charge Estimation	22
2.5	State of Health Estimation	22
2.6	Interactions Among Sub-Models	24
2.7	High-level Controller Integration	25
3	Digital Twin Validation	27
3.1	Model Parametrization	27
3.1.1	Electrical Model Parameters	28
3.1.2	Thermal Model Parameters	28
3.2	Case Studies for Validation	31
3.2.1	Residential Prosumer Profile	32
3.2.2	Microgrid Integration Profile	33
3.2.3	Thermal Stress Test Profile	35
3.2.4	Aging Validation Profile	37
II	Machine Learning Integration for Real-Time Monitoring and Adapta-	
	tion	41
4	Monitoring and Tracking of Energy Storage Systems	43
4.1	Online Monitoring and Drift Detection	44
4.1.1	Concept Drifts	44
4.1.2	Robust Anomaly Detection	46
4.1.3	Drift and Anomalies in Battery Systems	46
4.2	Existing Approaches	48
4.2.1	Parameter Identification and Online Adaptation	48
4.2.2	Adaptive Digital Twin Frameworks	48
4.2.3	Fault Detection and Novelty Identification	49
5	Cluster-based Real-Time Battery Tracking	51
5.1	Fundamentals of Online Learning	52
5.1.1	Drift Detection Methodologies	52
5.1.2	Adaptive Learning Strategies	53
5.2	Proposed Methodology	54
5.2.1	Monitoring the Parameter Space	54
5.2.2	The Fault Diagnosis Framework	54
5.2.3	The Proposed Algorithm	56
5.2.4	Discussion on Adaptation Dynamics	60
6	Experimental Evaluation	63
6.1	Experimental Setup	63
6.1.1	Operational Regimes Identification	64
6.1.2	Baselines	66
6.1.3	Scenarios Configuration	66
6.2	Case Studies	69
6.2.1	First Synthetic Scenario: Single Thermal Jump	70
6.2.2	Second Synthetic Scenario: Multiple Thermal Jumps	72
6.2.3	Real-World Scenario: Single Thermal Jump	74

III	Microgrid Energy Management through Reinforcement Learning	79
7	Energy Management in Microgrids	81
7.1	A Multi-Layered Control Scheme	82
7.1.1	Microgrid Topologies	82
7.1.2	Hierarchical Control Framework	82
7.1.3	Control Paradigms	83
7.2	Problem Formulation	84
7.2.1	Power Flow Model	85
7.2.2	Resource Modeling	86
7.2.3	Bidirectional Exchanges with the Main Grid	90
7.3	Existing Approaches	90
7.3.1	MPC-based Approaches	91
7.3.2	RL-based Approaches	94
7.3.3	Hybrid Approaches	95
8	A Reinforcement Learning Approach	97
8.1	Fundamentals of Reinforcement Learning	98
8.1.1	Introduction to Markov Processes	99
8.1.2	The Control Policy	99
8.1.3	Value Functions	100
8.1.4	From Tabular Methods to Function Approximation	101
8.1.5	Deep Reinforcement Learning	101
8.1.6	Proximal Policy Optimization (PPO)	102
8.2	RL-based Energy Management	103
8.2.1	Microgrid Setting	103
8.2.2	Physical Constraints	105
8.2.3	Controller's Objective	106
8.3	MDP Formulation	107
8.3.1	Observation Space	107
8.3.2	Action Space	108
8.3.3	Reward Function	108
8.4	Training Procedure	109
8.4.1	Exogenous and Endogenous Inputs	109
8.4.2	Algorithm and Interaction Loop	110
9	Experimental Evaluation	113
9.1	Experimental Setup	113
9.1.1	Datasets and Simulation Environment	113
9.1.2	Baseline Strategies	114
9.2	Case Study	115
9.2.1	Real-World Scenario Results	115
9.2.2	Sensitivity Analysis: Adaptability to Cost Shifts	118
9.3	Ablation Studies	119

IV Multi-Agent Reinforcement Learning for Renewable Energy Communities	121
10 Energy Management in Renewable Energy Communities	123
10.1 Problem Formulation	124
10.1.1 Node Level Formalization	124
10.1.2 Community Level Formalization	126
10.2 Existing Approaches	127
10.2.1 Rule-Based and Optimization Approaches	127
10.2.2 Centralized Reinforcement Learning	128
10.2.3 Multi-Agent Systems and Game Theoretic Approaches	128
11 A Multi-Agent Reinforcement Learning Approach	129
11.1 Fundamentals of Multi-Agent Reinforcement Learning	130
11.1.1 Game Theoretic Structures	131
11.1.2 The Multi-Agent Interaction Loop	131
11.1.3 Reduction to Single Agent	132
11.1.4 Challenges in Multi-Agent Reinforcement Learning	133
11.1.5 Training and Execution Modes	134
11.1.6 Opponent-Aware Learning	135
11.2 MARL-based Energy Community Management	136
11.2.1 Two-Phase Stochastic Game Formulation	136
11.2.2 Node Agent	137
11.2.3 REC Agent	139
11.3 The Proposed Solution: INAIA	140
11.3.1 The INAIA Algorithm	141
12 Experimental Evaluation	143
12.1 Experimental Setup	143
12.1.1 Datasets and Simulation Environment	144
12.1.2 Baseline Strategies	144
12.1.3 Evaluation Metrics	145
12.2 Case Studies	145
12.2.1 First Scenario: Scaling Number of Node Agents	146
12.2.2 Second Scenario: Demonstrating the Convenience of Joining the REC	148
12.3 Ablation Study	149
13 Discussion and Conclusion	153
13.1 Design and Validation of the Digital Twin Framework	154
13.2 Machine Learning for Real-Time Monitoring and Adaptation	154
13.3 Microgrid Energy Management through Reinforcement Learning	155
13.4 Multi-Agent Control for Renewable Energy Communities	155
Bibliography	157

Appendices	171
A Digital Twin Validation: Additional Details	173
A.0.1 DROPFLOW Performance Note	173
B Real-Time Monitoring and Adaptation: Additional Details	175
C Microgrid Energy Management: Additional Details	185
D Renewable Energy Communities Management: Additional Details	187
D.1 Incentives Structure	187
D.2 Additional Experimental Details	188
D.3 Network architectures and hyperparameters	188
D.3.1 Node agents	188
D.3.2 REC agent	188

List of Tables

3.1 DT framework validation results	33
3.2 Aging validation results	39
6.1 Performance on single thermal jump scenario	71
6.2 Performance on multiple thermal jumps	73
6.3 Performance on real-world scenario	76
12.1 Scalability results on varying number of agents	148
12.2 Fairness evaluation	149
A.1 EIG ePLB-C020 cell characteristics	173
A.2 Parameters for DT validation	174
B.1 Configuration parameters for real-time monitoring on the synthetic scenarios	176
B.2 Configuration parameters for real-time monitoring on the real-world scenario	176
C.1 Battery pack parameters	185
C.2 PPO parameters on MicrogridEnv	185
D.1 Scalability results on varying number of agents	188
D.2 Node agents training hyperparameters	188
D.3 REC agent training hyperparameters	190

List of Figures

2.1	DT architecture	20
2.2	First-order Thevenin model	21
2.3	Second-order Thevenin model	21
2.4	Thermal model	22
2.5	High-level controller integration	25
3.1	V_{OC} characterization points	29
3.2	Parameters V_{OC} and R_0	29
3.3	Parameters R_1 and C_1	30
3.4	Parameters R_1 , C_1 , R_2 and C_2	31
3.5	Parameter $\partial V_{OC}/\partial T$ versus SOC.	32
3.6	Temperature transient evolution.	32
3.7	PV profile validation	34
3.8	DER-TF profile validation	36
3.9	TJ profile validation	37
3.10	Aging validation: OCV curve	38
4.1	Taxonomy of concept drifts	45
5.1	Evolving cluster-based fault diagnosis system	55
6.1	DBSCAN identified clusters	64
6.2	Domain regions identified by DBSCAN	65
6.3	Current profile for synthetic scenarios	67
6.4	Temperature profiles for thermal jump scenarios	67
6.5	Voltage profiles for thermal jump scenarios	68
6.6	Current and temperature profiles for real-world scenario	69
6.7	Voltage tracking in single thermal jump scenario	70
6.8	Voltage prediction error in single thermal jump scenario	71
6.9	Cluster evolution over time in single thermal jump scenario	72
6.10	Voltage tracking in multiple thermal jumps scenario	73

List of Figures

6.11 Voltage prediction error in multiple thermal jumps scenario	74
6.12 Cluster evolution over time in multiple thermal jumps scenario	75
6.13 Voltage tracking in real-world scenario	76
6.14 Voltage prediction error in real-world scenario	77
6.15 Cluster evolution over time in real-world scenario	78
7.1 Three level hierarchical control scheme of MGs	83
7.2 Control paradigms of MGs	84
7.3 Microgrid power flow scheme	85
8.1 Markov Decision Process	100
8.2 RL controller interaction within MG	104
9.1 Cumulative average empirical reward	116
9.2 Gaps for trading and degradation components of the reward	117
9.3 Heatmap of action vs. demand	118
9.4 Empirical return distribution across validation profiles	119
9.5 Sensitivity analysis on energy price	119
9.6 Sensitivity analysis on replacement cost	120
9.7 Ablation study on the clipping term	120
10.1 REC scheme	125
11.1 MARL interaction with environment	133
12.1 First scenario: learning curves on self-consumption and social welfare.	147
12.2 Return comparison with and without REC	151
12.3 Ablation of the deterministic node actions	152
A.1 RSE lab setup	174
B.1 Zoomed-in view of the voltage signal in the single thermal jump scenario	177
B.2 Zoomed-in view of the voltage signal in the multiple thermal jumps scenario	177
B.3 Zoomed-in view of the voltage signal in the real-world scenario	178
B.4 Parameter evolution in single thermal jump synthetic scenario	179
B.5 Parameter evolution in multiple thermal jumps synthetic scenario	180
B.6 Parameter evolution in the real-world scenario	181
B.7 Cluster evolution in single thermal jump scenario	182
B.8 Cluster evolution in multiple thermal jumps scenario	183
B.9 Cluster evolution in real-world scenario	184
D.1 Energy consumption yearly profiles	189

List of Algorithms

1	ERNESTO's Evolving Clustering Routine	56
2	Online Model Adaptation	59
3	PPO-based Microgrid Control	110
4	Pseudocode of INAIA algorithm	141

List of Symbols and Acronyms

Mathematical Notation

$\mathbb{R}$	The set of real numbers
$\mathbb{R}^n$	n -dimensional real-valued vector space
$\mathbb{R}^{m \times n}$	Space of real-valued matrices with m rows and n columns
$\mathcal{X}$	Calligraphic uppercase letters denote sets or spaces
$ \mathcal{X} $	Cardinality of the set $\mathcal{X}$
x	Regular italic lowercase letters denote scalars
$\boldsymbol{x}$	Bold italic lowercase letters denote column vectors
$\boldsymbol{X}$	Bold uppercase letters denote matrices
$\boldsymbol{x}^\top, \boldsymbol{X}^\top$	Transpose of a vector or a matrix
x_i	The i -th element of vector $\boldsymbol{x}$
$\ \boldsymbol{x}\ _2$	Euclidean (L_2) norm of vector $\boldsymbol{x}$
$ x $	Absolute value of a scalar x
$:=$	Defined as (equality by definition)
$\hat{x}$	Hat denotes an estimated, predicted, or approximated value of x
x^*	Asterisk denotes an optimal value or target value of x
Δx	Difference or change in variable x
$\arg \min_x$	The argument x that minimizes the objective function
$\arg \max_x$	The argument x that maximizes the objective function
$\mathbb{P}(x)$	Probability of event x
$\mathbb{E}[X]$	Expected value of random variable X
$P(x y)$	Conditional probability of x given y
$x \sim \mathcal{D}$	Variable x is sampled from distribution $\mathcal{D}$
$\nabla f(x)$	Gradient of function f with respect to x

List of Symbols

v	Terminal voltage of the battery cell [V]
i	Applied current, positive for discharge [A]

List of Algorithms

V_{oc}	Open-circuit voltage (OCV) [V]
R_0	Series internal (ohmic) resistance [Ω]
R_1, R_2	Polarization (charge-transfer and diffusion) resistances [Ω]
C_1, C_2	Polarization capacitances [F]
τ	Time constant of the RC branch, defined as $R_1 C_1$ [s]
σ	State of charge of the battery cell [%]
ρ	State of health of the battery cell [%]
T_{core}	Core temperature of the battery cell [K]
T_{surf}	Surface temperature of the battery cell [K]
T_{amb}	Ambient temperature [K]
ϑ	Temperature associated with the battery cell [K]
C_{th}	Thermal capacitance of the battery [J/K]
R_{cond}	Thermal conduction resistance [K/W]
R_{conv}	Thermal convection resistance [K/W]
Q_{gen}	Heat generation rate inside the cell [W]
W	Size of the moving data window (number of time steps)
Ψ_k	The k -th nominal operational state
μ	Mean of a distribution
Σ	Covariance matrix of a distribution
n	Number of samples of a cluster
t	Time step index
$\mathcal{Z}_W$	Fault-free training dataset
Θ	Set of parameter vectors
ϕ	Generic model parameter vector
$\hat{\theta}$	Estimated physical parameter vector
ξ	Regularization hyperparameter controlling parameter inertia
D_M	Mahalanobis distance
$\mathcal{O}$	Buffer storing statistical outliers for drift detection
y_i	Measured system output (e.g., actual voltage)
$\hat{y}_i$	Predicted system output generated by the Digital Twin
H_0, H_1	Null and alternative hypotheses
$\Omega(\cdot)$	Potential density function
$\ \cdot\ _\lambda$	Spatio-temporal norm
$\mathcal{S}$	State space of the environment
$\mathcal{A}$	Action space available to the agent
P	State transition probability function
R	Reward function
γ	Discount factor
μ_0	Initial state distribution
π	Policy mapping observations to actions
V^π	State-value function under policy π
Q^π	Action-value function under policy π
A^π	Advantage function under policy π
s_t	State of the environment at time step t
a_t	Action taken by the agent at time step t

$r_{:,t}$	Specific reward component at time step t
$\mathcal{N}$	Set of all nodes (agents) within the Renewable Energy Community
N	Total number of nodes, $N = \mathcal{N} $
$\mathfrak{C}$	Centralized coordinator of the Renewable Energy Community
$\mathcal{N}_A$	Subset of actively participating nodes
$\mathcal{N}_P$	Subset of passively participating nodes
$\mathcal{O}_n$	Observation space for agent n in a POSG
$a_{n,t}$	Action taken by agent n at time step t
$r_{n,t}$	Instantaneous local reward for agent n at time t
R_n	Cumulative return or global reward function for agent n
U_t	Self-consumption at time step t
$\mathcal{W}$	Social welfare of the community
ω_t	Incentive distribution vector at time step t
$V_t^{\mathcal{N}_A}$	Portion of incentives distributed among active nodes [€]
$V_t^{\mathcal{N}_P}$	Portion of incentives distributed among passive nodes [€]
$r_{n,t}^{\text{REC}}$	Reward obtained by an independent node n not joining the REC
$\Delta r_{n,t}^{\text{loc}}$	Difference in local reward due to REC participation
$\Delta r_{n,t}$	Difference in total reward due to REC participation

List of Acronyms

AC	Alternating Current
AE	Autoencoder
AI	Artificial Intelligence
ANN	Artificial Neural Network
BESS	Battery Energy Storage System
BOL	Beginning Of Life
BMS	Battery Management System
CHP	Combined Heat and Power
CPU	Central Processing Unit
DBSCAN	Density-Based Spatial Clustering of Application with Noise
DC	Direct Current
DER	Distributed Energy Resource
DGU	Distributed Generation Unit
DNN	Deep Neural Network
DQN	Deep Q-Network
DRL	Deep Reinforcement Learning
DSO	Distribution System Operator
DT	Digital Twin
ECM	Equivalent Circuit Model
EKF	Extended Kalman Filter
EMS	Energy Management System
EOL	End Of Life
FDD	Fault Detection and Diagnosis
KS	Kolmogorov-Smirnov (test)

List of Algorithms

IoT	Internet of Things
LP	Linear Programming
LSTM	Long Short-Term Memory
LUT	Look-Up Table
MAE	Mean Absolute Error
MaAE	Maximum Absolute Error
MAPE	Mean Absolute Percentage Error
MARL	Multi-Agent Reinforcement Learning
MCD	Minimum Covariance Determinant
MDP	Markov Decision Process
MG	Microgrid
MILP	Mixed-Integer Linear Programming
ML	Machine Learning
MPC	Model Predictive Control
MSE	Mean Squared Error
OL	Online Learning
P2P	Peer-To-Peer
PCA	Principal Component Analysis
PCC	Point of Common Coupling
PPO	Proximal Policy Optimization
PRBS	Pseudo-Random Binary Sequence
PSO	Particle Swarm Optimization
PV	Photovoltaic
REC	Renewable Energy Community
RES	Renewable Energy Source
RL	Reinforcement Learning
RMSE	Root Mean Squared Error
RSE	Ricerca sul Sistema Energetico
RUL	Remaining Useful Life
SOC	State of Charge
SOE	State of Energy
SOH	State of Health
SG	Smart Grid
TCL	Thermostatically Controllable Load

CHAPTER 1

Introduction

Over the last few decades, the global energy landscape has undergone a profound transformation. The depletion of fossil fuels, coupled with growing environmental concerns, has accelerated the transition towards cleaner and more sustainable energy sources. Concurrently, the proliferation of digital technologies has catalyzed a paradigm shift in energy production, democratizing the sector and empowering consumers to evolve into active participants within the ecosystem. Whereas the legacy energy model relied on large-scale, centralized power generation, the contemporary landscape is characterized by the widespread deployment of distributed energy resources (DERs), such as rooftop photovoltaic (PV) systems, small-scale wind turbines, and energy storage systems. This evolution has effectively integrated the end-user into the loop of energy management, giving rise to the *prosumer* concept, meaning a consumer who also produces energy, often through local renewable sources. While this transition enables greater flexibility, grid resilience, and sustainability, it introduces significant operational challenges, necessitating innovative methodologies to manage grid complexity, ensure reliability, and optimize resource allocation.

Not coincidentally, the transformation of the energy sector is intrinsically linked with the digital revolution, which is characterized by the proliferation of advanced technologies such as the Internet of Things (IoT), big data analytics, cloud computing, and artificial intelligence (AI). These technologies are enabling new ways to monitor, analyze, and control energy systems, leading to improved efficiency and reduced costs. Consequently, the emergence of *smart grids* (SGs) represents a significant milestone in the evolution of energy systems.

SGs leverage digital technologies to achieve a more intelligent and responsive grid infrastructure, capable of adapting to stochastic renewable generation and optimizing power flows in real-time. Structurally, an SG is typically decomposed into smaller sub-

systems, known as *microgrids* (MGs), i.e., localized energy networks that can operate independently or in conjunction with the main grid. MGs integrate various DERs, energy storage systems, and controllable loads, enabling enhanced energy management and resilience at the local level. Furthermore, the SG architecture is fostering the development of *renewable energy communities* (RECs), which are collaborative networks of prosumers who collectively generate, consume, and share renewable energy resources (RESs). RECs promote local energy autonomy, social engagement, and environmental sustainability, aligning with broader societal goals of decarbonization and energy democratization.

From the perspective of this new energy system structure, the increasing flexibility and dynamism of SGs, MGs, and RECs necessitate advanced tools and methodologies to effectively managing the underlying complexity. Among the various digital technologies that have emerged to address these challenges, digital twins (DTs) have gained significant attention. DTs are powerful tools conceived to bridge the physical and digital worlds, providing a virtual representation of real-world systems that can be used for simulation, optimization, and decision-making. Moreover, the integration with machine learning (ML) techniques is a core feature of these frameworks. By leveraging ML algorithms, DTs can learn from historical data, adapt to changing conditions, and make predictions about future behavior. This synergy between DTs and ML has the potential to revolutionize various industries, including manufacturing, healthcare, transportation, and energy management (Tao et al., 2018; Lo et al., 2021; Zhang et al., 2022; Murillo et al., 2023).

1.1 Definition of Digital Twin

The conceptual roots of the *digital twin* can be traced back to NASA in the 1960s during the Apollo program. To manage assets in space where physical access was impossible, NASA utilized *physical* replicas of the space vehicles, called *twins*. Twins were employed both for training during flight preparation and to solve crucial situations during space missions. Notably, this approach proved critical during the Apollo 13 mission (1970); following the oxygen tank explosion, engineers used these ground-based simulators to model failure scenarios and validate rescue procedures before providing instructions to the crew (Rosen et al., 2015).

The theoretical conceptualization of DTs is detailed in the seminal whitepaper by Grieves (2014). Grieves first articulated the architecture in 2003 during an executive course on Product Lifecycle Management at the University of Michigan. At the time, Grieves introduced the concept as a "Mirrored Spaces Model", proposing a construct consisting of a physical system, a virtual system, and the information flow linking the two. It was not until a decade later that the specific nomenclature *digital twin* was coined, resulting from Grieves' collaboration with John Vickers at NASA (Grieves, 2011). Despite this foundational work, the concept of DT gained significant attention only after its formal operationalization and widespread adoption by NASA researchers in 2012. In their landmark report, Glaessgen and Stargel (2012) described the DT as «integrated multiphysics, multiscale simulation of a vehicle or system that uses the best available physical models, sensor updates, fleet history, etc., to mirror the life of its corresponding flying twin».

Since then, the DT paradigm has evolved, leading to various definitions based on specific application contexts (Kritzinger et al., 2018). A comprehensive taxonomy of DT definitions is provided by Barricelli et al. (2019), who identify 29 different definitions of the framework grouped according to some key features. Synthesizing these diverse perspectives, a DT can be generally defined as a virtual replica of a physical system, spanning the entire lifecycle of a product or process while remaining continuously synchronized with its real-world counterpart (Aheleroff et al., 2021). To achieve this, the framework leverages real-time data streams, integrating them with physics-based simulation models, ML techniques, and optimization algorithms, providing valuable insights for system-level decision-making (Tao et al., 2022).

Functionally, the DT operates on two complementary horizons. On one hand, it acts *reactively*: by running in parallel with the physical counterpart, it enables enhanced monitoring, diagnostics, and fault detection, thereby facilitating maintenance and prompt interventions (Errandonea et al., 2020). On the other hand, it supports *proactive* operations: the DT can execute optimization and control strategies within a hypothetical (*what-if*) virtual scenario before their physical deployment, allowing for the simulation of different conditions and critical events that might affect the system's lifecycle (Lim et al., 2020).

Structurally, while some works state that DTs should focus exclusively on simulation (Weyer et al., 2016; Gabor et al., 2016), the fundamental architecture typically conceptualized is a tri-partite framework, originally proposed by Grieves (2014), comprising three core domains: the *Physical Space*, the *Virtual Space*, and the *Data Linking Mechanism* that binds them (Qi and Tao, 2018; Tao et al., 2019a). Specifically:

- The **Physical Space** encompasses the actual assets and the associated sensory infrastructure. This layer is responsible for gathering heterogeneous data from sensors, actuators, and embedded controllers to capture the real-time state of the system.
- The **Virtual Space** serves as the high-fidelity mirror of the physical entity. It aggregates mathematical and physics simulation models, rule-based algorithms, and ML techniques to replicate physical behaviors with high accuracy.
- The **Data Link** facilitates the seamless, bidirectional flow of information. This critical interface ensures continuous synchronization: the *uplink* transmits real-time signals from the physical to the virtual world, while the *downlink* feeds simulation insights and control commands back to the physical actuators.

It is worth noting that while this three-dimensional model remains the foundational standard, recent literature has expanded this topology. Notably, Tao et al. (2019b) present a five-dimensional framework for product design, explicitly adding *Data* and *Services* as independent layers to address the growing complexity of big data management in modern industrial applications.

1.2 Why Machine Learning for Digital Twins?

Machine Learning is formally defined by Tom M. Mitchell as «the study of computer algorithms that allow computer programs to automatically improve through experi-

ence» (Mitchell, 1997). Fundamentally, ML constitutes a subset of the AI field comprising mechanisms that enable a computer, specifically an *artificial agent*, to *learn* through experience, i.e., information data generated by artificial or natural processes and affected by uncertainty and noise. Rather than relying on explicit, hard-coded instructions for specific tasks, ML algorithms construct mathematical models based on sample data to identify underlying patterns, execute predictions, and adapt to novel information.

The taxonomy of ML typically involves three main categories based on the nature of the problem they address. *Supervised learning* aims to map input data to known output labels to predict outcomes for new, unseen data by inferring hidden patterns in labeled (*training*) datasets (Bishop, 2007). Supervised learning encompasses tasks such as *regression*, if the output is continuous, and *classification*, if the output is categorical. A second paradigm is *unsupervised learning*, whose goal is to identify hidden patterns or structures in unlabeled data, without predefined output labels (Ghahramani, 2004). Common unsupervised learning tasks include *clustering*, which partitions data points into groups based on similarity, *anomaly detection*, which isolates outliers or deviations from nominal patterns, and *latent variable modeling*, which aims to discover underlying factors governing the observed data distribution. Finally, the third category is *reinforcement learning* (RL), which focuses on learning effective strategies to adopt in an unknown environment to maximize some utility function (Sutton and Barto, 2018). While supervised and unsupervised learning treat data as static, independent instances (*one-shot* decisions), RL is inherently sequential. Indeed, in RL, the agent faces a decision-making problem, where it must interact with an environment over time, influencing its dynamics and receiving feedback on the taken actions.

In the context of DTs, ML is not merely an auxiliary tool but a fundamental component on which the entire DT paradigm is built. Indeed, while traditional simulation models rely on predefined mathematical equations and physical laws to represent system behavior (*model-based*), ML-based DTs leverage *data-driven* approaches to learn complex relationships and non-linear dynamics directly from empirical observations. This shift from model-centric to data-centric paradigms enables DTs to capture intricate patterns, adapt to changing conditions, and make accurate predictions in scenarios where explicit modeling may be infeasible or impractical (Rasheed et al., 2020). However, the drawbacks of purely data-driven DTs include the need for large datasets, potential overfitting, and lack of interpretability. To address these challenges, *hybrid* approaches that combine model-based and data-driven techniques are increasingly being explored, leveraging the strengths of both paradigms to create more robust and effective DTs (Semeraro et al., 2023b).

Independent of the specific architectural choice for the DT simulator, ML can be deployed across various functional layers to enhance performance. For instance, supervised learning can be used to calibrate and validate simulation models, ensuring virtual entities accurately reflect the behavior of their physical counterparts. In parallel, unsupervised ML methods can facilitate real-time monitoring and anomaly detection. By analyzing latent patterns within sensor data, these algorithms can identify deviations from nominal behavior and trigger preemptive maintenance alerts before critical failures occur (Amruthnath and Gupta, 2018). Furthermore, regression-based approaches can support predictive analytics, allowing DTs to forecast future dynamics, which can be informative for maintenance scheduling, resource allocation, and decision-making. Most

notably, RL methods have emerged as a powerful tool for optimizing and deploying reliable control strategies. Unlike supervised methods, RL operates through *trial-and-error* interactions to maximize a cumulative reward signal. The DT environment is particularly advantageous for this paradigm: it provides a high-fidelity safe *sandbox* where the RL agent can explore diverse operational policies and converge toward an optimal control strategy without jeopardizing the integrity of the physical asset (Xia et al., 2021).

1.3 Digital Twins for Energy Systems

In recent years, the concept of DTs has gained significant traction in the energy sector, particularly in the context of battery energy storage systems (BESSs). In modern energy systems, BESSs provide the essential flexibility required to integrate intermittent RESs, stabilize grid frequency, and enable demand response mechanisms. Regardless of the specific operational domain, spanning from electric vehicles to stationary storage within MGs or broader SGs, the battery constitutes the critical node for energy buffering and regulation. Consequently, the performance, reliability, and longevity of this asset are important factors governing the overall techno-economic efficiency of the energy ecosystem (Yang et al., 2018).

For advancing the research on energy systems, designing and implementing accurate DTs for batteries is of paramount importance. However, this objective comes with significant challenges, related to many factors, such as the complex electrochemical dynamics governing battery operation, non-linear degradation mechanisms, but also application-specific requirements, real-time data acquisition, and computational constraints. Addressing these challenges necessitates a multidisciplinary approach, combining expertise in battery technology, data science, control theory, and system engineering (Semeraro et al., 2023c).

In this section, we describe with a *micro-to-macro* perspective how a DT framework for BESSs can be beneficial in the energy field. First, we provide a concise overview of the state-of-the-art methodologies for designing and implementing such DTs. Then, we discuss how ML techniques can be integrated into these frameworks to enhance their monitoring and adaptation capabilities. Next, we zoom out to the system level, illustrating how DTs can be employed for optimizing the energy management within MG networks. Finally, we go further, discussing the potential of DTs to facilitate the coordination and social engagement among multiple participants within RECs.

1.3.1 Design and Implementation of Digital Twins

The extant literature on DTs for battery systems is extensive and heterogeneous, reflecting the multifaceted nature of battery technologies and their applications (Semeraro et al., 2023c). However, in this dissertation, we abstract away from specific battery technologies and the fundamental electrochemical processes underpinning their micro-dynamics. Instead, the primary focus is placed on the architectural design and operational implementation of DT frameworks for BESSs, adopting a system-level perspective tailored for MG and SG integration. In this context, existing methodologies can be broadly categorized into distinct classes based on their modeling paradigm, i.e., *data-driven* and *model-based*, which present different characteristics depending on their application domain (Semeraro et al., 2023a).

Data-driven Approaches. Data-driven approaches utilize ML and artificial neural networks (ANN) to construct DTs without requiring explicit knowledge of the battery's internal physics. These methods, often termed as *model-free* approaches, learn system behavior directly from historical data. While they typically require substantial datasets, they are highly valuable when physical models are unavailable or prohibitively complex to derive. A seminal example is provided by [Severson et al. \(2019\)](#), who demonstrated the capability of data-driven pipelines to predict battery cycle life using early-cycle discharge data. Similarly, [Li et al. \(2021\)](#) designed a DT for a battery management system (BMS) using a multi-linear regression algorithm to enhance the state of charge (SOC) prediction accuracy. Comparable contributions are given by [Kharlamova et al. \(2022\)](#) and [Qu et al. \(2020\)](#), who employed neural networks for SOC and degradation forecasting. Another data-driven approach has been adopted by [He et al. \(2023a\)](#), who developed a DT capable of predicting mechanical structural integrity through a data-driven framework.

Model-based Approaches. Model-based approaches rely on explicit mathematical, physical, or electrical models of the battery. These methods generally exhibit better extrapolation capabilities and require less data than data-driven alternatives. However, they are highly sensitive to model selection accuracy, since inappropriate design choices can introduce significant approximation errors. Notable works include [Peng et al. \(2019\)](#), who integrated spacecraft battery models with real-world data for degradation assessment, [Xu et al. \(2022\)](#), which used computational fluid dynamics (CFD) for high-fidelity thermal management, and [Yang et al. \(2022\)](#), that studied battery aging and remaining useful life. [Li et al. \(2022\)](#) combined impedance and open-circuit voltage (OCV) models with the Cuckoo search algorithm for parameter estimation ([Gandomi et al., 2013](#)). More similarly to this work, [Wang et al. \(2021\)](#) proposed a Thevenin equivalent circuit model (ECM) with particle filtering. Furthermore, recent trends show a shift toward hybrid strategies: for instance, [Zhao et al. \(2022a\)](#) integrated second-order ECMs with an LSTM (long short-term memory) network and Kalman filtering to estimate the SOC of a li-ion battery.

Challenges and Gaps in Smart Grid Applications. Among existing works, most DTs target specific applications: automotive ([Li et al., 2021](#); [He et al., 2023a](#); [Wang et al., 2021](#)) or aerospace ([Peng et al., 2019](#)). Regarding MG and SG applications, [Park et al. \(2020\)](#) developed a data-driven DT to optimize energy storage scheduling and reduce electricity costs; however, their approach neglects critical thermal and degradation dynamics. [Tang et al. \(2021\)](#) proposed a cloud-based hybrid DT for multi-battery systems addressing voltage, SOC, and state of health (SOH) estimation, yet the work lacks comprehensive experimental validation. [Reniers and Howey \(2023\)](#) built a detailed physics-based simulator accounting for cell-to-cell variations, but the high computational complexity of the model presents a barrier to real-time integration with ML techniques. Finally, [Padmawansa et al. \(2023\)](#) designed a DT for MGs employing the Rainflow cycle-counting method for degradation assessment. While effective for offline analysis, this approach introduces significant computational overhead, rendering it less suitable for the real-time operation required by modern smart grids.

1.3.2 Real-Time Monitoring and Adaptation to Physical Systems

As a crucial element of the modern decentralized grid paradigm, BESSs must be strictly monitored and controlled to avoid critical failures and optimize overall longevity (Shang et al., 2024). Typically, indicators such as the SOC, SOH, and remaining useful life (RUL) are used to assess the battery's condition and inform operational decisions. However, accurately tracking these metrics is notoriously challenging due to the highly non-linear nature of internal electrochemical processes, which are strongly influenced by dynamic operating currents and fluctuating ambient temperatures (Wang et al., 2020).

Traditionally, to accurately measure the degradation indicators, it is often necessary to suspend the battery's normal operation, apply specific characterization profiles (e.g., standard charge/discharge cycles), and analyze the results offline. This approach is not only time consuming, but fundamentally impractical for real-world applications, especially in MG and SG contexts, where continuous, uninterrupted operation is essential for grid stability and economic viability. Therefore, it is imperative to develop methodologies that enable real-time monitoring and seamless adaptation to the physical system without requiring operational downtime.

An effective framework to address this challenge is the DT. By continuously synchronizing with the physical battery via sensor data streams, the DT acts as a virtual sensor, providing real-time tracking of the battery's internal states and allowing for the estimation of critical indicators on-the-fly (Das et al., 2024). Furthermore, by integrating ML techniques, the DT can outdo traditional mathematical modeling. Data-driven algorithms can identify latent degradation patterns, detect anomalies before they evolve into critical faults, and predict future performance trajectories (Khumprom and Yodo, 2019). However, the deployment of static ML models is insufficient for long-term battery monitoring. As a battery undergoes irreversible aging phenomena, such as Solid Electrolyte Interphase (SEI) growth and active material loss, its physical behavior fundamentally changes. In machine learning terminology, this phenomenon induces a *concept drift*, causing pre-trained, offline models to progressively degrade in accuracy as the physical asset deviates from its Beginning of Life (BOL) state (Peng et al., 2024).

To sustain high-fidelity tracking throughout the battery's entire lifecycle, the DT framework must be inherently adaptive. This necessitates the integration of Online Learning (OL) techniques. Unlike static models, OL algorithms continuously ingest new streams of operational data to recursively update the DT's internal parameters and logic. By actively identifying operational regime shifts and adapting to incremental degradation, an evolving DT framework ensures robust, lifelong synchronization with its physical counterpart (Erkoyuncu et al., 2020).

1.3.3 Control and Optimization of Microgrids

In the evolving landscape of decentralized energy systems, the increasing penetration of RESs has become a defining characteristic of modern power grids. RESs, such as solar, wind, hydroelectric, biomass, and geothermal energy, have emerged as critical components of global energy strategies due to their environmental benefits and their potential to reduce greenhouse gas emissions, being a sustainable alternative to fossil fuels (Ellabban et al., 2014). However, while their ability to generate electricity with minimal environmental impact is undisputed, their inherent intermittent nature and low

(or non-existent) inertial response present significant challenges to maintaining a stable, reliable, and high-quality power supply (Basit et al., 2020). To mitigate these instabilities, DERs are becoming fundamental within the distribution network. Comprising localized generation and storage systems, including rooftop solar panels, wind turbines, battery storage, and combined heat and power (CHP) units, DERs facilitate a transition toward a decentralized energy model that enhances grid flexibility, reduces transmission losses, and improves overall system resilience.

An essential component of this decentralized paradigm is the MG: a localized power network that can operate synchronously with the main electrical network (*grid-connected*) and seamlessly switch to a stand-alone mode (*islanded*) without service disruptions. By integrating RESs, DERs, conventional generations, and clusters of loads, MGs optimize local energy management, balance supply and demand more efficiently, and contribute to broader grid stability (Mariam et al., 2016). Their deployment is particularly beneficial in remote areas, industrial zones, and regions prone to natural disasters, where reliable energy access is crucial. From the perspective of the main utility grid, an MG operates as a single controlled load, which, having the generation situated close to the consumption, significantly reduces transmission losses.

Nevertheless, the stochastic nature of RES availability and load fluctuations introduces profound challenges to the secure and techno-economic operation of MGs. The complexity of real-time supply-demand balancing often necessitates additional operating reserves or backup generation, thereby increasing operational costs. This asks for the design and implementation of optimal MG operation schemes, as well as coordination and interaction mechanisms with the main grid. Crucially, these schemes must also account for the DERs contribution to ancillary services. If DERs merely displace energy from a central generation without providing equivalent flexibility and regulating capacity, the overall cost of operating the entire system will inevitably rise (Lopes et al., 2007).

Against this backdrop, the DT paradigm emerges as a promising tool for enhancing the design, validation, and operational management of MGs. As highlighted in Section 1.1, a DT serves as a high-fidelity virtual sandbox, providing a safe environment to analyze system behavior and test advanced control strategies without risking physical instability. Moreover, DTs facilitate seamless coordination between the MG and the main grid, allowing predictive and adaptive strategies to be validated under diverse operating conditions before deployment. The synergy between DTs and ML further amplifies this utility, enabling the application of advanced data-driven methods, such as model predictive control (MPC) (Mayne, 2014) and RL, for real-time optimization. These techniques can learn optimal policies for energy management, demand response, and ancillary services provision, continuously adapting to the dynamic conditions and uncertainties inherent in modern MG operations.

1.3.4 Energy Management for Renewable Energy Communities

Extending the scope beyond the single MG, the new decentralized paradigm is fostering the emergence of community-based energy models that leverage local RESs and DERs to better serve collective energy needs. A prominent manifestation of this trend is the *renewable energy community* (REC). RECs consist of aggregations of *prosumers* who collaborate to balance supply and demand at the local level. By orchestrating RESs and DERs, RECs promote local self-sufficiency, reduce reliance on the main grid, and lower

energy costs, thereby fostering both economic and environmental sustainability.

An operational backbone of a functional REC is the energy management system (EMS), the control layer responsible for regulating energy production, storage, and exchange both within and without the community. While traditional EMS architectures are often centralized, relying on a single decision-making unit (Aittahar et al., 2024; Palma et al., 2024), the scalability and privacy requirements of modern grids increasingly favor decentralized control strategies. With a decentralized EMS, decision-making is distributed among autonomous agents, each associated with a distinct community member. These agents optimize local objectives while coordinating to enhance the global performance of the community.

However, designing such systems presents several challenges. Beyond the classical limitations of decentralized control, such as partial observability and privacy preservation, RECs introduce a critical *socio-economical* conflict. Specifically, REC members are rational, self-interested entities primarily driven to minimize their individual costs. Without explicit incentives, they have little motivation to act cooperatively and adapt their behavior for the benefit of the community. In the European Union, and specifically within the Italian regulatory framework, financial incentives are provided for the *collective self-consumption* of renewable energy (Italian Government, 2019). Crucially, these regulations do not specify how to distribute such incentives among participants to effectively elicit cooperative behavior. This creates an urgent need for designing approaches capable of distributing incentives fairly, to effectively align individual utility with community-level efficiency (Moret and Pinson, 2019).

In this complex landscape, the DT paradigm offers a promising avenue for advancing REC management. By providing a virtual environment that accurately mirrors the physical participants in the community, DTs enable the testing and validation of decentralized control strategies under various scenarios. This capability is further amplified by the integration of advanced computational methodologies capable of handling multi-entity environments, whether through centralized coordination or decentralized consensus. In this context, techniques derived from *operational research*, *game theory*, and RL are paramount. These approaches enable the derivation of optimal policies for peer-to-peer energy sharing, demand response, and fair incentive distribution, all while strictly adhering to individual privacy constraints and promoting collective social welfare.

1.4 Research Objectives and Original Contributions

This dissertation is structured around four primary research objectives, organized into four distinct parts that mirror the logical evolution of the proposed framework. The central unifying theme is the *digital twin* paradigm, specifically tailored for BESSs deployed in stationary MG and SG applications. Notably, considered research objectives are designed to progressively build upon each other, creating the need to establish robust foundations before advancing to more complex system-level applications. Adopting a *micro-to-macro* perspective, we present the research trajectory from the granular modeling of individual assets to the orchestration of broader energy ecosystems.

1.4.1 Research Objectives

The research objectives interesting this dissertation are outlined as follows:

RO1. Design and implementation of a Digital Twin. The first objective focuses on designing a comprehensive DT framework for BESSs. The challenge is to develop a modular architecture that accurately captures the multifaceted dynamics of battery systems, capable of assuring scalability and customizability for different technologies and applications. The design must also account for the subsequent integration of ML techniques and their utilization in MG and REC contexts.

RO2. Integration of real-time monitoring and adaptation capabilities. The second objective aims to enhance the DT's fidelity and resilience through the incorporation of tracking and adaptation capabilities. Specifically, the focus is on developing algorithms for real-time monitoring, drift detection, and adaptive recalibration to ensure that the DT remains synchronized with its physical counterpart despite system degradation and changing operational conditions.

RO3. Design of control strategies for microgrid energy management. The third objective addresses the application of the DT framework for optimizing energy management within MGs. The goal is to leverage the DT as a simulation environment for training intelligent agents, enabling them to learn optimal control policies for battery operation, demand response, and minimizing costs associated with degradation and energy procurement.

RO4. Design of control strategies for renewable energy communities. The last objective extends the application of the DT framework to the broader context of RECs. The focus is the development of control strategies that enable individual prosumers to optimize their energy management while contributing to the overall efficiency and sustainability of the community. This involves addressing challenges related to coordination, privacy, and incentive alignment among multiple individual participants.

1.4.2 Original Contributions

The Ph.D. research presented in this dissertation has been funded by RSE S.p.A. in agreement with Politecnico di Milano.¹ RSE is an Italian research center specialized in energy systems, with a focus on renewable energy integration, grid modernization, and energy storage technologies. The collaboration with RSE has provided access to facilities, datasets, and industry expertise, facilitating and enriching the research process and outcomes. With them, we agreed on the research objectives over the three years and the expected contributions of this dissertation, with the aim of advancing the research about DTs for BESSs employed in MG and SG contexts. In the following, we summarize the original contributions of this dissertation, mapped to the corresponding research objectives.

¹Ricerca sul Sistema Energetico - RSE S.P.A., Via R. Rubattino 54, 20134 Milano, <https://www.rse-web.it/en/>

Part I: Design and Validation of the Digital Twin Framework

The first part of the dissertation (Part I) aims to satisfy research objective *ROI*. This section is devoted to the design and implementation of a novel DT framework for BESSs. The proposed architecture is characterized by a modular structure that integrates multiple simulation layers, each dedicated to specific battery dynamics, including electrical, thermal, and degradation processes. The framework incorporates an orchestrator module that synchronizes these layers and interfaces with external data sources, enabling real-time monitoring and control. The DT is validated through extensive experimental campaigns using a physical BESS prototype, demonstrating its accuracy in replicating real-world behavior across various operating conditions.

Part II: Machine Learning Integration for Real-Time Monitoring and Adaptation

In the second part of the dissertation (Part II), the focus shifts to enhancing the DT's adaptability through ML techniques. This part addresses the critical challenge of parameter drift, which can severely compromise the DT's fidelity over time due to battery aging and the shift of operational regimes. The proposed approach employs an active online learning strategy that associates a specific system model with each distinct operational regime. These models are recursively updated with newly estimated parameters. Crucially, the framework is capable of detecting distributional shifts by monitoring the frequency of statistical outliers, automatically triggering a model re-adaptation when a concept drift is confirmed. The results demonstrate that the integration of ML techniques significantly improves the DT's ability to maintain accurate synchronization with the physical battery, enabling real-time tracking and interpretability of the system dynamics.

Part III: Microgrid Energy Management through Reinforcement Learning

The third part of the dissertation (Part III) explores the problem of optimizing energy management in MGs. The DT serves as a high-fidelity simulation environment capable of integrating exogenous signals, such as load demand, renewable generation profiles, and market prices. Within this virtual setting, deep reinforcement learning (DRL) techniques can be employed to train intelligent agents. The proposed methodology involves formulating the energy management problem as a Markov Decision Process (MDP), where the DRL agent learns to make optimal decisions regarding battery charging and discharging actions. The DT provides a safe and controlled environment for training, allowing the agent to explore various strategies without risking physical asset integrity. The results demonstrate that the DRL-trained policies outperform traditional rule-based approaches in terms of cost savings and battery health preservation.

Part IV: Multi-Agent Reinforcement Learning for Renewable Energy Communities

In the last part of the dissertation (Part IV), the research employs the DT framework to reproduce a REC system. This part, grounded on the results of the previous sections, proposes a MARL approach, where each prosumer within the community is represented by an individual MG. The agents interact within a decentralized environment, learning to optimize their energy management strategies while considering both individual and collective objectives. The proposed framework addresses challenges such as coordination among agents, scalability, and privacy. Experimental results demonstrate that the

MARL approach effectively enhances overall community performance, promoting energy sharing and reducing reliance on external grid resources.

1.5 Overview

The remainder of this dissertation is structured in four parts, followed by a discussion on research contributions and on future directions. The detailed organization of the dissertation is described in the following.

Part I: Design and Validation of the Digital Twin Framework

This part is dedicated to the design, implementation, and experimental validation of a novel DT framework for BESSs. Concepts and results described in this part appear in [Salaorni et al. \(2025a\)](#).²

- Chapter 2 presents the architectural design of the proposed DT framework, detailing its modular structure and the integration of various simulation layers.
- Chapter 3 describes the experimental setup and validation procedures, showcasing the DT's performance in replicating real-world battery behavior.

Part II: Machine Learning Integration for Real-Time Monitoring and Adaptation

This part focuses on the integration of ML techniques to enable real-time adaptation of the DT, ensuring its accuracy over time despite parameter drift. Concepts and results described in this part are novel contribution of this dissertation.²

- Chapter 4 presents the problem of parameter drift in DTs and the motivation for integrating ML techniques. It also reviews existing approaches, highlighting challenges and gaps.
- Chapter 5 introduces fundamentals of online learning and concept drift detection, and details the proposed approach integrated within our DT framework.
- Chapter 6 presents experimental results demonstrating the effectiveness of the adaptation strategy, stressing the importance of trading off between model accuracy and interpretability.

Part III: Microgrid Energy Management through Reinforcement Learning

This part explores the application of DRL techniques for optimizing energy management in MGs using the DT as a training environment. Concepts and results described in this part appear in [Salaorni et al. \(2025b\)](#).³

- Chapter 7 frames the control problem of MG energy management and reviews existing approaches, with a focus on the potential of RL techniques.
- Chapter 8 introduces the fundamentals of RL and details the proposed methodology grounded on the DT framework.

²The relative codebase is available at <https://github.com/Daveonwave/ErNESTO-DT>.

³The relative codebase is available at <https://github.com/Daveonwave/ErNESTO-gym>.

- Chapter 9 showcases the performance of the RL approach in optimizing energy management within the MG, comparing it with baseline strategies and demonstrating its advantages in terms of cost savings and battery health preservation.

Part IV: Multi-Agent Reinforcement Learning for Renewable Energy Communities

This part extends the DT framework to RECs, proposing a MARL approach for decentralized energy management. Concepts and results described in this part belong to a paper currently submitted for publication at a Q1 peer-reviewed journal.⁴

- Chapter 10 presents the problem of energy management in RECs, highlighting the challenges related to coordination, privacy, and incentive alignment. It also reviews existing approaches, with a focus on the potential of MARL techniques.
- Chapter 11 introduces the fundamentals of MARL and details the proposed methodology for decentralized energy management within RECs, grounded on the DT framework.
- Chapter 12 showcases the performance of the MARL approach in optimizing energy management within the REC, demonstrating its effectiveness in promoting energy sharing and reducing reliance on external grid resources.

Finally, Chapter 13 summarizes the key findings of the dissertation, discusses the implications of the research, and outlines potential avenues for future works.

In the appendix, we report supplementary material, including additional experimental results, code snippets, and technical details that support the main content of the dissertation. In particular, in Appendix A we provide further details about the parameter identification procedures and the setting for the DT validation. In Appendix B, we report additional results about the online learning strategy, including hyperparameters tuning and clustering visualization. In Appendix C, we provide further details about the RL training procedure and the hyperparameters setting. Finally, in Appendix D, we report additional results about the MARL approach for RECs, including the hyperparameters setting and constant specification.

⁴The relative codebase is available at <https://github.com/saamur/INAIA-REC>.

Part I

**Design and Validation of the Digital
Twin Framework**

CHAPTER 2

Design of the Digital Twin Framework

In the previous chapter, we introduced the concept of DT, the historical development of this technology, and its relevance to the modern energy sector. Furthermore, we conducted a critical review of the state-of-the-art, identifying prevailing methodologies to design DT frameworks in the context of stationary BESS applications and highlighting the significant gaps that remain in current literature.

In this part of the dissertation, we aim to address the research objective *ROI* outlined in Section 1.4.1. Structurally, the development of a high-fidelity simulator for BESSs is a non-trivial undertaking. Although all lithium-ion batteries share fundamental electrochemical principles, their internal dynamics exhibit substantial variance depending on specific sub-chemistries (e.g., LFP, NMC, etc.), cell formats (cylindrical, prismatic, pouch), and manufacturing processes. Consequently, the formulation of a *universal* model remains an open challenge. A potential alternative would be the adoption of a purely data-driven approach, which theoretically allows for generalization among different battery types and the capture of complex non-linear dynamics without explicit physical derivation. Indeed, the availability of high-quality experimental data has improved significantly in the last decade. Since 2010, the NASA Prognostics Center of Excellence (PCoE) has released open-source datasets enabling the study of degradation under controlled conditions ([Saha and Goebel, 2010](#)). Similarly, the CALCE Battery Research Group at the University of Maryland has provided invaluable datasets covering various chemistries and aging profiles ([Center for Advanced Life Cycle Engineering \(CALCE\), 2010–2024](#)). More recently, NASA’s release of randomized battery usage profiles and electrochemical impedance spectroscopy (EIS) measurements ([Bole et al., 2022](#)) has further enriched the resources available for training AI models.

However, relying exclusively on a data-driven paradigm presents critical limitations for the design of a generalized BESS framework. First, while datasets are increasing

in volume, they remain largely confined to *cell-level* laboratory tests performed under strictly controlled environmental conditions (e.g., constant temperatures, standard cycling profiles). They rarely capture the stochastic, *system-level* thermal and electrical interactions typical of real-world operations. Second, pure data-driven models frequently struggle with generalization; functioning as *opaque* approaches, they may violate physical constraints when extrapolating for operating conditions outside their training distribution. For these reasons, the trend in the literature regarding DTs for stationary applications, as analyzed in Section 1.3, still favors *physics-based* or *hybrid* architectures. These approaches leverage physical laws to ensure robustness and interpretability, using ML techniques primarily for parameter identification or state estimation rather than for replacing the structural model entirely.

Nevertheless, a limitation of existing works adopting physics-based solutions is their tendency to focus on isolated *sub-models* (e.g., focusing solely on the electrical equivalent circuit) while neglecting coupled dynamics such as thermal evolution or aging mechanisms. Treated in isolation, these sub-models fail to capture the complex, multi-physics interactions that drive battery performance in the field.

To address these challenges, the primary objective of this dissertation is to design a *holistic* DT framework, named ERNESTO (ENERgy STOrage) Digital Twin, capable of simulating the complete BESS ecosystem. We propose a *modular* architecture that integrates distinct sub-models into a cohesive system. Concepts and analysis presented in this part are based on [Salaorni et al. \(2025a\)](#), which describes the design and validation of the proposed DT framework. In the following sections, we detail the design methodology, defining the role of each module and illustrating a representative configuration.

Chapter Outline. The chapter is organized as follows. First, in Section 2.1, we describe and justify the choice of adopting a modular architecture. Next, in Section 2.2, Section 2.3, Section 2.4, and Section 2.5, we describe the individual sub-models for electrical behavior, thermal dynamics, SOC estimation, and SOH estimation, respectively. Section 2.6 discusses the orchestration of these models within the DT framework, highlighting their interdependencies.

2.1 Design Philosophy: Modularity and Integration

In their comprehensive survey establishing guidelines for next-generation DTs, [Lim et al. \(2020\)](#) identify *modular architecture* as one of the eight foundational pillars for future DT development. This principle is not merely a software engineering convenience but a structural necessity for handling complex industrial systems. As demonstrated by [Guo et al. \(2019\)](#) in the context of factory design, modularity significantly enhances flexibility, reusability, and maintainability, decoupling individual functional components that can be updated or replaced without compromising the integrity of the entire system.

This paradigm is particularly relevant to the BESS domain, where diverse physical phenomena (electrical, thermal, and chemical) interact across different time scales. As highlighted in the literature analysis in Section 1.3.1, a significant gap exists in the current state-of-the-art: the lack of holistic DT frameworks that encapsulate these coupled dynamics. Most existing research concentrates on isolated *sub-models* ([Peng et al., 2019](#); [Li et al., 2022](#)), which, while accurate for specific tasks, fail to capture the

emergent behavior of the complete battery ecosystem.

2.1.1 The ERNESTO Framework

To address this fragmentation, we adopted a *divide et impera* (divide and conquer) design methodology. Rather than attempting to construct a monolithic model, which is inherently brittle and difficult to validate, we decompose the complex BESS problem into manageable, interacting sub-systems. Although this approach requires rigorous interface definition to manage signal coupling, it yields long-term benefits in terms of scalability and the ability to integrate heterogeneous technologies (Jia et al., 2022).

Therefore, we introduce ERNESTO, a physics-based, modular DT framework designed to simulate the electrical, thermal, and aging dynamics of lithium-ion BESSs. The architectural decision to rely on first-principles physics models is motivated by three key factors. First, *robustness*: physics-based models ensure consistency and plausible behavior even in operating regions outside standard testing conditions (granting extrapolation capability). Second, *interpretability*: physical models provide insights into underlying mechanisms, facilitating diagnostics and control parametrization. Third, *data scarcity*: as discussed previously, high-quality datasets capturing *system-level* dynamics are scarce. Physics-based foundations allow us to bridge this gap. Moreover, the modular design facilitates the generalization of the framework across different battery configurations by allowing individual sub-models to be swapped or adapted as needed. In addition, to ensure accuracy, the framework integrates renowned, experimentally validated sub-models from the literature. The holistic validation of these combined models is presented in Chapter 3.

2.1.2 Architectural Components

The architecture of ERNESTO is conceived with encapsulation and modularity at its core. Figure 2.1 illustrates the high-level schematic. The right side depicts the *Virtual Battery*, detailing the constituent sub-models (electrical, thermal, SOC, aging) and the multidirectional data flows required for their coupled operation. The left side represents the *External Environment*, emphasizing the framework’s capability to ingest real-world signals from the MG environment — a feature critical for the applications discussed in subsequent parts (Part III and Part IV).

The framework is coordinated by a central *Orchestrator*, which manages synchronization and data exchange across three distinct *Simulation Modes*:

1. **Driven Mode:** A *historical replay* mode that ingests external battery load time-series (e.g., CSV logs) to validate the virtual battery against real-world operations.
2. **Scheduled Mode:** An operation mode allowing users to define synthetic load profiles or charge/discharge commands via natural language instructions.
3. **Adaptive Mode:** An advanced operating mode integrating ML techniques. This mode enables the DT to autonomously detect parameter drift and update its internal state when operational regime shifts occur (detailed in Part II).

In the following sections, we will provide a detailed description of the sub-models embedded in the proposed framework. These models focus on system-level battery dynamics rather than detailed cell-level electrochemical processes. Nevertheless, enhanced

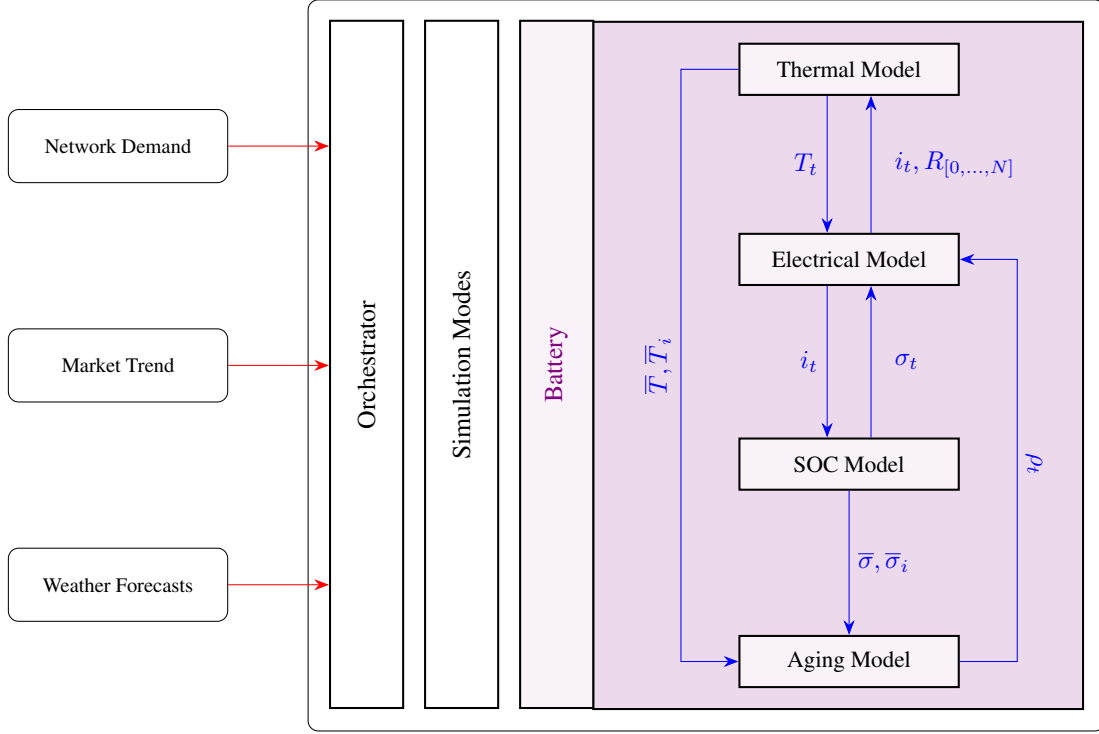


Figure 2.1: DT architecture. Schematic showing the framework's interconnected layers (right) and external data sources that the DT integrates to replicate the MG setting (left). The orchestrator, positioned above the battery object, synchronizes the modular battery sub-layers, enabling different simulation modes (driven and scheduled) and merges their outputs with external signals. It also incorporates learning algorithms to support continuous DT refinement.

versions incorporating electrochemical details can be easily integrated by complying with the DT structural requirements and specific sub-model interfaces.

2.2 The Electrical Model

The electrical model simulates the voltage response of the battery under dynamic load conditions. It employs an equivalent circuit representation based on Thevenin models, which balance computational efficiency with sufficient accuracy for system-level applications (Barbieri et al., 2020). Two variants are provided in ERNESTO: a first-order model (1-RC) and a second-order model (2-RC).

2.2.1 First-order Thevenin Model

The first-order Thevenin model consists of a series internal resistance R_0 , an RC branch containing resistance R_1 and capacitance C_1 , and an ideal voltage source V_{OC} representing the open circuit voltage (OCV). The terminal voltage is given by:

$$v = V_{OC} - v_R - v_{RC}. \quad (2.1)$$

Here, v_R denotes the voltage drop across R_0 , and v_{RC} is the voltage drop across the RC branch. This model is hereafter referred to as 1-RC. Figure 2.2 illustrates the circuit configuration.

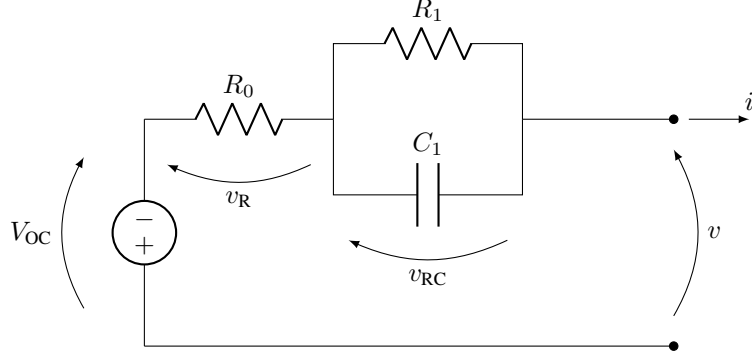


Figure 2.2: Circuitual representation of the first-order Thevenin ECM.

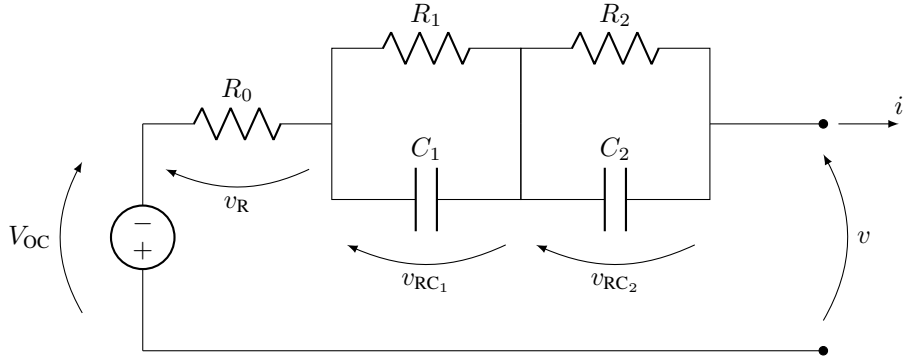


Figure 2.3: Circuitual representation of the second-order Thevenin ECM.

2.2.2 Second-order Thevenin model

The second-order variant mirrors the first-order structure but incorporates an additional RC stage with resistance R_2 and capacitance C_2 in series with the existing branch. The terminal voltage becomes:

$$v = V_{OC} - v_R - v_{RC_1} - v_{RC_2} , \quad (2.2)$$

where v_{RC_1} and v_{RC_2} represent voltage drops across the two RC branche, respectively. This variant achieves superior accuracy at the cost of increased parameter identification effort. This model is subsequently termed 2-RC. Figure 2.3 illustrates the circuit configuration.

2.3 The Thermal Model

The thermal model provided in ERNESTO, illustrated in Figure 2.4, is a renowned experimentally validated model (Barbieri et al., 2020, 2022; Cui et al., 2020b), that achieves a good balance between complexity and accuracy. It comprises a heat source Q_{gen} representing the thermal power generation, a thermal capacitor C_{th} , a conductive thermal resistance R_{cond} modeling the heat flow from the core to the surface, a convective thermal resistance R_{conv} capturing environmental heat exchange, and an ambient temperature source T_{amb} .

The model distinguishes between core and surface temperatures, denoted T_{core} and

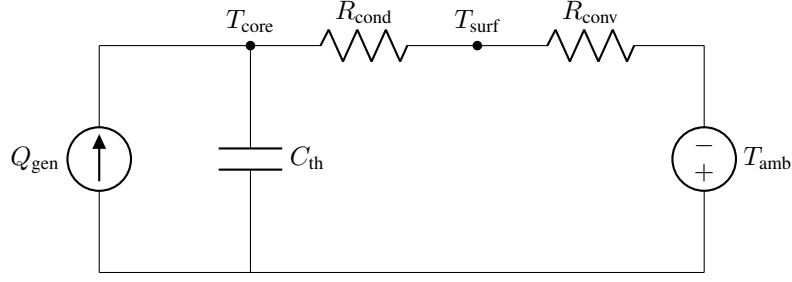


Figure 2.4: Circuitual representation of the thermal model.

T_{surf} . While the surface temperature is directly measurable via external sensors, core temperature is not readily accessible. Hence, we can approximate core temperature using surface temperature, relying on the assumption that the thermal conductivity between core and electrodes is significantly higher along its longitudinal axis than its transversal section. [Barbieri et al. \(2022\)](#) demonstrated this assumption holds when the current stays below 1C.

2.4 State of Charge Estimation

The SOC is a measure of the battery capacity that gives important information about the system's state. Formally, $\sigma_t \in [0, 1]$ describes the amount of battery charge with respect to the maximum capacity available C_t^{max} at time t :

$$\sigma_t = \frac{C_t}{C_t^{\text{max}}} . \quad (2.3)$$

where C_t is the current capacity. It is important to notice that the SOC cannot be directly measured online from the real system, but requires specific profile evaluation or proper estimation techniques ([Chang, 2013](#); [Mussi et al., 2021](#)).

In this work, to express the evolution of the SOC, we employ Coulomb Counting (CC) ([Ng et al., 2009](#)), a standard incremental real-time technique to compute the dynamics of the system. While computationally efficient and easy to implement, CC may suffer from accumulated errors in extended simulations. The method treats SOC as the integral of the current through time, normalized by the battery's nominal capacity C^{nom} as follows:

$$\sigma_t = \sigma_{t_0} - \frac{1}{C^{\text{nom}}} \int_{t_0}^t i_{\tau} d\tau . \quad (2.4)$$

In this dissertation, the CC method is employed in its time-discrete incremental version, which can be easily derived from the presented continuous formulation, and it appears as follows:

$$\sigma_t := \sigma_{t-1} - \frac{i_t}{C^{\text{nom}}} \Delta t . \quad (2.5)$$

The choice behind adopting the CC method is mainly due to its simplicity, which does not hinder our experiments. However, considering the framework's flexibility, there is no restriction against replacing the SOC estimation module with a different one.

2.5 State of Health Estimation

Battery degradation manifests as capacity loss or growth in the internal resistance. Capacity reduction causes energy fade, while resistance increase causes power fade. The SOH serves as an indicator of battery aging and can be defined in terms of either capacity degradation or resistance growth. In this dissertation, we indicate SOH as $\rho_t \in [0, 1]$ and we express it as a function of the capacity fade. Formally:

$$\rho_t = \frac{1}{1-p} \frac{C_t^{\max}}{C^{\text{nom}}} - \frac{p}{1-p}, \quad (2.6)$$

where $p \in [0, 1]$ is the rate of capacity at which the battery can no longer be utilized for the chosen application. The SOH reflects remaining operational life for the intended application and reveals degradation impacts across the battery's aging process and its components (Lyu et al., 2024). Hence, an accurate SOH estimator, or aging model, is essential to be considered during a reliable DT design, as it directly influences electrical model parameters over time. According to Swarnkar et al. (2023), conventional SOH methods either lack online capability or require prohibitively long full charge-discharge cycles. To not interrupt the battery application, a DT can be used, for example, to simulate full charging/discharging cycles, thus acquiring the information needed for an accurate SOH estimate (Qu et al., 2020). In addition, this solution is also suited for those applications where the battery is dynamically and often partially charged and discharged.

In this dissertation, we adopt the semi-empirical model by Xu et al. (2018), which combines the theoretical foundations of Li-Ion chemistry with experimental data. The model decomposes degradation into two components: *calendar* and *cyclic* aging. The calendar aging represents the stress factor caused by the unavoidable passage of time. The cyclic aging derives from the direct utilization of the battery in terms of applied charging/discharging profiles. The authors describe the aging model as a factorization of time and cyclic effects. In particular, each cycle is conceived as a single independent *stress* event. The overall stress function $f_{d,t}$ is a linear combination of both calendar $f_{\text{cal},t}$ and cyclic $f_{\text{cyc},t}$ aging. Formally:

$$f_{\text{cal},t} = f_{\text{cal}}(t, \bar{\sigma}, \bar{T}), \quad (2.7)$$

$$f_{\text{cyc},t} = \sum_i^{n_{\text{cyc}}} n_i f_{\text{cyc}}(\delta_i, \bar{\sigma}_i, \bar{T}_i), \quad (2.8)$$

$$f_{d,t} = f_{\text{cal},t} + f_{\text{cyc},t}, \quad (2.9)$$

where t is the time of battery life for which we need to estimate the cumulative degradation, $\bar{\sigma}$ and $\bar{\sigma}_i$ represent overall and per-cycle, $\bar{T}$ and $\bar{T}_i$ represent mean temperatures mean SOC levels, respectively, δ_i is the discharge depth of cycle i , and n_{cyc} and n_i identify total number of cycles and the cycle type, i.e., whether cycle i is a full or half cycle ($n_i \in \{1, 0.5\}$). Notably, we emphasize that the model depends on electrical and thermal sub-models that provide SOC and temperature measurements.

From the definition of the overall stress function $f_{d,t}$, it is possible to extract information about the system's degradation D_t as follows:

$$D_t = 1 - \alpha_{\text{SEI}} e^{-\beta_{\text{SEI}} f_{d,t}} - (1 - \alpha_{\text{SEI}}) e^{f_{d,t}}, \quad (2.10)$$

where $\alpha_{\text{SEI}} \in \mathbb{R}_0^+$ quantifies the amount of charge capacity eroded during the solid electrolyte interface (SEI) film formation, while $\beta_{\text{SEI}} \in \mathbb{R}_0^+$ represents the relative impact of usage and temperature. Notably, under extreme conditions, the degradation may follow different evolutions over time. We refer to [Leng et al. \(2015\)](#) for an extensive analysis on the impact of high temperatures. It is also worth considering that, in Equation (2.10), non-zero SEI parameters work only with brand new batteries. Indeed, since the initial rapid phase of SEI film formation typically stabilizes after approximately 10% of the battery’s useful life has elapsed, α_{SEI} must be carefully calibrated to the actual aging dynamics of the cell to accurately estimate its SOH.

A significant limitation of this aging model arises from the need to compute charging and discharging cycles. The original approach by [Xu et al. \(2018\)](#) used the Rainflow cycle counting algorithm ([Matsuishi and Endo, 1968](#)), a widely-used fatigue-analysis algorithm capable of identifying closed and half cycles within a time-series of data. While such a method is particularly effective with a pre-loaded dataset, existing Python implementations ([Janiszewski, 2022](#)) are unsuited for iterative step-by-step simulations, which will enable the integration of RL methods later (Part III). Unlike prior studies ([Padmawansa et al., 2023](#); [Mussi et al., 2021](#)) that adopt the presented aging model in its original form, we introduce a novel stateful version of the Rainflow algorithm, called DROPFLOW ([Salaorni, 2024](#)). DROPFLOW is designed to facilitate stepwise aging computation, significantly reducing the computational overhead. Indeed, while Rainflow gets a quadratic computational complexity if used in step-by-step simulation, DROPFLOW presents a linear complexity since it stores the previously computed cycles at each step. Further algorithmic details and validation of the method are provided in Appendix A.0.1.

2.6 Interactions Among Sub-Models

While a modular architecture simplifies the design and validation of individual sub-models, it is crucial to accurately capture and orchestrate the interactions among them to build a holistic DT framework. In Figure 2.1 we depict battery structure and model interdependencies. For instance, the parameters of the electrical model depend on temperature (provided by the thermal model), SOC (computed using the Coulomb Counting method), and SOH (estimated by the aging model). Furthermore, while the thermal model influences resistances and capacitances, these components dissipate heat back to the thermal model, forming a feedback loop. Similar couplings exist between aging-thermal and aging-SOC pairs, as aging calculation requires temperature and SOC inputs. Conversely, computed SOH modulates electrical model parameters over time.

To accurately capture these interactions, proper preliminary parameterization is required. The methodology entails collecting reference data points across the system’s operational range to construct multi-dimensional dependency maps for each model parameter. This can be done, for example, by fitting a regression model, such as a polynomial function or neural network, to the experimental data. During runtime simulation, the DT can then query these maps to retrieve parameter values based on current state variables (e.g., temperature, SOC, SOH). This approach ensures that the DT accurately reflects the non-linear coupled dynamics of the physical battery system.

In this specific work, given the sparsity of the available experimental dataset, we employ a lookup table (LUT) approach with linear interpolation for parameter retrieval,

as detailed in Section 3.1. However, the framework is architecturally designed to be agnostic to the mapping method, allowing for the seamless integration of more sophisticated regression techniques.

2.7 High-level Controller Integration

The architectural design of ERNESTO is expressly conceived to facilitate its integration within a broader MG control framework. For this purpose, it is structured to interface with external decision-making modules to simulate the BESS's behavior under realistic operational conditions. For example, considering the MG energy management problem, ERNESTO can be employed in conjunction with a high-level controller tasked with optimizing energy flows across the MG distribution network. While the controller receives generation and demand profiles from renewable energy sources and loads, it relies on the DT to simulate the BESS's response to various control actions. In particular, the DT aids the controller through two mechanisms:

1. **Training phase:** The DT provides a safe, risk-free simulation environment for training advanced control strategies, such as DRL agents. This allows the controller to explore the entire state-action space and learn optimal policies without jeopardizing the physical asset (Salaorni et al., 2025b).
2. **Operation Phase (Online/Shadow Mode):** During real-time operation, the DT runs in parallel with the physical system. It serves to monitor discrepancies between the expected virtual state and the actual physical state, enabling the detection of anomalies and providing reliable state estimates to support informed control decisions.

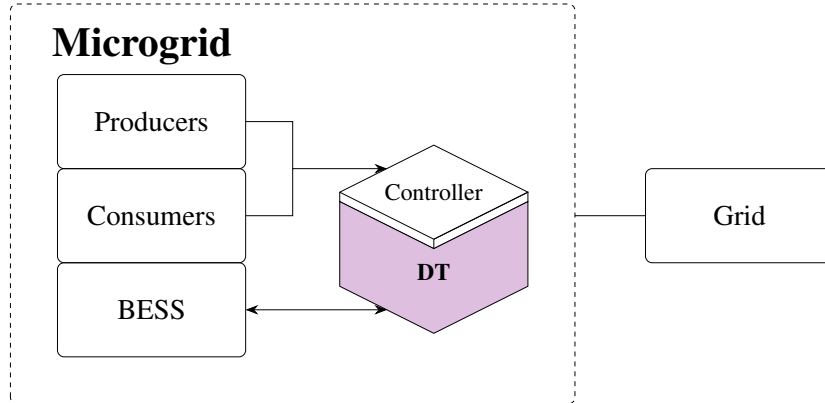


Figure 2.5: Control problem schema within an MG. The high-level controller operates under various scenarios, optimizing energy flows based on generation, demand, and grid parameters. It leverages the DT to simulate BESS responses, facilitating informed decision-making and control action deployment to the physical system. The DT acts to maintain alignment between virtual and physical systems through continuous feedback or as a training environment throughout what-if simulations.

Figure 2.5 illustrates this integration. The controller directly engages the DT as the primary decision maker within the MG, ingesting generation and demand profiles, BESS state, and grid parameters (e.g., energy prices), while exploring future scenarios via the DT. Control actions are subsequently deployed to the physical BESS.

Crucially, to serve as an effective training environment for data-driven algorithms, the DT framework must satisfy specific computational requirements. Beyond maintaining high physical fidelity, it must support *fast* execution to accelerate agent learning. Furthermore, it must expose a standard interface for *discrete time-step execution*, compliant with widely adopted RL standards such as the Gymnasium API ([Towers et al., 2025](#)), enabling the agent to interact with the virtual environment in a step-wise manner.

CHAPTER 3

Digital Twin Validation

In Chapter 2, we have introduced the design of ERNESTO to model the virtual representation of a BESS, describing the components of its modular architecture and the interactions among them. In this chapter, we present the experimental validation of the framework. We selected four different scenarios to evaluate the DT's accuracy in replicating the physical system's behavior, both with synthetic and real-world data. Moreover, we evaluate the performance of the single thermal and aging sub-models to highlight the need for their integration within the DT framework.

Chapter Outline. The chapter is organized as follows. First, in Section 3.1 we describe the experimental procedure for retrieving model parameters. Then, in Section 3.2, we illustrate the selected case studies and discuss the results obtained.

3.1 Model Parametrization

This section outlines the experimental methodology for model calibration and validation, detailing procedures for extracting electrical and thermal parameters. Data have been provided by RSE researchers and collected in RSE's laboratory. In particular, test profiles were executed on a physical lithium-ion NMC cell, gathering voltage, current, and temperature data. From these datasets, we derived the internal parameters of the circuital DT models as explained in the following.

The device employed as subject for the experiments is an EIG ePLB-C020 lithium-ion NMC cell, whose specifications appear in Table A.1. Experiments employed a Chroma 17011 bidirectional cycler (24 channels, 5V - 100A) controlled remotely. The cell was housed in a thermal chamber (TB-TS Series, FDM), maintaining constant temperature. Figure A.1 depicts the experimental configuration of RSE's laboratory.

3.1.1 Electrical Model Parameters

As outlined in Section 2.2, we implemented two variants of the Thevenin equivalent circuit model: a first-order model (1-RC) and a second-order model (2-RC). Both models require parameterization of resistive and capacitive elements, which depend on SOC and temperature (we do not consider SOH dependence here since the cell is new).

The parameters of the Thevenin models were determined at five different reference temperatures (273 K, 283 K, 293 K, 303 K, 313 K). For each temperature, the cell was first fully charged using a constant current-constant voltage (CC-CV) method. This involved charging at a constant current of 10 A (0.5 C) until the upper cut-off voltage was reached, followed by a voltage hold phase until the current decreased below 200 mA (0.01 C). Once the cell reached 100% SOC, it was left at rest for 2 hours to reach thermodynamic equilibrium. Subsequently, a discharge current of 20 A (1 C) was applied until the SOC decreased of 10%. The cell was then allowed to rest for another 2 hours at this new SOC level. This discharge-relaxation sequence was iterated every 10% SOC step until 0% was reached. Finally, the cell was fully recharged using the CC-CV method, and the thermal chamber was set to the next reference temperature. This procedure yielded voltage, current, and temperature datasets across five temperatures and eleven SOC breakpoints (from 0% to 100%).

The OCV V_{OC} at each SOC level was extracted as the voltage measured at the end of the 2-hour rest phase (Figure 3.1). The parameters R_0 , R_1 , R_2 , C_1 , and C_2 , related to both the first-order and the second-order Thevenin models, were obtained by fitting the voltage transient curve during the relaxation phase. During this phase, the voltage response can be formulated as:

$$v_t = V_{OC,t} + R_0 \Delta i + \sum_{k=1}^{n_{RC}} R_k \Delta i \left(1 - e^{-\frac{1}{R_k C_k}} \right), \quad (3.1)$$

where Δi is the current step and n_{RC} is the number of RC branches of the selected Thevenin model (with $i = 1$ for 1-RC and $i = 2$ for 2-RC). The characterization results are illustrated in Figure 3.2, Figure 3.3, and Figure 3.4. Consistent with established literature, internal resistance drops as temperature decreases (Barcellona et al., 2022a). Specifically, the ohmic resistance R_0 exhibits weak SOC dependence, whereas polarization resistances R_1 and R_2 show a marked increase at low SOC levels (Sakile and Sinha, 2022; Gomez et al., 2011). Furthermore, capacitances C_1 and C_2 demonstrate a positive correlation with temperature, peaking in the mid-SOC region (Tran et al., 2021).

3.1.2 Thermal Model Parameters

The thermal model described in Section 2.3 utilizes a circuit-based representation to approximate heat transfer phenomena within the battery cell. The model architecture relies on two resistive parameters, R_{cond} (conduction) and R_{conv} (convection), and a thermal capacity C_{th} . To accurately simulate thermal evolution, these parameters must be determined alongside the internal entropic variation, defined as $\partial V_{OC}/\partial T$.

The total thermal power generated within the cell, $Q_{gen} \in \mathbb{R}$, is the sum of two contributions: an irreversible component, Q_{irr} , arising from Joule heating (Ohmic losses), and a reversible component, Q_{rev} , associated with the entropy change of the electrochemical reactions. The irreversible heat is derived from the electrical model

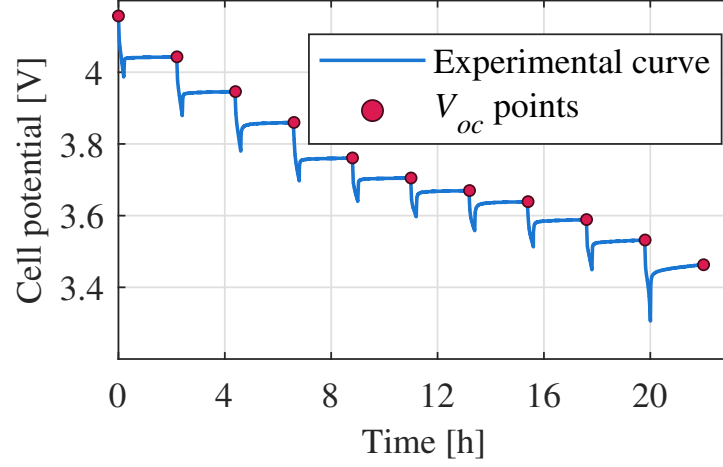


Figure 3.1: V_{OC} points collected during the model characterization at the end of each rest phase.

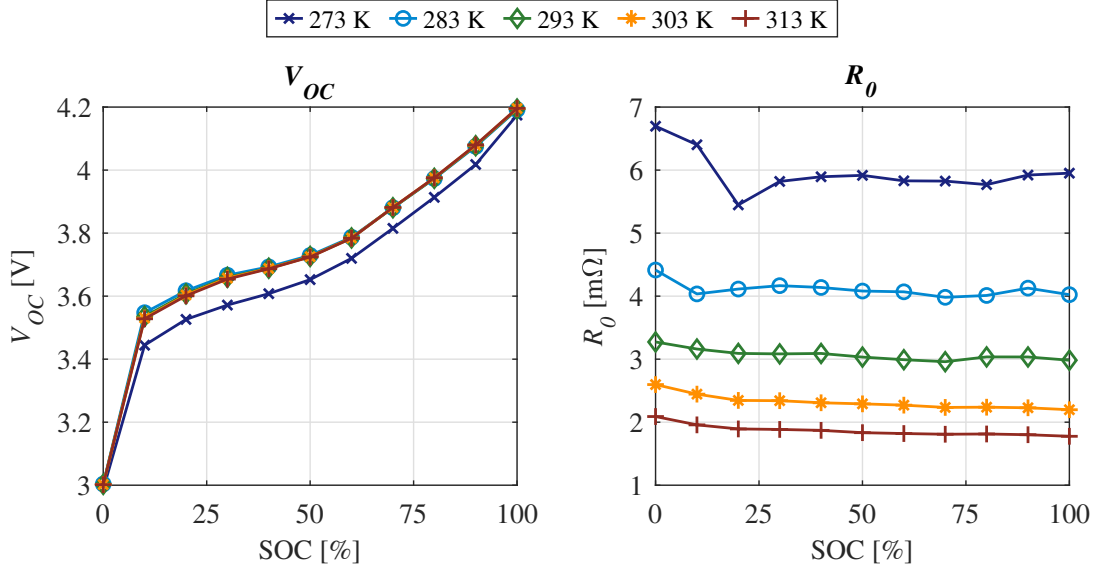


Figure 3.2: Parametrization of V_{OC} and R_0 for 1-RC and 2-RC models at 5 reference temperatures and 11 SOC breakpoints from 0% to 100% with 10% increments.

states as follows:

$$Q_{\text{irr}} = R_0 i^2 + \sum_{k=1}^{n_{\text{RC}}} R_k i_{R_k}^2, \quad (3.2)$$

where $n_{\text{RC}} \in \mathbb{N}$ denotes the number of RC branches in the selected Thevenin equivalent circuit. The total generated thermal power is then computed as:

$$Q_{\text{gen}} = Q_{\text{irr}} + \underbrace{T_{\text{core}} \frac{\partial V_{OC}}{\partial T}}_{Q_{\text{rev}}} i. \quad (3.3)$$

The entropic coefficient $\frac{\partial V_{OC}}{\partial T}$ is a function of the battery SOC. It was determined

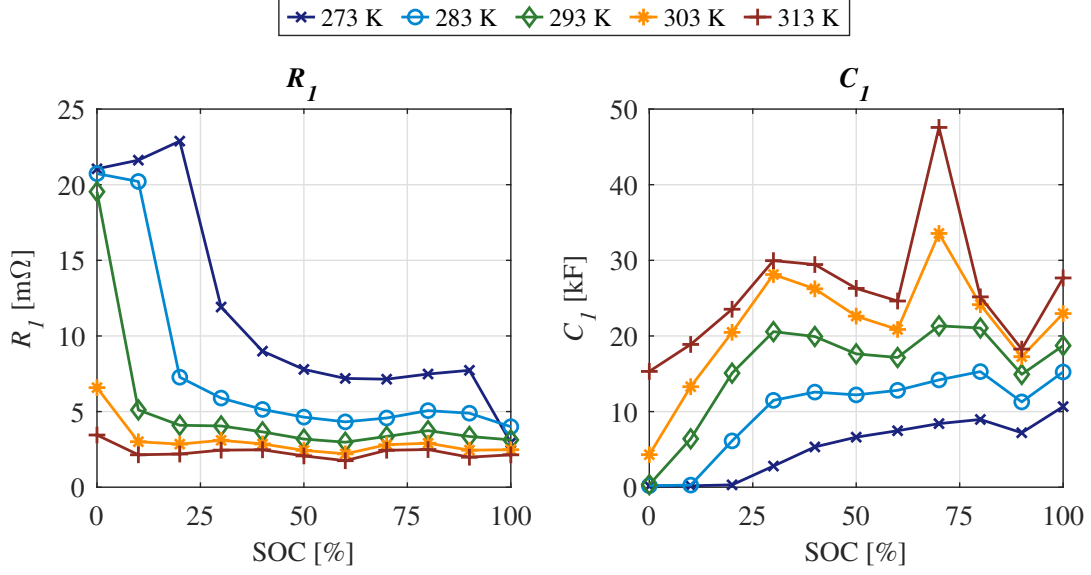


Figure 3.3: Parametrization of R_1 and C_1 for 1-RC model at 5 reference temperatures and 11 SOC breakpoints from 0% to 100% with 10% increments.

experimentally by analyzing the V_{OC} variations across different temperatures, reported in Section 3.1.1. Specifically, the gradient of V_{OC} with respect to temperature was computed at each discrete SOC level. The resulting profile is illustrated in Figure 3.5.

The thermal resistances R_{cond} and R_{conv} are determined based on the cell's geometry and material properties. Under the assumption of steady-state heat flow, they are defined as:

$$R_{cond} = \frac{T_{core} - T_{surf}}{Q_{gen}} \approx \frac{l}{\lambda_{cell} A_{int}}, \quad (3.4)$$

$$R_{conv} = \frac{T_{surf} - T_{amb}}{Q_{gen}} = \frac{1}{A_{cell} h}, \quad (3.5)$$

where $l \in \mathbb{R}^+$ is the characteristic distance between the core and surface, $\lambda_{cell} \in \mathbb{R}_0^+$ is the thermal conductivity of the cell material, $A_{int} \in \mathbb{R}^+$ is the internal cross-sectional heat transfer area, $A_{cell} \in \mathbb{R}^+$ is the total external surface area, and $h \in \mathbb{R}$ is the convective heat transfer coefficient.

The geometric parameters (l , A_{int} , A_{cell}) were measured directly from the physical cell. The thermal conductivity λ_{cell} was adopted from Cui et al. (2020a), based on the analysis of a similar NMC cell format. The computed R_{cond} was found to be extremely small, rendering the thermal gradient between the core and surface negligible. This confirms the validity of the assumption $T_{surf} \approx T_{core}$, as discussed in Section 2.3.

The convective coefficient $h \in \mathbb{R}$ and the total thermal capacity $C_{th} \in \mathbb{R}^+$ were identified experimentally following the procedure detailed by Barcellona et al. (2022b). The cell was subjected to a sequence of 1C charge and discharge pulses in free air, with current direction alternating every 5 minutes. This excitation induces a controlled thermal transient without violating voltage constraints. Once the system reached a quasi-steady-state temperature (as shown in Figure 3.6), the thermal storage term vanishes

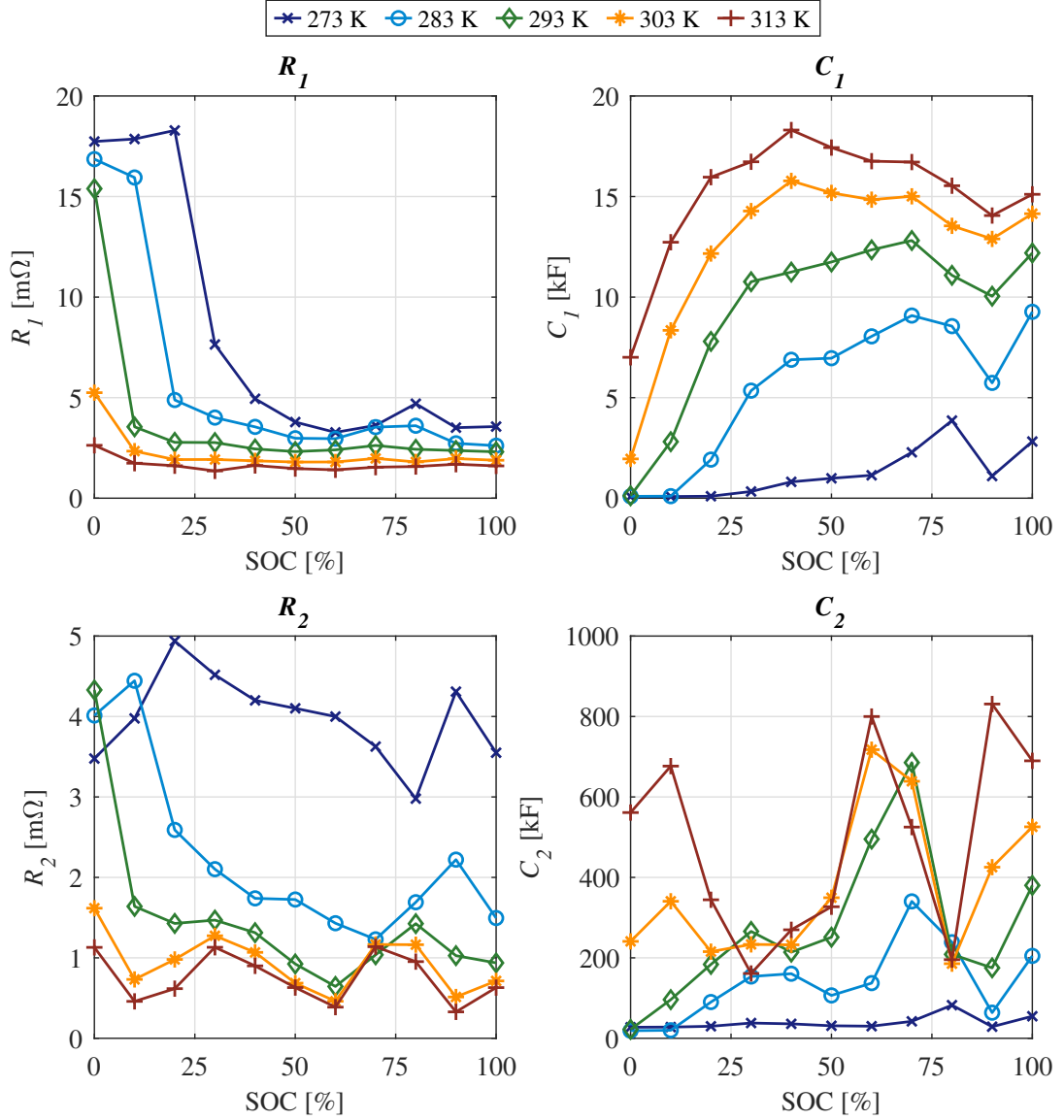


Figure 3.4: Parametrization of R_1 , C_1 , R_2 and C_2 for 2-RC model at 5 reference temperatures and 11 SOC breakpoints from 0% to 100% with 10% increments.

($C_{th} \frac{dT}{dt} \approx 0$). In this condition, the capacitive branch acts as an open circuit, allowing the direct determination of R_{conv} and consequently the convective coefficient, which was identified as $h = 12.02 \text{ W m}^{-2} \text{ K}$. Finally, the thermal capacity C_{th} was derived from the time constant of the transient response, yielding a value of $C_{th} = 838 \text{ J K}^{-1}$.

3.2 Case Studies for Validation

In this section, we present the experimental campaign conducted to validate the configuration of the ERNESTO framework detailed in Chapter 2. The DT's predictive performance is evaluated across four distinct use cases, each specifically designed to stress-test different dimensions of the model's fidelity:

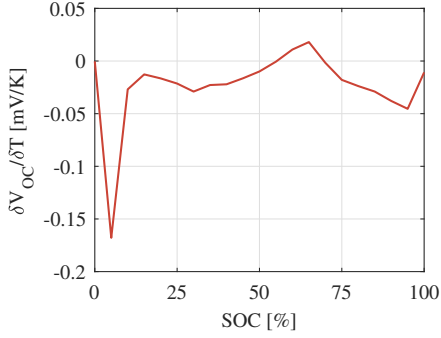


Figure 3.5: Parameter $\partial V_{OC}/\partial T$ versus SOC.

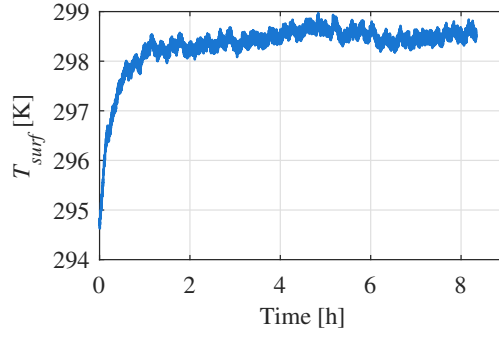


Figure 3.6: Temperature transient evolution.

- (i) **Residential Prosumer Profile (Section 3.2.1):** A dynamic power profile derived from a residential building equipped with photovoltaic (PV) generation. This scenario aims to replicate real-world operating conditions, assessing the DT's ability to track voltage and temperature variations under typical load and generation fluctuations.
- (ii) **Microgrid Integration (Section 3.2.2):** A profile of DER data collected at the RSE's test facility (DER-TF). This profile reflects the power dynamics encountered during MG integration.
- (iii) **Thermal Stress Test (Section 3.2.3):** A specialized test profile designed to provoke a rapid thermal transient, or *thermal jump* (TJ). This scenario is critical for assessing the accuracy of the electro-thermal coupling and the model's ability to track core temperature deviations.
- (iv) **Aging Validation (Section 3.2.4):** A dataset comprised of characterized aged battery cells. This scenario serves to validate the long-term degradation tracking and SOH estimation capabilities of the framework.

To rigorously quantify the tracking accuracy between the Digital Twin and the physical reference, three statistical metrics are employed: the Mean Absolute Error (MAE) to assess average tracking performance, the Mean Absolute Percentage Error (MAPE) to evaluate relative deviations, and the Maximum Absolute Error (MaAE) to identify worst-case divergences, which are critical for safety monitoring. The quantitative results for each experiment are summarized in Table 3.1, while a comprehensive overview of the configuration parameters adopted for each simulation scenario is provided in Table A.2. From a computational perspective, the system models are implemented in discrete-time form, approximating continuous derivatives via the backward finite difference method.

3.2.1 Residential Prosumer Profile

In this experiment, we simulate the operation of a BESS integrated with a residential PV system. The reference power profile (*PV profile* in plots and tables) was selected from real-world German residential data presented in [Tjaden et al. \(2015\)](#) with simulated PV generation. The PV production was modeled assuming a standard rooftop installation area of 30 m², utilizing historical irradiation and temperature data for the German region ([European Commission - Joint Research Centre, 2022](#)). To validate the Digital

3.2. Case Studies for Validation

Experiment		Voltage			Temperature		
Type	Test	MAE[V]	MAPE[%]	MaAE[V]	MAE[K]	MAPE[%]	MaAE[K]
PV	1-RC	0.0185	0.50	0.0448	0.1231	0.04	0.5443
PV	2-RC	0.0186	0.50	0.0446	0.1231	0.04	0.5443
DER-TF	1-RC	0.0072	0.18	0.0166	0.0994	0.03	0.4673
DER-TF	2-RC	0.0072	0.18	0.0165	0.0994	0.03	0.4673
TJ	1-RC w/o temp	0.01568	0.41	0.0460	-	-	-
TJ	2-RC w/o temp	0.0161	0.42	0.0461	-	-	-
TJ	1-RC	0.0102	0.27	0.0302	0.5309	0.17	6.4563
TJ	2-RC	0.0108	0.28	0.0295	0.5309	0.17	6.4563

Table 3.1: DT framework validation results for voltage and temperature tracking in the different scenarios. Metrics of MAE, MAPE, and MaAE are reported.

Twin at the unit level, the resulting net power profile was downscaled to match the nominal capacity of a single cell.

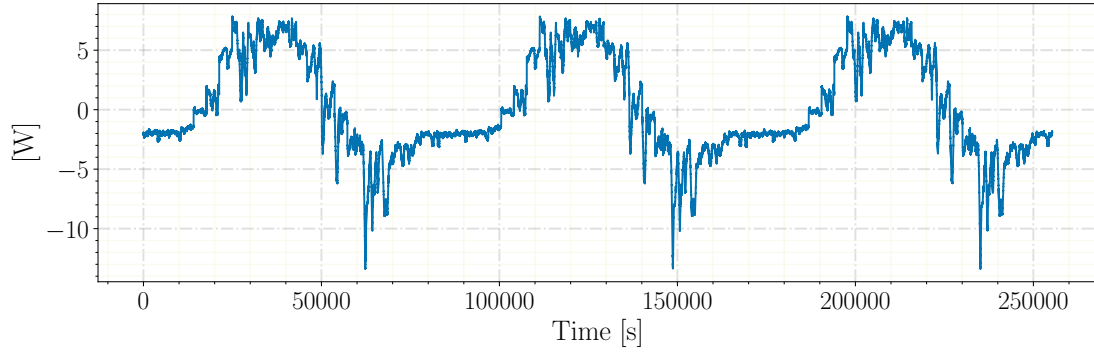
The experimental validation was conducted on an NMC cell at a constant ambient temperature of 293 K using the experimental setup described in Section 3.1. The test protocol proceeds as follows: first, the cell was fully charged via the CC-CV method; subsequently, it was discharged with a constant current of 10 A until 20% SOC was reached. Finally, the scaled residential PV profile, shown in Figure 3.7a, was applied.

As illustrated in Figure 3.7a, the resulting profile exhibits a peak power magnitude below 15 W, but it is characterized by high-frequency, irregular oscillations. These fluctuations are typical of PV systems operating under stochastic real-world irradiance conditions (e.g., cloud passing), providing a robust stress test for the model’s dynamic response.

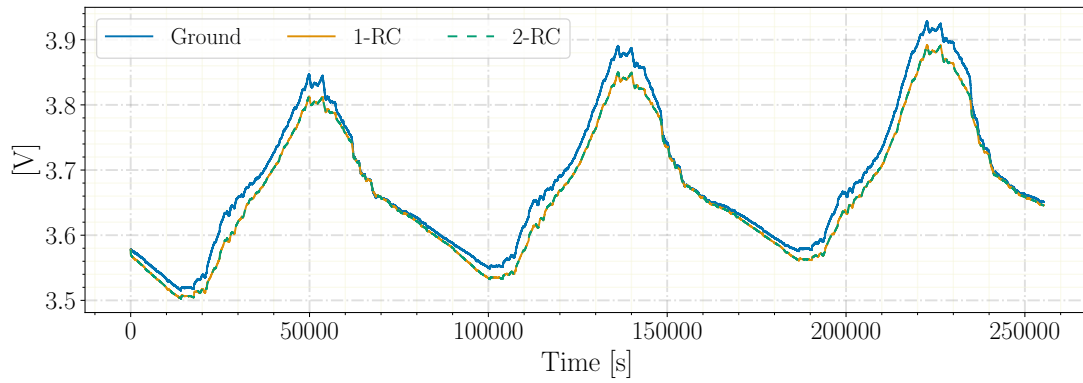
In Figure 3.7b and Figure 3.7c, we present the validation results, comparing the experimental ground truth (labeled *Ground*) against the trajectories estimated by the DT using both the first-order (1-RC) and second-order (2-RC) Thevenin ECMs. Quantitative analysis reveals that both models achieve remarkably high fidelity. The tracking error is minimal with both architectures yielding an MAPE of 0.50% for voltage and 0.04% for core temperature. The MaAE is constrained to approximately 0.044 V and 0.54 K, respectively. Notably, the response curves of the 1-RC and 2-RC models are nearly indistinguishable in the plots. This convergence suggests that for this specific class of load profiles, characterized by relatively low C-rates despite high variability, the dominant dynamics are adequately captured by the first-order time constant, rendering the additional complexity of the second RC branch redundant (Wang et al., 2021). Comprehensive metrics are detailed in Table 3.1.

3.2.2 Microgrid Integration Profile

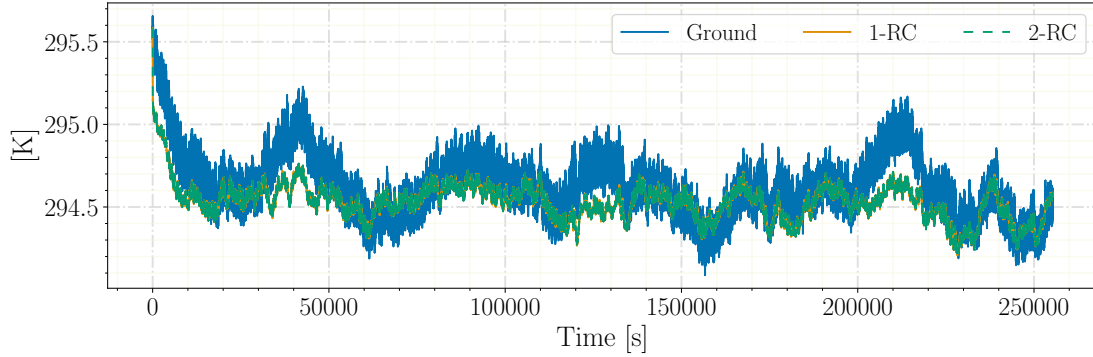
The second validation scenario utilizes data collected from the RSE’s test facility for experiments on Distributed Energy Resources, located at the RSE headquarters in Milan. The test facility (Sandroni et al., 2016) is a low-voltage (400 V), three-phase experimental MG covering an area of approximately 20 000 m². The generation infrastructure includes various RESs, comprising multiple PV panels and a micro wind turbine, with a total renewable generation capacity of 45 kW. Additionally, an internal combustion generator, in a cogeneration configuration powered by natural gas, provides an extra 50 kW of active power. The facility also features a range of energy storage batteries, including Lithium-



(a) PV power profile.



(b) Comparison of voltage signal.



(c) Comparison of temperature signal.

Figure 3.7: Power profile and signals comparison for the residential prosumer profile experiment. The test was conducted with a profile covering almost 3 days of operational data. Results for both 1-RC and 2-RC models show high fidelity tracking capabilities of the real-world signals.

Ion, Redox Flow, Lead-Acid, and Sodium-Nickel Chloride (NaNiCl) technologies. The total bidirectional storage capacity is 180 kV A, with an energy capacity of 230 kW h. The network is interconnected with controllable load banks capable of simulating varying resistive, capacitive, or reactive loads, with active and reactive power up to 100 kW. Finally, the test facility features a low-voltage direct current (DC) MG, interconnected with the main alternating current (AC) MG. The DC MG includes an electrical load, energy storage systems, and a PV simulator. The interconnected network is equipped with multiple switches and disconnectors, allowing for flexible reconfiguration of the

MG topology, which can act either in grid-connected or islanded mode.

For this experiment, we utilize a historical operation profile where the MG controller aimed to minimize the power exchange with the external grid, operating in a peak shaving/self-consumption mode (Lazzari and Bella, 2023). This operational regime places significant dynamic demands on the storage system, which must alternate between quasi-steady-state intervals and high-frequency transient regulation. To validate the DT at the cell level, the power profile of the facility’s BESS was downscaled to match the nominal capacity of a single reference cell, as illustrated in Figure 3.8a. This scaled profile was then physically applied to the test cell using the experimental setup described in Section 3.1.

The DT tracking performance in these settings is visualized in Figure 3.8b and Figure 3.8c. Here, we compare the ground-truth experimental data against the results of *1-RC* and *2-RC* DT models for voltage and temperature, respectively. Also in this scenario, ERNESTO exhibits robust tracking capabilities. As evidenced in Table 3.1, the predictive accuracy surpasses that of the residential use case. Specifically, the voltage estimation achieves a MAPE of just 0.18% and a MaAE of 0.016 V. Thermal tracking accuracy, instead, remains consistent with previous results. Notably, the negligible performance gap between the first- and second-order models reinforces the observation that, for standard MG regulation profiles, the *1-RC* architecture provides a sufficient balance between computational complexity and fidelity.

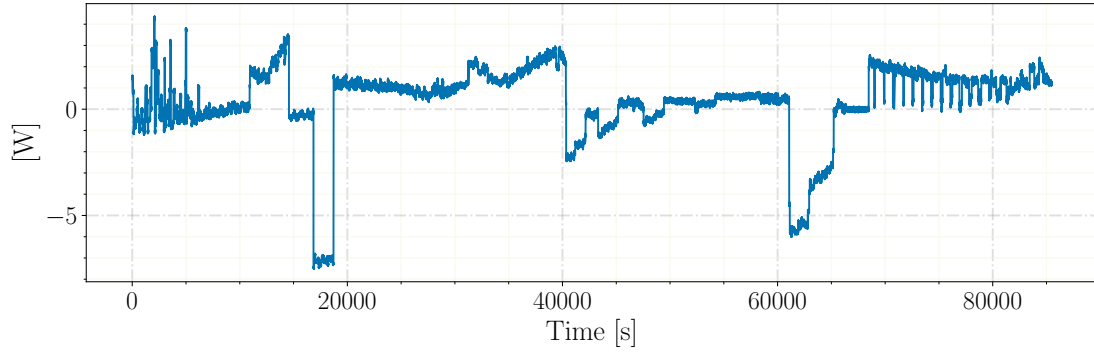
3.2.3 Thermal Stress Test Profile

While the previous experiments focused on evaluating the DT’s tracking capabilities under standard conditions, this specific analysis aims to assess the marginal gain provided by using the selected sub-models. Specifically, while the electrical model is essential for accurately replicating the system’s operational behavior, the benefit coming from incorporating a thermal model may not be immediately clear. To address this question, a *thermal jump* experiment was conducted, i.e., a scenario with a significant ambient temperature change. This experiment was designed to examine the impact of thermal dynamics on battery performance and assess whether integrating a thermal model provides meaningful improvements in simulation accuracy.

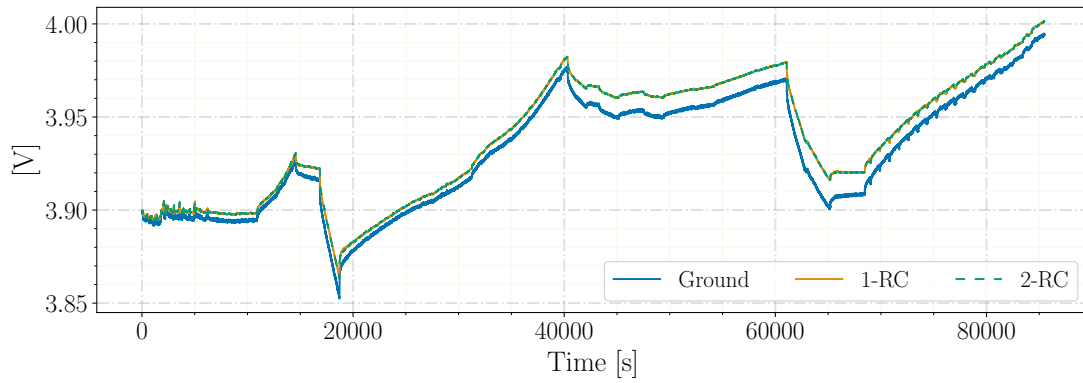
In this scenario, the battery was fully charged using a CC-CV protocol, then discharged at 10 A down to 3.9 V. Afterward, the battery underwent charge and discharge steps of 20 minutes with a current of 4 A. At 19 500 seconds from the beginning of the test, i.e., after more than five hours, the temperature set-point of the thermal chamber was changed from 313 K to 283 K. This step change is visualized in Figure 3.9a.

To evaluate the impact of the thermal module, we compared two configurations: the complete ERNESTO framework, where electrical parameters for both *1-RC* and *2-RC* models are coupled with the simulated core temperature, estimated with the model presented in Section 2.3, and a restricted isothermal version of the DT (named *w/o temp*) where electrical parameters are locked to the initial ambient temperature, effectively severing the electro-thermal feedback loop.

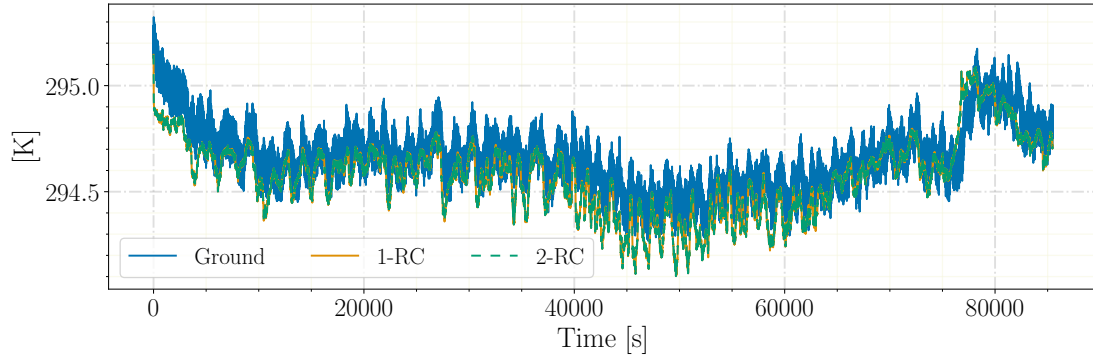
The comparative results for voltage estimation are presented in Figure 3.9b for the *1-RC* architecture and Figure 3.9c for the *2-RC* architecture. Initially, both the coupled and isothermal configurations exhibit negligible tracking errors, as the cell temperature remains close to the initial reference ambient temperature. However, a



(a) DER-TF power profile.



(b) Comparison of voltage signal.



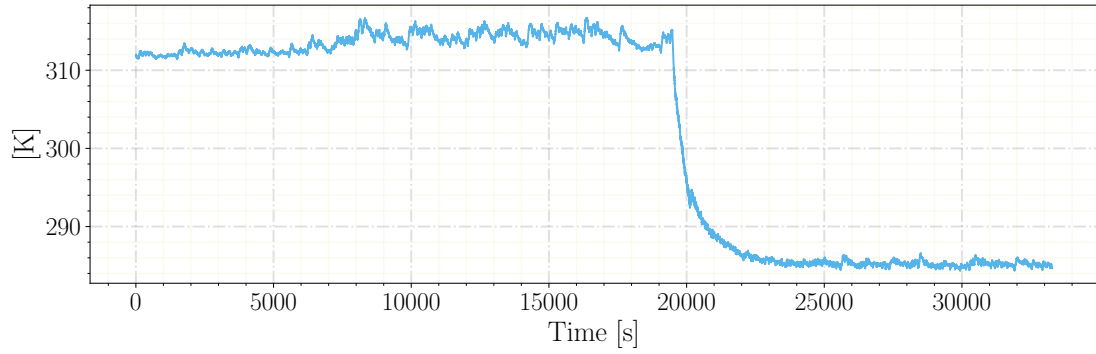
(c) Comparison of temperature signal.

Figure 3.8: Power profile and signals comparison for the DER-TF profile experiment. The test was conducted with a profile covering about one day of operational data. Results for both 1-RC and 2-RC models show high fidelity tracking capabilities of the real-world signals.

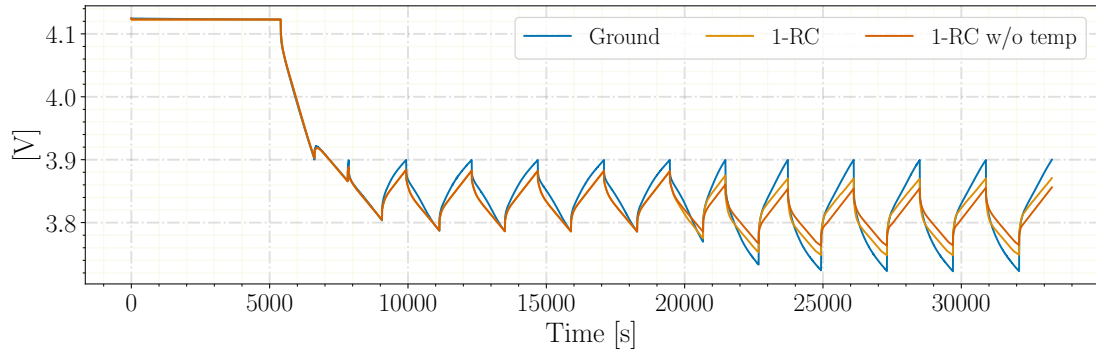
distinct divergence occurs immediately following the thermal jump at approximately 19 500 seconds. As the physical cell cools, its internal impedance rises significantly. The coupled DT accurately tracks this physical phenomenon, adjusting its resistance parameters in real-time. In contrast, the isothermal model continues to operate using parameters neglecting the thermal effects, leading to a systematic underestimation of voltage drops.

Quantitatively, this divergence is substantial. While the coupled models maintain a stable MAPE of approximately 0.27% throughout the entire test, the isothermal baselines

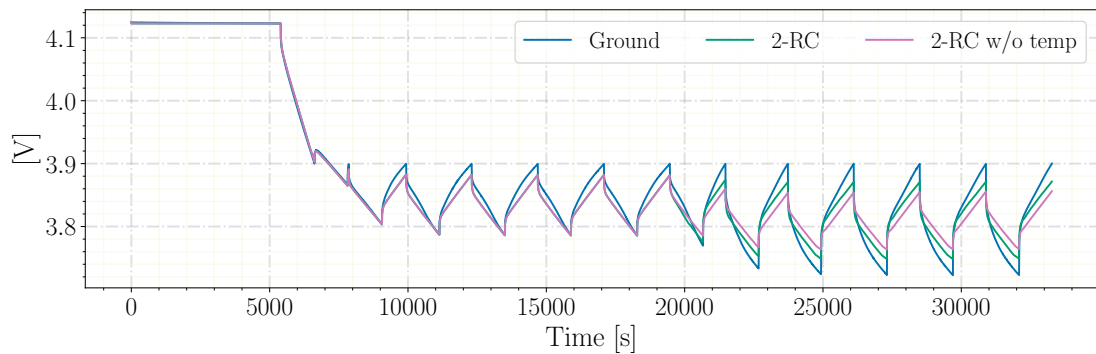
suffer significant degradation in accuracy. Specifically, in the region following the temperature shift, the error of the uncoupled models spikes to 0.72% for the 1-RC and 0.73% for the 2-RC architectures. These results empirically validate the architectural necessity of the thermal sub-model. For BESS applications exposed to outdoor environments or variable cooling conditions, neglecting electro-thermal coupling results in significant state estimation errors.



(a) Ambient temperature dynamics.



(b) Comparison on voltage signal between versions of 1-RC coupled with thermal model and in isothermal conditions.



(c) Comparison on voltage signal between versions of 2-RC coupled with thermal model and in isothermal conditions.

Figure 3.9: Ambient temperature dynamics and signals comparison for the TJ profile experiment. The test was conducted with a profile covering about 9 hours of operational data, with a sudden temperature change applied at around 5.4 hours. Results show the importance of including the thermal model to accurately track voltage signals under thermal stress conditions.

3.2.4 Aging Validation Profile

The final validation scenario focuses on assessing the DT's capability to accurately model battery aging phenomena over extended operational periods. The test is performed using data obtained from an accelerated aging experiment conducted on an NMC cell from the same batch as the one used for electrical and thermal characterization. This dataset consists of three characterization phases interspersed with two aging periods. The characterization phases are conducted by applying the profile described in Section 3.1.

Following the methodology proposed in [Barcellona et al. \(2023\)](#), the V_{OC} points extracted during the characterization phase are plotted as a function of the absolute state of discharge q , expressed in Ah. The variable q is computed by integrating the current throughput over time:

$$q = \frac{1}{3600} \int_{t_i}^{t_f} i_{\tau} d\tau + q_{t_i}, \quad (3.6)$$

where t_i and t_f define the integration window. By interpolating these discrete points, the OCV curve with respect to the absolute state of discharge $V_{OC}(q)$ is reconstructed, as illustrated in Figure 3.10. The maximum capacity of the battery at any given aging stage is physically defined as the value of q corresponding to the lower cut-off voltage limit ([Barcellona et al., 2023](#)). The aging phases consist of a series of charge/discharge cycles performed at a current of 18 A, constrained between 10% and 90% of SOC. Both the characterization and aging phases were conducted in a controlled environment at 298 K.

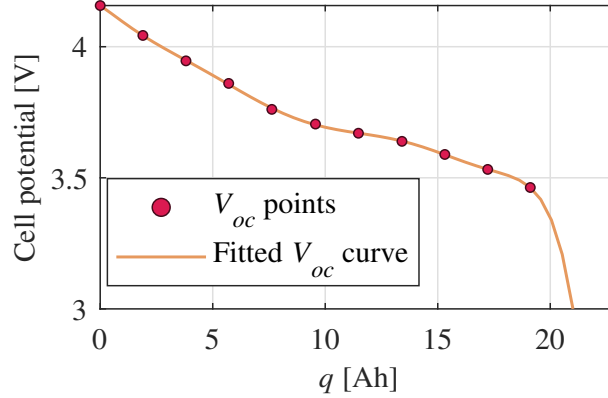


Figure 3.10: Reconstruction of the OCV curve as a function of the absolute state of discharge q at different aging stages. The maximum capacity at each stage is identified as the value of q corresponding to the lower cut-off voltage.

It is important to note that the test cell was not a fresh cell, but had previously undergone preliminary cyclic and calendar aging. Consequently, the simulation assumes that the degradation component associated with the initial SEI layer formation has already stabilized. As established in literature ([Xu et al., 2018](#)), initial SEI growth is a transient phenomenon dominant only in the early lifecycle. Accordingly, the model parameter α_{SEI} in Equation (2.10) is initialized to zero. All other aging parameters are detailed in Table A.2.

The validation of the aging model is presented in terms of the estimated SOH in Table 3.2. The framework, operating with fully coupled electrical and thermal dynamics,

3.2. Case Studies for Validation

Check-up	Ground	DT	Error[%]
0	1.00	1.00	0.00
I	0.9832	0.9779	0.54
II	0.9559	0.9539	0.20

Table 3.2: Aging validation results with corresponding prediction error on SOH. The experiment, conducted to validate the aging model, covers a total period of about 70 days, with two aging phases interspersed with three characterization phases (at checkpoints 0, I, and II).

demonstrates high fidelity in tracking capacity fade. The SOH estimation error remains consistently below 0.6%, empirically validating the hypothesis regarding the stability of the SEI layer. The marginal residual error is attributable to the natural propagation of uncertainties from the electrical and thermal sub-models.

To assess the long-term stability of the system, we evaluate the voltage and temperature tracking errors at two reference points along the aging profile: the midpoint (approximately after one month) and the endpoint (approximately after 70 days). The electrical model exhibits remarkable stability, with the MAE increasing only by 0.0115 V over the 40-day interval, corresponding to a MAPE increase of 0.35%. In contrast, the thermal model shows a slight decrease in temperature estimation error (MAE -1.0716 K, and MAPE -0.25%), remaining within the same order of magnitude throughout the profile. These metrics confirm ERNESTO’s robustness against strictly monotonic error accumulation (drift), ensuring reliable long-term performance for long-term prognostic applications.

Part II

Machine Learning Integration for Real-Time Monitoring and Adaptation

CHAPTER 4

Monitoring and Tracking of Energy Storage Systems

In the first part of the dissertation, we presented a DT framework for BESSs and we validated its effectiveness in mirroring and simulating the physical counterpart's behavior. However, as highlighted in [Semeraro et al. \(2023a\)](#), a DT is not just a static replica of the physical system or a mere simulation environment. Instead, it should be capable of spanning the entire lifecycle of the physical asset, continuously adapting to changes and maintaining high fidelity over time. Indeed, the potential discrepancies between the DT and the physical system may compromise the effectiveness of the DT in different tasks, such as decision-making and control, predictive maintenance, fault detection, and overall system optimization. Moreover, having an accurate and up-to-date model of the BESS allows for better anticipation of potential issues and improved system reliability, making it possible to recognize early signs of faults and take proactive measures to mitigate their impact.

However, monitoring and tracking the physical process is a challenging task, especially when dealing with real-world systems influenced by non-stationary conditions. As we previously discussed, BESSs are subject to various factors (degradation, environmental conditions, and operational modes) that can alter their behavior over time. At the same time, model parameters associated with the new dynamics are not easily accessible due to *verification latency*, i.e., the delay between observing a change in the physical system and updating the DT accordingly. This is because internal state indicators of the BESS, such as capacity and internal resistance, are only observable through a controlled full-discharge test, which is operationally disruptive and rarely available at runtime ([Tan et al., 2025](#)). This introduces a temporal gap between the DT prediction of the behavioral shift, its verification on the real-system and the model update.

In this part of the dissertation, we study, design, and empower ERNESTO with

real-time tracking mechanisms, trying to answer to objective *RO2*, described in Section 1.4.1. In particular, the goal is to equip the DT with the ability to detect deviations from its expected behavior and recalibrate model parameters to adapt to the current operating dynamics. This involves adopting different statistical procedures, including ML-based methodologies, making the model-based DT presented in Part I a proper hybrid approach, leveraging the capabilities of data-driven methods. The work presented in this part is a novel contribution of this dissertation, which will bring to the submission of a journal paper.

Chapter Outline. The chapter is organized as follows. In Section 4.1, we introduce the concept of online monitoring and drift detection, discussing the challenges associated with non-stationary data streams. Next, in Section 4.2, we review existing methods for battery parameter estimation, monitoring, and recalibration, highlighting benefits and limitations.

4.1 Online Monitoring and Drift Detection

In non-stationary environments, such as energy storage systems operating under varying load profiles and degradation states, static models inevitably degrade over time. To maintain system reliability, it is strictly necessary to implement *online monitoring* frameworks that process data streams in real time. This section provides a formal overview of the fundamental concepts related to monitoring, specifically focusing on distinguishing between *concept drift* (system evolution) and *anomalies* (faults or outliers).

4.1.1 Concept Drifts

The phenomenon of the underlying data distribution changing over time is formally known as *concept drift* (Gama et al., 2004). In a supervised learning context, let $X_t \in \mathbb{R}^d$ be the input feature vector (e.g., current, temperature) and y_t be the target variable (e.g., voltage, SOC) at time t . The data stream is modeled as a sequence of pairs $\{(X_t, y_t)\}_{t=1}^{\infty}$ generated by a joint probability distribution $P_t(X, y)$. The concept drift is defined as the change in this joint distribution between two time steps (Gama et al., 2014; Lu et al., 2018), formally:

$$\exists t : P_t(X, y) \neq P_{t+1}(X, y). \quad (4.1)$$

Using the Bayesian decomposition $P_t(X, y) = P_t(y|X)P_t(X)$, we can categorize the nature of the change into two distinct phenomena (Moreno-Torres et al., 2012):

- **Virtual Drift (Covariate Shift):** Changes occur only in the input distribution $P_t(X)$, while the posterior relation $P_t(y|X)$ remains constant. In battery systems, this effectively corresponds to a change in the *operating conditions* (e.g., a shift from summer to winter temperatures) without a change in the underlying physics of the system.
- **Real Concept Drift:** Changes occur in the posterior distribution $P_t(y|X)$. This implies a fundamental change in the system's behavior (e.g., internal resistance increase due to aging), necessitating a model update to recover predictive performance.

Drifts are further categorized based on their temporal dynamics, which dictate the urgency and type of the adaptive response. As illustrated in Figure 4.1, common drift types include:

- **Abrupt Drift:** A sudden, instantaneous shift in the distribution which can be associated with events like sensor faults or sudden load changes.
- **Incremental Drift:** A continuous, slow evolution over time that may reflect gradual modifications in system behavior, such as battery degradation.
- **Recurrent Drift:** Concepts that reappear periodically over time, such as seasonal phenomena.
- **Gradual Drift:** A transition period where the process oscillates between the old and new distributions before stabilizing.

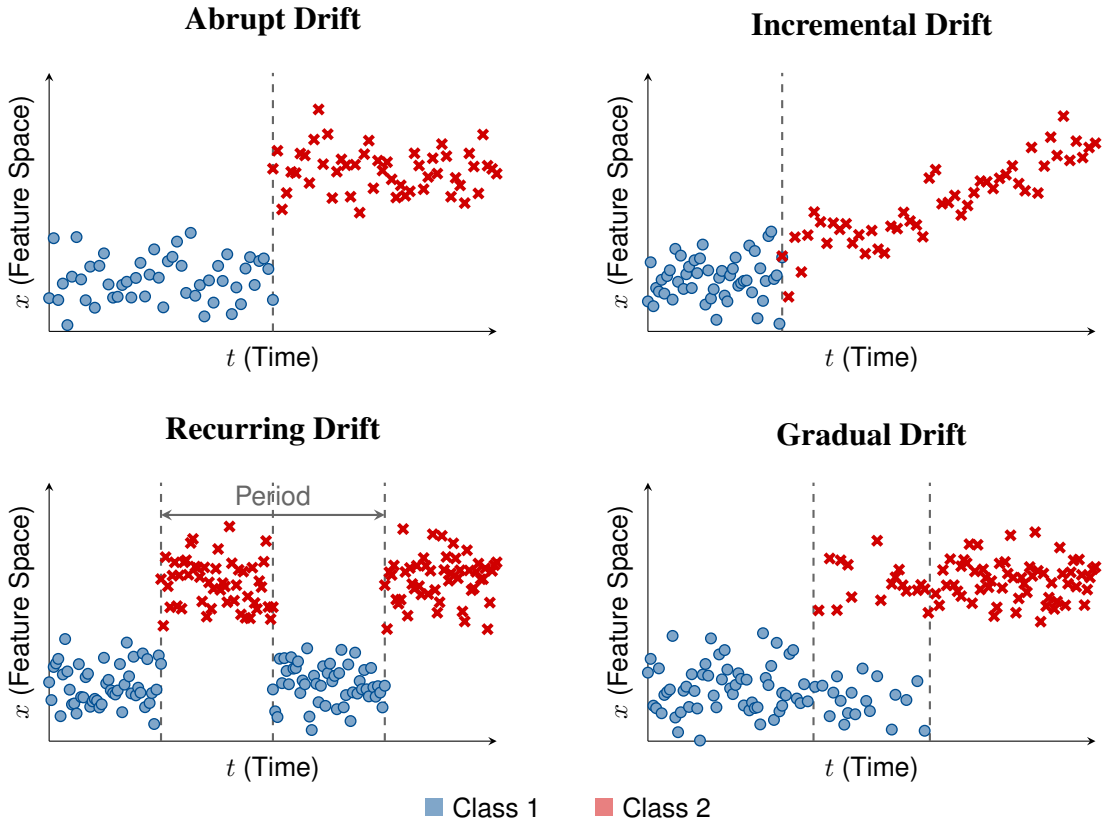


Figure 4.1: Taxonomy of concept drifts in data streams based on temporal dynamics. We illustrate abrupt, incremental, gradual, and recurrent drifts by splitting the points into two classes, representing the distributions before and after each drift.

In Figure 4.1, we visualize these drift types in a one-dimensional feature space for clarity. Each subplot illustrates how the data distribution evolves over time, highlighting the distinct characteristics of each drift type.

Each drift type may require different detection and adaptation strategies. For instance, abrupt drifts may necessitate immediate model retraining, whereas incremental or gradual drifts may be better handled through continuous adaptation mechanisms.

Finally, recurrent drifts could benefit from storing and reusing previously learned models corresponding to known concepts.

4.1.2 Robust Anomaly Detection

While concept drift refers to the evolution of the underlying data distribution, *anomaly detection* focuses on identifying individual data instances that deviate significantly from the majority of the data. In real-world systems, these deviations often serve as early indicators of faults, cyber-attacks, or the beginning of rapid degradation phenomena.

Formally, given a dataset $\mathcal{D}$ generated by a probability distribution $P(\cdot)$, an anomaly is an observation x_t such that $P(x_t)$ is below a threshold τ . However, in complex time-series data, simple probabilistic thresholds are often insufficient. The literature typically categorizes anomalies into three distinct types (Chandola et al., 2009):

- **Point Anomalies:** This is the most straightforward type, where a single data point significantly deviates from the global distribution. For instance, a sudden spike in battery voltage that exceeds the nominal operating range.
- **Contextual Anomalies:** If data points are considered in a specific context, some points may be anomalous only within that context. For example, a battery temperature of 40 °C is nominally valid, but if observed during a resting period, it indicates a thermal runaway or sensor fault.
- **Collective Anomalies:** A sequence of data points that are individually valid but form an anomalous pattern when considered together (e.g., a voltage signal that remains perfectly constant for minutes, indicating a frozen sensor).

4.1.3 Drift and Anomalies in Battery Systems

In the specific domain of BESS, the abstract concepts of *drift* and *anomalies* map directly to distinct physical phenomena, such as *degradation* or intermittent sensor data losses. Understanding this physical correspondence is crucial for designing monitoring systems that can differentiate between a battery entering a new operational regime (requiring model adaptation) and one failing (requiring maintenance).

Virtual and Real Drifts in Batteries

In a supervised learning context where a model $f : \mathcal{X} \rightarrow \mathcal{Y}$ predicts battery states (e.g., terminal voltage or SOC), general drift is defined as a change in the joint probability $P_t(X, Y)$. As mentioned above, this can be decomposed into two distinct phenomena with different physical interpretations: *virtual drifts*, corresponding to changes in operating conditions (covariate shift), and *real concept drifts*, corresponding to the irreversible evolution of electrochemical parameters (aging).

Virtual Drift. Virtual drift in BESS is characterized by a change in the input feature distribution $P_t(X)$ while the underlying system dynamics $P(Y|X)$ remain constant. This is typically induced by external factors such as seasonal temperature fluctuations or variations in load profiles. Since battery performance is highly non-linear with respect to temperature, a shift in $P_t(X)$ can lead to drastic changes in model performance without

any actual degradation in the battery's health. For instance, a voltage prediction model trained exclusively on summer data (high T_{amb}) may significantly underperform when deployed in winter (low T_{amb}) due to the unlearned increase in internal resistance at low temperatures. This type of drift is often *recurrent* (e.g., annual seasonal patterns) or *gradual* (e.g., weekly weather changes). However, *abrupt* virtual drifts may occur during extreme weather events or grid outages, necessitating rapid domain adaptation.

Real Concept Drift. Real concept drift occurs when the posterior distribution $P_t(Y|X)$ changes, meaning the relationship between inputs and outputs has fundamentally altered. In BESS, this is the mathematical manifestation of *aging*. Even if the operating conditions X (i.e., current, temperature) remain identical, the target output Y (e.g., terminal voltage, capacity) evolves due, for example, to *impedance rise* or *capacity fade*. In the former case, the internal resistance (R_{int}) increases due to SEI film growth. Consequently, for the same input current and temperature (X), the observed terminal voltage (y) decreases. A model trained on BOL data will systematically over-predict the voltage, accumulating error over time. In the latter case, the loss of active lithium inventory shifts the OCV curve relative to the total throughput. This represents a fundamental change in the system's dynamics, invalidating static models.

In BESS, this drift is predominantly *incremental* and monotonic, following known degradation trajectories (Barré et al., 2013). However, *abrupt drift* can occur during maintenance events, such as the replacement of a module, which instantaneously resets the health parameters.

Anomaly Detection as Fault Diagnosis

While drift represents a legitimate evolution of the system's physics, *anomalies* in BESS typically correspond to *unwanted* or *pathological* states that deviate from the expected electrochemical behavior. Anomalies generally fall into two categories relevant to safety, namely sensor/communication faults, which are *point anomalies*, and thermal/electrical faults, i.e., *contextual anomalies*.

The first category includes data artifacts, such as a voltage sensor reading 0 V due to a loose wire, or missing packets in the BMS stream. These must be filtered out to prevent them from corrupting the model during online learning. The second category comprises anomalies that are precursors to catastrophic failure. For instance, an internal short circuit may manifest as a slightly higher self-discharge rate or a localized temperature rise during rest. While a temperature of 35 °C is normal during operation, it is an anomaly if observed while the current is zero for an extended period. Detecting these phenomena requires multivariate correlations rather than simple thresholds.

Distinguishing Drift from Anomalies

A critical challenge in BESS monitoring is the *ambiguity zone* where distinguishing drift from anomalies is non-trivial. For example, *lithium plating* can cause a sudden, sharp drop in voltage during charging. If interpreted as an *anomaly* (outlier), the data might be discarded, missing a critical safety warning. If interpreted as *drift* (aging), the model might adapt to it, essentially learning that lithium plating is normal behavior. Therefore, robust BESS monitoring requires a hierarchical approach: strictly rejecting

point anomalies (physics violations) while carefully evaluating persistent structural changes as potential degradation (drift).

4.2 Existing Approaches

In this section, we review the state of the art regarding the monitoring and tracking of physical systems' evolution. We specifically analyze existing methods and DTs that implement parameter estimation, adaptive learning strategies, and drift detection mechanisms in the context of BESS.

4.2.1 Parameter Identification and Online Adaptation

The literature on battery monitoring focuses on addressing the observability problem: estimating internal, unmeasurable states (SOC, SOH, internal resistance, and capacity) from external measurements (voltage, current, temperature). We categorize existing works into two main groups: *static calibration* methods, which characterize the model before deployment, and *online estimation* methods, which continuously adapt the model during operation.

Static Calibration and Optimization. Works in this category focus on identifying the optimal parameters of an ECM, typically during an offline training phase or strictly controlled initialization, as we did in Section 3.1. [Hossain et al. \(2019\)](#) employ parameter identification theory to characterize a first-order Linear Time-Invariant (LTI) model, treating the battery as a static system over short intervals. Similarly, [Ma et al. \(2020\)](#) propose a method to find optimal Thevenin parameters by minimizing the deviation in steady-state response curves corresponding to the power flow computation. Solutions based on genetic algorithms are also used, for instance, by [Peng et al. \(2018\)](#), which adopt Particle Swarm Optimization (PSO) to identify model parameters. Notably, while these methods provide accurate baselines, they are inherently limited by their *static nature*: these solutions are useful at training time to calibrate the model before deployment, but may be less suitable for online adaptation due to computational complexity or the need for specific load profiles.

Online Estimation and Self-Adaptation. To address aging and environmental variability, recent research has shifted toward *adaptive observers* that update parameters in real-time, realizing a self-adapting model. A foundational approach is the use of dual Extended Kalman Filters (EKF) ([Plett, 2004](#)), which estimate states and parameters concurrently. [Yun et al. \(2019\)](#) improved upon standard recursive methods by using lower-upper factorization to accelerate the solution of high-dimensional linear systems for online identification. Data-driven approaches are also gaining traction. [Cadini et al. \(2019\)](#) proposed a Multi-Layer Perceptron (MLP) to predict battery RUL, using Particle Filtering (PF) to update the network's weights online as new data arrives. Similarly, [Mussi et al. \(2021\)](#) introduced the Voltage Dynamic Based SoC Estimation (VDBSE), which employs a sliding-window learning method to dynamically adjust Thevenin parameters and predict the SOC. However, a major drawback of these methods is their *passive* nature. Lacking a formal *change detection* layer, they risk interpreting transient sensor faults or measurement noise as genuine physical evolutions, thereby potentially

corrupting the model.

4.2.2 Adaptive Digital Twin Frameworks

While parameter estimation is mature, integrating these techniques into a comprehensive DT framework is a more recent development (Das et al., 2024). Li et al. (2020) proposed a cloud-based DT that acts as a virtual BMS, employing a hybrid model (ECM + PSO) for online SOH estimation. More recently, Fu et al. (2024) developed a self-adaptive DT combining PSO with an EKF. In this framework, the DT continuously minimizes the error between predicted and actual voltage to update its internal parameters. A critical limitation in these frameworks is the handling of *verification latency*. Most adaptive DTs assume that the true state is eventually available to correct the model. In reality, capacity fade is unobservable during standard grid operations, making supervised adaptation difficult without identifying specific health indicators from partial data.

4.2.3 Fault Detection and Novelty Identification

While adaptation mechanisms handle gradual drifts, *Fault Detection and Diagnosis (FDD)* is required to identify abrupt anomalies or safety-critical failures. The FDD literature is vast and generally divided into supervised and unsupervised paradigms (Liu et al., 2023; Hong et al., 2024; Adasah et al., 2024).

Supervised Classification. Standard approaches rely on learning from labeled historical data containing known fault patterns. Zhao et al. (2017) and Zhang et al. (2020) use Deep Neural Networks (DNNs) to classify specific failure modes (e.g., internal short circuits, sensor faults), while Wu et al. (2017) and Ran et al. (2018) apply rule-based fuzzy logic systems. However, these methods suffer from a major limitation: they require exhaustive labeled datasets of all possible faults, which are rarely available in real-world applications.

Unsupervised Novelty Detection. In dynamic environments where future fault types are unknown a priori, unsupervised *novelty detection* is preferred (Pimentel et al., 2014). These methods learn the manifold of nominal state operation and flag deviations as anomalies. One common branch of research involves *Reconstruction-Based* methods. Algorithms, such as Autoencoders (AE) or LSTMs, are trained to reconstruct normal operational data, with high reconstruction errors indicating a fault. For instance, Li et al. (2025) employ a stacked autoencoder to extract features and detect inconsistencies in battery voltage curves without requiring fault labels.

The second branch, most relevant to this thesis, involves *Clustering-Based* approaches. A significant contribution is provided by Nie et al. (2025), who propose a fault detection algorithm based on multi-dimensional feature clustering fusion. By integrating an adaptive sliding time window, they transform the binary classification problem into a continuous *fault-free scoring* problem, evaluated through a hierarchical warning mechanism with mild and hard thresholds. Peng et al. (2022) advanced fleet monitoring by introducing Operating Condition-aware (OC-aware) feature extraction, which decouples external variability from internal faults. Most notably, Peng et al. (2024) extended this work to explicitly account for an evolving environment. They use an adaptive anomaly detector that distinguishes between transient noise and concept drift.

Despite these advancements, a critical gap remains: existing unsupervised works typically focus on rejecting anomalies or flagging them for inspection. Few frameworks leverage *evolving clustering* to actively incorporate these coherent anomaly groups into the model as new, valid operating modes (e.g., a degraded but safe state). This capability, which transforms identified drift into knowledge through active learning, is the core focus of our proposed framework.

CHAPTER 5

Cluster-based Real-Time Battery Tracking

The previous chapter established the critical need to equip DTs with robust monitoring and adaptation capabilities. After formulating the drift detection problem and reviewing the relevant literature, we identified a significant research gap: the absence of unsupervised methods that can reliably distinguish between transient sensor anomalies and genuine system degradation.

In this chapter, we present the design and implementation of our proposed solution to address this challenge. Specifically, we frame the adaptation problem as a two-phase process that distinguishes noise from drift with statistical rigor in a fully unsupervised manner. The first phase focuses on *drift identification*, in which the system continuously collects estimates of model parameters to monitor the distribution of the data stream. By leveraging clustering techniques, it aggregates anomalous samples to detect the emergence of new operating modes. The second phase involves *model recalibration*, where, upon confirming a drift, the system triggers an *active* adaptation step by recalibrating the model using information gathered from the identified cluster. This methodology can be formally categorized as a *window-based, active learning approach employing evolving clustering*.

Notably, the evolution of the system is tracked in the *parameter space* rather than the output space. This design choice is motivated by the observation that different operational regimes and degradation states manifest as distinct geometric patterns in the parameter manifold, allowing for more effective characterization of drifts. Moreover, this framework allows the DT to treat sustained deviations not merely as faults, but as novel, valid operating modes (e.g., aged states), enabling the system to adapt and maintain accurate tracking of the battery's dynamics over its entire lifecycle.

Chapter Outline. This chapter is organized as follows. In Section 5.1, we review the

fundamental concepts of online learning, focusing on drift detection methodologies and adaptive learning strategies. Then, in Section 5.2, we detail the proposed cluster-based real-time battery-tracking framework, including the monitoring strategy, fault-diagnosis architecture, and the overall algorithmic procedure.

5.1 Fundamentals of Online Learning

In this section, we describe the most common methods adopted for drift detection. Then, we provide a brief distinction between adaptation strategies, classified as *passive* and *active*.

5.1.1 Drift Detection Methodologies

Detecting drift generally involves monitoring the error rate or a statistical property of the data stream. Standard techniques used in the literature can be broadly classified into three categories: *statistical tests*, *window-based approaches*, and *error monitoring methods*.

Statistical Tests. Among statistical tests, a simple, standard method is the *Page-Hinkley (PH) Test* (Page, 1954), a sequential analysis technique for detecting abrupt changes in the mean of a signal. It is widely used in signal processing due to its low computational footprint. The test tracks the cumulative difference between observed values and their empirical mean:

$$m_t = \sum_{i=1}^t (x_i - \bar{x}_t - \delta), \quad (5.1)$$

where $\bar{x}_t$ is the running mean up to time t , and δ is a tolerance parameter that improves robustness against noise. The detector tracks the maximum value observed so far, $M_t = \max_{i \leq t} m_i$. A drift is detected if the difference exceeds a threshold λ :

$$M_t - m_t > \lambda. \quad (5.2)$$

While extremely efficient and easy to implement, the PH test has some limitations heavily dependent on the manual tuning of parameters λ and δ and it is specifically designed for *abrupt* changes, demonstrating struggles with *gradual* and *incremental* drifts, as the running mean $\bar{x}_t$ may slowly adapt to the change without triggering the alarm.

Another widely used statistical test is the *Kolmogorov-Smirnov (KS) Test* (Massey Jr, 1951), a non-parametric statistical test that compares the empirical cumulative distribution functions (ECDF) F of two data windows. It calculates the maximum vertical distance between the two cumulative distributions as follows:

$$D_{n,m} = \sup_x |F_{ref}(x) - F_{curr}(x)|, \quad (5.3)$$

where $F_{ref}(x)$ and $F_{curr}(x)$ are the ECDFs of the reference window (size n) and the current window (size m), respectively, and $\sup_x$ denotes the supremum over all x . A drift is signaled if $D_{n,m}$ exceeds a critical value defined by the desired significance level α . The KS test is more flexible than PH, as it can detect changes also in the shape and

variance of the distribution, not just the mean. This makes it suitable for identifying a wider range of drift types, including gradual and incremental changes. However, it is computationally more intensive because it requires maintaining and comparing two data batches.

Window-Based Approaches. Window-based techniques maintain a memory of recent samples to estimate current distribution statistics. The state-of-the-art method is *ADaptive WINdowing* (ADWIN) (Bifet and Gavalda, 2007). Unlike fixed-size sliding windows, ADWIN dynamically adjusts its window size W to effectively balance the *stability-plasticity* dilemma. The algorithm repeatedly checks whether the window W can be split into two sub-windows, W_0 (older) and W_1 (newer), such that the difference in their means is statistically significant. In particular, it evaluates

$$|\mu_{W_0} - \mu_{W_1}| > \epsilon_{cut},$$

where ϵ_{cut} is a threshold defined based on the Hoeffding inequality, which depends on the sizes of W_0 and W_1 , and a confidence parameter δ . If this condition holds, W_0 is discarded, effectively signaling a concept shift. The main advantages of ADWIN are its ability to automatically adapt the window size to data stream characteristics and its strong theoretical guarantees regarding false-positive rates. The drawbacks include higher computational and memory overhead compared to simpler methods, as it requires maintaining and evaluating multiple sub-windows.

Error Monitoring Methods. Error-monitoring methods focus on tracking the learner's performance over time rather than on the raw data distribution. A common approach is to use a control chart, such as the *Cumulative Sum* (CUSUM) chart (Shams et al., 2011), which accumulates deviations of the prediction error from its expected value. When the cumulative sum exceeds a predefined threshold, a drift is signaled.

Another popular method is the *Drift Detection Method* (DDM) (Gama et al., 2004), which models the error as a Binomial random variable. It tracks the minimum observed error rate (p_{min}) and standard deviation (ς_{min}), with p_t and ς_t denoting the current error rate and standard deviation at time t . DDM defines two distinct thresholds, namely the *warning level* and the *drift level*. The former, i.e., $p_t + \varsigma_t \geq p_{min} + 2\varsigma_{min}$ indicates a potential drift and the system may start buffering data or querying labels; the latter, i.e., $p_t + \varsigma_t \geq p_{min} + 3\varsigma_{min}$ confirms the drift and triggers the model retraining or reset. While this method is intuitive and easy to implement, it requires labeled data for error computation, making it less suitable for unsupervised scenarios, and it may struggle with very slow drifts where the error rate changes gradually over time.

5.1.2 Adaptive Learning Strategies

Once a monitoring system is in place, the core challenge becomes determining *how* and *when* to update the model to maintain alignment with the evolving data stream. In the literature (Ditzler et al., 2015), adaptation strategies are broadly categorized into two paradigms: *passive* and *active* learning.

Passive Adaptation (Continuous Evolution). Passive approaches assume that the data distribution is continuously changing, whether or not a drift has been explicitly

detected. These methods continuously update the model parameters with each new incoming sample (or batch). Common techniques include *Online Gradient Descent*, *Recursive Least Squares* (RLS) with forgetting factors, and ensemble methods in which member weights are dynamically adjusted based on recent performance. These methods are highly effective at handling *gradual* and *incremental* drifts (e.g., battery aging) because they smoothly track changes. However, they may suffer from *lag* when facing *abrupt* drifts and can be computationally expensive, as the update step is performed continuously.

Active Adaptation (Trigger-Based). Active approaches (also known as *Trigger-Based* or *Informed* adaptation) rely on explicit *drift detection* mechanisms (such as ADWIN or PH) to decide when to intervene. In this paradigm, the model remains static as long as the detector reports a stable distribution. When a drift signal is detected, the system either re-trains the model on a recent window of data or creates a new model to replace the old one. With respect to passive ones, active methods are superior for handling *abrupt* drifts as they can react immediately to significant statistical shifts. They are also computationally efficient during stable periods. However, their performance is heavily dependent on the detector’s sensitivity: a missed detection implies no adaptation, leading to rapid performance degradation.

5.2 Proposed Methodology

In this section, we present our proposed methodology for real-time battery tracking integrated into the DT framework. In particular, we describe and motivate the design choices behind the adopted approach and the overall algorithm structure.

5.2.1 Monitoring the Parameter Space

An intuitive approach to monitoring the evolution of dynamical systems is to observe the residual error between the model prediction and the real system output. In the context of BESS, this translates into monitoring the voltage error between the battery model and the real battery voltage. However, this approach has some limitations. First, it is highly sensitive to sensor noise, leading to false alarms. Second, it heavily depends on the instantaneous load profile, which can be confounded with actual changes in the system’s health. For example, a sudden current spike can cause a transient voltage drop that may be misinterpreted as a fault, even though the battery is functioning normally.

To overcome these limitations, we propose tracking the system’s evolution in the *parameter space* (e.g., R_0, R_1, C_1). This is preferred over the output space because it effectively decouples the fast dynamics of the load profile (*virtual drift*) from the slow variation of electrochemical properties (*real concept drift*). While voltage residuals indicate that a mismatch exists, parameter shifts reveal *why* it occurs, providing a stationary feature space in which healthy operation manifests as dense clusters, significantly enhancing the robustness of unsupervised learning algorithms.

5.2.2 The Fault Diagnosis Framework

Our solution builds upon the theoretical framework of a *fault diagnosis system* (FDS), a monitoring tool used to detect, isolate, and identify faults in complex systems (Isermann,

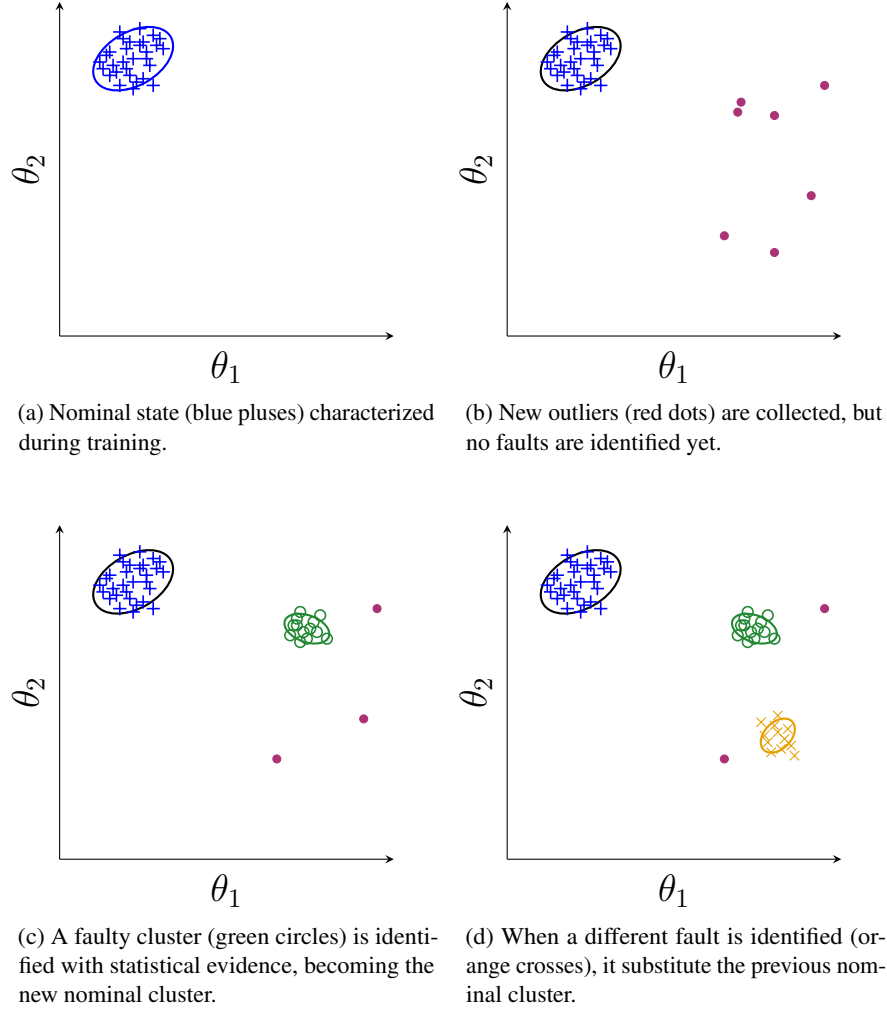


Figure 5.1: Evolving cluster-based fault diagnosis system. The system collects parameter estimates over time, forming clusters that represent different operational regimes. New estimates are compared against the previously identified nominal cluster to detect anomalies. If a sufficient number of anomalies accumulate, they form a new cluster, indicating a drift and triggering model adaptation.

2005; Gertler, 2019; Venkatasubramanian et al., 2003). Standard FDSs require two static components: a *nominal model*, representing the fault-free state collected during a training phase, and a *fault dictionary*, i.e., a database of known fault signatures used to classify anomalies. However, in real-world battery systems, these requirements are rarely met. A static fault-free model is invalid because the *normal* system dynamics evolve continuously. Furthermore, constructing a comprehensive fault dictionary is infeasible, as it requires a priori knowledge of every possible fault mode, assuming no unseen anomalies will arise.

To address these challenges related to the non-stationary nature of battery dynamics, we adopt an evolving-clustering solution inspired by the learning framework of Alippi et al. (2014). In this paradigm, the fault dictionary is not pre-compiled but is *generated at runtime*, while the initial training phase serves only to initialize the first valid cluster (BOL state). Critically, our approach diverges from Alippi et al. (2014) in how it interprets structural changes. While standard methods often assume a time-invariant

setting where any new cluster represents a fault or a recurrent state, we explicitly account for the *irreversible degradation process* of batteries. In our framework, when the data stream drifts and forms a stable new cluster, it is not necessarily flagged as a fault. Instead, as illustrated in Figure 5.1, if the shift follows a physically consistent degradation trajectory, the new cluster is promoted to a *new valid operating mode*. This allows the DT to adapt to aging rather than rejecting it. This distinction is vital for BESS: unlike mechanical systems, where faults might be recurrent (e.g., a valve sticking and unsticking), battery aging is a monotonic drift in the parameter space. Therefore, the system must support an *open-ended* learning structure that continuously incorporates new knowledge without risking forgetting the valid history of the degradation path.

5.2.3 The Proposed Algorithm

In this section, we outline the operational procedure of our proposed monitoring framework. The algorithm executes a continuous control loop that alternates between two principal subroutines: *cognitive fault detection* and *online model adaptation*. A formal description of the overall logic is provided below, with the high-level pseudocode presented in Algorithm 1.

Algorithm 1 ERNESTO’s Evolving Clustering Routine

Input: Initial training data $\mathcal{Z}_M$

- 1: Estimate initial parameter cluster $\Theta_0 \leftarrow \text{OFFLINEIDENTIFICATION}(\mathcal{Z}_M)$
- 2: Compute statistics $\Psi_0 \leftarrow (\text{mean}(\Theta_0), \text{cov}(\Theta_0), M, t_0)$
- 3: $\Psi \leftarrow \{\Psi_0\}$ ▷ List of known states
- 4: $k \leftarrow 0$ ▷ Index of active state
- 5: $t \leftarrow 0$ ▷ Adaptation epoch index
- 6: $\mathcal{O} \leftarrow \emptyset$ ▷ Outlier buffer
- 7: Update DT model $\theta \leftarrow \mu_{\Psi_0}$
- 8: **while** DT is running **do**
- 9: Collect data window $\mathcal{Z}_W = \{(\mathbf{x}_h, y_h)\}_{h=1}^W$
- 10: $\hat{\theta}_t \leftarrow \text{PARAMETERESTIMATION}(\mathcal{Z}_W)$
- 11: is_outlier $\leftarrow \text{COGNITIVEFAULTDETECTION}(\hat{\theta}_t, \Psi_k)$
- 12: **if not** is_outlier **then**
- 13: $\Theta_k \leftarrow \Theta_k \cup \{\hat{\theta}_t\}$ ▷ Update nominal cluster
- 14: Update statistics $\Psi_k \leftarrow (\text{mean}(\Theta_k), \text{cov}(\Theta_k), |\Theta_k|, t)$
- 15: **else**
- 16: $\mathcal{O} \leftarrow \mathcal{O} \cup \{\hat{\theta}_t\}$ ▷ Buffer outlier
- 17: $\Psi_{\text{new}} \leftarrow \text{ONLINEMODELADAPTATION}(\mathcal{O})$
- 18: **if** $\Psi_{\text{new}} \neq \text{NULL}$ **then**
- 19: $\Psi \leftarrow \Psi \cup \{\Psi_{\text{new}}\}$ ▷ Update known states
- 20: $k \leftarrow k + 1$ ▷ Switch to new active state
- 21: $\mathcal{O} \leftarrow \emptyset$ ▷ Clear buffer (or partial clear)
- 22: **end if**
- 23: **end if**
- 24: $\theta \leftarrow \mu_k$ ▷ Recalibrate DT
- 25: $t \leftarrow t + 1$ ▷ Update adaptation epoch index
- 26: **end while**

Initialization and Main Loop

The framework relies on an initial nominal state Ψ_0 , representing the BOL snapshot of the system. We formally define a state Ψ_k as a tuple containing the statistical descriptors of a valid operating mode:

$$\Psi_k = \langle \mu_k, \Sigma_k, n_k, t_k \rangle, \quad (5.4)$$

where μ_k and Σ_k are the centroid (mean vector) and covariance matrix of the parameter cluster, n_k is the support (number of samples), and t_k is the timestamp of the last update. The global history of the system is tracked via the sequence of identified states $\Psi = \{\Psi_0, \dots, \Psi_k\}$. Ψ_0 is initialized during a preliminary training phase using a fault-free dataset $\mathcal{Z}_M = \{(\mathbf{x}_t, y_t)\}_{t=1}^M$, where $\mathbf{x}_t$ denotes the input feature vector and y_t the corresponding output. This yields a set of initial parameter vectors $\Theta_0 = \{\hat{\theta}_1, \dots, \hat{\theta}_M\}$, from which μ_0 and Σ_0 are computed.

During operation, the algorithm estimates a parameter vector $\hat{\theta}_t$ from incoming non-overlapping data windows of size W . This estimate is evaluated against the current active nominal state Ψ_k (the most recent valid mode). If $\hat{\theta}_t$ falls within the acceptance region of Ψ_k , the state statistics are updated incrementally (Lines 13-14). If it falls outside, it is classified as an anomaly and buffered in an outlier set $\mathcal{O}$ (Line 16). Unlike Alippi et al. (2014), we do not maintain a complete fault dictionary; instead, we store only the anomalous samples for clustering.

When the outlier buffer $\mathcal{O}$ accumulates sufficient density to satisfy a validity criterion, a new state Ψ_{new} is generated. This new valid mode is appended to the history Ψ , and it becomes the active nominal state (Lines 19-20). The buffer $\mathcal{O}$ is then cleared (fully or partially) to prepare for future anomaly collection (Line 21). After each iteration, the DT model is recalibrated using the centroid of the current active state μ_k (Line 24), ensuring continuous alignment with the evolving system dynamics.

While Algorithm 1 summarizes the complete logic of the proposed framework, it is designed to be agnostic to the specific implementation of the subroutines (namely OFFLINEIDENTIFICATION, PARAMETERESTIMATION, COGNITIVEFAULTDETECTION, and ONLINEMODELADAPTATION), which will be detailed in the following sections. This brings the flexibility to adapt the framework to different application contexts, spanning from passive monitoring to active adaptation or even fault detection, by selecting appropriate algorithms for parameter estimation, fault detection, and model adaptation.

Moreover, the algorithm is integrated into the ERNESTO *adaptive mode* simulator (presented in Section 2.1.2), wrapping the core physics engine to enable real-time alignment. The implementation leverages asynchronous execution for the drift detection routines, allowing efficient parameter estimation over sliding windows without disrupting the real-time simulation step.

OFFLINEIDENTIFICATION Routine

The offline identification procedure initializes the first nominal state Ψ_0 using a verified, fault-free dataset $\mathcal{Z}_M$. Consider our setting in Part I: it involves estimating the BOL parameters of the Thevenin model, specifically $\theta = [R_0, R_1, C_1]^\top$, as detailed in Section 3.1.

This process yields a dataset of parameter vectors Θ_M distributed across the operational domain (i.e., varying SOC levels and temperatures). Since battery parameters are

inherently state-dependent, a simple global average would fail to capture the system's baseline dynamics accurately. To address this, a clustering algorithm, such as *Density-Based Spatial Clustering of Applications with Noise* (DBSCAN) (Ester et al., 1996), can be used to segment the initial estimates into coherent groups.

While multiple clusters may be identified, each corresponding to a distinct valid operating region (e.g., high SOC vs. low temperature), we select the largest cluster whose centroid is closest to the desired experimental conditions to define the primary initial state. The statistics μ_0 (mean vector) and Σ_0 (covariance matrix) are then computed from this selected cluster, establishing the baseline Ψ_0 against which future drift is measured.

PARAMETER ESTIMATION Routine

At each iteration of the main loop, the algorithm estimates the local parameter vector $\hat{\theta}_t$ from the incoming data window $\mathcal{Z}_W$. This is achieved by minimizing a regularized MSE objective function defined as follows:

$$\hat{\theta}_t = \arg \min_{\phi} \left[\frac{1}{W} \sum_{i=1}^W (y_i - \hat{y}_i(\phi, \mathbf{x}_i))^2 + \xi \|\phi - \theta_{t-1}\|_2^2 \right], \quad (5.5)$$

where y_i is the measured voltage, $\hat{y}_i(\phi, \mathbf{x}_i)$ is the model output given parameters ϕ and input features $\mathbf{x}_i$, and θ_{t-1} is the parameter vector from the previous time step.

The regularization term, weighted by the hyperparameter ξ , specifically penalizes large deviations from the previous estimate, fostering smoother updates. The optimization is performed using the L-BFGS-B algorithm (Zhu et al., 1997), a quasi-Newton method chosen for its ability to handle bound constraints (e.g., $R_0 > 0$) and its efficiency within the real-time constraints of the DT framework.¹

COGNITIVE FAULT DETECTION Routine

The cognitive fault detection mechanism evaluates each new parameter estimate $\hat{\theta}_t$ against the active nominal state Ψ_k . This is achieved by computing the *Mahalanobis Distance* (MD) (McLachlan, 1999), a statistical measure that explicitly accounts for data correlation and covariance. Specifically, the MD between the new estimate and the centroid of the nominal cluster is calculated as:

$$D_M(\hat{\theta}_t) = \sqrt{(\hat{\theta}_t - \mu_k)^\top \Sigma_k^{-1} (\hat{\theta}_t - \mu_k)}. \quad (5.6)$$

To determine the validity of the estimate, we define a dynamic threshold τ_M derived from the *F-distribution*. Unlike simple asymptotic thresholds (e.g., Chi-square) often used in literature, this approach accounts for the finite sample size of the cluster, providing a robust statistical bound even during the early stages of cluster formation. The critical value is given by:

$$D_M^2(\hat{\theta}_t) \cdot \frac{n_k(n_k - d)}{d(n_k^2 - 1)} \leq F_{d, n_k - d, \alpha_s}, \quad (5.7)$$

¹To ensure numerical stability and faster convergence, the parameters are normalized to a common scale (e.g., $[0, 1]$) prior to optimization, mitigating the impact of differing physical magnitudes (e.g., $R \approx 10^{-3} \Omega$ vs. $C \approx 10^4 \text{ F}$).

which implies the critical threshold τ_M for the Mahalanobis Distance is:

$$\tau_M = \sqrt{\frac{d(n_k^2 - 1)}{n_k(n_k - d)}} F_{d, n_k - d, \alpha_s}. \quad (5.8)$$

where d is the dimensionality of the parameter space, n_k is the current support of the nominal cluster, and $F_{d, n_k - d, \alpha_s}$ is the upper α_s -critical value of the F-distribution with d and $n_k - d$ degrees of freedom. Note that the factor $(n_k^2 - 1)/n_k = (n_k - 1)(n_k + 1)/n_k$ includes the correction term $(n_k + 1)/n_k$, explicitly accounting for the uncertainty of the single future observation $\hat{\theta}_t$ relative to the estimated cluster statistics. This ensures rigorous outlier detection even when the nominal cluster size n_k is small.

The classification rule is straightforward: if $D_M(\hat{\theta}_t) > \tau_M$, the estimate is classified as an anomaly (outlier) falling outside the $(1 - \alpha_s)$ confidence region; otherwise, it is accepted as an inlier and used to update the nominal cluster statistics.

ONLINEMODELADAPTATION Routine

The *Online Model Adaptation* routine is triggered whenever the outlier buffer $\mathcal{O}$ is populated with a new anomaly. The primary objective of this subroutine is to assess whether the accumulated outliers represent a statistically significant *concept drift*, thereby warranting the creation of a new valid operating mode. This is achieved through a rigorous three-stage procedure: *statistical drift confirmation*, *spatio-temporal clustering*, and *robust statistics estimation*. The complete logic is detailed in Algorithm 2.

Algorithm 2 ONLINEMODELADAPTATION

Input: Outlier buffer $\mathcal{O}$, Nominal state Ψ_k , Threshold α_{ks} , Grid parameters (r, λ)

```

1: Project data:  $\mathcal{Z}_{\text{proj}} \leftarrow \text{PCA}(\mathcal{O} \cup \text{samples}(\Psi_k))$ 
2:  $p_{\text{val}} \leftarrow \text{KS-TEST}(\mathcal{Z}_{\text{proj}}[\mathcal{O}], \mathcal{Z}_{\text{proj}}[\Psi_k])$ 
3: if  $p_{\text{val}} > \alpha_{ks}$  then
4:   return NULL  $\triangleright$  No significant drift detected
5: else
6:    $\tilde{\Theta} \leftarrow \text{MOUNTAINMETHOD}(\mathcal{O}, r, \lambda)$ 
7:   if  $|\tilde{\Theta}| \leq d$  then
8:     return NULL  $\triangleright$  Cluster too sparse
9:   else
10:     $\bar{\Theta} \leftarrow \text{MCD}(\tilde{\Theta})$ 
11:    if  $|\bar{\Theta}| > d + 1$  then
12:       $\Psi_{\text{new}} \leftarrow (\text{mean}(\bar{\Theta}), \text{cov}(\bar{\Theta}), |\bar{\Theta}|, t_{\text{now}})$ 
13:      return  $\Psi_{\text{new}}$ 
14:    else
15:      return NULL  $\triangleright$  Cluster too sparse or singular
16:    end if
17:  end if
18: end if

```

Statistical Drift Confirmation. The procedure begins with a statistical test to determine whether the distribution of the buffered outliers $\mathcal{O}$ differs significantly from that of the current nominal state Ψ_k . Specifically, we rely on the KS test, a univariate, non-parametric test used to assess if two samples are drawn from the same underlying distribution.

As a preliminary step, we apply *Principal Component Analysis* (PCA) to project the data onto the first principal component of the nominal cluster, to obtain a one-dimensional representation, preserving the direction of maximum variance while filtering out noise. Then, we perform a Two-Sample KS test on the projected data. The test evaluates the null hypothesis H_0 that the ECDFs of the buffer and the nominal state are drawn from the same underlying distribution:

$$H_0 : F_O^{\text{PCA}} = F_{\Psi_k}^{\text{PCA}} \quad \text{vs.} \quad H_1 : F_O^{\text{PCA}} \neq F_{\Psi_k}^{\text{PCA}}, \quad (5.9)$$

If H_0 is rejected at significance level α_{ks} , the drift is confirmed, and the algorithm proceeds to cluster the outliers.

Spatio-Temporal Clustering. To identify coherent structures within the confirmed drift, we employ the *Mountain Method* (Yager and Filev, 1994). This grid-based clustering algorithm is particularly well-suited for our application due to its robustness to noise and ability to identify cluster centers without *a priori* knowledge of the number of clusters.

To capture the physical nature of battery degradation, where changes are expected to be continuous in time, we utilize a specialized *spatio-temporal norm* $\|\cdot\|_\lambda$. The potential density function at a candidate grid center $c_j \in \mathbb{R}^d$ is defined as:

$$\Omega(c_j) = \sum_{h=1}^{|\mathcal{O}|} \exp \left(-\frac{\|c_j - \hat{\theta}_h\|_\lambda^2}{2r^2} \right), \quad (5.10)$$

where r is the radius parameter controlling the neighborhood granularity. The distance metric $\|\cdot\|_\lambda$ combines Euclidean distance in the parameter space with temporal proximity:

$$\|c_j - \hat{\theta}_h\|_\lambda^2 = \lambda \frac{\|c_j - \hat{\theta}_h\|^2}{2d} + (1 - \lambda) \frac{(t_j - t_h)^2}{t}, \quad (5.11)$$

where $\lambda \in [0, 1]$ balances the spatial and temporal components, t_j and t_h are the timestamps associated with the center c_j and the sample $\hat{\theta}_h$, respectively, and t is the current time step. This formulation promotes the formation of clusters that are strictly *spatially coherent* and *temporally contiguous*. The algorithm iteratively selects the cluster center c^* that maximizes the potential density:

$$c^* = \arg \max_{c_j} \Omega(c_j). \quad (5.12)$$

Robust Statistics Estimation. Once a candidate cluster $\tilde{\Theta}$ is identified around c^* , we must estimate its statistical descriptors ($\mu_{\text{new}}, \Sigma_{\text{new}}$). To ensure these statistics are not corrupted by residual noise or outliers within the buffer, we employ the *Minimum Covariance Determinant* (MCD) estimator (Rousseeuw and Driessen, 1999). MCD identifies the subset of samples $\bar{\Theta} \subseteq \tilde{\Theta}$ of size h (where $d < h \leq |\tilde{\Theta}|$) that minimizes the determinant of the covariance matrix. This effectively isolates the most relevant coefficient vectors of the new operating mode. If the resulting core subset satisfies the dimensionality constraint $|\bar{\Theta}| \geq d + 1$, a new nominal state is formalized as $\Psi_{\text{new}} = \langle \text{mean}(\bar{\Theta}), \text{cov}(\bar{\Theta}), |\bar{\Theta}|, t \rangle$ and added to the set of nominal states Ψ . Until the completion of the next adaptation phase, this will be the new DT state.

5.2.4 Discussion on Adaptation Dynamics

The proposed framework effectively implements a *hybrid* adaptation strategy. It combines *passive adaptation*, i.e., the continuous tracking of local mean shifts within the current nominal cluster Ψ_k , with the *active detection* of structural changes via the outlier buffer. This duality renders the method highly efficient in tracking *gradual* and *incremental* drifts, as the clustering mechanism naturally accumulates slow-moving parameter estimates until they form a statistically dense region.

A distinctive feature of our approach is the handling of the outlier buffer $\mathcal{O}$ post-adaptation. While old outliers may be retained in the buffer after a new cluster is formed, their influence on future adaptation cycles is naturally mitigated by the *temporal awareness* embedded in the spatio-temporal clustering norm included in Equation (5.11). Older samples effectively reduce their relevance accordingly to the weight λ , ensuring that new clusters are primarily driven by recent system behavior.

Furthermore, the framework’s sensitivity is governed by explicit structural constraints rather than arbitrary heuristics. Specifically, the sparsity checks (Algorithm 2, Lines 7 and 11) ensure that only statistically significant densities are promoted to valid operating modes. These thresholds (e.g., minimum cluster size d) can be tuned empirically: stricter thresholds reduce the risk of false positives, while more lenient thresholds accelerate the detection of subtle drifts.

Finally, regarding *abrupt drifts*, the system exhibits an inherent *verification latency*. A sufficient density of samples must accumulate in the new region of the parameter space before a valid cluster is recognized. While this latency is beneficial for robustness, preventing the model from reacting to transient sensor noise, it imposes a delay in adapting to instantaneous changes. Future developments could mitigate this by integrating rapid change-point detection algorithms (e.g., CUSUM) to trigger an emergency adaptation mode for safety-critical shifts.

CHAPTER 6

Experimental Evaluation

In previous chapters, we introduced the problem of enabling ERNESTO with tracking and drift detection capabilities through online learning techniques. In particular, we present the adaptive routine, designed and integrated into ERNESTO, embedded as a high-level module that orchestrates the normal DT simulation loop. In this chapter, we report the experimental evaluation of the proposed approach, starting from the identified operational regimes, the design of the case studies, and the results obtained. The experiments are designed to evaluate the effectiveness of the proposed online learning routine for tracking the system’s evolution and detecting drifts, while maintaining accurate predictions and reliable monitoring. We also compare our approach with baseline methods to demonstrate its advantages in terms of robustness and interpretability in dynamic environments.

Chapter Outline. This chapter is organized as follows. In Section 6.1, we describe the experimental setup, including the identification of operational regimes, the baselines used for comparison, and the configuration of the scenarios. In Section 6.2, we present the results of the case studies conducted on both synthetic and real-world scenarios, evaluating the performance of our approach in tracking the system’s evolution and detecting drifts.

6.1 Experimental Setup

In this section, we describe the experimental setup used to evaluate the proposed online learning routine integrated into ERNESTO. We first outline the general settings for the experiments, the procedure for collecting nominal data, and the assumptions regarding the system’s behavior.

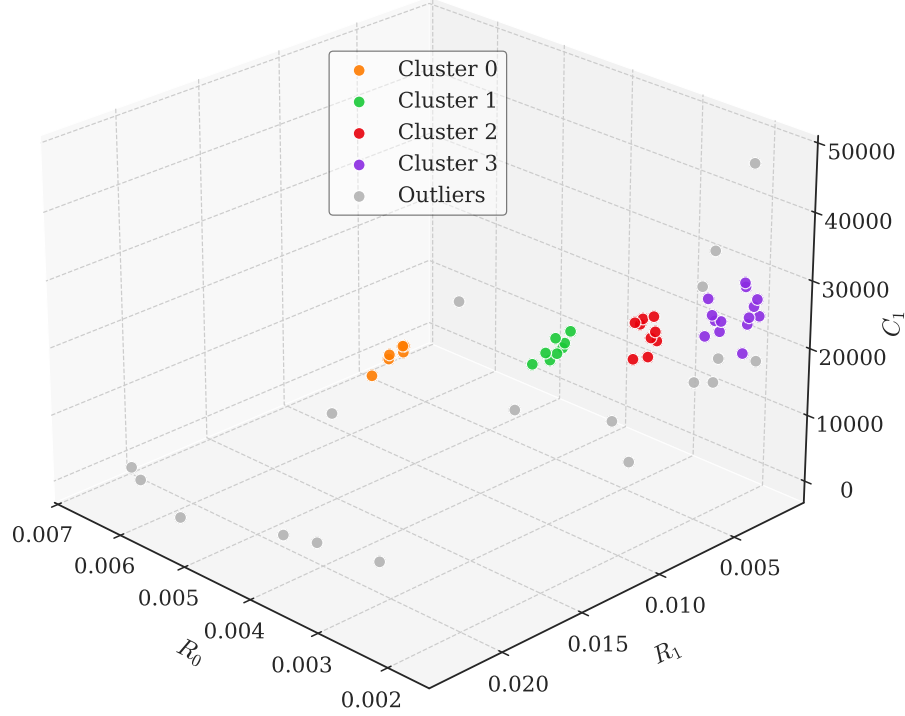


Figure 6.1: DBSCAN clustering of the battery characterization data.

6.1.1 Operational Regimes Identification

To identify the system's operational regimes, we apply the DBSCAN algorithm (Ester et al., 1996). DBSCAN is a density-based clustering method that groups points that are densely packed, while marking points that lie alone in low-density regions as outliers. In this context, we apply DBSCAN to the battery characterization data presented in Section 3.1, which describes the dynamics of the internal Thevenin parameters, i.e., $\{R_0, R_1, C_1\}$, over the SOC-temperature domain. The identified clusters correspond to different regions of the SOC-temperature domain and describe the nominal operating regimes of the system at the start of its life, as data were collected on a fresh battery before the onset of degradation phenomena. It should be noted that the characterization test requires charging and discharging the battery across various SOC-temperature levels, inherently causing minor system degradation. However, for these experiments, we set the BOL immediately after the characterization phase, treating the initial aging as negligible.

The output of the DBSCAN algorithm is illustrated in Figure 6.1, demonstrating that the data points are successfully grouped into distinct clusters. The clustering procedure is configured with $\epsilon = 0.3$, which defines the maximum distance between two samples to be considered in the same neighborhood, and with a minimum neighborhood size of 5, determined through a preliminary grid search tuning procedure.

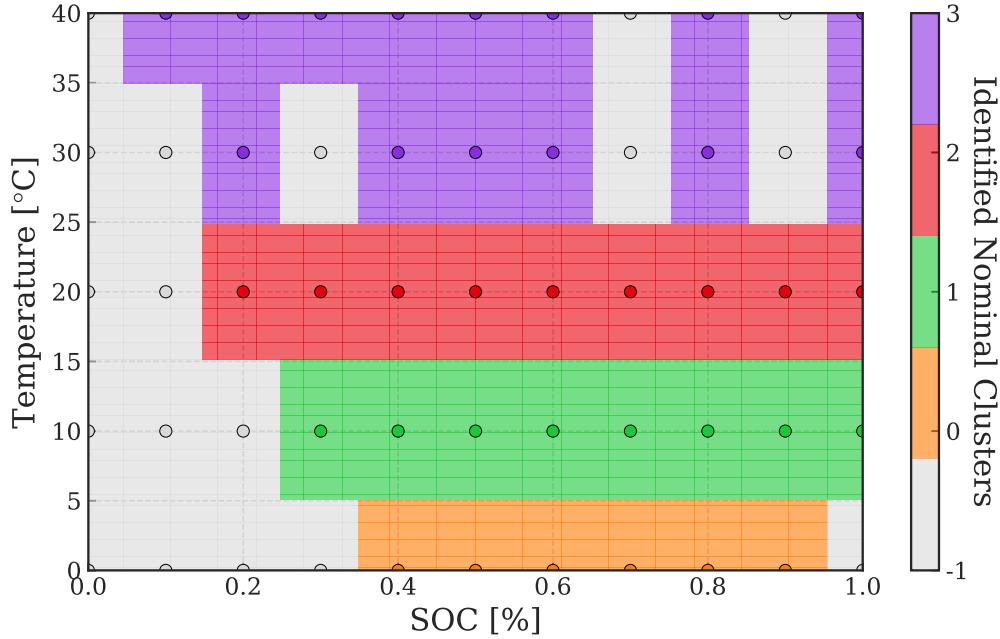


Figure 6.2: Domain-specific interpretation of the identified clusters.

An important aspect emerging from Figure 6.2 is that the main factor influencing the clustering process is temperature rather than SOC, in alignment with the literature on battery modeling. Indeed, the domain is split into four horizontal regions, highlighting that within the SOC range of (0.2, 1.0), parameter variance is particularly sensitive to temperature. Conversely, at extreme SOC levels, parameters cannot be clustered as cleanly since their behavior becomes highly non-linear and less consistent, as previously highlighted in Figure 3.2 and Figure 3.3.

In our experiments, we use one of the identified clusters as the battery’s initial nominal regime, setting the cluster’s centroid as the initial model parameter vector. According to Algorithm 1, ERNESTO estimates new parameters in real time by optimizing the model over fixed, non-overlapping windows of incoming data, and subsequently evaluates whether the estimated vector belongs to the active nominal cluster. As the cardinality of the outlier set increases, ERNESTO may trigger a re-adaptation of the model, meaning that a concept drift has occurred. Hence, the adaptive routine triggers the creation of a new cluster (i.e., a new operational regime) from the outlier set. If successful, this new cluster becomes the active nominal reference, representing the real-world system’s most reliable state.

While in this context, the identified clusters are rarely recurrent due to degradation mechanisms, this procedure helps us clearly interpret the system’s evolution. As demonstrated in the following sections, each new cluster represents a distinct state of the BESS lifecycle. This can be leveraged to simulate precise system conditions or, when coupled with a fault detection system, to trigger maintenance alarms, effectively balancing tracking reactivity with historical interpretability.

6.1.2 Baselines

To evaluate the performance of the proposed *evolving clustering* routine, we compare it against three baseline methods:

- *Static Model*: A fixed model that does not adapt to changes in system behavior. It is trained strictly on the initial nominal data and remains unchanged throughout the simulation (specifically, the parameter vector remains the centroid of the initial cluster), serving as a lower-bound reference to highlight the necessity of learning strategies.
- *Cluster Shift*: This baseline passively adapts the nominal cluster using new parameter estimations computed over every non-overlapping window $\mathcal{Z}_W$. It does not classify samples as outliers or inliers; instead, it continuously updates the parameter distribution. Consequently, the nominal cluster slowly shifts towards the current operational regime without discarding past parameters.
- *Memoryless Adaptation*: A highly reactive approach where the model is entirely retrained over every non-overlapping window $\mathcal{Z}_W$. Every new estimate immediately replaces the model's parameter vector, ignoring the system's previous states. No concept of clustering or historical memory exists in this baseline.

With respect to these baselines, our active method positions itself between the *memoryless adaptation* and *cluster shift* approaches. Indeed, the *evolving clustering* approach is more conservative than *memoryless adaptation*, as it only triggers retraining when a drift is detected rather than at fixed intervals. This means that our method could be slower to adapt to abrupt changes, but it is more robust to noise and less prone to overfitting, which is particularly important in real-world scenarios. At the same time, the *evolving clustering* approach is more reactive than the *cluster shift* method, which maintains the entire parameter history and passively adapts the nominal cluster without forgetting outdated parameters, which is not ideal in the presence of abrupt changes. Above all, our method explicitly enhances interpretability and fosters structured fault detection, capabilities entirely absent in passive baseline approaches.

6.1.3 Scenarios Configuration

To evaluate the performance of the proposed online learning routine, we design a set of scenarios that simulate different conditions and changes in the battery's behavior. These scenarios are designed to test the routine's adaptability and robustness across both synthetic and real-world conditions.

In particular, our objective is to replicate an abrupt change that can be perceived as a sensor fault, which is a common issue in real-world applications. We simulate this by introducing a thermal jump in the system, following an abrupt modification of the thermal environment, which causes a sudden change in the battery's behavior. All the scenarios revolve around this artifact. Specifically, we designed two synthetic scenarios, one undergoing a single thermal jump, and another one undergoing multiple thermal jumps (i.e., consecutive malfunctions of the thermal sensor). We also apply the same configuration with one thermal jump to real-world data, collected by altering the thermal chamber's setpoint during the experiment. We will describe the details of these scenarios in the following.

We recall that, in this work, our objective is to *track* the system’s evolution, not to strictly classify fault types (fault isolation). However, with our approach, the creation of a new nominal cluster naturally serves as a highly reliable drift alert, whereas isolation mechanisms require contextualized analysis of parameter evolution. At the same time, we were not completely able to evaluate our method on real-world incremental drift related to degradation. Indeed, for this specific experiment, we would have needed more extensive data, which were not possible to collect within the RSE testbed facility. While aging is inherently present in the real-world validation scenario and affects signal clarity, it is omitted by default in our synthetic experiments to focus the evaluation on regime shift adaptation.

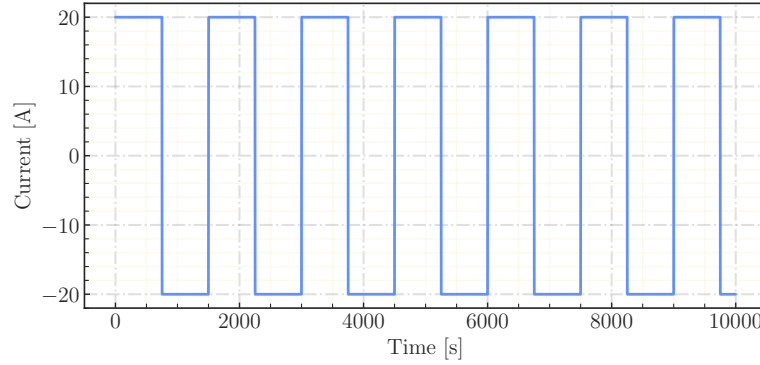
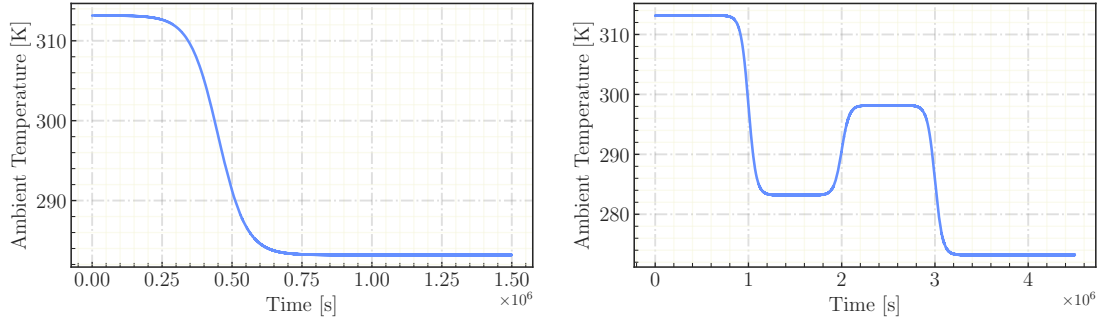


Figure 6.3: Current profile used in the synthetic scenarios, consisting of a square wave alternating between a discharge current of -20 A and a charging current of 20 A.



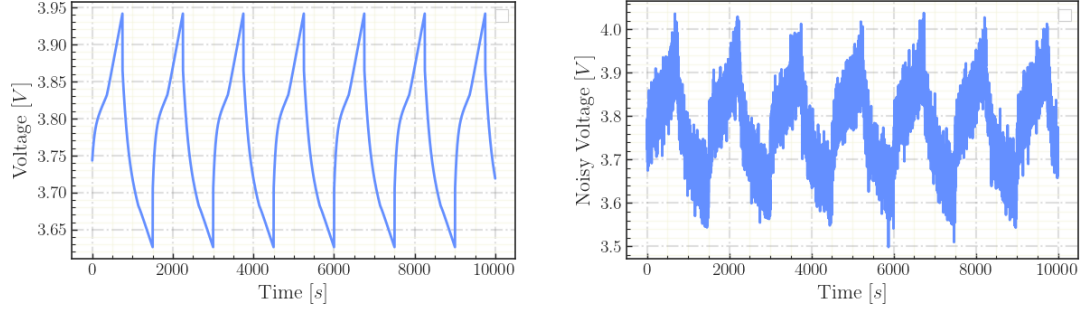
(a) Single thermal jump scenario.

(b) Multiple thermal jumps scenario.

Figure 6.4: Temperature profiles for the simulated scenarios. (a) The single jump scenario starts with a constant temperature of 313 K for the first $300\,000$ seconds, followed by a drop to 283 K. (b) The multiple jumps scenario varies dynamically, starting at 313 K and stepping to 283 K, 298 K, and finally 273 K at intervals of $1\,000\,000$ seconds.

Synthetic Scenarios Configuration

The synthetic scenarios are constructed by selecting specific current and temperature profiles and simulating the DT’s battery response using ERNESTO. The obtained results serve as the ground truth for evaluating the adaptive routine’s performance. This makes sense after validating the DT model, presented in Section 3.2, where we show



(a) Voltage profile without noise.

(b) Voltage profile with noise.

Figure 6.5: Voltage profiles for the simulated scenarios. (a) The voltage profile without noise shows the ideal response of the battery to the current and temperature changes, highlighting the effects of the thermal jumps on the voltage. (b) The voltage profile with noise includes random fluctuations to simulate measurement noise and real-world conditions, providing a more realistic scenario for evaluating the performance of the adaptive routine.

that the model accurately reproduces the battery's behavior under various conditions.

Specifically, we adopted a current profile consisting of a square wave with a period of 1500 seconds, alternating between a discharge current of -20 A and a charging current of 20 A, presented in Figure 6.3. Regarding the temperature profile, we differentiate the two scenarios as follows. In the single thermal jump scenario, we start with a constant temperature of about 313 K for the first $300\,000$ seconds, then we introduce a thermal jump to almost 283 K for the remaining time, as depicted in Figure 6.4a. In the multiple thermal jumps scenario, we have three different thermal jumps, as shown in Figure 6.4b: from the same initial temperature of 313 K to almost 283 K at $1\,000\,000$ seconds, then to 298 K at $2\,000\,000$ seconds, and finally down to 273 K at $3\,000\,000$ seconds.

To mirror real-world operations, we introduce Gaussian noise to the simulated voltage measurements. The additive noise is calibrated using a Signal-to-Noise Ratio (SNR) approach, governed by the parameter ξ (in decibels). The noisy voltage $\tilde{v}_t$ at time t is computed as:

$$\tilde{v}_t = v_t + \varepsilon_t, \quad \varepsilon_t \sim \mathcal{N}(0, \sigma_v^2),$$

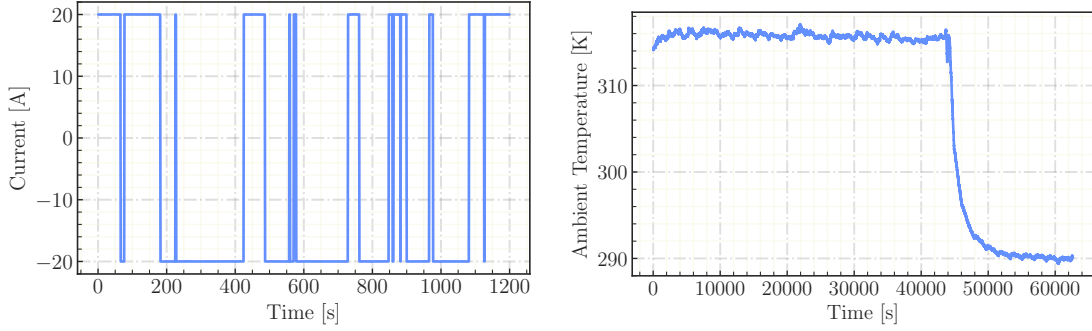
where the standard deviation σ_v is proportional to the root mean square of the clean voltage signal v_t over a window of size W , scaled by the SNR parameter ξ :

$$\sigma_v = \frac{\sqrt{\frac{1}{W} \sum_{w=1}^W v_w^2}}{10^{\frac{\xi}{20}}}.$$

For these experiments, we adopted $\xi = 40$ dB. The effect of this noise on the voltage profile is illustrated in Figure 6.5b, allowing us to evaluate the robustness of the adaptive routine under realistic, uncertain conditions.

Real-World Scenario Configuration

The real-world scenario is designed to evaluate the performance of the proposed adaptive routine on actual battery data collected in the RSE's testbed facility. The data were collected by changing the thermal chamber's setpoint during the experiment, thereby



(a) Current profile for the real-world scenario.

(b) Temperature profile for the real-world scenario.

Figure 6.6: Current and temperature profiles for real-world scenario. (a) Current profile used in the real-world scenario, consisting of a PRBS signal between a discharge current of -20 A and a charging current of 20 A. (b) Temperature profile for the real-world scenario, showing a thermal jump from about 315 K to almost 290 K, similar to the first synthetic scenario.

introducing a thermal jump similar to that simulated in the first synthetic scenario, illustrated in Figure 6.6b. However, the profile after the thermal jump here is shorter, lasting approximately $15\,000$ steps, requiring a rapid adaptation of the model to the new regime with fewer samples at disposal to confirm the drift.

Unlike the synthetic scenarios, the imposed current profile is generated using a Pseudo-Random Binary Sequence (PRBS) signal, where the current alternates between a minimum of -20 A and a maximum of 20 A. The variable governing the PRBS signal is the time interval between two consecutive switches, which is uniformly sampled in the range from 1 s to 10τ , where $\tau = R_1 C_1$ is the assumed time constant of the Thevenin model's RC branch. Finally, to guarantee SOC balancing, each portion of the profile is concatenated with its inverted counterpart, ensuring that the net charge transferred is zero. In Figure 6.6a we report the resulting current profile, lasting approximately $60\,000$ seconds, with a step duration of 0.1 s.

Since the real-world profile differs significantly from the synthetic ones, it is crucial to highlight the additional challenges inherent to the use of empirical data. Beyond the limited data availability and the shorter duration of the experimental profile, the initial parameters and exact internal state of the battery at the start of the experiment are not perfectly known, conversely from the synthetic scenarios. Moreover, empirical measurements are subject to sensor noise and inherent physical phenomena, such as degradation. Collectively, these factors foster a highly complex and noisy operational environment that must be carefully considered when evaluating the robustness and performance of the chosen adaptation strategy, which may seem less effective but more reliable in real-world conditions.

6.2 Case Studies

This section details the results of our experimental campaign. We assess tracking performance primarily by measuring the voltage prediction errors between the DT and the ground truth. We will adopt the Root Mean Square Error (RMSE) and Mean Absolute Error (MAE) as our performance metrics. We also evaluate the system's responsiveness to abrupt changes and, crucially, the interpretability provided by the adopted adaptation

strategies acting within the parameter space. Notably, all the evaluated approaches are integrated within the same DT framework, ensuring a fair comparison of their tracking capabilities under identical conditions, and demonstrating the versatility of the ERNESTO platform in accommodating various online learning strategies. Finally, in addition to these presented results, we report in Appendix B additional visualizations, including the comparison of employed parameters and a zoomed-in analysis of the voltage tracking performance during the experiment, which further confirm the insights drawn from the main results presented here.

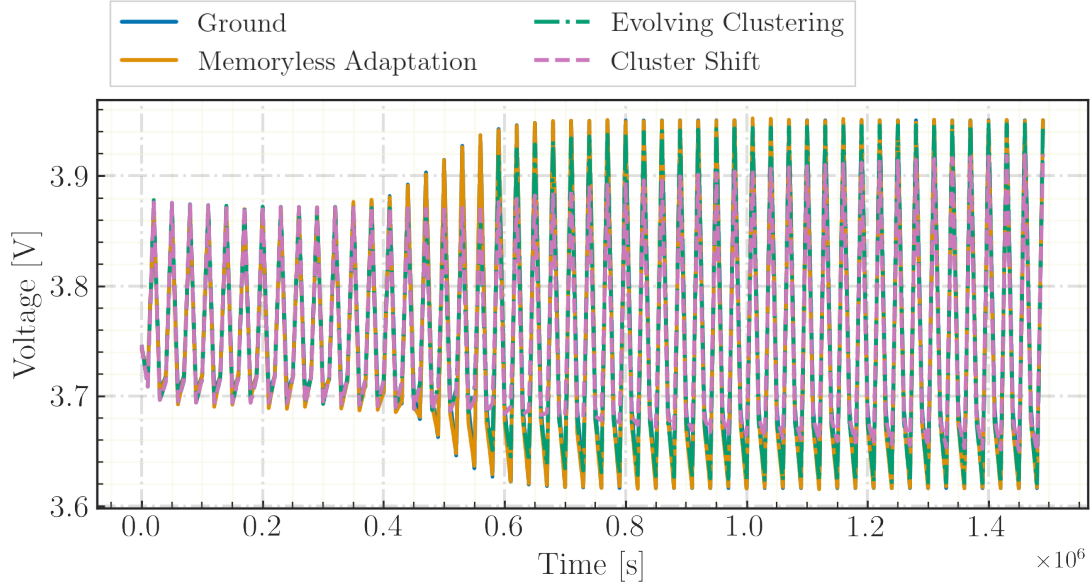


Figure 6.7: Voltage tracking performance of the proposed adaptive routine compared to the static model and passive evolution baselines in the single thermal jump scenario. The plot shows the predicted voltage versus the actual voltage over time, highlighting the model’s ability to adapt to changes in temperature. For better visualization, we reported a zoomed-in view of a portion of the profile in Figure B.1.

6.2.1 First Synthetic Scenario: Single Thermal Jump

The single thermal jump scenario is designed to evaluate the model’s responsiveness to an isolated, abrupt operational shift influencing temperature, which can be perceived as a sensor fault. We report in Table 6.1 the global voltage tracking performance metrics, i.e., RMSE and MAE, for the different adaptation methods. The *static model* exhibits the highest tracking error, confirming that a fixed parameterization is entirely unsuited for dynamic environments. The *cluster shift* baseline improves upon this, reducing the RMSE to 0.0409 V. However, its preservation of the entire parameter history becomes a critical disadvantage after the change. Passively averaging the parameters of the new thermal state with those of the old state yields a compromised, inaccurate distribution that converges slowly. The *memoryless adaptation* baseline achieves the lowest numerical tracking error. By operating with a zero-memory horizon, the model instantly discards its past state and overfits entirely to the most recent data window, ensuring immediate voltage alignment after the jump. Our proposed active approach, the *evolving*

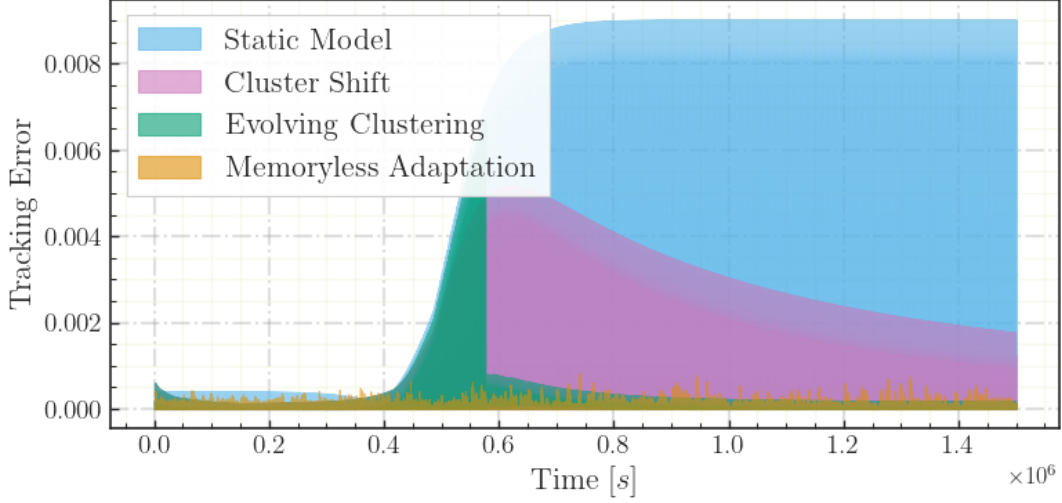


Figure 6.8: Voltage prediction error over time for the single thermal jump scenario. The plot compares the error of the proposed adaptive routine with those of the static model and passive evolution baselines, demonstrating its effectiveness in reducing error after the thermal jump.

clustering method, delivers highly robust tracking, achieving an RMSE of 0.0191 V and an MAE of 0.0126 V. While there is a numerical gap between our approach and the *memoryless* baseline, this difference is a deliberate consequence of the framework’s design. Our active method has an inherent *verification latency*: when an abrupt thermal jump occurs, the algorithm conservatively buffers incoming parameter anomalies, delaying recalibration until a statistically significant density accumulates to confirm genuine concept drift rather than transient noise. Crucially, this statistical conservatism is the cornerstone of the system’s diagnostic capability. The memoryless approach operates as an opaque strategy, blindly overwriting its internal state and erasing all historical context. In contrast, the *evolving clustering* routine safely preserves the original nominal cluster and formalizes the new thermal state as a distinct, mathematically defined operating mode.

Table 6.1: Comparison of voltage tracking performance metrics on single thermal jump scenario

Adaptation Method	RMSE [V]	MAE [V]
<i>Static Model</i>	0.0687	0.0571
<i>Cluster Shift</i>	0.0409	0.0347
<i>Memoryless Adaptation</i>	0.0041	0.0027
<i>Evolving Clustering (Ours)</i>	0.0191	0.0126

Note: Best values are highlighted in bold. Lower values indicate better tracking accuracy.

These dynamics are visually confirmed in the step-wise analyses. Figure 6.7 and Figure 6.8 illustrate how our method successfully realigns with the ground truth shortly after the thermal shock, while the passive *cluster shift* method suffers from severe adaptation lag. Conversely, the *memoryless adaptation* baseline achieves near-instantaneous

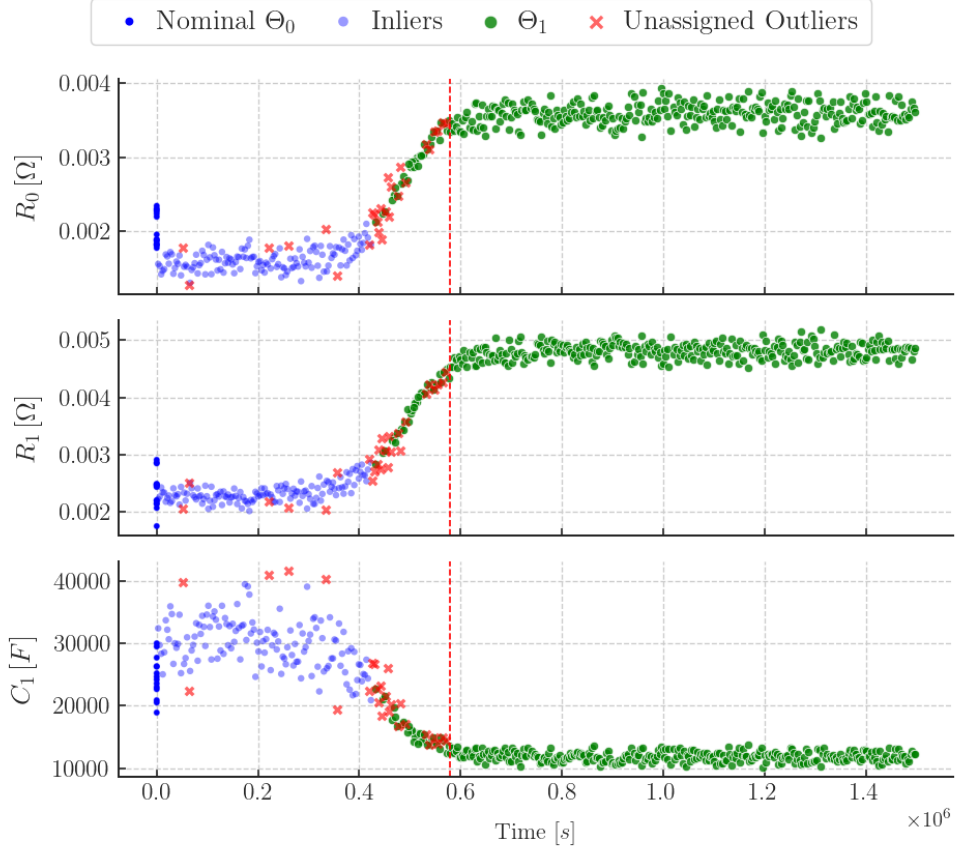


Figure 6.9: 2D visualization of the parameter evolution over time in the single thermal jump scenario. The plot shows the trajectory of the estimated parameters, highlighting the point at which a drift is detected, and a new cluster is formed (vertical dashed line), indicating the model’s adaptation to the new regime.

realignment, but at the cost of complete loss of historical context. This trade-off between reactivity and interpretability is fundamental to our framework’s design philosophy. Such a concept is demonstrated by Figure 6.9. Here, the three-dimensional parameter space is projected onto a 2D plane for visualization purposes, showing the trajectory of the estimated parameters over time. The vertical dashed line indicates the point at which a drift is detected and a new cluster is formed. This visual representation clearly illustrates the evolution of the system’s state, marked by the creation of a new cluster that transforms abstract prediction errors into actionable system knowledge.

6.2.2 Second Synthetic Scenario: Multiple Thermal Jumps

The multiple thermal jump scenario represents a significantly harsher stress test, evaluating the algorithms’ ability to manage repeated, abrupt operational shifts without suffering from model collapse or catastrophic forgetting.

Table 6.2 reports the global performance metrics for this scenario. Unsurprisingly, the *static model* exhibits the highest cumulative error, as its initial parameterization becomes

Table 6.2: Comparison of voltage tracking performance metrics on multiple thermal jumps scenario

Adaptation Method	RMSE [V]	MAE [V]
<i>Static Model</i>	0.1060	0.0798
<i>Cluster Shift</i>	0.0721	0.0533
<i>Memoryless Adaptation</i>	0.0051	0.0034
<i>Evolving Clustering (Ours)</i>	0.0284	0.0149

Note: Best values are highlighted in bold. Lower values indicate better tracking accuracy.

entirely obsolete after the first thermal shock. The *cluster shift* baseline demonstrates a severe limitation here, yielding a high RMSE of 0.0721 V. Because it continuously averages new estimations into a single global cluster, its infinite memory becomes a critical liability. With each successive thermal jump, the passive cluster is dragged across the parameter space, roughly approximating the distinct operational modes into a single, inaccurate distribution. While the *memoryless adaptation* baseline once again achieves the lowest numerical error by instantaneously overfitting to new conditions, our proposed *evolving clustering* method maintains highly robust tracking performance and drastically outperforms the passive baseline. As discussed in the previous scenario, the numerical gap between our method and the memoryless tracker is due to the cumulative verification latency incurred at each successive jump.

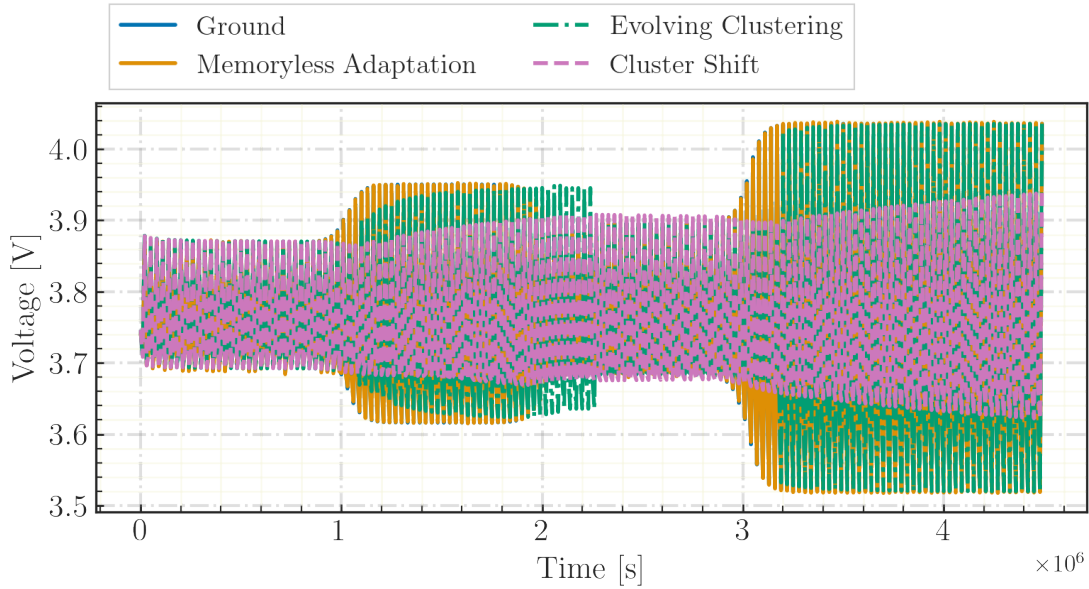


Figure 6.10: Voltage tracking performance of the proposed adaptive routine compared to the static model and passive evolution baselines in the multiple thermal jumps scenario. The plot shows the predicted voltage against the actual voltage over time, highlighting the model's ability to adapt to the changes in temperature. For better visualization, we reported a zoomed-in view of a portion of the profile in Figure B.2.

However, this statistical conservatism is precisely what enables the DT to act as a reliable diagnostic tool. As illustrated in Figure 6.12, while the memoryless baseline

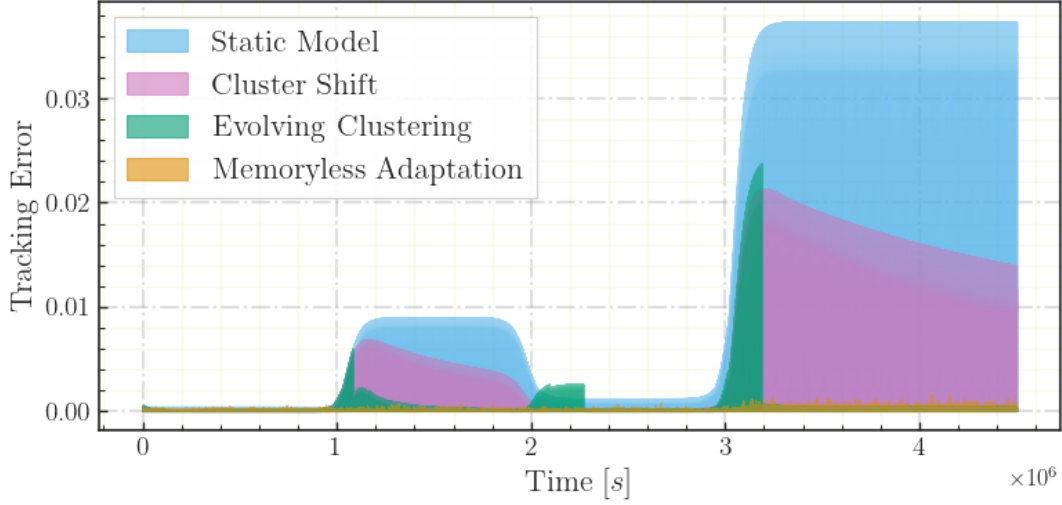


Figure 6.11: Voltage prediction error over time for the multiple thermal jumps scenario. The plot compares the error of the proposed adaptive routine with that of the static model and passive evolution baselines, demonstrating its effectiveness in reducing error after each thermal jump.

provides no historical context, our *evolving clustering* routine dynamically partitions the battery’s lifecycle into physically meaningful operating modes. Here, it is even more evident the value of the interpretability our method provides. Each new cluster represents a distinct state of the BESS lifecycle, which can be leveraged to simulate precise system conditions, providing a broader range of what-if analysis. Finally, the voltage-tracking results (Figure 6.10) and error profiles (Figure 6.11) confirm that the method reliably detects and adapts to each cascading drift, preserving the structural integrity required for long-term health monitoring.

6.2.3 Real-World Scenario: Single Thermal Jump

Transitioning to the real-world dataset, the framework must navigate additional operational challenges, including unmodeled sensor noise, mild inherent battery aging, and limited data availability. Furthermore, lacking an exact characterization of the initial parameters, we initialized the model using the synthetic BOL cluster. Consequently, we expect an initial adaptation lag as the model collects sufficient data to adjust the nominal cluster to the true physical reality of the testbed. This is confirmed by the initial tracking error observed in Figure 6.13 and Figure 6.14, where all methods exhibit a significant error at the start of the experiment. However, while the *memoryless* baseline rapidly overfits to the new conditions, our *evolving clustering* method takes a more cautious approach, gradually adjusting the nominal cluster as it accumulates evidence of the true operational regime.

Another critical aspect to consider is the lack of prior knowledge regarding the periodicity of the load profile. While in the synthetic scenarios we could tailor the data window size to the exact profile period to ensure smooth parameter estimation, this information is typically unavailable in real-world applications. In this specific case, a preliminary autocorrelation analysis of the current profile revealed a dominant period of

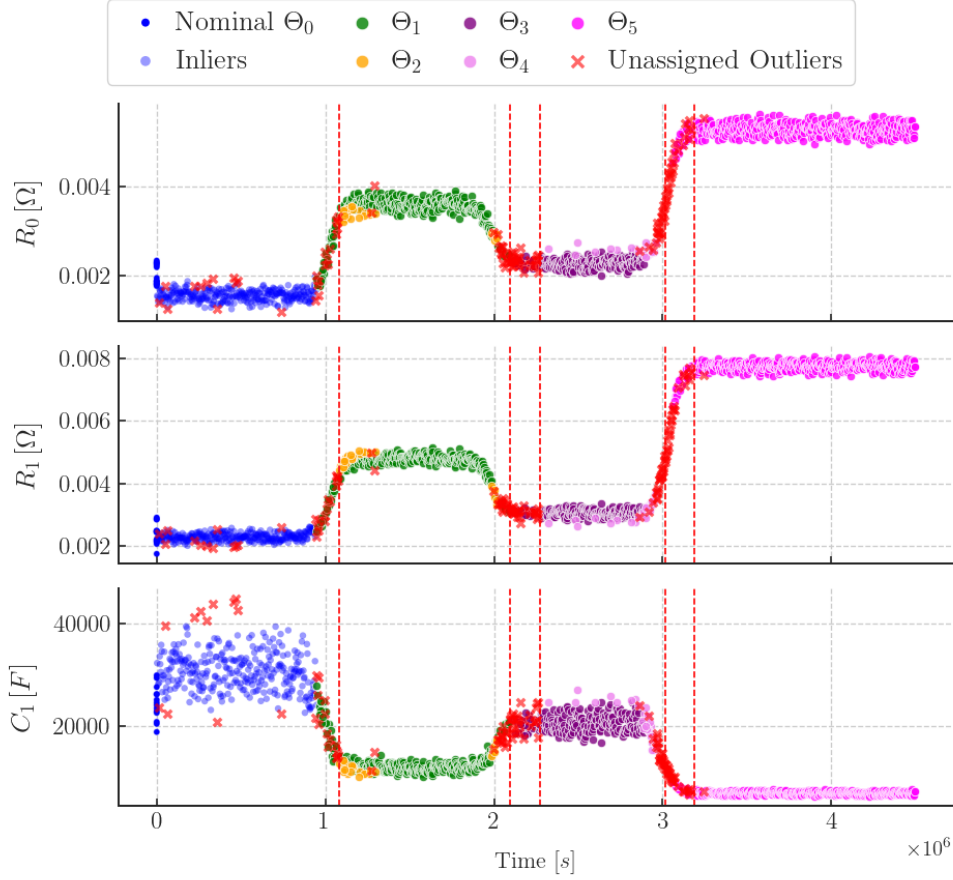


Figure 6.12: 2D visualization of the parameter evolution over time in the multiple thermal jumps scenario. The plot shows the trajectory of the estimated parameters, highlighting the point at which a drift is detected, and a new cluster is formed (vertical dashed lines), indicating the model’s adaptation to the new regime.

approximately 18 000 steps. However, utilizing this value would result in prohibitively large data windows, hindering the proper evaluation of the presented method due to insufficient data. Consequently, we empirically identified a suitable window size of 2700 steps. This dimension is sufficiently large to guarantee reliable parameter estimation, yet small enough to allow the model to effectively adapt to dynamic changes in the operating conditions.

In Table 6.3 we report the voltage tracking performance of tested algorithms. As expected, the *static model* yields the worst performances. Interestingly, in this real-world environment, our proposed *evolving clustering* method performs poorly than both the *memoryless adaptation*, still the best performing, and *cluster shift* baselines, with an RMSE of 0.0628 V. This is mainly due to the initial calibration mismatch, which is handled more rapidly by the other methods, as illustrated in Figure 6.13 and Figure 6.14. However, carefully observing Figure 6.14, it is evident that our method, after successfully recovering from the initial misalignment (at about 16 000 steps), achieves a significant error reduction for the remainder of the experiment. From that point onward, it maintains a tracking error comparable to, and occasionally even lower

Table 6.3: Comparison of voltage tracking performance metrics on the real-world scenario.

Adaptation Method	RMSE [V]	MAE [V]
Static Model	0.1349	0.1313
Cluster Shift	0.0406	0.0320
Memoryless Adaptation	0.0276	0.0212
Evolving Clustering (Ours)	0.0628	0.0470

Note: Best values are highlighted in bold. Lower values indicate better tracking accuracy.

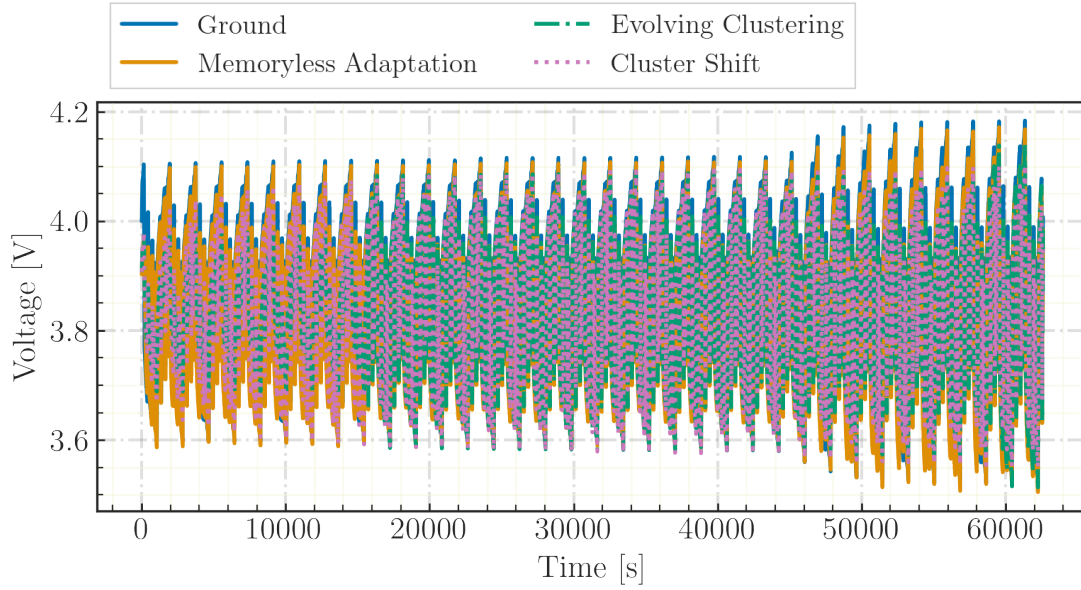


Figure 6.13: Voltage tracking performance of the proposed adaptive routine compared to the static model and passive evolution baselines in the real-world scenario. The plot shows the predicted voltage against the actual voltage over time, highlighting the model’s ability to adapt to the changes in temperature. For better visualization, we reported a zoomed-in view of a portion of the profile in Figure B.3.

than, the *memoryless* baseline. This confirms the robustness of the *evolving clustering* routine even in real-world conditions characterized by partial information about the battery’s conditions and load profile.

Despite the initial transient error, the *evolving clustering* method offers interpretability of the system’s evolution and dynamics. As shown in Figure 6.15, the initial calibration mismatch mainly impacts the estimated ohmic resistance R_0 . This triggers the creation of a new nominal cluster, a mechanism that, in an extensive real-world deployment, is essential for correctly aligning the DT with the physical battery. Moreover, following the occurrence of the thermal jump, the algorithm successfully detects the concept drift and generates a new cluster representing the updated operational regime. To ensure smooth parameter estimation during this experiment, we assigned a higher weight to the regularization term ξ introduced in Equation (5.5). This adequately compensates for the mismatch between the data window size and the signal’s periodicity, which would

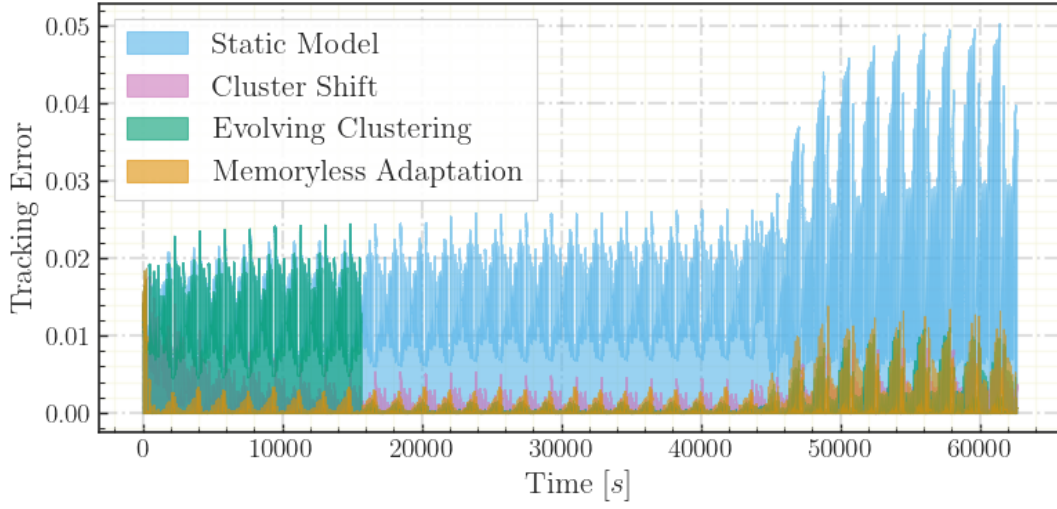


Figure 6.14: Voltage prediction error over time for the real-world scenario. The plot compares the error of the proposed adaptive routine with the static model and passive evolution baselines, demonstrating the effectiveness of the adaptive routine in reducing the error after each thermal jump occurs.

have otherwise led to noisy parameter estimates. Overall, the algorithm successfully tracks the system's evolution, following a coherent trajectory in the parameter space that reflects the physical behavior of the Thevenin components described in Figure 3.2 and Figure 3.3.

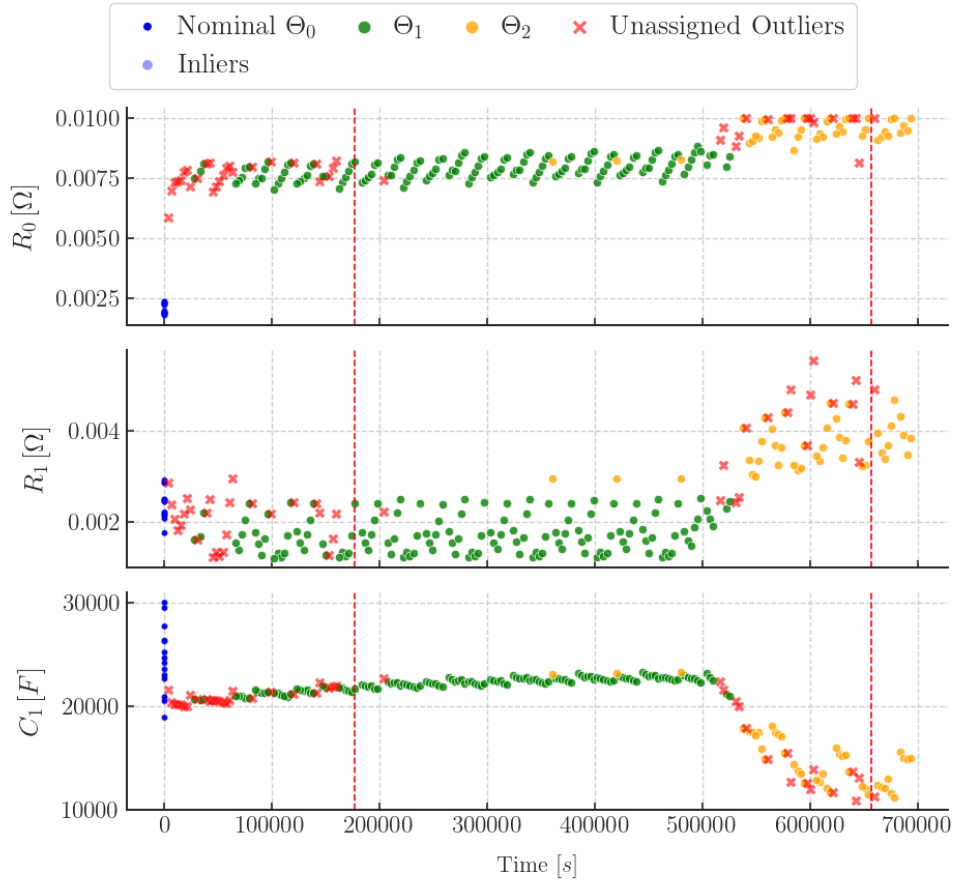


Figure 6.15: 2D visualization of the parameter evolution over time in the real-world scenario. The plot shows the trajectory of the estimated parameters, highlighting the point at which a drift is detected, and a new cluster is formed (vertical dashed lines), indicating the model's adaptation to the new regime.

Part III

Microgrid Energy Management through Reinforcement Learning

CHAPTER 7

Energy Management in Microgrids

While in previous chapters we focused on the BESS as the main component of interest within the DT framework, in this chapter we shift our attention to the broader context of the MG, with the aim of satisfying research objective *RO3*, described in Section 1.4.1. However, the BESS remains a crucial element within the MG, playing a significant role in its operation and management. Indeed, the BESS provides essential services such as energy storage, load balancing, and grid stabilization, which are vital for the efficient functioning of the MG. Because the battery serves as the primary point of control, leveraging a validated and reliable DT, such as ERNESTO, directly facilitates the development of advanced energy management strategies for the MG as a whole.

In this chapter, we will explore the elements that compose a typical MG, the control scheme underlying its operation, and the mathematical formulation of the energy management problem. Next, we will discuss various control strategies adopted in the literature to address this problem and their respective advantages and disadvantages. Notably, this is a necessary preamble to both the methodological developments in the context of a single MG energy management presented in Chapter 8 and the following extension to RECs, described in Part IV, where the members of the community are represented by single MGs. The work presented in this part has been published in ([Salaorni et al., 2025b](#)).

Chapter Outline. The chapter is organized as follows. In Section 7.1, we present the typical control scheme adopted for MG energy management. In Section 7.2, we formulate the energy management problem, detailing the power flow model, the controller's objectives, and the modeling of dispatchable resources within the MG. Finally, in Section 7.3, we review the state-of-the-art methods for MG energy management, highlighting their strengths and limitations.

7.1 A Multi-Layered Control Scheme

In this section, we will delineate the structural classification of typical Microgrids (MGs) and detail the hierarchical control architectures commonly adopted for their management.

7.1.1 Microgrid Topologies

Broadly, MGs can be categorized into three primary topologies based on their bus architecture: alternating current (AC) MGs, direct current (DC) MGs, and hybrid AC/DC MGs. AC MGs (Mohammed et al., 2019) are the most prevalent type, primarily due to their compatibility with existing power systems and infrastructure. They utilize AC power for generation, distribution, and consumption, making them suitable for residential, commercial, and industrial retrofit applications. DC MGs (Al-Ismail, 2021), on the other hand, are gaining traction in specific domains such as data centers, electric vehicle charging stations, and systems inherently reliant on native DC generation (e.g., solar panels). By eliminating AC/DC conversion stages, DC MGs offer superior efficiency and simplified control for integrating DC-native sources and electronic loads. Hybrid AC/DC MGs (Liu et al., 2011) combine both AC and DC buses within a single ecosystem, interconnected via bidirectional interlinking converters. While this architecture introduces additional control complexity, it offers maximum flexibility, allowing diverse energy sources and loads to be integrated at their native voltage levels without unnecessary conversion steps.

7.1.2 Hierarchical Control Framework

Regardless of the specific topology, every MG requires an effective control strategy to ensure stability and meet design objectives. It is widely recognized that MG management involves multiple-objective optimization spanning different, varying technical areas, time scales, and physical layers (Bidram and Davoudi, 2012; Meng et al., 2017). Consequently, the literature has converged on a *multi-layered hierarchical framework*, which decomposes control tasks into three distinct levels, as illustrated in Figure 7.1. Specifically:

1. **Primary control (local control):** Operating on a fastest timescale (milliseconds), this level focuses on the immediate regulation of local power, voltage, and current. It typically follows the setpoints provided by higher-level controllers and executes control actions through interface power converters.
2. **Secondary control (MG-level control):** Operating on a slower timescale (seconds to minutes), this level compensates for the steady-state deviations introduced by the primary control. It addresses system-wide challenges, including power quality regulation, synchronization with the external grid for seamless reconnection, and coordination among the MG units.
3. **Tertiary control (system-level control):** Operating at the lowest timescale (minutes to hours), this level is responsible for overall operative optimization, cost-effective performance, system-wide management, and regulatory compliance.

within the MG. Other existent taxonomies, such as [Olivares et al. \(2014\)](#), view tertiary control as primarily responsible for coordinating multiple MGs or subsystems within the hosting grid, effectively serving as a hosting-grid control layer, delegating the operational and economic optimization of power and energy flows to the secondary level. In this dissertation, we adopt the classification described by [Meng et al. \(2017\)](#), where the tertiary control focuses on the *internal* optimization of the MG.

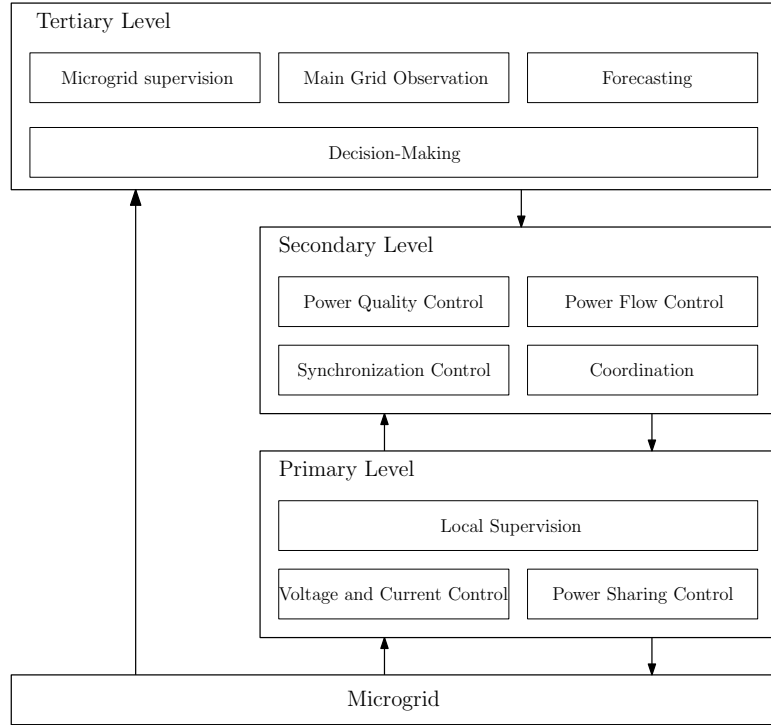


Figure 7.1: Three level hierarchical control scheme proposed by [Meng et al. \(2017\)](#) for the energy management in MGs.

7.1.3 Control Paradigms

Each logical level within the presented hierarchical framework can be implemented using different control paradigms: centralized, decentralized, distributed, or a hybrid hierarchical combination, as illustrated in Figure 7.2. These architectures differ fundamentally in their reliance on a central controller and the extent of communication required between the MG components. The main characteristics of each architecture are summarized as follows:

- **Centralized control:** This paradigm (Figure 7.2a) features a central control unit that collects data from all essential MG components to execute control and management procedures for secure and efficient operation. The advantages of centralized control include strong observability of the entire system and straightforward implementation. However, a significant drawback is the single-point-of-failure issue, as the breakdown of the central controller leads to the loss of all functions. Furthermore, it can suffer from reduced flexibility and scalability and necessitates

considerable computational resources. Consequently, centralized control is generally more suitable for small-scale, localized MGs.

- **Decentralized control:** In this architecture (Figure 7.2b), each unit regulates itself using only local information, without requiring communication links to peers. The advantage of this approach is the absence of real-time communication requirements and *plug-and-play* capability. However, the lack of coordination between local regulators limits the possibility of achieving globally coordinated behaviors.
- **Distributed control:** This structure (Figure 7.2c) eliminates the central coordinator in favor of a peer-to-peer network. Local controllers communicate only with immediate neighbors to coordinate their actions based on both local measurements and reach a consensus on global objectives (e.g., average voltage restoration). This paradigm combines the resilience of decentralized systems with the coordination capabilities of centralized ones. However, achieving effective coordination necessitates well-designed communication protocols and consensus algorithms to enable agents to reach a unified understanding of shared information in a distributed manner.
- **Hierarchical control:** As modern energy systems grow in complexity, not all functions can be effectively executed in a distributed or decentralized manner. Consequently, a hybrid hierarchical control structure (Figure 7.2d) is becoming a standard configuration. This approach assigns specific paradigms to each logical layer: *primary control* is typically managed by local controllers and kept decentralized (for speed and reliability), while *secondary* and *tertiary functions* are implemented using traditional centralized or distributed schemes (to handle global optimization). This hybrid approach leverages the strengths of each paradigm and maximizes the trade-off between local responsiveness and global efficiency.

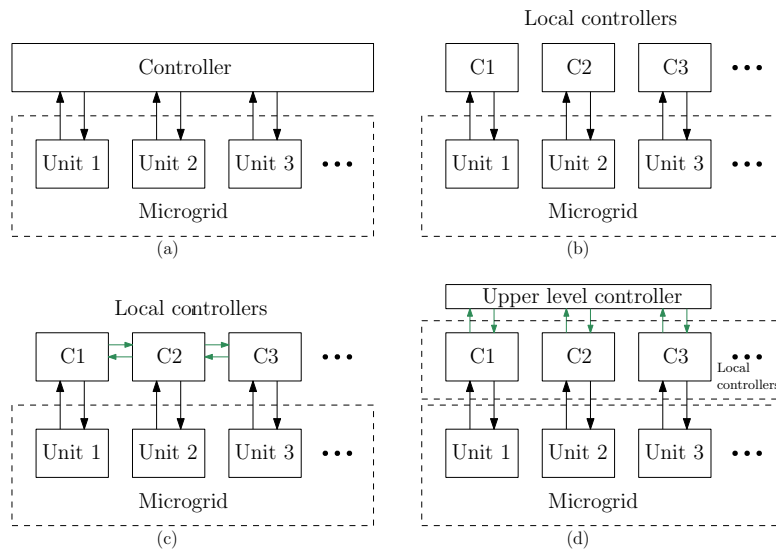


Figure 7.2: Control paradigms defined by Meng et al. (2017). (a) Centralized. (b) Decentralized. (c) Distributed. (d) Hierarchical. Green arrows denote communication flows.

7.2 Problem Formulation

In this dissertation, we focus on the design of a *tertiary-level controller* for MG energy management, as intended by Meng et al. (2017). The primary objective is to optimize power and energy flows within the MG ecosystem while adhering to economic and technical constraints. In particular, we model the power exchange at the Point of Common Coupling (PCC) between the MG and the main utility grid. This framework serves as a generalized tertiary control layer capable of accounting for operational limits and costs of MG components, market conditions, grid constraints, and long-term objectives. Moreover, it is designed to be adaptable to specific operational contexts, including both grid-connected and islanded modes.

In this section, we first describe the power flow framework and the controller's objectives. Then, we model the dispatchable MG components and their operational constraints.

7.2.1 Power Flow Model

The MG operation is modeled over a finite time horizon (typically one day), defined as $\mathcal{T} = \{1, \dots, T\}$, discretized into $T \in \mathbb{N}$ time slots of duration $\Delta t \in \mathbb{R}$. The MG is connected to the main grid or other MGs, and typically includes various *dispatchable* resources $\mathcal{D}$, such as batteries, thermostatically controlled loads (TCLs), or generators like diesel internal combustion engines or microturbines, and *non-dispatchable* sources $\mathcal{N}$, like wind turbines and solar PV systems.

We define the (average) power $p_{\text{net},t}$ exchanged with the main grid at time t as the algebraic sum of all internal power injections:

$$p_{\text{net},t} = \sum_{i \in \mathcal{D}} p_{i,t}^{\mathcal{D}} + \sum_{j \in \mathcal{N}} p_{j,t}^{\mathcal{N}}, \quad (7.1)$$

where $p_{i,t}^{\mathcal{D}}, p_{j,t}^{\mathcal{N}} \in \mathbb{R}$ denote the power output of the i -th dispatchable and j -th non-dispatchable unit at time t , respectively. Adopting the standard load convention, $p > 0$ denotes the power consumption (load) when it is absorbed by the unit, and $p < 0$ denotes the power generation (source) when it is injected into the grid.

An example of the power exchange within the system is provided in Figure 7.3, where a dispatchable load, a non-dispatchable load, a battery, a non-dispatchable generator, and a dispatchable generator (from left to right) contribute to the overall net power of the MG.

Typically, the MG and the main grid coordinate in advance, by means of the distribution system operator (DSO), to establish a power exchange profile $\bar{\mathbf{p}} = [\bar{p}_1, \dots, \bar{p}_T]$. Indeed, defining in advance $\bar{p}_t$ for each time slot t ensures that the MG neither injects surplus energy into the network nor experiences a shortage that could lead to unmet demand. This contributes to minimizing energy losses and maintaining a stable and reliable exchange with the main grid. The controller's objective is to track this reference, aiming to maintain:

$$p_{\text{net},t} = \bar{p}_t, \quad \forall t \in \mathcal{T}. \quad (7.2)$$

To achieve this, the controller must adjust for any discrepancies between the actual and forecasted power outputs of non-dispatchable units in $\mathcal{N}$, as well as account for possible

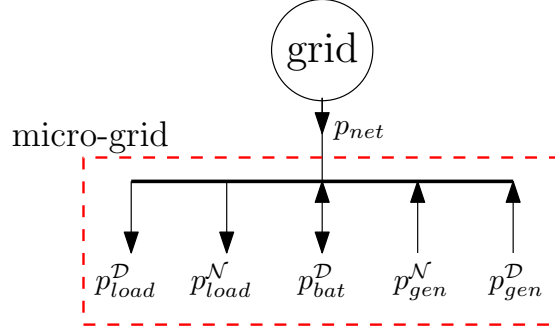


Figure 7.3: Schematic view of the power exchange flows between an MG and the main grid with different dispatchable and non-dispatchable units.

variations in energy prices. This involves determining the optimal power allocation for the flexible units in $\mathcal{D}$ while ensuring compliance with their respective operational limits. Some remarks are needed. First, note that when $\bar{p}_t = 0$ at each instant t , the control objective effectively promotes full self-consumption within the MG. This aligns with the typical requirements of islanded operation, where local power balance must be maintained without external grid support. Second, depending on the nature of the specific unit involved, the power values $p_{\cdot,t}^{\mathcal{D}}$ and $p_{\cdot,t}^{\mathcal{N}}$ may need to satisfy specific physical constraints. We will discuss some of them in Section 7.2.2.

The optimality of power dispatch is assessed based on unit-specific operational costs, such as fuel consumption for generators and degradation costs for batteries, alongside the economic value of energy exchanges with the main grid. Additionally, imbalance penalties resulting from deviations from the scheduled power profile are considered and quantified by:

$$\Delta p_t = \bar{p}_t - p_{net,t}, \quad \forall t \in \mathcal{T}. \quad (7.3)$$

Other relevant objectives may also be included, such as those discussed in Sections 7.3.1 and 7.3.2, which detail additional cost components of interest.

7.2.2 Resource Modeling

This section presents possible modeling of flexible dispatchable resources ($\mathcal{D}$) within the MG and their corresponding operational dynamics, physical constraints, and cost functions. Specifically, we detail models for generators, batteries, and TCLs. We refer the interested reader to [Palma-Behnke et al. \(2013\)](#) for forecasting models of non-dispatchable sources.

Since we will refer solely to dispatchable units, we will omit the apex $\mathcal{D}$ to identify power and objectives associated with such resources. However, we will keep a suitable subscript to distinguish between different resources.¹

Dispatchable Generator. A dispatchable generator is a power source that can adjust its output based on demand and operational constraints. Examples include diesel generators, microturbines, and fuel cells. These generators are typically employed to provide backup power during outages or to meet peak demand periods.

¹For example, we will refer to p_{gen} for a generator power and p_{bat} for a battery power, instead of $p_{gen}^{\mathcal{D}}$ and $p_{bat}^{\mathcal{D}}$.

The power output p_{gen} for a generic dispatchable generator is commonly subject to a constraint of the following kind:

$$\ell_{\text{gen}}^p \leq p_{\text{gen},t} \leq u_{\text{gen}}^p, \quad \forall t \in \mathcal{T}, \quad (7.4)$$

ensuring it stays within the predefined minimum $\ell_{\text{gen}}^p \in \mathbb{R}$ and maximum $u_{\text{gen}}^p \in \mathbb{R}$ power limits, respectively. The values ℓ_{gen}^p and u_{gen}^p can correspond to either safe operational boundaries or preferred performance ranges. For instance, generators are typically not operated exactly at their extreme safety thresholds. Instead, conservative operating margins are established to allow flexibility for increasing or decreasing power output in response to grid demands.

To prevent mechanical stress, ramp-rate constraints can also be imposed to limit the rate of change of power output. Formally, we require that:

$$\ell_{\text{gen}}^r \leq p_{\text{gen},t} - p_{\text{gen},t-1} \leq u_{\text{gen}}^r, \quad \forall t \in \mathcal{T}, \quad (7.5)$$

where $\ell_{\text{gen}}^r \in \mathbb{R}$ and $u_{\text{gen}}^r \in \mathbb{R}$ are the lower and upper bounds for the ramp-rate constraints, respectively. In this case, the initial power value $p_{\text{gen},0}$ is assumed to be known.

In the case of a small-scale fuel-based generator utilized within an MG, the main operational cost arises from fuel usage $\mathcal{J}_{\text{fuel}}$, which is commonly modeled as a quadratic function of the generated power. Formally:

$$\begin{aligned} \mathcal{J}_{\text{fuel}}(p_{\text{gen},t}) &= c_t^{\text{fuel}} f_{\text{gen}}(p_{\text{gen},t}) \\ &= c_t^{\text{fuel}} [\alpha p_{\text{gen},t}^2 - \beta p_{\text{gen},t} + \gamma], \end{aligned} \quad (7.6)$$

where c_t^{fuel} represents the fuel price, $f_{\text{gen}}(p_{\text{gen},t})$ denotes the fuel consumption rate as a function of the production, and the coefficients $\alpha, \beta, \gamma \in \mathbb{R}$ characterize the cost of the generator output (Nguyen and Crow, 2015). Specifically, $\alpha > 0$ guarantees that the marginal cost increases with production; $\beta \geq 0$ ensures the cost is non-decreasing since $p_{\text{gen},t}$ is negative, so that each additional kW of production increases the unit cost; and $\gamma = 0$ ensures zero cost in the absence of production. The parameter c_t^{fuel} encompasses fuel expenses at time t as well as additional costs, such as transportation and on-site storage. Furthermore, the cost model in Equation (7.6) can be applied to fuel cells and microturbines, though with different parameter values. For instance, in the work by Parisio et al. (2014), the quadratic cost in Equation (7.6) is approximated using the maximum of a set of affine functions, achieving this simplification without the need to introduce binary variables. A similar approach is adopted by Palma-Behnke et al. (2013), where it is used to approximate a non-convex cost function. Higher-order polynomial cost functions have also been proposed, e.g., by Sachs and Sawodny (2016), in which a third-degree polynomial function of power was introduced.

In addition to the fuel consumption term in Equation (7.6), the total cost of operating a fuel-based generator may also include operation and maintenance (O&M) expenses, as well as startup and shutdown costs (Parisio et al., 2014; Hu et al., 2023). The overall cost can thus be expressed as:

$$\begin{aligned} \mathcal{J}_{\text{gen}}(p_{\text{gen},t}) &= \mathcal{J}_{\text{fuel}}(p_{\text{gen},t}) + \\ &\quad + c_t^{\text{om}} \delta_{\text{gen},t} + \end{aligned}$$

$$\begin{aligned}
 &+ c_t^{\text{su}} [\delta_{\text{gen},t} - \delta_{\text{gen},t-1}] + \\
 &+ c_t^{\text{sd}} [\delta_{\text{gen},t-1} - \delta_{\text{gen},t}],
 \end{aligned} \tag{7.7}$$

where $\delta_{\text{gen},t} \in \{0, 1\}$ denotes the generator's status (ON/OFF), and c_t^{om} , c_t^{su} and c_t^{sd} represent the O&M, time-dependent startup and shutdown costs, respectively.

Finally, to mitigate frequent switching and enforce minimum utilization or rest periods, minimum up (T^{up}) and minimum down (T^{down}) time constraints can be imposed:

$$\begin{aligned}
 \delta_{\text{gen},t} - \delta_{\text{gen},t-1} &\leq \delta_{\text{gen},\tau}, \quad \forall \tau \in \{t+1, \dots, t+T^{\text{up}}-1\} \\
 \delta_{\text{gen},t-1} - \delta_{\text{gen},t} &\leq 1 - \delta_{\text{gen},\tau}, \quad \forall \tau \in \{t+1, \dots, t+T^{\text{down}}-1\}
 \end{aligned} \tag{7.8}$$

These constraints ensure that once the generator is switched ON (or OFF), it must remain in that state for at least the specified duration (Parisio et al., 2014).

Battery Storage System. A battery unit acts like a prosumer, since it is capable of bidirectional power flow. Consistent with our sign notation, $p_{\text{bat}} > 0$ denotes power absorption by the device (charging), while $p_{\text{bat}} < 0$ denotes power withdrawal (discharging). Beyond the satisfaction of power and ramp-rate constraints as given in Equation (7.4) and Equation (7.5), for a battery, it may also be necessary to impose limits on the minimum and maximum amount of stored energy. Let us first define the state of energy (SOE) dynamics with the following recursive equation:

$$e_{\text{bat},t+1} = \zeta e_{\text{bat},t} + \eta_t p_{\text{bat},t} \Delta t, \quad \forall t \in \mathcal{T}. \tag{7.9}$$

In this expression, $e_{\text{bat},t} \in \mathbb{R}$ represents the energy content of the battery at time t , Δt denotes the duration of the discrete-time slot, and $\zeta \in (0, 1]$ the self-discharge coefficient. Notably, most lithium-ion batteries exhibit a self-discharge rate $(1 - \zeta)$ ranging between 0.5% and 3% per month. Finally, η_t is defined as:

$$\eta_t = \begin{cases} \eta^+, & p_{\text{bat},t} > 0 \quad (\text{charging}), \\ 1/\eta^-, & p_{\text{bat},t} < 0 \quad (\text{discharging}). \end{cases} \tag{7.10}$$

Here, $\eta^+, \eta^- \in (0, 1]$ denote the charging and discharging efficiencies, respectively (Parisio et al., 2014; Hu et al., 2023). Note that in discharging mode ($p_{\text{bat}} < 0$), the term $1/\eta^-$ implies that the energy drawn from the battery is higher than the energy delivered to the grid, accounting for efficiency losses.

Typically, batteries are never operated at their extreme charge levels due to degradation effects, which become more pronounced at very high or low SOC levels. Moreover, operating near these limits can lead to issues such as overvoltage, undervoltage, or stability concerns, particularly at low charge levels (Zhu et al., 2021). To mitigate these risks, energy constraints are imposed. For example:

$$\ell_{\text{bat}}^e \leq e_{\text{bat},t} \leq u_{\text{bat}}^e, \quad \forall t \in \mathcal{T}, \tag{7.11}$$

ensure that the energy content stays within the predefined minimum $\ell_{\text{bat}}^e \in \mathbb{R}$ and maximum $u_{\text{bat}}^e \in \mathbb{R}$ energy limits, respectively. Note that by unrolling the dynamics of $e_{\text{bat},t}$ in Equation (7.9), the constraint in Equation (7.11) can be equivalently expressed in terms of power.

The energy dynamics described in Equation (7.9), together with the mode-dependent efficiencies defined in Equation (7.10), are typically handled by separating charging ($p_{\text{bat}} > 0$) and discharging ($p_{\text{bat}} < 0$) power flows and introducing auxiliary binary variables to indicate the active operational mode at time t (Parisio et al., 2014). However, an alternative formulation proposed by Vignali et al. (2022) expresses the same dynamics without the need for binary mode indicators. Specifically, the model in Equation (7.9), with $\eta_{\text{bat},t}$ defined as in Equation (7.10), can be equivalently written as:

$$e_{\text{bat},t+1} = \zeta \cdot e_{\text{bat},t} + \Delta t \min \left\{ \eta^+ p_{\text{bat},t}, \frac{1}{\eta^-} p_{\text{bat},t} \right\}. \quad (7.12)$$

This formulation naturally eliminates the need for binary variables by inherently capturing the transition between charging and discharging modes through a piecewise expression, being rather advantageous for convex optimization if relaxed appropriately.

As for dispatchable generators, also for batteries, the operational cost is typically modeled with a stress-based formulation as a quadratic function (He et al., 2023b). Formally:

$$\mathcal{J}_{\text{bat}}(p_{\text{bat},t}) = c_t^{\text{bat}} (p_{\text{bat},t})^2, \quad (7.13)$$

where the coefficient c_t^{bat} may explicitly incorporate the cost per kWh for charging, heat dissipation losses during both charging and discharging, and maintenance-related expenses depending on the specific battery design (Nguyen and Crow, 2015). Moreover, it can be interpreted as the cost associated with making 1 kWh of storage capacity available, which is influenced by the battery's replacement cost and its total lifetime cycling capacity, which depends on factors such as the rated depth of discharge, nominal capacity, and expected lifespan. The quadratic term in Equation (7.13) accurately represents the instantaneous stress associated with high currents, both during charging and discharging.

Alternative modeling of the cost has been used by Hu et al. (2023), in which the cost is considered linearly dependent on the exchanged power. Formally:

$$\mathcal{J}_{\text{bat}}^{\text{lin}}(p_{\text{bat},t}) = c_t^{\text{bat}} |p_{\text{bat},t}|. \quad (7.14)$$

In this case, considering the cumulative cost over a specified time horizon, the cost function penalizes the total amount of energy exchanged over time. A positive weighted combination of the cost terms in Equation (7.13) and Equation (7.14) preserves the convexity of the resulting cost function.

Thermostatically Controlled Load. TCLs are electrical appliances, such as air conditioners, heat pumps, electric water heaters, and refrigerators, having inherent thermal inertia that allows them to function as "virtual" energy storage systems. By modulating their power consumption while maintaining temperatures within a user-defined comfort band, they can provide flexibility similar to a battery. The temperature evolution of a TCL is typically modeled using a first-order difference equation, formalized as:

$$\vartheta_{\text{tcl},t+1} = \xi \vartheta_{\text{tcl},t} + (1 - \xi)(\vartheta_{a,t} - \chi p_{\text{tcl},t}), \quad \forall t \in \mathcal{T}, \quad (7.15)$$

where $\vartheta_{\text{tcl},t}$ is the indoor temperature, $\vartheta_{a,t}$ is the ambient temperature, and $p_{\text{tcl},t} \geq 0$ is the average power consumption obtained by switching the TCL between off (zero power)

and on (nominal power P_n). The parameters $\xi, \chi \in \mathbb{R}$ are parameters characterizing the thermal inertia and efficiency of the controlled zone, with $\chi > 0$ for a cooling unit and $\chi < 0$ for a heating unit. Moreover, to ensure user comfort, TCL's temperature must remain within specified limits. Formally, we can impose temperature constraints of the form:

$$\ell_{\text{tcl}}^\vartheta \leq \vartheta_{\text{tcl},t} \leq u_{\text{tcl}}^\vartheta, \quad \forall t \in \mathcal{T}, \quad (7.16)$$

where $\ell_i^\vartheta \in \mathbb{R}$ and $u_i^\vartheta \in \mathbb{R}$ represent minimum and maximum temperature limits, respectively.

Starting from Equation (7.15), by multiplying all terms by a factor C_{th}/η , we can express it in terms of the thermal energy, defined as:

$$e_{\text{tcl},t} = \frac{C_{\text{th}}}{\eta} \vartheta_{\text{tcl},t}. \quad (7.17)$$

Here, C_{th} is the thermal capacitance modeling the amount of thermal energy that can be stored in the zone, and η is the coefficient of performance. Similarly, the temperature limits in Equation (7.16) can be equivalently expressed as energy constraints, as shown in Equation (7.11). Therefore, TCLs can be regarded as generalized energy storage systems and thus represented using the model in Equation (7.9), with power and capacity constraints in the form of Equation (7.4) and Equation (7.11).

Given that the nominal power at time t , P_t^{nom} , required to keep a TCL at its set-point ϑ_s is defined as:

$$P_t^{\text{nom}} = \frac{\vartheta_{a,t} - \vartheta_s}{\chi}, \quad (7.18)$$

where the operating cost of a TCL can be defined as the discomfort caused by deviating from the desired set-point. Formally, this can be expressed as:

$$\mathcal{J}_{\text{tcl}}(p_{\text{tcl},t}) = c^{\text{tcl}} (p_{\text{tcl},t} - P_t^{\text{nom}})^2, \quad (7.19)$$

with $c^{\text{tcl}} > 0$ being the cost to maintain the thermal comfort. A deviation of the power $p_{\text{tcl},t}$ from P_t^{nom} leads to an approximate temperature change, defined as $\Delta\vartheta \approx \chi|p_{\text{tcl},t} - P_t^{\text{nom}}|$. By choosing $c^{\text{tcl}} = \alpha\chi^2$, with $\alpha > 0$, the cost function can be normalized so that a specified temperature deviation results in a proportional increase in cost.

7.2.3 Bidirectional Exchanges with the Main Grid

When operating in grid-connected mode, the MG can both export energy to and import energy from the main grid. This bidirectional energy exchange can be viewed as an additional source of flexibility available to the controller. It enables cost optimization of MG operations by, for instance, taking advantage of favorable market prices to sell excess energy or to charge storage systems in preparation for future energy deficits.

Power constraints in the form of Equation (7.4) can be used to model the minimum and maximum allowable power flow through the transmission line. Electricity purchase costs and sale revenues can be modeled as follows:

$$\mathcal{J}_{\text{grid}}(p_{\text{net},t}) = c_t^{\text{buy}} \max\{p_{\text{net},t}, 0\} - c_t^{\text{sell}} \max\{-p_{\text{net},t}, 0\}, \quad (7.20)$$

where the coefficients $c_t^{\text{buy}}, c_t^{\text{sell}} \in \mathbb{R}_0^+$ represent the time-varying unitary buying and selling prices, respectively. Finally, by observing that $c_t^{\text{buy}} \geq c_t^{\text{sell}}, \forall t \in \mathcal{T}$, Equation (7.20)

can be expressed in the more compact form as follows:

$$\mathcal{J}_{\text{grid}}(p_{\text{net},t}) = \max \left\{ c_t^{\text{buy}} p_{\text{net},t}, c_t^{\text{sell}} p_{\text{net},t} \right\}, \quad (7.21)$$

where the max operator automatically accounts for whether the MG is importing or exporting power, correctly assigning costs or revenues. Alternatively, to explicitly distinguish between purchasing and selling modes, an auxiliary binary variable can be introduced, as proposed by (Parisio et al., 2014).

7.3 Existing Approaches

In this section, we provide an overview of how the control problem formalized in Section 7.2 has been addressed in the automatic control and systems literature. Historically, the dominant paradigm for the energy management problem in MGs has been *Model Predictive Control* (MPC) (Richalet, 1976; Mayne, 2014).

Initially developed in the 1970s, MPC has evolved into the gold standard for constrained multivariable control. The fundamental principle of MPC is the use of an explicit predictive model to forecast the future evolution of system states and outputs over a finite time horizon. At each sampling instant, an optimization problem is solved to determine a sequence of control actions that minimizes a cost function while satisfying system constraints. Crucially, MPC operates on a *Receding Horizon* strategy: only the first element of the computed control sequence is immediately applied. At the next time step, the system state is measured (or estimated), the prediction horizon is shifted forward, and the optimization is repeated. Interestingly, this concept mirrors the intuitive decision-making process followed by humans. For example, a skilled chess player evaluates future scenarios by simulating several possible moves and counter-moves before selecting the one that seems most advantageous. This *closed-loop* implementation provides inherent robustness against disturbances. Its widespread adoption in the energy sector is primarily due to its ability to rigorously handle hard constraints (e.g., battery SOC limits, grid capacities) and non-linear dynamics directly within the controller formulation.

More recently, *Reinforcement Learning* (RL) techniques (Sutton and Barto, 2018) have also been proposed to address the MG energy management problem. RL is a subfield of ML that focuses on training intelligent agents to make sequential decisions by interacting with a stochastic environment modeled as a Markov Decision Process (MDP). Unlike MPC, which requires a pre-engineered physical model, standard RL is *model-free*: the agent learns an optimal control policy by maximizing a cumulative reward signal through trial-and-error interaction with the physical (or virtually simulated) system. This enables the agent to capture complex, hard-to-model phenomena directly from data. We provide a rigorous theoretical treatment of RL in Section 8.1.

Notably, MPC and RL offer complementary strengths. First, while MPC relies on the accuracy of the internal physical model, which may degrade performance if it mismatches reality (system identification error), RL learns directly from the system's response, but requires massive amounts of training data. Second, MPC is computationally intensive *online*, as it requires solving an optimization problem at every decision step. RL shifts this burden *offline*; once the policy (e.g., a neural network) is trained, execution

is extremely fast, making it ideal for real-time applications. Third, MPC provides mathematical guarantees on constraint satisfaction. RL typically requires auxiliary mechanisms (e.g., reward shaping or projection layers) to ensure safety constraints are respected during operation.

In summary, MPC is generally preferred when accurate models are available, and safety is the main priority, while RL excels in highly stochastic, partially observable environments where adaptability and real-time execution speed are critical. The following subsections (Section 7.3.1 and Section 7.3.2) review representative works from both categories.

7.3.1 MPC-based Approaches

Among MPC approaches, in the literature, we can identify three main groups: those focusing on grid-connected MGs, those addressing islanded MGs, and those on networked MGs. In the following, we report key contributions for every category.

Grid-connected MGs. In grid-connected scenarios, the primary objective is economic optimization (Hu et al., 2023), intended as balancing market revenues against operational costs (fuel, degradation, and startup penalties). The presence of hybrid dynamics, such as continuous power flows and binary generator states, naturally leads to *Mixed-Integer Linear Programming* (MILP) formulations.

Parisio et al. (2014) present a foundational MILP-MPC framework for minimizing operating costs, including fuel consumption, generator startup and shutdown, maintenance expenses, and revenues from energy sales, along with a load curtailment penalty to model user discomfort. Embedded in a hierarchical control architecture, the MILP-MPC layer functions centrally for long-term optimization, while subordinate controllers address fast electrical dynamics such as voltage and frequency regulation. Building on this, Clarke et al. (2016, 2018) emphasize robustness under disturbances and uncertainties by introducing artificial SOC constraints to prevent violation of operational limits. Moreover, they integrate terminal costs in the objective function to preserve battery energy at the end of the prediction horizon, supporting long-term system sustainability. To explicitly address asset longevity, Ju et al. (2017) incorporates detailed degradation models. For batteries, degradation costs are expressed as nonlinear functions dependent on full capacity and DOD, capturing the wear from deep and frequent cycling, translating long-term capital degradation into short-term operational costs. In contrast, they model supercapacitor degradation using a simple linear time-based function, reflecting the higher durability and suitability for high-frequency power balancing.

To mitigate the computational burden of MILP solvers, Pippia et al. (2019) propose a *Rule-Based MPC* (RBMPC). This technique pre-assigns binary decision variables via heuristic rules derived by economic signals, such as electricity prices, renewable generation, and load profiles, reducing the MILP problem to a Linear Program (LP) and substantially lowering computational burden with minimal performance loss. To better manage uncertainties, stochastic MPC solutions have been proposed. Kou et al. (2019) utilized a two-layer hierarchical *Stochastic MPC* (SMPC) to balance EV charging demands under uncertain wind generation, approximating the problem as a Quadratic Program (QP) for computational efficiency. Similarly, Parisio et al. (2016) applied two-stage stochastic programming to address economic and environmental operation

management, minimizing expected costs of correction actions and/or emissions of atmospheric pollutants.

Islanded MGs. When an MG operates in islanded mode, disconnected from the main power grid, the control task becomes considerably more challenging. Without external grid support, the MG must independently balance supply and demand, maintain voltage and frequency stability, and manage the uncertainties associated with the RES variability and load fluctuations (Lopes et al., 2006). These conditions place significantly higher demands on MPC in terms of robustness, responsiveness, and operational reliability. In response to these challenges, the literature has explored a variety of strategies to adapt and improve MPC for islanded operations. These include hierarchical control architectures, which layer decision-making across timescales, robust control strategies that guard against worst-case uncertainties, and stochastic approaches that exploit probabilistic forecasts.

In the work by Nahata et al. (2020), the authors propose a hierarchical MPC-based EMS governing power setpoints and operational modes for distributed generation units (DGUs), such as batteries, PVs, and dispatchable generators. This supervisory layer seeks to minimize generation costs and improve DGU efficiency by reducing unnecessary mode switching, such as between charging/discharging or ON/OFF states, while the secondary control layer determines the voltage setpoints by optimizing the power flow. Similarly, Nair and Costa-Castelló (2020) utilize MPC to maximize operational efficiency and renewable energy utilization of an isolated MG while minimizing the degradation of a hybrid ESS. They design the cost function to enhance ESS longevity by penalizing SOC variations, as well as fuel cell power output and ramping rates. Additionally, soft constraints on ESS capacity limits are included to ensure feasibility.

While hierarchical MPC structures help manage system complexity and different time scales, they often require additional mechanisms to address uncertainty. To this end, *Robust MPC* (RMPC) formulations have been proposed. Cai et al. (2020) specifically focus on the problem of restoring critical loads in a disconnected MG, where the distribution system is reconfigured into multiple autonomous MGs, and it is necessary to ensure continuity of service to priority loads. Here, the main challenge lies in the uncertainty associated with renewable generation output, whose variability complicates accurate forecasting. To tackle this, they propose a two-layer robust optimization model, resilient to the "worst-case" realization of renewable generation.

While robust methods guarantee safety, they can be overly conservative. SMC approaches, instead, offer probabilistic alternatives, balancing risk and performance. Hans et al. (2015) propose a *Scenario-Based Stochastic MPC* (SMPC) approach that explicitly incorporates the probabilistic nature of forecasts by representing uncertainty through a computationally tractable scenario tree, which is generated and reduced using Monte Carlo simulations. An SMPC solution is also proposed by Polimeni et al. (2021), within a two-layer architecture where the control strategy is formulated as piecewise affine policies, which provide correction rules based on real-time feedback from the observed disturbances.

Networked MGs. The concept of networked MGs, i.e., systems composed of multiple electrically interconnected AC and DC MGs, has gained significant attention due to its

potential to enhance the integration of RES and accommodate higher load demands. Besides these benefits, the coordination and optimization of power flow in networked MGs introduce greater complexity (Chen et al., 2020). To address this, decentralized and distributed architectures and cooperative schemes have emerged.

One such approach is a two-layer control architecture proposed by Bonassi et al. (2021), which divides the distribution network into distinct, non-overlapping clusters. At the lower layer, each cluster operates a local *Decentralized MPC* (DMPC) that autonomously manages active power fluctuations. A centralized supervisor coordinates inter-cluster power transfers via controllable DC links only when a local controller requires external support. A DMPC-based framework is also explored by Baker et al. (2016), where a DMPC strategy coordinates storage systems and stochastic renewable resources across multiple control areas, by focusing on scalability and efficient resource sharing. Furthermore, Zhao et al. (2022b) integrate robust optimization, forming a *Distributed Robust MPC* (DRMPC) framework. Targeting a network of isolated MGs featuring an internal energy market, this approach ensures feasibility under "worst-case" uncertainty scenarios while maintaining real-time adaptability.

Finally, Parisio et al. (2017) present a cooperative MPC-based distributed coordination algorithm, designed to ensure constraint satisfaction, promote cooperation, and guarantee fairness within a networked MGs system with shared DERs. A distinct advantage of the framework is its ability to guarantee cost savings for each participating MG by incentivizing collaboration within a three-phase iterative process.

7.3.2 RL-based Approaches

The literature offers a wide array of RL techniques and perspectives for MG energy management. The predominant formulation models the problem as an optimal power scheduling task typically approached from an *MG-centric* perspective. In this paradigm, the control agent is embedded within the MG, often paired with the BESS. Most works adopting this paradigm rely on centralized control and apply either traditional tabular RL methods or more advanced DRL algorithms to control the amount of energy to store or retrieve from the BESS. Few other works address the problem with a *Grid-centric* perspective by controlling the DSO to decide the energy retail price.

Tabular Methods. The first set of methods relies on the tabular representation of the MDP, considering the states and actions as a finite set. This allows the use of value-based RL techniques to solve them. Within this setting, Kuznetsova et al. (2013) implement a two-step-ahead tabular Q-learning algorithm (Watkins and Dayan, 1992) to achieve optimal power scheduling within an MG system. Their approach modeled a simplified environment including electricity prices, demand, the battery SOC, and forecasts of wind turbine generation. The action space included decisions for two step-ahead time instants, consisting of boolean signals that tell whether to use or not the battery. Instead, the work by Foruzan et al. (2018) propose a multi-agent framework in which multiple independent Q-learning agents represent buyer and seller entities participating in an auction-based energy market within an MG. Each agent independently learns its bidding strategy based on local observations and expected outcomes. More specifically, the system is able to learn the amount of energy to trade and the associated bid price. The agents are designed to maximize individual utility functions, which reflect the profit derived from successful

market participation or the cost incurred through fallback interactions with the main grid in the event of rejected bids. Later on, [Zhou et al. \(2022\)](#) extends the Q-learning framework to a more sophisticated setting that incorporates multiple dispatchable and non-dispatchable energy resources. In their setup, the agent observes estimated resource values generated by a deep learning-based forecasting module.

Continuous State Space. Given the inherently continuous nature of the MG control problem, more recent works predominantly employ DRL techniques, as discretizing continuous observation or action spaces often leads to suboptimal approximations. A frequently adopted method is the Deep Q-Network (DQN) algorithm ([Mnih, 2013](#)), which, while effective, still requires a discretized action space. For example, [Chen and Bu \(2019\)](#) investigate the use of DQN in a multi-MG trading scenario, where each MG agent independently optimizes its energy trading strategy. Their model assumes a peer-to-peer (P2P) electricity market. The agents seek to maximize a utility function encompassing trading profits and dependency on a centralized power supply.

In contrast, [Domínguez-Barbero et al. \(2020\)](#) apply DQN to manage energy resources in an islanded MG. Their setup includes hydrogen and battery storage systems, a diesel generator, and photovoltaic generation, to minimize operational costs associated with non-dispatchable generation and unmet energy demand. Further advancing real-world applicability, [Kolodziejczyk et al. \(2021\)](#) apply DQN to real consumption and generation data, using two representative simulation periods (a winter week and a summer week). Their observation space comprises 24-hour window vectors of net demand, battery SOC, electricity price, and time-of-day features. The action space is discretized to represent various percentages of power purchased from the grid, scaled by the maximum charging capacity of the storage system.

[Nakabi and Toivanen \(2021\)](#) conduct a comparative study of various DRL algorithms within an extended MG control scenario. Interestingly, their experimental setting incorporates TCLs, which introduce additional flexibility into the energy management process. The RL agent is tasked with its conventional responsibilities, such as managing battery dispatch and the direct control of TCLs. For this reason, the agent's observation space is augmented with additional environmental information, such as external temperature, which influences thermal demand. The discrete action space is expanded to include both the battery control actions and set-point commands for TCL energy allocation.

Continuous State and Action Spaces. Some recent studies move beyond traditional *value-based* methods to address scenarios involving continuous action spaces, which are more representative of real-world MG operations. For instance, [Liu et al. \(2021\)](#) and [Guo et al. \(2022\)](#) apply the Deep Deterministic Policy Gradient (DDPG) ([Lillicrap, 2015](#)) and Proximal Policy Optimization (PPO) ([Schulman et al., 2017b](#)) algorithms, respectively, to the MG energy dispatching problem. Their primary objective is to minimize operational costs, including those associated with generator usage, BESS, and power exchanges with the main grid. By dropping the discretization over the action space, these methods allow for more granular control of energy flows, leading to improved performance and adaptability in dynamic environments.

Grid-centric Methods. While the majority of RL-based MG energy management

literature adopts an MG-centric perspective, focusing on local control strategies, a subset of works explores a Grid-centric approach. These methods aim to optimize the behavior of the DSO, an external entity responsible for ensuring the reliable and secure operation of the electrical grid. In these scenarios, the DSO often assumes the role of market maker, tasked with dynamically setting energy prices to influence the behavior of connected MGs in a desirable manner. For instance, [Du and Li \(2020\)](#) propose a Monte Carlo-based RL algorithm to train a DSO to optimize retail pricing strategies within a multi-MG setting. The RL agent in this framework is exclusively responsible for price determination, while the behavior of individual MGs is modeled using deep learning architectures, resulting in a hierarchical control structure. A similar multi-level approach is presented by [Shojaeighadikolaei et al. \(2021\)](#), who adopt a multi-agent framework based on DQN. In their model, a central agent represents the DSO, which is responsible for setting dynamic energy prices, and additional agents correspond to network *prosumers*, which must decide among charging, discharging, or idling their battery systems in response to both local conditions and the DSO's pricing signals.

7.3.3 Hybrid Approaches

Finally, recent research has explored hybrid frameworks that integrate both RL and MPC techniques to capitalize on their complementary strengths. These approaches aim to take the best of both worlds: the model-based precision and constraint-handling capabilities of MPC alongside the adaptability and data-driven learning of RL. However, in the context of MG energy management, such hybrid methods remain relatively underexplored. Hereafter, we outline some promising directions identified in the literature.

A first direction involves using MPC to generate expert demonstrations for RL agents. In this setup, MPC can supervise RL by providing expert trajectories that are then used as expert demonstrations to *warm-start* an RL policy via imitation learning (e.g., behavior cloning or policy distillation). The RL agent can then fine-tune this policy online, adapting to non-modeled dynamics or disturbances that the original MPC model failed to capture.

Another approach integrates safety mechanisms inspired by MPC's constraint-handling capabilities, as proposed by [Patra and Pradhan \(2025\)](#). These *safe-RL* methods augment the policy with action filters, *safety shields*, or specialized critics designed to penalize or prevent predicted constraint violations. This provides feasibility safeguards akin to MPC while retaining the model-free adaptability of RL.

A third strategy uses RL to reduce the computational complexity of the MPC's online optimization. For instance, in the context of MG systems, [Da Silva et al. \(2025\)](#) integrate MPC and RL for the control of mixed-logical dynamical systems. Here, RL is employed to simplify an MILP into an LP by fixing the discrete decision variables, reducing the computational burden of MPC during online operation.

A Reinforcement Learning Approach

While in the previous chapter we provided an overview of the state-of-the-art methods for MG energy management, in this chapter, we present our proposed RL-based approach that directly builds upon the ERNESTO framework introduced in Part I.

As mentioned in Section 7.3.2, RL has recently gained significant attention as a promising solution for the MG energy management problem. However, a significant gap in much of the early RL works is the omission or oversimplification of battery aging and degradation effects, which represent a high economic cost when considering battery replacement and are critical for realistic long-term planning. A partial solution to these shortcomings is offered by [Sui and Song \(2020\)](#), which addresses the problem of scheduling charge, discharge, and resting periods with multiple batteries through a multi-agent approach. Since the study aims to minimize the system degradation, the authors adopt an aging model influenced solely by temperature while ignoring other degradation factors related to usage, such as DOD, SOC, and current rates. More recent studies have started to incorporate these aspects into the problem formulation. For example, [Mussi et al. \(2024\)](#) employs Fitted Q-Iteration (FQI) ([Ernst et al., 2005](#)) to train a policy that maximizes revenues from energy trading while explicitly accounting for battery degradation. However, their model assumes a static energy market with fixed buying and selling prices, ignores ambient temperature variations (which significantly influence battery aging), and restricts control to discrete action spaces.

In this chapter, we employ the capabilities of ERNESTO to propose a novel RL-based approach that overcomes these limitations by incorporating a comprehensive battery degradation model, operating in a dynamic energy market with time-varying prices and supporting continuous action spaces for more granular control. We will frame the MG energy management problem as an MDP ([Puterman, 2014](#)) and describe how we leverage RL techniques to learn an optimal control policy that maximizes economic

profits while minimizing battery degradation costs over time.

Chapter Outline. The chapter is organized as follows. In Section 8.1, we provide a brief introduction to RL concepts relevant to our approach. Next, in Section 8.2, we describe the specific MG scenario to which the RL controller is applied. In Section 8.3, we formulate the energy management problem as an MDP, detailing the state, action, and reward structures. Finally, in Section 8.4, we describe the RL algorithm and the learning procedure, including the integration with ERNESTO framework.

8.1 Fundamentals of Reinforcement Learning

Reinforcement Learning refers to a class of techniques within the broader field of ML that is learning from experience the optimal policy to follow in dynamic environments. The term *reinforcement learning* is derived from psychology, where it is described as «a consequence applied that will strengthen an organism’s future behaviour whenever that behaviour is preceded by a specific antecedent stimulus» (Skinner, 1953). In other words, RL is a method of teaching humans and animals through explicit feedback, known as *rewards*. These rewards serve as evaluations of the subject’s actions and guide behavior toward a desired outcome. The classic example is the training of a pet. To train a dog to sit, one might offer a treat each time the dog performs the correct action. In this way, the dog learns to associate correct behavior with a positive outcome (i.e., the reward-treat), while incorrect behavior yields a null or negative outcome.

Getting back to the AI field, the objective of RL is to train intelligent agents by positively reinforcing desirable actions and discouraging undesirable ones (Sutton and Barto, 2018). This can be achieved by maximizing that positive feedback over time, prioritizing, depending on the task, a farsighted perspective. The behavior of an agent reflects a strategy, called *policy*, that determines the action to perform based on the agent’s current configuration and the surrounding environment.

A major inspiration for the development of RL came from the fields of dynamic programming and optimal control, pioneered by Richard Bellman in the 1950s. Bellman’s introduction of the *Bellman equations* provided the mathematical foundation for decision-making problems and laid the groundwork for the first class of RL methods, known as *value-based* approaches. Notable among these were the contributions of Sutton (1988), with the Temporal Difference learning algorithm, and Watkins and Dayan (1992), with the development of Q-learning. These algorithms allowed agents to learn from incomplete sequences of experience, improving sample efficiency and eliminating the need for an explicit model of the environment. During the same period, another important class of methods, called *policy gradient* algorithms, began to emerge. These approaches enabled the direct optimization of policies, making them particularly suitable for high-dimensional or continuous action spaces. A turning point in the field occurred around 2010 with the integration of neural networks into RL frameworks. This gave rise to the era of *Deep Reinforcement Learning*, enabling agents to achieve superhuman performance in complex domains such as Atari games, chess, and Go. These breakthroughs sparked a renewed wave of interest and research in RL, leading to its application in a wide range of challenging tasks, including robotic manipulation, autonomous driving, financial decision-making, and natural language processing.

8.1.1 Introduction to Markov Processes

To formalize the problem of learning in a decision-making framework, RL approaches rely on the concept of the Markov Decision Process (MDP). This mathematical framework describes the problem $\mathcal{M}$ as a tuple:

$$\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, P, R, \gamma, \mu_0 \rangle, \quad (8.1)$$

where:

- $\mathcal{S} \subseteq \mathbb{R}^n$ is the *state space*, representing the set of all possible environment configurations, i.e., the representations of the environment as perceived by the agent. Based on this information, the agent selects actions to fulfill its objective.
- $\mathcal{A} \subseteq \mathbb{R}^m$ is the *action space*, which defines the set of all possible control inputs the agent can execute to interact with the environment. Such actions may lead to a modification (complete or partial) of the current environment state and are evaluated with a direct scalar feedback signal from the environment (the *reward*).
- $P : \mathcal{S} \times \mathcal{A} \rightarrow \Delta(\mathcal{S})$ is the *transition kernel*. We denote $P(s'|s, a)$ as the probability density of transitioning to state s' given state s and action a . In many practical scenarios, the transition model is unknown or only partially known.
- $R : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ is the *reward function*, $R(s, a)$, i.e., a mapping from state-action pairs to a real-valued outcome. This function evaluates the immediate consequence of executing an action a in a specific state s and plays a pivotal role in guiding the agent's learning process. It should be carefully designed to reflect the objectives of the control task, as it is essentially derived from the problem's objective function.
- $\gamma \in [0, 1)$ is the *discount factor*, which ensures the convergence of infinite-horizon sums and balances the trade-off between immediate and future rewards.
- $\mu_0 \in \Delta(\mathcal{S})$ is the *initial state distribution* stating where the decision-making process starts.

The defining characteristics of an MDP are the *Markov property*, which states that the future state depends only on the current state and action, not on the sequence of events that preceded it. Formally, this can be expressed as:

$$P(s_{t+1}|s_t, a_t, s_{t-1}, a_{t-1}, \dots, s_0, a_0) = P(s_{t+1}|s_t, a_t). \quad (8.2)$$

An MDP unfolds through the continuous interaction between the agent and the environment over discrete time steps $t = 0, 1, 2, \dots$. At each time step t , the agent observes the current state $s_t \in \mathcal{S}$, selects an action $a_t \in \mathcal{A}$ according to its policy, and receives a reward $r_t = R(s_t, a_t)$. The environment then transitions to a new state s_{t+1} based on the transition dynamics defined by $P(s_{t+1}|s_t, a_t)$. This process, described in Figure 8.1, continues iteratively, forming a trajectory of states, actions, and rewards.

8.1.2 The Control Policy

In this setting, a policy $\pi : \mathcal{S} \rightarrow \mathcal{A}$ is a mapping from states to actions, i.e., a rule that selects an action (or defines a distribution over the available actions) in each state. The

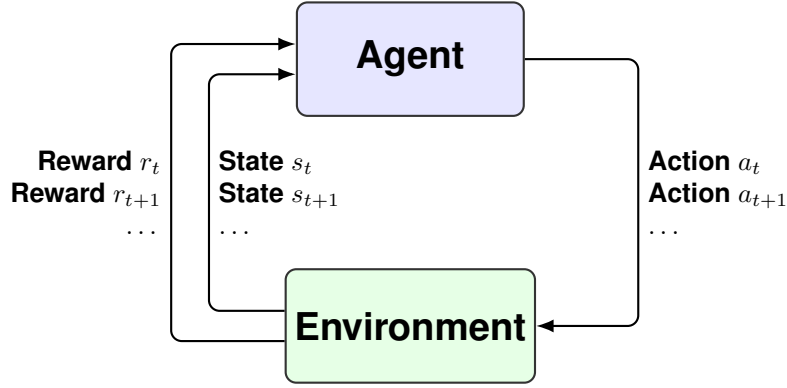


Figure 8.1: Interaction between the agent and the environment in an MDP framework. At each time step t , the agent observes the current state s_t , selects an action a_t according to its policy, receives a reward r_t , and transitions to a new state s_{t+1} .

objective of RL is to find the policy that maximizes the expected cumulative reward over time. This is typically expressed in terms of the expected return, which is the discounted sum of future rewards. Formally, the optimal policy $\pi^* \in \Pi$, where Π is the space containing all the possible policies, can be defined as

$$\pi^* = \arg \max_{\pi} \mathbb{E}_{s_0 \sim \mu_0, a_t \sim \pi(\cdot | s_t)} \left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \right], \quad (8.3)$$

where $\gamma \in [0, 1)$ is the discount factor, which determines the relative importance of immediate versus future rewards, and the expected value is taken w.r.t. the initial state distribution μ_0 and the policy π . The choice of γ should reflect the problem's time horizon, which can be either finite or infinite.

The objective of the RL agent is to identify an optimal policy π^* that maximizes the expected discounted cumulative return:

$$J(\pi) = \mathbb{E}_{\tau \sim \pi} \left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t) \right], \quad (8.4)$$

where the expectation is taken over the trajectory $\tau = (s_0, a_0, s_1, a_1, \dots)$ induced by the policy π and the system dynamics P .

8.1.3 Value Functions

To evaluate and improve the available candidate policies, RL relies on the Bellman equations, which provide recursive formulations of value functions. Two central functions in this framework are:

- The *state-value* function $V^{\pi}(s)$: the expected return starting from state s and following policy π :

$$V^{\pi}(s) = \mathbb{E}_{\pi} \left[\sum_{k=0}^{\infty} \gamma^k R(s_{t+k}, a_{t+k}) \middle| s_t = s \right]. \quad (8.5)$$

- The *action-value* function $Q^\pi(s, a)$: the expected return starting from state s , taking action a , and then following policy π :

$$Q^\pi(s, a) = \mathbb{E}_\pi \left[\sum_{k=0}^{\infty} \gamma^k R(s_{t+k}, a_{t+k}) \middle| s_t = s, a_t = a \right]. \quad (8.6)$$

These functions satisfy the recursive property known as the *Bellman Expectation Equation*, which decomposes the value into the immediate reward plus the discounted value of the successor state:

$$V^\pi(s) = \sum_{a \in \mathcal{A}} \pi(a|s) \sum_{s' \in \mathcal{S}} P(s'|s, a) [r(s, a) + \gamma V^\pi(s')], \quad (8.7)$$

$$Q^\pi(s, a) = \sum_{s' \in \mathcal{S}} P(s'|s, a) \left[r(s, a) + \gamma \sum_{a' \in \mathcal{A}} \pi(a'|s') Q^\pi(s', a') \right]. \quad (8.8)$$

It is possible to show that the optimal policy $\pi^* = \arg \max_\pi \mathbb{E}_{\mu_0} [V^\pi(s)]$ is the one that maximizes the state value function across all states s , so that $V_{\pi^*}(s) \geq V^\pi(s)$ for all $s \in \mathcal{S}$ (Sutton and Barto, 2018). Similar definitions can be provided using the action value function $Q^\pi(s, a)$.

8.1.4 From Tabular Methods to Function Approximation

Classical algorithms such as Q-Learning (Watkins and Dayan, 1992) and Value Iteration (Sutton and Barto, 2018) assume that the state space $\mathcal{S}$ is finite and small enough to store $V^\pi(s)$ or $Q^\pi(s, a)$ in a lookup table. However, most control problems typically involve continuous variables, leading to an infinite or combinatorially explosive state space, a phenomenon known as the *curse of dimensionality*.

To address this, we employ *function approximation*. Instead of storing values for every state, we approximate the value function using a parameterized model $V_{\mathbf{w}}(s) \approx V^\pi(s)$, where $\mathbf{w} \in \mathbb{R}^d$ is a vector of learnable weights (with $d \ll |\mathcal{S}|$). The objective is to minimize the MSE between the approximate value and the true target:

$$L(\mathbf{w}) = \mathbb{E}_{s \sim \rho} [(V^\pi(s) - V_{\mathbf{w}}(s))^2], \quad (8.9)$$

where ρ is the state visitation distribution. Traditionally, linear approximations were used ($V_{\mathbf{w}}(s) = \mathbf{w}^\top \phi(s)$). However, these require manual feature engineering (ϕ) and often fail to capture the complex, non-linear dynamics.

8.1.5 Deep Reinforcement Learning

Deep Reinforcement Learning (DRL) replaces manual function approximators with *Deep Neural Networks* (DNNs). A DNN acts as a universal function approximator, capable of learning high-level abstract representations directly from raw sensor inputs. Depending on the target of the approximation, DRL algorithms are categorized into:

- **Value-Based Methods:** these algorithms aim to learn a value function that estimates the expected return for each action-state pair by approximating $Q^*(s, a) \approx Q_{\mathbf{w}}(s, a)$. The policy is derived implicitly by selecting $\arg \max_a Q_{\mathbf{w}}(s, a)$. Such

methods are generally well-suited for discrete action spaces, but struggle in continuous ones due to the need to enumerate or discretize actions. Notable examples include Q-Learning (Watkins and Dayan, 1992) and Deep Q-Networks (DQN) (Mnih, 2013).

- **Policy-Based Methods:** Directly parameterize the policy $\pi_{\theta}(a|s)$ via a neural network with weights θ . They optimize the objective $J(\theta)$ via gradient ascent: $\theta_{k+1} = \theta_k + \alpha \nabla_{\theta} J(\theta)$. These methods are naturally suited for continuous or high-dimensional action spaces and can represent stochastic policies. These approaches can learn the policy directly by optimizing the expected return through gradient ascent. However, they typically suffer from higher variance and are less sample-efficient. Common examples include REINFORCE (Williams, 2004) and GPOMDP (Baxter and Bartlett, 2001).
- **Actor-Critic Methods:** A hybrid approach where an *Actor* network π_{θ} learns the policy and a *Critic* network V_w estimates the value function to reduce the variance of the gradient estimator. Notable examples are Proximal Policy Optimization (PPO) (Schulman et al., 2017b), Deep Deterministic Policy Gradient (DDPG) (Lillicrap, 2015), and Soft-Actor Critic (SAC) (Haarnoja et al., 2018).

For a more detailed overview of RL, we recommend well-known monographs by Sutton and Barto (2018), Agarwal et al. (2019), and Szepesvari (2010).

8.1.6 Proximal Policy Optimization (PPO)

Among Actor-Critic methods, Proximal Policy Optimization (PPO) (Schulman et al., 2017b) is the current state-of-the-art for continuous control tasks. PPO addresses the instability inherent in standard policy gradient methods, where a single large update step can degrade the policy performance catastrophically.

PPO derives from the Trust Region Policy Optimization (TRPO) algorithm (Schulman et al., 2017a), which enforces a constraint on the Kullback-Leibler (KL) divergence between the old and new policies during updates. Specifically, it maximizes the expected advantage while ensuring that the updated policy does not deviate too far from the previous one. Formally, the TRPO optimization problem is:

$$\begin{aligned} \max_{\theta} \quad & \mathbb{E}_{s \sim \rho^{\pi_{\theta_{\text{old}}}}, a \sim \pi_{\theta_{\text{old}}}} \left[\frac{\pi_{\theta}(a|s)}{\pi_{\theta_{\text{old}}}(a|s)} \hat{A}_t \right] \\ \text{s.t.} \quad & \mathbb{E}_{s \sim \rho^{\pi_{\theta_{\text{old}}}}} [D_{KL}(\pi_{\theta_{\text{old}}}(\cdot|s) || \pi_{\theta}(\cdot|s))] \leq \delta, \end{aligned} \quad (8.10)$$

where $\hat{A}_t$ is an estimator of the advantage function, and δ is a small positive constant defining the trust region size. While effective, TRPO requires solving a complex constrained optimization problem at each iteration, which can be computationally expensive.

Conversely, PPO implements the same *trust region* update by constraining the probability ratio between the new policy, parametrized by θ , and the old policy, parametrized by θ_{old} . Let $r_t(\theta)$ denote this ratio:

$$r_t(\theta) = \frac{\pi_{\theta}(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)}. \quad (8.11)$$

In its original version, the PPO parameters are updated by maximizing a composite objective function that balances policy improvement, value estimation accuracy, and exploration. The total loss function L^{PPO} is defined as:

$$L_t^{\text{PPO}}(\theta) = \hat{\mathbb{E}}_t \left[\underbrace{L_t^{\text{CLIP}}(\theta)}_{\text{Policy Loss}} - c_1 \underbrace{L_t^{\text{VF}}(\theta)}_{\text{Value Loss}} + c_2 \underbrace{S[\pi_\theta](s_t)}_{\text{Entropy Bonus}} \right], \quad (8.12)$$

where L^{VF} is the squared-error loss for the value function, such that:

$$L_t^{\text{VF}}(\theta) = (V_\theta(s_t) - V_t^{\text{target}})^2, \quad (8.13)$$

where V_t^{target} is the target value computed using the observed rewards and bootstrapped estimates. Moreover, $S[\pi_\theta](s_t)$ is an entropy bonus to encourage exploration, and c_1, c_2 are coefficients balancing the contributions of each term.

The most significant part of the Equation (8.12) is the clipped surrogate objective L_t^{CLIP} , which ensures that policy updates do not deviate excessively from the previous policy $\pi_{\theta_{\text{old}}}$. Specifically, L^{CLIP} is defined as:

$$L^{\text{CLIP}}(\theta) = \hat{\mathbb{E}}_t \left[\min \left(r_t(\theta) \hat{A}_t, \quad \text{clip}(r_t(\theta), 1 - \epsilon, 1 + \epsilon) \hat{A}_t \right) \right], \quad (8.14)$$

where $\hat{A}_t$ is the *Generalized Advantage Estimate* (GAE), which measures how much better action a_t is with respect to the average baseline, $\text{clip}(\cdot)$ is the function limiting the probability ratio $r_t(\theta)$ to the interval $[1 - \epsilon, 1 + \epsilon]$, and ϵ is a hyperparameter (typically 0.2) controlling the size of the trust region.

The intuition is as follows: if the advantage $\hat{A}_t$ is positive, we want to increase the probability of the action ($r_t > 1$), but only up to a limit $(1 + \epsilon)$ to prevent over-fitting to a single batch of data. Conversely, if the action was poor ($\hat{A}_t < 0$), we decrease its probability, but bounded by $1 - \epsilon$. This lower-bound constraint ensures monotonic policy improvement and robust convergence.

8.2 RL-based Energy Management

In this section, before presenting the mathematical formulation of the problem as an MDP in Section 8.3, we first describe the specific MG operational setting and the controller's objectives. Notably, while in Section 7.2 we provided a generalized description of the MG ecosystem, including dispatchable generators and TCLs, here we narrow our focus to a stochastic grid-connected architecture. Specifically, we aggregate all renewable sources into a single generation profile and all loads into a single demand profile. This abstraction allows us to isolate the core challenge of the study: optimizing the usage of the BESS under uncertainty, without loss of generality regarding the specific mix of underlying assets.

8.2.1 Microgrid Setting

Energy management within an MG network can be naturally formulated as a sequential decision-making problem with continuous state and action spaces, as the system variables, such as power flows, energy storage levels, and demand, are represented by

continuous signals. Although the operation of an MG is theoretically an *infinite-horizon* process, the *cyclo-stationary* nature of the underlying signals (e.g., daily solar cycles, weekly consumption patterns) allows us to approximate it as an episodic task. This enables the training of policies over representative finite time windows that generalize effectively to continuous operation.

In this dissertation, we consider a standard topology where a single BESS serves as the primary flexibility asset to balance local generation, demand, and grid exchange (Mussi et al., 2024). As depicted in Figure 8.2, the MG is a node connected to the external main grid E . It comprises an overall renewable generation source G , an overall consumption D , and a BESS B . In such an architecture, the RL agent acts as the high-level supervisory controller embedded within the BESS. At each time step t , it observes the system state (e.g., current SOC, net demand) and determines the optimal power allocation to achieve specific objectives, such as minimizing costs or maximizing self-consumption.

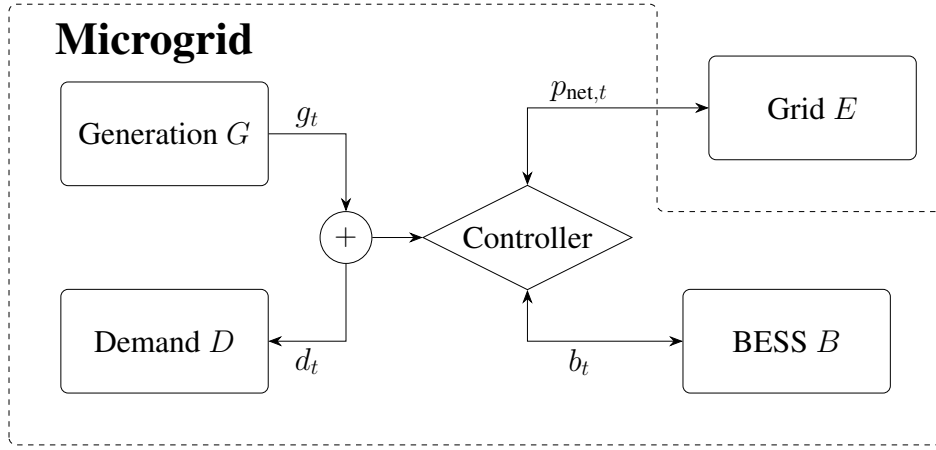


Figure 8.2: Schema of the interactions between core MG elements. Single-edge arrows indicate a unidirectional positive exchange, while double-edge arrows represent bidirectional ones.

From Figure 8.2, we can easily understand the power balance within the system. We define the instantaneous *power surplus*, $p_{\text{surplus},t} \in \mathbb{R}$, as the mismatch between renewable generation and load:

$$p_{\text{surplus},t} = g_t - d_t, \quad (8.15)$$

where $g_t \in \mathbb{R}^+$ is the aggregated power generated by MG’s RESs, and $d_t \in \mathbb{R}^+$ is the total power demand of loads and users.

This surplus dictates the fundamental control logic:

- if $p_{\text{surplus},t} < 0$ (deficit), the controller must decide how much of the shortage to cover with the battery versus buying from the grid;
- if $p_{\text{surplus},t} > 0$ (excess), the controller must decide how much excess energy to store versus selling to the grid.

Ideally, generation and demand would be synchronized. However, in practice, due to the stochastic nature of weather and user behavior, these profiles frequently diverge. Indeed, power generated by RESs strictly depends on weather conditions, season, and time of day. On the other hand, consumption profiles follow users’ behavior within a residential context (Kong et al., 2019), and they do not synchronize well with generation

patterns, often creating the *duck curve phenomenon* where peak generation does not align with peak demand (Kosowatz, 2018).

The whole power balance within the MG is described by the following equation:

$$p_{\text{net},t} = p_{\text{surplus},t} - b_t = (g_t - d_t) - b_t, \quad (8.16)$$

where $p_{\text{net},t} \in \mathbb{R}$ is the net power exchanged with the main grid. Notably, the net power $p_{\text{net},t}$ reflects the one described in Section 7.2.1 as the summation of all dispatchable and non-dispatchable generation. Here, $b_t \in \mathbb{R}$ represents the battery power, which is positive when charging and negative when discharging.¹

It is worth explicitly noting that the presented MG modeling relies on a simplified grid topology, effectively abstracting away physical power line capacity constraints. This formulation essentially addresses the aggregate production and consumption balance over the grid. While this aggregation of productions and loads provides a highly tractable environment for policy optimization, it represents a structural limitation: it inherently masks local network bottlenecks, which in a real-world deployment could lead to line overloads requiring specific mitigation. Consequently, integrating physical grid topology and power flow constraints into the proposed framework represents a relevant direction for future work.

8.2.2 Physical Constraints

At each time t , the controller has to choose the value b_t of the power to be exchanged with the BESS. However, the chosen action should be compliant with the physical constraints of the BESS since it is not possible to overcharge or excessively drain it. Moreover, there exist physical limitations on the amount of power it can exchange over a limited time period.

First, to ensure compliance with these constraints, the power b_t must be bounded with power limits presented in Equation (7.4). In particular, we have:

$$b^{\text{dch}} \leq b_t \leq b^{\text{ch}}, \quad (8.17)$$

where $b^{\text{dch}} \in \mathbb{R}^-$ and $b^{\text{ch}} \in \mathbb{R}^+$ denote the maximum discharging and charging power, respectively.

Second, to prevent overcharging or deep discharging of the BESS, the action b_t must also satisfy limits on the SOE dynamics, to avoid violating the minimum and maximum SOC thresholds as in Equation (7.11). Recalling the definition of SOC from Equation (2.5), at each time t , we can derive its discrete-time evolution in function of the stored energy over time as follows:

$$\sigma_{t+1} = \sigma_t + \frac{b_t \eta_t \Delta t}{e_t^{\text{max}}}, \quad (8.18)$$

where $\sigma_0 \in [0, 1]$ is the value for the SOC at initial time, e_t^{max} is the maximum internal energy at time t , and Δt is the considered interval of time in hours. To ensure that the next state σ_{t+1} satisfies the safety constraints $\sigma_{\min} \leq \sigma_{t+1} \leq \sigma_{\max}$, we impose:

$$\sigma_{\min} \leq \sigma_t + \frac{b_t \eta_t \Delta t}{e_t^{\text{max}}} \leq \sigma_{\max}.$$

¹Notice that b_t corresponds to p_{bat} described in Section 7.2.2.

By solving this inequality for b_t , we derive the dynamic power limits induced by the battery's energy state:

$$\frac{(\sigma_{\min} - \sigma_t)e_t^{\max}}{\eta_t \Delta t} \leq b_t \leq \frac{(\sigma_{\max} - \sigma_t)e_t^{\max}}{\eta_t \Delta t}. \quad (8.19)$$

Consequently, the power applicable to the BESS at any time t must adhere to the intersection of the power limit (Equation (8.17)) and the SOC limits constraints (Equation (8.19)).

8.2.3 Controller's Objective

In the presented setting, the primary goal of the RL agent is to learn the optimal control policy π that maximizes the economic profits of the MG operator over a finite time horizon $\mathcal{T}$. Specifically, the objective function comprises two competing economic drivers: the *market profit*, which entails maximizing revenues from energy arbitrage (selling high, buying low) and self-consumption, and the *degradation cost*, which involves minimizing the costs associated with battery aging. Formally, we define the *total profit* as the objective function $\mathcal{J}(\pi)$ as the sum of the market profit and the degradation cost over the length T of the time horizon as follows:

$$J_{\mathcal{T}}(\pi) = \mathbb{E}_{\tau \sim \pi} \left[\sum_{t=0}^{T-1} (r_{\text{trad},t} + r_{\text{deg},t}) \right], \quad (8.20)$$

where the expectation is taken over the stochastic trajectory τ induced by the policy π and the environment dynamics. Moreover, $r_{\text{trad},t} \in \mathbb{R}$ is the instantaneous profit gained from trading energy with the market, and $r_{\text{deg},t} \in \mathbb{R}_0^-$ is the cost associated with battery degradation.

Market Profit. The immediate economic gain depends on the net power exchanged with the grid, $p_{\text{net},t}$. Recall from Section 8.2.1 that $p_{\text{net},t} > 0$ implies injection (selling) and $p_{\text{net},t} < 0$ implies absorption (buying). Ideally, the MG sells at the feed-in tariff c_t^{sell} and buys at the retail price c_t^{buy} (where typically $c_t^{\text{sell}} < c_t^{\text{buy}}$). The trading component of the profit $r_{\text{trad},t}$ is thus defined as:

$$r_{\text{trad},t} = \begin{cases} c_t^{\text{sell}} \cdot p_{\text{net},t} \cdot \Delta t & \text{if } p_{\text{net},t} \geq 0 \quad (\text{selling}) \\ c_t^{\text{buy}} \cdot p_{\text{net},t} \cdot \Delta t & \text{if } p_{\text{net},t} < 0 \quad (\text{buying}) \end{cases}, \quad (8.21)$$

Note that when buying ($p_{\text{net},t} < 0$), the trading reward is negative, representing a cost incurred by the MG. Using the notation of positive and negative parts, where for a scalar x we define the positive part $x^+ = \max(0, x)$ and the negative part $x^- = \min(0, x)$, we can rewrite Equation (8.21) in a more compact form:

$$r_{\text{trad},t} = (c_t^{\text{sell}} \cdot p_{\text{net},t}^+ + c_t^{\text{buy}} \cdot p_{\text{net},t}^-) \cdot \Delta t. \quad (8.22)$$

Degradation cost. To include the cost of aging, we penalize the loss of SOH. As modeled in Section 2.5, the SOH $\rho_t \in [0, 1]$ decreases monotonically due to stress

cycles and passing of time (*cyclic* and *calendar* aging). The marginal degradation cost is proportional to the fraction of life consumed during the considered interval. Formally:

$$r_{\text{deg},t} = \frac{\rho_t - \rho_{t-1}}{1 - \rho_{\text{EOL}}} c^{\text{bat}}, \quad (8.23)$$

where ρ_t and ρ_{t-1} are the battery SOH values at time t and $t - 1$, respectively, $\rho_{\text{EOL}} \in (0, 1)$ is the SOH of the battery at its EOL (typically in the range 60 – 80% of the SOH), and $c^{\text{bat}} \in \mathbb{R}^+$ is the replacement cost for the BESS. Since the SOH at time t is always strictly lower than the previous step, i.e., $\rho_t < \rho_{t-1}$, the term $r_{\text{deg},t}$ is always negative and acts as a penalization term.

8.3 MDP Formulation

In this section, we formalize the presented problem as an MDP, identifying observation space, action space and reward function for the presented sequential-decision process.

8.3.1 Observation Space

In the presented setting, the observation space includes both endogenous and exogenous variables. Endogenous (or *controllable*) variables generally refer to the internal state of the BESS, including indicators such as SOC and battery temperature, whose evolution can be deterministically determined by the agent's actions. On the other hand, exogenous (or *uncontrollable*) variables encompass both deterministic inputs, such as market price signals known in advance, and stochastic forecasts, such as solar or wind generation, load demand profiles, and weather conditions. Unlike endogenous signals, exogenous signals evolve independently of the agent's action and are generated by the environment external to the agent. Temporal variables (e.g., time of day, day of the week) are also important to capture temporal patterns in consumption and generation behavior.

Hence, we can formalize the observation space s_t as the vector of the following signals:

$$s_t = \left(\sigma_t, \vartheta_t, \widehat{d}_t, \widehat{g}_t, c_t^{\text{buy}}, c_t^{\text{sell}}, t_{\text{day}}, t_{\text{year}} \right), \quad (8.24)$$

where:

- σ_t is the storage unit SOC at time t ;
- ϑ_t is the current battery temperature;
- $\widehat{d}_t$ is the estimate of energy demand d_t at time t ;
- $\widehat{g}_t$ is the estimate of energy generation g_t at time t ;
- c_t^{buy} and c_t^{sell} are the buying price and the selling price of the energy market at time t , respectively;
- t_{day} is the temporal value capturing the intra-day patterns. Specifically, it can be define as $t_{\text{day}} = (\cos(\varphi_t^d), \sin(\varphi_t^d))$, where $\varphi_t^d \in [0, 2\pi]$ is a variable representing the angular position of the clock in a day, given by $\varphi^d = \frac{2\pi\tau_d}{\mathcal{T}_d}$, where $\tau_d \in [0, \mathcal{T}_d]$ is the current time of the day in seconds and $\mathcal{T}_d$ is the total number of seconds in a day;

- t_{year} is the temporal value capturing the inter-day patterns. Specifically, it can be define as $t_{\text{year}} = (\cos(\varphi_t^y), \sin(\varphi_t^y))$, where $\varphi_t^y \in [0, 2\pi]$ is a variable representing the angular position of the current time over the entire year, given by $\varphi^y = \frac{2\pi\tau_y}{\mathcal{T}_y}$, where $\tau_y \in [0, \mathcal{T}_y]$ is the current time in seconds and $\mathcal{T}_y$ is the total number of seconds in a year.

Some remarks are in order. First, the battery SOC σ_t and temperature ϑ_t variables represent the internal state of the BESS component. Their joint evolution determines the degradation of the system, as described in Section 2.5. Second, $\hat{d}_t$, $\hat{g}_t$, c_t^{buy} , and c_t^{sell} enclose the information related to the exogenous factors influencing the system, i.e., energy consumption, generation, and buying/selling price. Ideally, the decision should be driven by the actual demand d_t and generation g_t at time t , which are unknown at the time the agent is making a decision. In our approach, we use the corresponding values at time $t - 1$ since they represent a sufficiently accurate estimate as $\hat{d}_t$ and $\hat{g}_t$.² Conversely, we assume that the energy prices c_t^{buy} and c_t^{sell} are known in advance. While modern European energy markets feature active intra-day trading, day-ahead market clearing prices remain the primary reference for energy management scheduling, making this assumption a standard and robust approximation in the literature. Finally, the observation space is enriched with a specific transformation of the current time variables φ^d and φ^y to take into account the day-night and seasonal periodicity related to the intra-day and seasonal variability of both energy sources and MG demands.

8.3.2 Action Space

The action space defines the agent's decision-making capabilities, typically involving the dispatch strategy of energy within the MG. In a generic MG scenario, an action at time t can be described as a vector of commands such that:

$$\mathbf{a}_t = [a_{i,t}]_1^{|\mathcal{D}|} := [p_{i,t}^{\mathcal{D}}]_1^{|\mathcal{D}|} = \mathbf{p}_t^{\mathcal{D}}, \quad (8.25)$$

where $i \in \mathcal{D}$ represents every dispatchable unit in the network. An element $a_{i,t}$ of the vector represents the amount of power that has to be provided to or withdrawn from the considered dispatchable unit, i.e., $p_{i,t}^{\mathcal{D}}$. Notice that the decision-making process is influenced by the stochasticity of $\mathbf{p}_t^{\mathcal{N}}$, i.e., the power injected or absorbed by the non-dispatchable entities. Thus, the goodness of the chosen action strictly depends on the estimation accuracy of non-dispatchable units' profiles.

In our specific setting, we model the action space as the continuous action variable $a_t \in [0, 1]$, representing the proportion of surplus energy $p_{\text{surplus},t}$ to dispatch (or withdraw) to (from) the single dispatchable unit, i.e., the BESS. Formally, we have:

$$b_t = a_t \cdot p_{\text{surplus},t}, \quad (8.26)$$

$$p_{\text{net},t} = (1 - a_t) \cdot p_{\text{surplus},t}. \quad (8.27)$$

Depending on the sign of $p_{\text{surplus},t}$, the action a_t acquires a different meaning. If $p_{\text{surplus},t} > 0$, and therefore $b_t \geq 0$ and $p_{\text{net},t} \geq 0$, it regulates the proportion between the power used to charge the battery and the one sold to the main grid. Conversely, if $p_{\text{surplus},t} < 0$,

²More complex approaches can be used to get a better approximation of such profiles, e.g., using the techniques presented in Rastkar et al. (2024)

the action a_t regulates the proportion of demanded energy that will be taken from the energy storage and the one bought from the market, thus $b_t \leq 0$ and $P_{\text{net},t} \leq 0$.

We remark that b_t has to adhere to constraints described in Equation (8.17) and Equation (8.19). Hence, the chosen action must be clipped to feasible values so as not to actuate unfeasible or harmful actions. Such a restriction avoids actions that otherwise would be disallowed by the BMS and fosters the agent to learn within a feasible operating region.

8.3.3 Reward Function

The reward function is tightly connected to the controller’s objective presented in Section 8.2.3. In our setting, the reward function mimics the components of *market profit* and *degradation cost*, thus guiding the agent to learn policies that optimize economic performance while considering battery health. Formally, the instantaneous reward at time t is defined as:

$$r_t = r_{\text{trad},t} + r_{\text{deg},t} + \omega r_{\text{clip},t}. \quad (8.28)$$

While $r_{\text{trad},t}$ and $r_{\text{deg},t}$ are exactly the same terms defined in Equation (8.21) and Equation (8.23), respectively, the term $r_{\text{clip},t}$ is a penalty to discourage the agent from selecting actions that violate the physical constraints of the BESS, as detailed in Section 8.2.2. In practice, this component penalizes actions that require clipping due to constraint violations, thereby encouraging the agent to learn feasible policies. The weight $\omega \in \mathbb{R}^+$ regulates the influence of this penalty on the learning process. We formally define $r_{\text{clip},t} \in \mathbb{R}_0^-$ as:

$$r_{\text{clip},t} = -\max \left\{ 0, b_t + \frac{(\sigma_t - \sigma_{\max})}{\Delta t} v_t C_t^{\max}, \frac{(\sigma_{\min} - \sigma_t)}{\Delta t} v_t C_t^{\max} - b_t \right\}. \quad (8.29)$$

In particular, this penalty quantifies the extent to which the action a_t would lead to a violation of the SOC constraints defined in Equation (8.19), which, in our setting, are stricter than the power limits in Equation (8.17). Here, the energy limits are expressed in terms of power: we recall that $C_t^{\max}$ and v_t are the maximum internal capacity and voltage of the battery at time t , respectively, and Δt is the time interval between two consecutive decision steps.

8.4 Training Procedure

In this section, we describe the learning procedure adopted to train the RL agent leveraging ERNESTO as the simulation environment. The training process is based on the iterative interaction between the agent and the virtual MG to collect trajectories of experience, which are subsequently used to optimize the control strategy. As an RL agent, we employ PPO (Schulman et al., 2017b), a state-of-the-art method adopted in various real-world applications due to its robustness and sample efficiency (Berner et al., 2019; Peng et al., 2020). With respect to other RL methodologies like DQN, PPO is capable to handle continuous action spaces, which is essential in our setting where the action variable a_t is continuous. Moreover, PPO’s clipped surrogate objective ensures stable policy updates, which is crucial when learning in complex environments with high-dimensional state spaces and stochastic dynamics. The full pseudocode of the training procedure is provided in Algorithm 3.

Chapter 8. A Reinforcement Learning Approach

Algorithm 3 PPO-based Microgrid Control

Input: Data profiles: Demand set $\{\mathbf{d}^{(i)}\}_{i=1}^M$, generation $\mathbf{g}$, prices $\mathbf{c}_{\text{buy}}, \mathbf{c}_{\text{sell}}$, ambient temperature ϑ_a
Input: Models: ERNESTO simulator $\text{DT}(\cdot)$, Actor π_θ , Critic V_ϕ
Input: Hyperparams: Episodes n_{ep} , Horizon T , Learning rates $\alpha_\theta, \alpha_\phi$, GAE λ , update epochs k_{epochs}

- 1: **for** episode $1, \dots, n_{ep}$ **do**
- 2: Sample demand profile $\mathbf{d}^{(i)} \sim \text{Unif}(\{\mathbf{d}^{(i)}\}_{i=1}^M)$
- 3: Initialize state s_1 (with $\sigma_1, \vartheta_1, \rho_1$)
- 4: Initialize forecast/buffer $\mathcal{B} \leftarrow \emptyset$
- 5: **for** $t = 1, \dots, T$ **do**
- 6: Compute forecasts $\hat{d}_t, \hat{g}_t$ and construct state s_t
- 7: Sample action $a_t \sim \pi_\theta(\cdot | s_t)$
- 8: Compute BESS power $b_t \leftarrow a_t(g_t - d_t^{(i)})$
- 9: Evolve BESS through ERNESTO: $(\sigma_{t+1}, \vartheta_{t+1}, \rho_{t+1}) \leftarrow \text{DT}(\sigma_t, \vartheta_t, K_t, b_t)$
- 10: Compute net power $p_{\text{net},t} \leftarrow (1 - a_t)(g_t - d_t^{(i)})$
- 11: Compute Reward r_t (Eq. 8.28)
- 12: Store transition (s_t, a_t, r_t, s_{t+1}) in $\mathcal{B}$
- 13: Update state $s_t \leftarrow s_{t+1}$
- 14: **end for**
- 15: Compute Advantages $\hat{A}_t$ using GAE(λ) for all $t \in \mathcal{B}$
- 16: **for** epoch $1, \dots, k_{\text{epochs}}$ **do**
- 17: Update θ by maximizing $L^{\text{CLIP}}(\theta)$ (Eq. 8.14)
- 18: Update ϕ by minimizing $L^{\text{VF}}(\phi)$ (Eq. 8.13)
- 19: **end for**
- 20: **end for**

8.4.1 Exogenous and Endogenous Inputs

The training procedure requires a set of exogenous inputs to simulate the MG environment. Specifically, for the finite time horizon $\{1, \dots, T\}$, we need:

- A set of M demand profiles $\{\mathbf{d}^{(i)}\}_{i=1}^M$, where each profile $\mathbf{d}^{(i)} := (d_1^{(i)}, \dots, d_T^{(i)})$ represents the energy consumption over the time horizon;
- A generation profile $\mathbf{g} := (g_1, \dots, g_T)$, representing the aggregated energy produced by the RESs within the MG;
- Market prices sequences $\mathbf{c}^{\text{buy}} := (c_1^{\text{buy}}, \dots, c_T^{\text{buy}})$ and $\mathbf{c}^{\text{sell}} := (c_1^{\text{sell}}, \dots, c_T^{\text{sell}})$, representing the buying and selling prices of energy at each time step;
- An ambient temperature profile $\vartheta_a := (\vartheta_{a,1}, \dots, \vartheta_{a,T})$, which influences the thermal dynamics of the BESS.

These inputs can be sourced from historical data, synthetic generation models, or a combination of both, depending on the specific application and availability of data. In addition to exogenous inputs, the training procedure also requires the initial internal state of the BESS, including the initial SOC, temperature, and SOH. These endogenous variables are configured directly from ERNESTO's interface before starting the experiment and are updated at each time step according to the battery configuration chosen by the user. Further details on datasets and battery configurations are provided in Section 9.1.

8.4.2 Algorithm and Interaction Loop

Differently from the original implementation of PPO (Schulman et al., 2017b), state-of-the-art implementations (Huang et al., 2022) utilize separate networks for the policy and value function approximations, splitting the *actor* $\pi_\theta(a|s)$, with parameters θ , which outputs the distribution over actions given the state, and the *critic* $V_\phi(s)$, with parameters ϕ , which estimates the expected discounted return from state s .

At the beginning of each episode, the environment is reset by sampling a random demand profile $d^{(i)}$ and initializing the BESS state (Lines 2–3). The agent then interacts with the environment for the full horizon T , collecting a trajectory $\tau = \{s_t, a_t, r_t, s_{t+1}\}_{t=1}^T$ (Lines 6–13). In particular, the procedure involves computing forecasts for demand and generation to construct the current state s_t , sampling an action a_t from the policy using the estimated demand $\hat{d}_t^{(i)}$ and generation g_t , computing the BESS power b_t based on the action and the actual surplus $p_{\text{surplus},t}$. Then, the BESS state is evolved using the ERNESTO simulator. Hence, after computing the grid power $p_{\text{net},t}$, the reward r_t is calculated according to Equation (8.28), and the transition is stored in a buffer $\mathcal{B}$ for later use. Unlike other methods that update parameters at every step, PPO collects a batch of experience (a full episode or multiple episodes) and then, after the estimation of the advantage function $\hat{A}_t$, performs a mini-batch update via Stochastic Gradient Descent, specifically using the Adam optimizer (Kingma and Ba, 2017), to maximize the clipped objective and minimize the value loss, respectively (Lines 15–19).

CHAPTER 9

Experimental Evaluation

In Chapter 8 we have formalized the control problem of MG energy management as an MDP and proposed a model-free RL-based approach to solve it. In this chapter, we want to experimentally validate the proposed approach in a realistic scenario, adopting real-world historical data. We compare our solution with common rule-based strategies and state-of-the-art RL-based controllers taken from the literature. We also conduct ablation studies to analyze how our approach behaves when varying some of the problem’s key parameters.

Chapter Outline. The chapter is organized as follows. In Section 9.1, we describe the experimental setup, including the datasets and BESS models used. In Section 9.2, we present and discuss the results of our experiments, highlighting the performance of the proposed approach compared to baseline methods. Finally, in Section 9.3, we conduct ablation studies.

9.1 Experimental Setup

In this section, we report details about employed datasets, the BESS model, and the baseline strategies used for comparison.

9.1.1 Datasets and Simulation Environment

We validate our approach using real-world data in an Italian residential MG scenario. The dataset includes PV generation profiles, ambient temperature data, energy market prices, and household consumption profiles comprising years from 2015 to 2019, using the first 4 years for training and the last one for testing. Specifically:

- **PV Generation Profiles:** We use data from a 3 kW PV system, providing hourly generation values averaged across Italy (Pfenninger and Staffell, 2016).
- **Demand Profiles:** We consider 398 realistic household consumption profiles from various Italian regions (Fioriti et al., 2022), split into 370 for training and 28 for testing.
- **Energy Market Prices:** Hourly energy prices are obtained from the Italian Gestore Mercati Energetici (GME) (Gestore dei Mercati Energetici (GME), 2015-2020), accounting for taxes and management costs.
- **Ambient Temperature Data:** Hourly ambient temperature data averaged across Italy are sourced from (Staffell et al., 2023).

The simulation environment is implemented using the Gymnasium framework (Towers et al., 2024), allowing for easy integration with an RL algorithm. Gymnasium acts as a wrapper around the ERNESTO, which is used to accurately simulate BESS dynamics, by computing the evolution of SOC, temperature, and SOH over time. The BESS modeled in the experiments is a battery pack built by connecting multiple lithium-ion NMC cells, specifically 3 parallel strings of 96 cells in series, resulting in a total of 288 cells. Further details about the cell characteristics are reported in Table C.1.

For these experiments, we configure ERNESTO in the following way. The electrical evolution is modeled using a 1-RC Thevenin ECM (Figure 2.2), the thermal dynamics are simulated using the thermal model presented in Section 2.3, while the aging model is based on the empirical formulation proposed by Xu et al. (2018), which accounts for both calendar and cyclic aging effects, presented in Section 2.5.

9.1.2 Baseline Strategies

In the following experiments, we compare our PPO-based RL solution, from now on denoted as RL^* , with several baseline strategies, including both deterministic rule-based and RL-based approaches taken from the literature. The idea is to compare our method with both trivial but reasonable strategies and with more complex approaches that have been proposed in the literature to solve similar problems.

Rule-based Strategies:

- **X-Ys Policies:** These strategies allocate $X\%$ of the available surplus power to the BESS and the remaining $Y\%$ to the grid. We test 20-80, 50-50, and 80-20 configurations.
- **OnlyGrid (OG):** This strategy corresponds to the 0-100 configuration, where the MG solely interacts with the main grid without utilizing the BESS.
- **BatteryFirst (BF):** This strategy corresponds to the 100-0 configuration, where the MG prioritizes BESS usage before interacting with the grid.

RL-based Strategies:

- **RL-base:** This strategy replicates the approach by Mussi et al. (2024), training the agent with fixed ambient temperature and market prices. Different from their

approach employing FQI and discrete actions, here we keep their assumptions with PPO and continuous actions.

- *RL-base+*: This strategy extends *RL-base* by incorporating evolving market dynamics while maintaining a fixed ambient temperature, to capture the benefits of market information in the learning process.

Additional details about hyperparameters can be found in Table C.2.

9.2 Case Study

In this section, we present the results of our experimental evaluation in the realistic MG scenario described in Section 9.1. We assess the performance of the proposed *RL** approach against the baseline strategies introduced earlier. The evaluation focuses on the empirical reward achieved over the testing period, considering both trading and degradation costs.

We compare each control policy π in terms of average (over the n_{test} validation profiles) empirical reward over the time horizon T . Specifically, we define the empirical reward $R_t(\pi)$ at time t as:

$$R_t(\pi) = \frac{1}{n_{\text{test}}} \sum_{i=1}^{n_{\text{test}}} \sum_{h=1}^t \left[r_{\text{trad},h}^{(i)} + r_{\text{deg},h}^{(i)} \right], \quad (9.1)$$

where $r_{\text{trad},h}^{(i)}$ and $r_{\text{deg},h}^{(i)}$ are the market profit and degradation cost corresponding to the i -th demand profile $\mathbf{d}^{(i)}$ at time h . We also compared the separate contribution of trading and degradation to the final reward as follows:

$$R_{\square,t}(\pi) = \frac{1}{n_{\text{test}}} \sum_{i=1}^{n_{\text{test}}} \sum_{h=1}^t r_{\square,h}^{(i)} \quad (9.2)$$

with $\square \in \{\text{trad}, \text{deg}\}$. Furthermore, we evaluate the gap of a specific policy π with respect to a given baseline π^B as $\Delta R_{\square}(\pi, \pi^B)$ defined as:

$$\Delta R_{\square}(\pi, \pi^B) := R_{\square}(\pi) - R_{\square}(\pi^B). \quad (9.3)$$

Positive values for $\Delta R_{\square}(\pi, \pi^B)$ corresponds to larger performance of π than π^B .

9.2.1 Real-World Scenario Results

The results of our experimental evaluation are summarized in Figure 9.1, which illustrates the cumulative average empirical reward over the testing period for each control strategy. It is worth noting that, on average, none of the strategies achieves a positive total return. This was expected since the cost of buying energy is always larger than the selling price, and the generation profile produces energy in excess of the demand only 39.6% of the time over the year of testing. Moreover, the reward also includes the degradation component, which is strictly negative since it is related to BESS usage and passing of time.

Among the analysed approaches, our proposed *RL** solution provides the best average results over the time horizon. Within the central months of the year (approximately

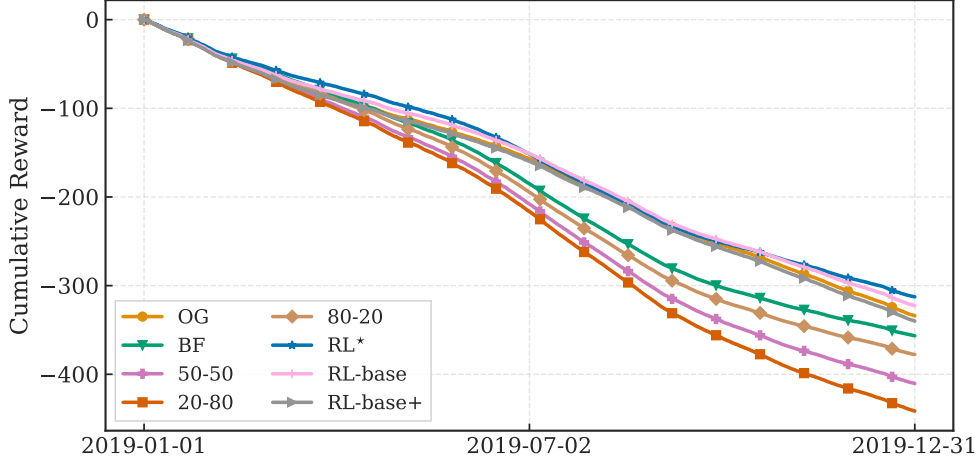


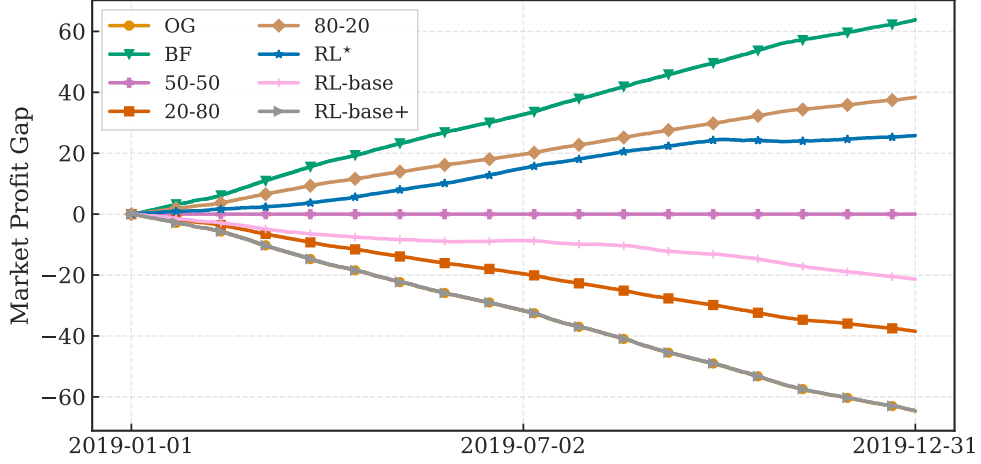
Figure 9.1: Cumulative average empirical reward: average empirical reward $R_t(\pi)$ over the 28 test profiles for each considered strategy π as a function of time t . Confidence intervals are omitted for the sake of clarity.

June to September), the performances have been on par or slightly worse than those of previous RL-based solutions and OG. However, at the end of the time horizon, the RL^* method reduces the cost by 3.2% with respect to $RL-base$ and by 6.7% with respect to OG , i.e., the best among the rule-based strategies. The strong OG performance, in this scenario, is primarily due to the relatively high cost of BESS replacement, which in turn brings high degradation costs for those methods that rely more heavily on BESS utilization.

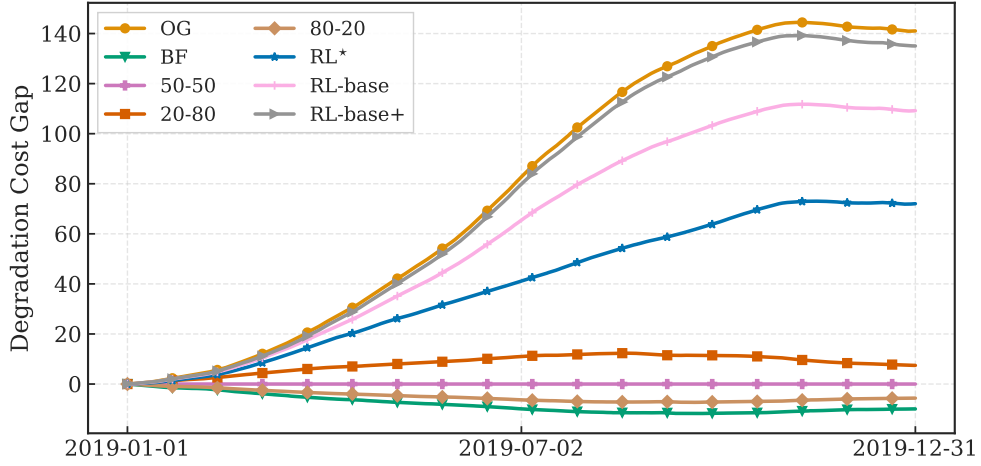
By inspecting Figure 9.2a and Figure 9.2b, we can infer that similar rewards can be achieved with completely different strategies. Let us compare the OG strategy, namely the best among the rule-based ones, with ours. The degradation gap $\Delta R_{deg,t}(OG, 50-50)$ for OG with respect to $50-50$ is the largest one, meaning that it is the method using the BESS the least. However, the corresponding trading gap $\Delta R_{trad,t}(OG, 50-50)$ is negatively compensating the previous advantage. Conversely, the RL^* approach balances the two gaps, highlighting a substantial usage of the BESS.

This can also be observed in Figure 9.3, which presents a heatmap showing how RL^* plays different actions in response to varying demand levels throughout the test year. Each tile represents the logarithm of the number of times an action is chosen.¹ This figure highlights that as the demand rises, the action progressively increases the amount of power the algorithm is addressing to the BESS. Using such a strategy, RL^* obtains a larger overall reward than OG . These results suggest that trivial rule-based approaches might not be flexible enough for this setting since the best performances are achieved using a strategy that can adapt to external environmental changes.

¹For visualization reasons, we report the logarithm of the number of occurrences of the actions.



(a) Market profit gap with respect to 50-50.



(b) Degradation cost gap with respect to 50-50.

Figure 9.2: Gaps for trading and degradation components of the reward: (a) trading gap $\Delta R_{\text{trad},t}(\pi, 50-50)$ and (b) degradation gap $\Delta R_{\text{deg},t}(\pi, 50-50)$ for each considered strategy π with respect to the 50-50 baseline.

Now, let us focus on the comparison of the three RL-based approaches. Even if their empirical rewards are similar, by inspecting their gaps in Figure 9.2a and Figure 9.2b, we infer that *RL-base* and *RL-base+* tend to trade more with the market than *RL** (smaller trade gap), and they make less intensive usage of the BESS (larger degradation gap). This phenomenon is more evident for *RL-base+*, whose policy is almost equal to the OG. This might be because it employs approximate information about the degradation, overestimating the associated cost. This highlights how the joint use of information about the energy market and the ambient temperature is crucial to allow *RL** to learn effective strategies for the MG.

In Figure 9.4, we provide boxplots representing the empirical distribution of the reward $R_{\square,t}(\pi)$ at the end of the time horizon T of the considered strategies over the different demand profiles. The figure shows that returns significantly fluctuate across testing profiles. This is because the yearly demand for different profiles spans from 1.50 MWh to 5.05 MWh over the test set. However, we can see that the median of

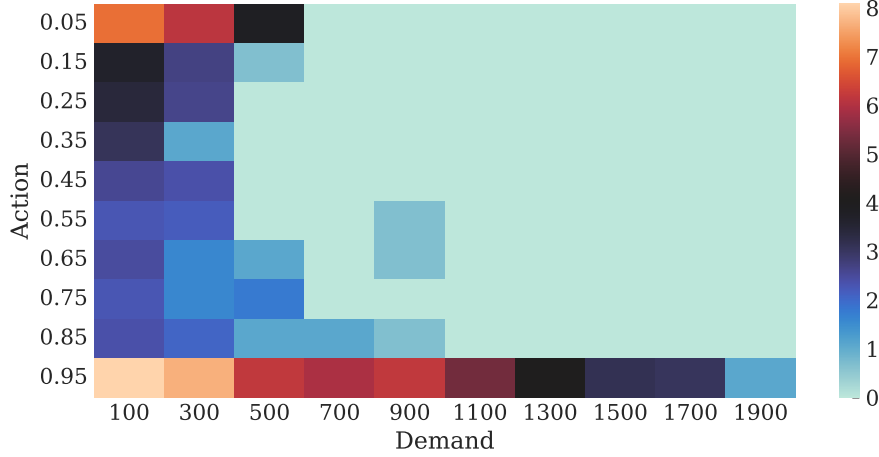


Figure 9.3: Heatmap of action vs. demand: each tile represents the logarithm of the number of times an action is chosen by RL^* for a given demand level over the test year. As the demand increases, the action tends to allocate more power to the BESS.

the reward for RL^* is close to the 75% percentiles of most rule-based algorithms and has lower variability with respect to the OG one. Even when compared with the other RL -based approaches, RL^* provides the largest median value for the reward and the smallest spread (having a smaller box and shorter whiskers). This corroborates the idea that the proposed approach can provide more consistent results over the considered profiles.

Finally, a paired t-test to check if the difference between the rewards of RL^* and those of the other methods is significantly greater than zero provided us with a p-value (overall) of 0.0052, proving that there is strong statistical significance that our method is consistently performing better than the others.

9.2.2 Sensitivity Analysis: Adaptability to Cost Shifts

We evaluated the robustness of our approach across various scenarios with modified energy prices and battery costs. In particular, we performed experiments in which we considered a multiplicative term $\kappa > 0$ for both the selling and buying prices, i.e., we used κc^{sell} and κc^{buy} in place of c^{sell} and c^{buy} , respectively (during the training and testing of our algorithms). This modification mimics the increase ($\kappa > 1$) or decrease ($\kappa < 1$) of the energy price caused by geopolitical events or natural catastrophes, e.g., the Covid-19 (Krarti and Aldubyan, 2021).

Figure 9.5 show the results of the reward $R_T(\pi)$ with different values of the parameter $\kappa \in \{0.1, 0.5, 1, 1.5, 2\}$. When the energy cost is small, the most profitable strategy is the OG , since the financial loss from market transactions diminishes. Conversely, as κ increases, the optimal strategy shifts toward BF , since purchasing energy becomes prohibitively expensive, making the battery degradation a more cost-effective choice. Moving to RL strategies, RL^* outperforms RL -base for $\kappa > 0.5$ and is on par with all the other approaches for all the values of κ . Figure 9.6 presents a similar analysis, examining the impact of varying the replacement cost $c^{\text{bat}} \in \{200, 1000, 3000, 5000, 10000\}$ €. The results confirm the same insights regarding RL^* behavior as those observed in the

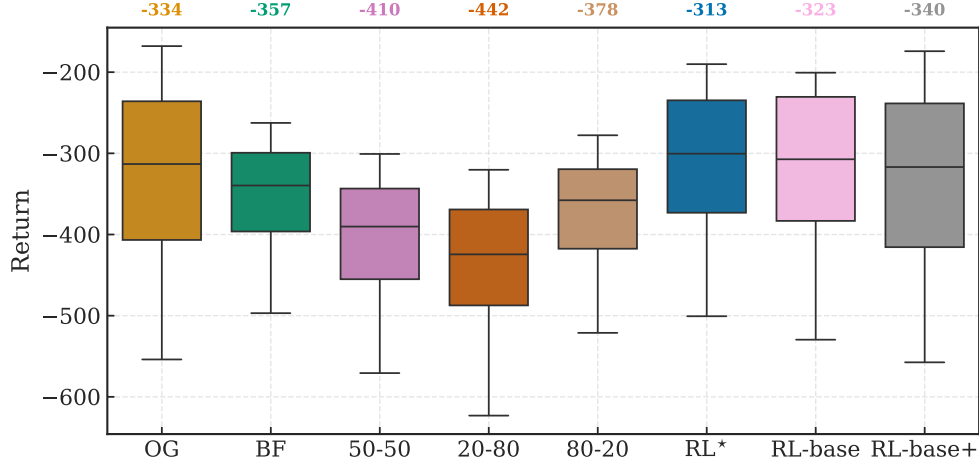


Figure 9.4: Empirical return distribution across validation profiles: boxplots representing the empirical distribution of the reward $R_{\square, T}(\pi)$ at the end of time horizon T of the considered strategies over the different 28 demand profiles. Above each boxplot, the average return is reported.

experiment on κ variation. These results, regarding both market and replacement cost alterations, corroborate that our approach can be adapted to different environmental conditions.

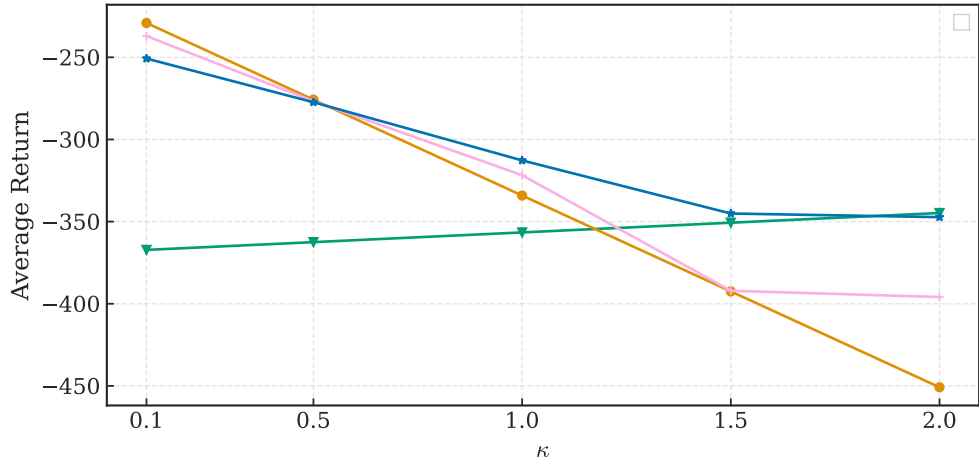


Figure 9.5: Sensitivity analysis on energy price: the plot shows the return of the most relevant methods at the end of the time horizon for different values of κ .

9.3 Ablation Studies

We study the influence of the clipping term on the learning of the optimal policy. Specifically, $r_{clip,t}$, serves as both a penalty for unsafe actions that violate structural constraints and a mechanism that discourages excessive battery usage in favor of direct interactions with the market. We compare the reward at the end of the time horizon $\mathbb{R}_T(\pi)$ for different values of the parameter, specifically $\omega \in \{0, 0.01, 0.05, 0.1, 0.25, 0.5, 1, 2\}$.

The optimal value chosen for RL^* is $\omega = 0.1$. Figure 9.7 shows that, on average,

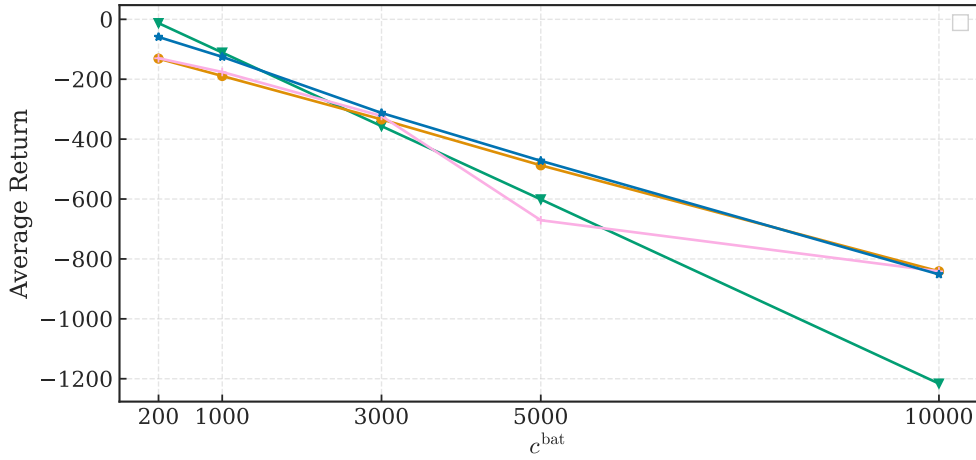


Figure 9.6: Sensitivity analysis on replacement cost c^{bat} : the plot shows the return of most relevant methods at the end of the time horizon for different values of c^{bat} .

the configuration with $\omega = 0.1$ (the one used for the main experiment) provides the best results. However, even using values far from the optimal one, the average reward is still larger than that of most of the rule-based strategies and is below 6.7% to the *OG* one. This shows how, to achieve optimal performance, you must carefully tune the ω parameter, but misspecification of such a parameter is not too crucial to obtaining significant results.

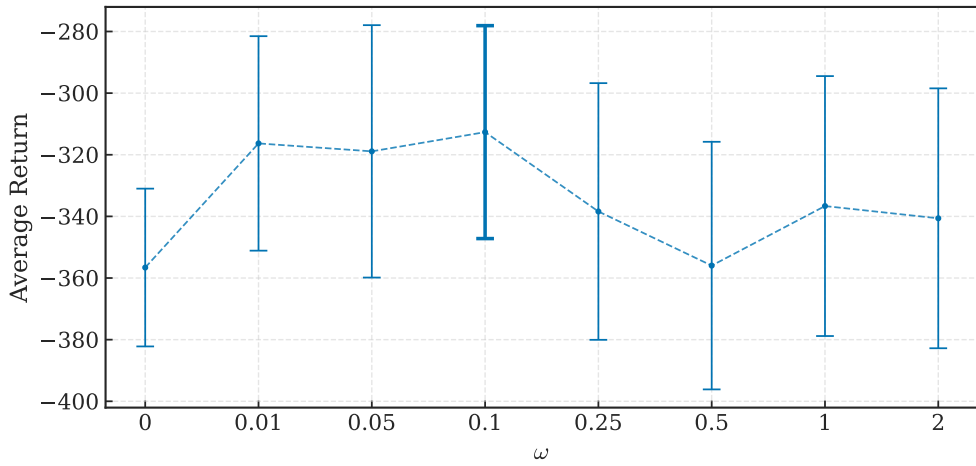


Figure 9.7: Ablation study on the clipping term: average empirical return for different values of ω . Error bars represent the standard deviation over the 28 test profiles.

Part IV

**Multi-Agent Reinforcement Learning
for Renewable Energy Communities**

CHAPTER 10

Energy Management in Renewable Energy Communities

In this dissertation, we have decided to follow a progressive *micro-to-macro* trajectory: beginning with the granular modeling of individual BESS assets and their internal components in Part I, we proceeded by integrating monitoring and adaptation layers, and finally expanding to the single-agent control of MGs in Part III. In this final part, we further broaden the perspective to encompass the system-level orchestration of *Renewable Energy Communities* (RECs).

Energy communities are a recent concept, born in 2018 when the European Parliament and Council approved the directive introducing the legal framework of RECs on the EU level ([European Parliament and European Council, 2018](#)). The directive aims to increase decentralized renewable energy production, in line with the EU's climate goals of reducing greenhouse gas emissions and increasing the share of renewables in the energy mix. By promoting local generation and consumption of energy, RECs help reduce transmission losses and ease the operational complexity of transmitting energy across large distances to balance supply and demand, challenges that can increase both physical strain on the grid and overall system costs.

In this chapter, we set the background for the study of energy management within RECs, aiming to fulfill the research objective *RO4* reported in Section 1.4.1, concerning the use of the DT to develop coordination strategies that benefit complex multi-MG systems. In particular, we provide a formalization of the problem, defining the system model and objectives at both the node and community levels. Then, we review existing approaches to energy management in RECs, highlighting their limitations and the gaps that our proposed method aims to address.

Chapter Outline. This chapter is organized as follows. In Section 10.1, we formalize the

energy management problem within an REC, defining the system model and objectives at both the node and community levels. Then, in Section 10.2, we review existing approaches to energy management in RECs, highlighting their limitations and the gaps that our proposed method aims to address.

10.1 Problem Formulation

A REC is configured as a network of prosumers who jointly manage energy production, consumption, and storage to enhance exchange efficiency, improve local grid stability, and reduce individual energy costs. Notably, while the general concept of REC is widely accepted across countries, not all nations have established specific regulatory frameworks to govern its operation, leading to variations in definitions and incentive structures. In this dissertation, we consider an REC operating under the Italian regulatory framework, as defined by national decrees ([Italian Government, 2019](#); [Italian Parliament, 2020](#); [Italian Government, 2020](#)), which govern, in particular, the distribution of financial incentives for shared energy usage. These incentives are proportional to the *self-consumption*, defined as «*the minimum, in each hourly interval, between the amount of energy produced and injected in the network by the renewable power generators and the energy withdrawn by all associated final customers*».

Apart from these incentives for REC members, everything else stays the same as in the single MG setting in Section 7.2, where the REC is not in place: every household continues to pay the bill to its energy provider, and when injecting energy in the network, it will receive the corresponding revenue. The prices for buying and selling energy are not tied to the REC and may differ from household to household. Hereafter, we provide the formalization of the system in two layers: at the *node level* (individual MG prosumers) and the *community level* (the aggregator). For each, we provide a formal description and highlight the associated objectives.

10.1.1 Node Level Formalization

In this section, we describe the structure and objectives of individual nodes within the REC. For the single node, everything remains the same as in the single MG setting described in Section 7.2, with the addition of the incentive term in the objective function, which is determined at the community level.

Node Structure. Let $\mathcal{N}$ denote the finite set of distinct entities participating in the REC. We denote a generic node by $n \in \mathcal{N}$. The cardinality of the set is given by $N := |\mathcal{N}|$. A node of the REC typically represents a residential MG, e.g., a household, with local energy loads, and potentially equipped with a PV implant and a storage system. Prosumers equipped with a BESS belong to the partition $\mathcal{N}_A$ of controllable or *active* nodes, as they are capable of participating in energy optimization decisions. Conversely, $\mathcal{N}_P$ is the set of prosumers without storage capabilities, defined as *passive*, contributing to the community primarily through consumption and local generation. Thus, $\mathcal{N} := \mathcal{N}_A \cup \mathcal{N}_P$.

Figure 10.1 illustrates the physical configuration of an REC, where all nodes are connected to the national grid in the same way as non-REC members. REC membership is defined contractually, not physically, and does not alter the standard conditions

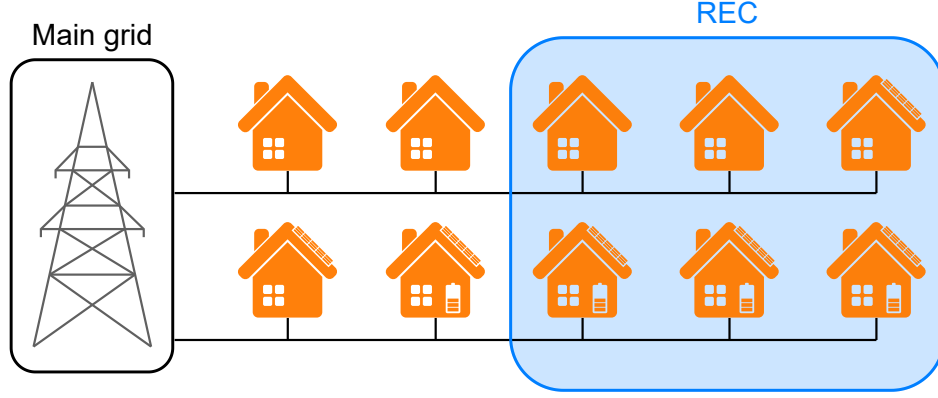


Figure 10.1: Example of an REC. All participating households are physically connected to the national grid like any other user. Members may be equipped with renewable energy sources (such as solar panels, like in the second row of houses) and/or a BESS (last four houses). In the picture, 3 out of 6 REC members are active nodes (on the second row) since they are equipped with a storage system, while the others (on the first row) are passive nodes.

under which each household interacts with the external grid. As such, each household exchanges energy with the grid following the usual individual pricing and contractual mechanisms.

At each time step t , each node n follows the same dynamics defined in Section 7.2 for single MGs. In particular, the net power $p_{n,t}$ exchanged with the grid by node $n \in \mathcal{N}$ is described by Equation (7.1) as the sum of all dispatchable and non-dispatchable power components. However, considering the simplified model adopted in Section 8.2, with aggregated generation and demand signals, and a single BESS unit, the power balance at node n can be expressed as:

$$p_{n,t} = g_{n,t} - d_{n,t} - b_{n,t}, \quad (10.1)$$

where $g_{n,t} \in \mathbb{R}$ and $d_{n,t} \in \mathbb{R}$ are the exogenous power signals of the energy renewable generation and demand, respectively, and $b_{n,t} \in \mathbb{R}$ is the energy charged ($b_{n,t} > 0$) into or discharged ($b_{n,t} < 0$) from the battery ($b_{n,t} = 0$ for *passive* nodes). A positive value of $p_{n,t}$ indicates that the node is injecting power into the grid (i.e., selling energy), while a negative value implies that it is withdrawing energy from the grid (i.e., purchasing energy).

An *active* node $n \in \mathcal{N}_A$ must choose a control strategy which represents the load to apply to the battery at each time step t . Even for RECs, this process is subject to the same constraints defined in Section 7.2, including power limits (Equation (8.17)) and SOC bounds (Equation (8.19)).

Node Objective. The primary objective of each household is to minimize its electricity-related expenses, including energy bills and the operational costs associated with PV systems and battery storage. Participation in an REC does not alter this fundamental goal. However, a household may willingly deviate from its typical cost-minimizing strategy to adopt a more cooperative behavior that benefits the community if the financial incentives provided by the REC are sufficiently attractive. Formally, given a period of time T , each active node n aims to optimize the BESS management to maximize its

cumulative economic utility J_n , defined as:

$$J_n = \mathcal{J}_n^{\text{MG}} + \sum_{t=1}^T r_{n,t}^{\text{inc}}, \quad n \in \mathcal{N}_A, \quad (10.2)$$

where, the term $\mathcal{J}_n^{\text{MG}}$ represents the objective of the single MG defined in Section 7.2, which combines exchanges with the grid from Equation (7.21) and battery degradation costs from Equation (7.13), and the additional component $r_{n,t}^{\text{inc}} \in \mathbb{R}_0^+$ captures the incentive achieved for contributing to the community's overall self-consumption. The latter term is a portion of the total incentive received by the REC, which is determined by the central coordinator.

Furthermore, to evaluate the overall fulfillment of individual household goals within the REC, we employ the *social welfare* $\mathcal{W} \in \mathbb{R}$. This metric, originally defined in the game theory field (Nash, 1950), is formalized as follows:

$$\mathcal{W} := \sum_{n \in \mathcal{N}_A} J_n. \quad (10.3)$$

In practice, $\mathcal{W}$ serves as an aggregate indicator of REC participants' overall satisfaction. In this dissertation, we use it to assess the benefits of participation relative to non-participation, thereby quantifying the net advantage of cooperative energy management from the perspective of individual households.

10.1.2 Community Level Formalization

In this section, we describe the structure and objectives of the REC controller, which operates at the community level. The REC controller is responsible for managing the collective incentives and coordinating the behavior of individual nodes to maximize the community's utility.

Community Structure. At the community level, the REC receives financial incentives from the government based on its collective self-consumption. According to the decree defined by Italian Government (2019), the *self-consumption* is the minimum, in each hourly interval, between the total injected energy and the energy withdrawn by all REC participants. Formally, we define the power injection from node $n \in \mathcal{N}$ as $p_{n,t}^+ := \max(p_{n,t}, 0)$ and the power withdrawn from the grid as $p_{n,t}^- := |\min(p_{n,t}, 0)|$. The self-consumed energy at the community level at time t is thus given by:

$$U_t := \min \left\{ \sum_{n \in \mathcal{N}} p_{n,t}^+, \sum_{n \in \mathcal{N}} p_{n,t}^- \right\}. \quad (10.4)$$

This value describes the coordination between production and consumption among participants and determines the financial incentives received by the REC at time instant t . Following Italian regulatory decrees, we model the financial incentives V_t received by the REC at time t as:

$$V_t := U_t (\alpha + \beta + \varphi_t), \quad (10.5)$$

where $\alpha \in \mathbb{R}_0^+$ is a cost refund term, addressing the spared management and repair costs of the grid authority, $\beta \in \mathbb{R}_0^+$ is a fixed coefficient of the actual incentives, and $\varphi_t \in \mathbb{R}_0^+$

is a time-dependent component based on market energy prices.¹ Further details about the incentive coefficients are provided in Appendix D.1.

Community Objective. The fundamental challenge of the REC controller is to indirectly maximize U_t . Since the controller cannot force nodes to charge or discharge their batteries, it must use the incentive revenue V_t to steer them. At each time step t , the REC controller determines an allocation vector, defined as:

$$\boldsymbol{\omega}_t := [\omega_{n,t}]_{n \in \mathcal{N}},$$

where $\omega_{n,t} \in [0, 1]$ represents the share of the total incentive allocated to node n , subject to $\sum_{n \in \mathcal{N}} \omega_{n,t} \leq 1$. Consequently, the individual incentive for each node n is:

$$r_{n,t}^{\text{inc}} = \omega_{n,t} V_t. \quad (10.6)$$

Hence, the goal of the community layer is to design an incentive allocation strategy for $\boldsymbol{\omega}_t$ that effectively shapes the individual agents' utility functions. By dynamically modulating the component $r_{n,t}^{\text{inc}}$ within the individual objective J_n , the REC controller aligns the prosumers' self-interested optimization with the maximization of the global self-consumption U_t .

10.2 Existing Approaches

The literature on energy management for RECs has evolved significantly, shifting focus from single-user optimization to complex, multi-agent scenarios, with increasing emphasis on autonomy, scalability, and coordination. Existing approaches can be broadly categorized into rule-based strategies, optimization techniques, and learning-based methods. In this section, we review these contributions to contextualize our proposed solution, highlighting the gaps regarding incentive alignment and decentralized coordination.

10.2.1 Rule-Based and Optimization Approaches

Early research primarily focused on rule-based heuristics and deterministic optimization. [Pasqui et al. \(2023\)](#) compared individual versus collective self-sufficiency strategies, demonstrating that community-focused heuristics can reduce grid reliance without compromising local performance. Similarly, [Mustika et al. \(2022\)](#) explored two rule-based and two optimization-based methods, each either household- or community-oriented, highlighting the benefits of community organization over isolated household control. Specific to decentralized architectures, [Chreim et al. \(2023\)](#) designed a multi-agent architecture for shared BESS management in residential energy communities, where agents operate on predefined rules and limited communication, aiming to reduce energy costs, minimize transmission losses, and preserve user privacy. The system is designed to operate without a centralized optimization or learning entity, emphasizing interpretability and compatibility with Internet-of-Energy infrastructures.

¹The term α is due to the fact that, thanks to the RECs, the grid authority does not need to transport energy over long distances within the national grid, since RECs foster localized self-consumption, allowing for the reduction of transportation and management costs.

While rule-based strategies offer interpretability and low computational cost, they lack the adaptability to handle the stochastic nature of RESs. Conversely, traditional optimization approaches (e.g., MILP) provide theoretical optimality but often struggle with scalability and require precise forecasts and system models, and high computational effort, which are not always available in real-time scenarios.

10.2.2 Centralized Reinforcement Learning

To overcome the rigidity of heuristics and the model-dependency of optimization, RL has been widely adopted. In the work by [Aittahar et al. \(2024\)](#), a centralized controller is devoted to optimizing the energy management to minimize total electricity bills for community members, which can exchange energy internally within the REC or externally with the grid, incurring different fees. The paper compares MPC with a centralized RL agent trained via PPO ([Schulman et al., 2017c](#)), showing that while MPC performs better, it is computationally demanding at runtime, whereas RL offers faster execution. Other works by [Guiducci et al. \(2023\)](#) and [Palma et al. \(2024\)](#) use centralized RL to schedule BESS operations to maximize daily social welfare, defined as the sum of all node costs and revenues. Training is guided by a MILP formulation, used as a baseline, which, however, overlooks households' self-interest and assumes all BESS units are empty at the start and end of each day.

Overall, the presented centralized approaches evidently face two significant limitations: they present a single point of failure and privacy risk, and they often assume perfect consensus within the community, neglecting the self-interested behavior of individual prosumers.

10.2.3 Multi-Agent Systems and Game Theoretic Approaches

To better reflect the decentralized structure of RECs, recent works have shifted toward MARL and game-theoretic formulations. [Zhou et al. \(2019\)](#) propose a market-based model in which intelligent households, trained via Fuzzy Q-learning ([Berenji, 1994](#)), optimize battery usage and trade energy in response to dynamic pricing within a private network. Tariffs within the REC, determined by the energy supply-demand ratio, are lower than those for purchasing from the external grid and higher than those for selling surplus energy. Similarly, [Prasad and Dusparic \(2019\)](#) study a setting where a group of buildings aims to achieve net-zero grid withdrawal by trading energy internally and externally. Their agents, trained with DQN ([Mnih et al., 2015](#)), share a global reward that encourages collaboration but overlook self-interested behaviors. Furthermore, [Zang and Kim \(2021\)](#) introduce a peer-to-peer market brokered by a central *energy trade supporter* trained with Q-learning ([Watkins and Dayan, 1992](#)). While this entity serves to match bids in a two-phase market process and maximizes its arbitrage profit, it also manages a *community energy storage*. Finally, to address the conflict between individual and global goals, leader-follower dynamics have been explored. [Lai et al. \(2022\)](#) modeled the REC as a Stackelberg game, with a central REC controller as the leader and households as followers. The game is formulated so that at each step, the leader determines the energy price for subsequent time slots based on the followers' demand profiles. In response, followers adjust their expected demand profiles based on the pricing and forward them to the controller, which then elaborates a new pricing strategy. This process iterates until

a Stackelberg equilibrium is reached. While effective, this approach is computationally intensive and requires high communication overhead.

While these works foster prosumer autonomy and collaboration, reliance on such value-based methods necessitates discrete action spaces, which often limit expressiveness and can be insufficient for the fine-grained control required by modern BESSs.

CHAPTER 11

A Multi-Agent Reinforcement Learning Approach

In this chapter, we present our solution for the energy management problem in RECs, addressing the formulation introduced in Section 10.1. Building on the results of Part III, where we demonstrated the efficacy of combining DTs and RL agents for single-MG energy management, we now extend this paradigm to a multi-entity domain. As highlighted in recent works (Ponse et al., 2024; Esan and Shareef, 2025; Legrene et al., 2025), RL is gaining traction in energy management settings due to its ability to learn optimal strategies through interaction with complex, stochastic environments, without requiring an explicit analytical model of the system dynamics.

However, as discussed in Section 10.2, applying standard RL techniques to RECs presents unique challenges. More specifically, many existing approaches rely on centralized control schemes, which are poorly scalable and offer limited privacy protection, or assume overly simplified models of battery behavior and incentives, thereby limiting their ability to capture the complexities of real-world, regulation-driven REC contexts. Given its inherently distributed architecture, an REC is best modeled as a multi-agent system in which autonomous prosumers pursue individual objectives. Consequently, Multi-Agent Reinforcement Learning (MARL) (Zhang et al., 2021; Oroojlooy and Hajinezhad, 2023) emerges as the natural framework for this problem. Indeed, MARL can overcome the limitations of centralized approaches by enabling decentralized decision-making, thus enhancing scalability and privacy. Moreover, integrating this framework with DT technologies enables accurate modeling of complex system dynamics and incentive mechanisms, facilitating the development of more effective energy management strategies.

Nevertheless, learning in a multi-agent environment is non-trivial. The main obstacles include the *non-stationarity* introduced by multiple agents learning simultaneously and the *credit assignment problem*, which makes it challenging to discern the contribution

of individual agents' actions to the overall system performance. Standard independent learners typically fail to account for this adaptation, often leading to convergence failures or suboptimal equilibria. To address these challenges, we draw inspiration from *opponent-aware learning*. Algorithms like *Learning with Opponent-Learning Awareness* (LOLA) (Foerster et al., 2018) enable agents to anticipate their peers' learning steps and shape their own policies to guide the collective toward better outcomes.

Yet, applying standard LOLA directly to an REC is problematic. A fully decentralized implementation in which every household models the learning process of every other household is computationally prohibitive and privacy-intrusive. Therefore, in this chapter, we propose a novel algorithm tailored to the hierarchical REC architecture, called *INcentive Allocation for Interest Alignment* (INAIA). In our framework, a central, community agent learns to influence the learning dynamics of decentralized, prosumer agents via *node-aware learning*. This approach preserves prosumer autonomy and scalability while effectively guiding the system toward cooperative equilibria.

Chapter Outline. The chapter is organized as follows. In Section 11.1, we summarize the fundamentals of MARL, focusing on the aspects relevant to our proposed solution. Next, in Section 11.2, we provide a detailed formulation of the REC energy management problem. Finally, in Section 11.3, we introduce our proposed algorithm.

11.1 Fundamentals of Multi-Agent Reinforcement Learning

Multi-Agent Reinforcement Learning (MARL) extends the single-agent RL framework to stochastic environments populated by multiple interacting agents. Unlike the single-agent case, where a solitary learner optimizes its policy against a stationary background, MARL involves a set of n agents, each learning a policy π_i through a *trial-and-error* process to maximize their individual cumulative returns, which can be the expression of either a selfish or a shared objective. The transition from single-agent to multi-agent learning is driven by two primary use cases: the *decomposition of intractable problems* and *inherently decentralized constraints*.

The former arises when the complexity of a task exceeds the capabilities of a single agent. In large-scale systems (like RECs), controlling all assets via a single central agent leads to an exponential explosion of the action space. For example, supposing a problem where we need to control three robots with 6 actions each, we would have $6^3 = 216$ joint actions. MARL decomposes this into smaller, tractable decision problems, e.g., three agents with 6 actions each. However, this introduces other challenges, such as coordination or communication.

The latter case concerns scenarios where centralized control is impractical or impossible due to physical, technical, or privacy constraints. Many real-world applications, such as autonomous driving or wide-area grid control, lack the reliable communication infrastructure required for centralized control (Gronauer and Diepold, 2022). In these settings, agents must learn decentralized policies that can be executed locally based solely on local observations.

Further details on the MARL framework are provided in well-known overviews, such as Zhang et al. (2021) and Oroojlooy and Hajinezhad (2023), as well as in textbooks like Albrecht et al. (2024). In this section, we summarize the key concepts necessary

to understand the decentralized energy management strategy developed in subsequent chapters.

11.1.1 Game Theoretic Structures

An important aspect of MARL is characterizing agents' interactions through the lens of game theory. Depending on the alignment or conflict of agents' objectives, different game structures emerge, influencing the learning dynamics and the nature of equilibria.

Three primary categories of games are commonly studied in MARL:

- *Cooperative (Dec-POMDP)*: All agents share a single global reward function, i.e., $R_1 = \dots = R_N \equiv R_{\text{global}}$, where R_n denotes the individual reward function of agent $n \in \{1, \dots, N\}$. The problem essentially reduces to finding a joint policy that maximizes the common expected return (Oliehoek et al., 2016).
- *Zero-Sum (Competitive)*: In two-agent games, the reward functions are strictly opposed, such that $R_1 = -R_2$. One agent's gain is exactly the other's loss. This symmetry leads to specific mathematical properties (such as the minimax equilibrium) that simplify theoretical analysis (Littman, 1994).
- *General-Sum (Mixed)*: Agents have distinct, potentially conflicting, but not strictly opposite goals. This setting is the most complex and realistic, as it captures a wide range of interactions, including cooperation, competition, and negotiation (Hu and Wellman, 2003).

In our setting, the interactions among agents can be modeled as a general-sum game, where each agent aims to minimize its own utility function, while also considering the overall community welfare.

11.1.2 The Multi-Agent Interaction Loop

In the single-agent setting, the MDP framework suffices to model the interaction between the agent and the environment. Conversely, in the multi-agent case, we need to extend this to capture the dynamics introduced by multiple learning entities. The most general formalism is the so-called *Partially Observable Stochastic Games* (POSGs) (Hansen et al., 2004), an extension of the *Stochastic Game* defined by Shapley (1953). Formally, a POSG is defined by the tuple:

$$\mathcal{M}_{\text{POSG}} := \langle \mathcal{I}, \mathcal{S}, \mathcal{A}, P, \Omega, \mathcal{O}, \mathcal{R}, \gamma \rangle, \quad (11.1)$$

where:

- $\mathcal{I} = \{1, \dots, N\}$ is the *set of agents*.
- $\mathcal{S}$ is the *global state space*, describing the full configuration of the environment.
- $\mathcal{A} = \mathcal{A}_1 \times \dots \times \mathcal{A}_N$ is the *joint action space*. At each time step t , the joint action is $\mathbf{a}_t = (a_{1,t}, \dots, a_{N,t})$.
- $P : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow [0, 1]$ is the *transition function*. The next state s_{t+1} depends on the current state and the *joint action*: $s_{t+1} \sim P(\cdot | s_t, \mathbf{a}_t)$.

- $\Omega = \{\Omega_1, \dots, \Omega_N\}$ represents the *observation spaces*, i.e., the portions of the environment state that each agent can perceive.
- $\mathcal{O} : \mathcal{S} \times \mathcal{I} \rightarrow \Omega$ is the *observation function*. Each agent i receives a local observation $o_{i,t} \in \Omega_i$ which correlates with the state s_t , typically capturing only partial information (partial observability).
- $\mathcal{R} = \{R_1, \dots, R_N\}$ is the set of *reward functions*, where $R_i : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$. Agent i receives a reward $r_{i,t} = R_i(s_t, \mathbf{a}_t)$.
- $\gamma \in [0, 1]$ is the discount factor.

While in game theory, POSGs are treated as *complete knowledge games* (Owen, 2013), in MARL we usually operate at the other end of the knowledge spectrum: agents typically do not know the reward functions of other agents, nor even their own reward function; and agents have no knowledge of the state transition and observation functions, but they have to learn optimal strategies solely through interaction with the environment.

The way in which MARL agents make experience of the environment is similar to the single-agent case. As illustrated in Figure 11.1, at each time step t , every agent i observes the partial observation $o_{i,t}$ of the state s_t and selects an individual action $a_{i,t}$. Collectively, these form the *joint action* $\mathbf{a}_t = (a_{1,t}, \dots, a_{N,t})$. The environment dynamics are driven by this joint action, transitioning the system to a new state and yielding individual rewards $r_{i,t}$ for each agent. Hence, the trajectory of the system over time is defined as $\tau = (s_0, \mathbf{a}_0, s_1, \mathbf{a}_1, \dots)$, and each agent has to learn a policy $\pi_i(a_i|o_i)$ to maximize its own expected discounted return. Formally:

$$J_i(\boldsymbol{\pi}) = \mathbb{E}_{\tau \sim \boldsymbol{\pi}} \left[\sum_{t=0}^{\infty} \gamma^t R_i(s_t, \mathbf{a}_t) \right], \quad (11.2)$$

where the notation $J_i(\boldsymbol{\pi})$ highlights that the return depends on the joint policy $\boldsymbol{\pi} = (\pi_1, \dots, \pi_N)$. While the return is defined individually for each agent, the environment's evolution is inherently coupled through the joint actions.

This creates a fundamental dependency: an agent's reward and the future state depend on the actions of all agents in the system. It is easy to show that the optimal strategy for one agent is contingent on the strategies of all other agents. At the same time, it is more difficult to interpret the effect of a single agent's action on the environment, as it is confounded by the actions of others (a.k.a. *credit assignment problem*).

11.1.3 Reduction to Single Agent

To manage the complexity of MARL, various approaches reduce the multi-agent problem to a series of single-agent problems. The two primary paradigms are *central learning* and *independent learning* (de Witt et al., 2020). In approaches employing fully *central learning*, a single agent controls the joint action $\mathbf{a}$ of all agents. This transforms the multi-agent problem into a single-agent MDP with an exponentially larger action space. While this simplifies the learning process and avoids non-stationarity, it suffers from poor scalability and requires full observability of the global state, which is often impractical in real-world scenarios. On the other hand, *independent learning* (IL) treats each agent as an autonomous learner, optimizing its own policy π_i based solely on its local

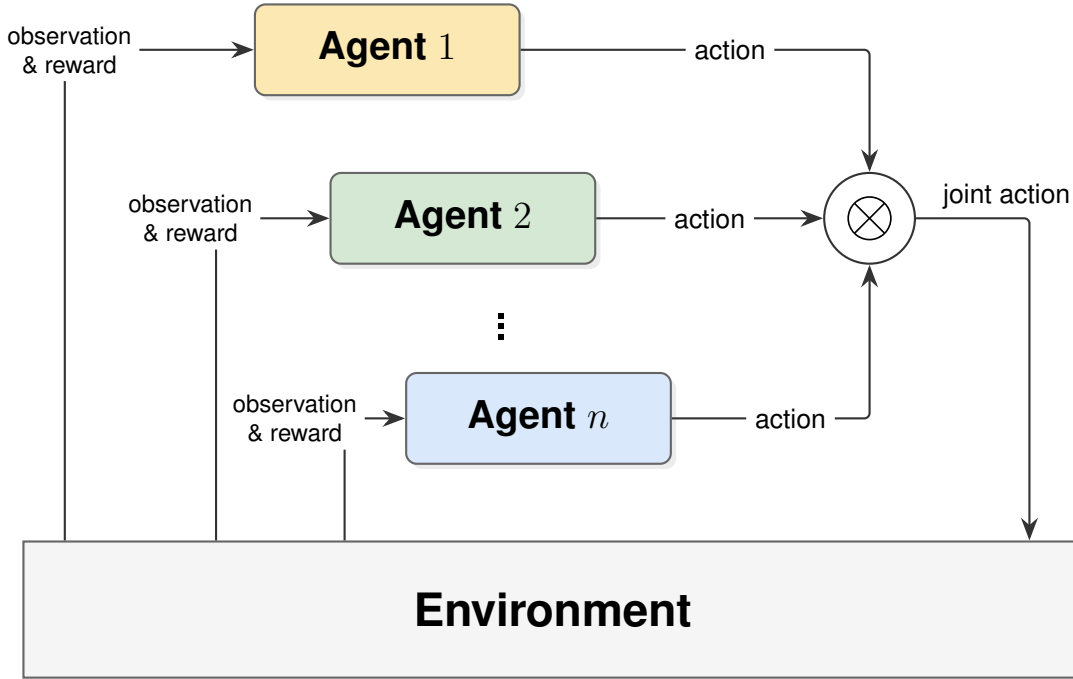


Figure 11.1: Schematic representation of the interaction loop in a MARL setting. Each agent i observes the environment state (or a partial observation) and selects an action a_i . The joint action $\mathbf{a} = (a_1, \dots, a_n)$ influences the environment's transition to a new state and yields individual rewards r_i for each agent. This interdependence creates a complex dynamic where each agent's optimal strategy is contingent on the actions of all other agents.

observations and rewards. Each agent treats other agents as static components of the environment and applies single-agent algorithms (e.g., Independent Q-Learning). While scalable, IL ignores the non-stationarity problem. It often fails to converge in cooperative settings because agents cannot distinguish between the environment's stochasticity and peers' exploratory actions. Thus, to foster coordination, we need to arrange some sort of communication among agents.

11.1.4 Challenges in Multi-Agent Reinforcement Learning

MARL algorithms inherit some challenges from single-agent RL, such as unknown environment dynamics, exploration-exploitation trade-offs, non-stationarity from bootstrapping, and temporal credit assignment. However, the multi-agent context introduces additional complexities inherent to the presence of multiple learning entities.

Non-Stationarity. A central challenge in MARL is the non-stationarity of the environment from the perspective of individual agents. This instability can lead to cyclic dynamics, whereby a learning agent adapts to the changing policies of other agents, who in turn adapt to the learning agent's updated policy. This phenomenon differs from the non-stationarity encountered in single-agent RL (such as shifting target values during bootstrapping); in MARL, the effective transition dynamics of the environment itself are continuously changing. From the perspective of agent i , the other agents $j \neq i$ are part of the environment. As agents update their policies π_j during training, the environment's

effective transition dynamics change:

$$P(s'|s, a_i) = \sum_{\mathbf{a}_{-i}} P(s'|s, a_i, \mathbf{a}_{-i}) \prod_{j \neq i} \pi_j(a_j|o_j). \quad (11.3)$$

Since π_j evolves over time, the optimal policy for agent i effectively chases a *moving target*, violating the Markov assumption required for standard convergence guarantees.

Multi-Agent Credit Assignment. Credit assignment in RL is the problem of determining which past actions have contributed to the received rewards. In multi-agent RL, there is an additional question: whose actions among multiple agents contributed to the rewards? To distinguish the two problems, the first type is usually called *temporal credit assignment*, while the second is called *multi-agent credit assignment*. The problem of multi-agent credit assignment is especially prominent in common-reward settings, since each reward is applied indiscriminately to each agent, effectively leaving it to the agents to disentangle the effects of everyone's actions on the rewards. This can lead to situations in which an agent's action is repeatedly reinforced even when receiving a positive reward to which it made no contribution. To resolve multi-agent credit assignment, an agent needs to understand the effect of its own actions on the reward it receives, versus the effect of other agents' actions. Ascribing values to joint actions, as in central learning, is a useful way to disentangle each agent's contribution to the received reward.

Scalability. As the number of agents increases, the joint action space grows exponentially, leading to the so-called *curse of dimensionality*. This makes it infeasible to learn value functions or policies that depend on the full joint action space. Additionally, communication and coordination overheads increase with more agents, complicating the learning process. While exponential growth due to the number of agents is an important challenge in MARL, it is not unique to MARL. Single-agent reductions of the problem, such as central learning, still face exponential growth in the number of actions.

11.1.5 Training and Execution Modes

MARL algorithms can be classified based on the information during training and execution. During training, it may be necessary to access the global state and actions of all agents to stabilize learning. In other approaches, agents may have access only to local observations and rewards. Similarly, during execution, agents may act based on global information or be restricted to local observations. Based on these criteria, we can identify three main execution modes.

Centralized Training and Execution. In *centralized training and execution*, the learning of agents' policies uses information or mechanisms that are centrally shared among all agents both during training and execution, such as local observations, actions, rewards, value functions, or even policies. In practice, in this setting, the concept of partial observability is often ignored, assuming that all agents have access to the full global state s during both training and execution. This allows for joint policy learning, where a single policy $\pi(\mathbf{a}|s)$ is learned for the entire system. While this approach can yield optimal joint strategies, it suffers from scalability issues and is often impractical in real-world scenarios due to communication and privacy constraints. Examples of this

approach include *Centralized Q-Learning* (Littman, 1994) and *Joint Action Learning* (JAL) (Claus and Boutilier, 1998).

Decentralized Training and Execution. In *decentralized training and execution*, each agent learns its policy $\pi_i(a_i|o_i)$ independently, using only its local observations o_i and rewards r_i during both training and execution. This approach treats other agents as part of the environment, leading to non-stationarity challenges. While decentralized methods are scalable and respect privacy constraints, they often struggle with convergence and coordination, especially in cooperative settings. Examples include *Independent Q-Learning* (Tan, 1997) and *Independent PPO* (Schulman et al., 2017b).

Centralized Training with Decentralized Execution. *Centralized Training with Decentralized Execution* (CTDE) has emerged as the dominant paradigm in modern MARL, effectively bridging the gap between centralized optimization and decentralized control. This framework leverages the distinction between the information available during *training* (often performed in a simulator or controlled environment) and *execution* (or *deployment*). During the training phase, the algorithm leverages global information, such as the full system state and joint actions across all agents, to stabilize learning and mitigate non-stationarity. However, the resulting policies are constrained to rely solely on local observations, ensuring they remain executable in a fully decentralized manner during execution. In Actor-Critic architectures, this is typically achieved by training a centralized critic that estimates the joint value function $Q(s, a)$ while optimizing the decentralized actors $\pi_i(a_i|o_i)$. This allows the critic to correctly evaluate complex joint interactions, providing stable gradients to actors that, once trained, operate independently. Most relevant examples include *Multi-Agent Deep Deterministic Policy Gradient* (MADDPG) (Lowe et al., 2017), *Value Decomposition Networks* (VDN) (Sunehag et al., 2017), and QMIX (Rashid et al., 2020).

11.1.6 Opponent-Aware Learning

Standard independent learning approaches often fail in general-sum games because they ignore the non-stationarity introduced by the simultaneous adaptation of other agents. To address this, in this dissertation, we will adopt the paradigm of *opponent-aware learning*, which explicitly models how an agent’s policy update influences its peers’ future policies. In particular, we focus on the *Learning with Opponent-Learning Awareness* (LOLA) algorithm (Foerster et al., 2018), which modifies an agent’s learning objective to explicitly account for the learning dynamics of others.

Consider a two-agent game where agents 1 and 2 optimize parameters θ_1 and θ_2 to maximize their respective expected returns $V_1(\theta_1, \theta_2)$ and $V_2(\theta_1, \theta_2)$. In a naive independent learning setting (e.g., Independent PPO), agent 1 assumes θ_2 is fixed and updates θ_1 via gradient ascent:

$$\theta_1^{k+1} = \theta_1^k + \alpha \nabla_{\theta_1} V_1(\theta_1, \theta_2), \quad (11.4)$$

where α is the step size.

LOLA modifies this process by allowing agent 1 to anticipate the learning step of agent 2, which is also a learning agent who will update its policy in response to the current state of the game. Hence, agent 1 aims to optimize its objective *after* this

anticipated update. The LOLA modified objective for agent 1 becomes:

$$V_1^{\text{LOLA}}(\theta_1, \theta_2) \approx V_1(\theta_1, \theta_2 + \Delta\theta_2(\theta_1)), \quad (11.5)$$

where $\Delta\theta_2(\theta_1) = \eta \nabla_{\theta_2} V_2(\theta_1, \theta_2)$ is the anticipated naive update of agent 2 with learning rate η . To derive the update rule, we need a first-order Taylor expansion of V_1 around the current parameters (θ_1, θ_2) :

$$V_1(\theta_1, \theta_2 + \Delta\theta_2) \approx V_1(\theta_1, \theta_2) + (\Delta\theta_2)^\top \nabla_{\theta_2} V_1(\theta_1, \theta_2). \quad (11.6)$$

Substituting the definition of $\Delta\theta_2$:

$$V_1^{\text{LOLA}} \approx V_1(\theta_1, \theta_2) + \eta (\nabla_{\theta_2} V_2(\theta_1, \theta_2))^\top \nabla_{\theta_2} V_1(\theta_1, \theta_2). \quad (11.7)$$

To find the update rule for agent 1, we compute the gradient of this objective with respect to θ_1 . Applying the product rule to the second term yields two components:

$$\nabla_{\theta_1} V_1^{\text{LOLA}}(\theta_1, \theta_2) \approx \nabla_{\theta_1} V_1 + \eta \frac{\partial(\nabla_{\theta_2} V_2)^\top}{\partial \theta_1} \nabla_{\theta_2} V_1 + \eta (\nabla_{\theta_2} V_2)^\top \frac{\partial \nabla_{\theta_2} V_1}{\partial \theta_1}. \quad (11.8)$$

Crucially, the LOLA algorithm makes a simplifying assumption to isolate the opponent-shaping behavior. As stated in (Foerster et al., 2018), the dependence of $\nabla_{\theta_2} V_1$ on θ_1 (the third term) is dropped during the backward pass. This is because the algorithm focuses specifically on how agent 1 can influence the *learning step* of the opponent ($\Delta\theta_2$), rather than how agent 1 changes the sensitivity of its own value function.

Discarding the third term, we arrive at the specific *LOLA Learning Rule*:

$$\mathbf{f}_{\text{LOLA}}^1 = \underbrace{\nabla_{\theta_1} V_1(\theta_1, \theta_2)}_{\text{Naive Gradient}} + \underbrace{\eta (\nabla_{\theta_2} V_1(\theta_1, \theta_2))^\top \nabla_{\theta_1} \nabla_{\theta_2} V_2(\theta_1, \theta_2)}_{\text{Correction Term}}, \quad (11.9)$$

where the correction term captures the mixed second-order derivatives. Intuitively, this term asks: «"How does a change in my parameters θ_1 alter the gradient of my opponent $\nabla_{\theta_2} V_2$?"» By ascending this gradient, agent 1 steers agent 2 towards policy updates that are beneficial for agent 1, often fostering cooperative equilibria that naive learners fail to discover.

11.2 MARL-based Energy Community Management

In this section, starting from the formulation of the setting provided in Section 10.1, we formalize the REC management problem as a *Two-Phase Stochastic Game* (TPSG), a modification of the common POSG described in Section 11.1.2. This game involves the presence of two types of players: *node agents*, i.e., prosumers participating in the REC, and the *REC agent*, which acts as a centralized authority to regulate the allocation of incentives. For each of these agent types, we will provide descriptions of the observation and action spaces and the reward functions. Finally, we will introduce INAIA, the algorithm we designed to address the presented problem.

11.2.1 Two-Phase Stochastic Game Formulation

We model the REC management problem as a TPSG. In the first phase of each learning step t , a set of decentralized node agents acts simultaneously based on their local

observations. In the second phase, a centralized community agent observes the global state and the joint actions of the nodes, then selects its own action. Thus, the environment transitions to the next state, proceeding with the following learning step $t + 1$, where the two phases will be iterated again. Notably, in our setting, the decentralized agents correspond to the set of *active* nodes $\mathcal{N}_A$, while the centralized agent represents the community-level controller, denoted by symbol $\mathfrak{C}$.

Formally, we present the game as a variation of a POSG, defined by the tuple:

$$\mathcal{M}_{\text{TPSG}} := \langle \mathcal{I}, \mathcal{S}, \mathcal{A}, P, \mathcal{R}, \Omega, \mathcal{O}, \gamma \rangle,$$

where the elements differing from the standard POSG definition are described as follows:

- $\mathcal{I} := \mathcal{N}_A \cup \{\mathfrak{C}\}$ is the set of agents involved in the TPSG;
- $\mathcal{A} := \times_{i \in \mathcal{I}} \mathcal{A}_i = \mathcal{A}_{\mathcal{N}_A} \times \mathcal{A}_{\mathfrak{C}}$ is the joint action space of the agents, where $\mathcal{A}_{\mathcal{N}_A} := \times_{n \in \mathcal{N}_A} \mathcal{A}_n$;
- $P : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow \Delta(\mathcal{S})$ is the state transition function, mapping the current state $s \in \mathcal{S}$ and joint action $\mathbf{a} = (\mathbf{a}_{\mathcal{N}_A}, a_{\mathfrak{C}})$, with $\mathbf{a}_{\mathcal{N}_A} \in \mathcal{A}_{\mathcal{N}_A}$ and $a_{\mathfrak{C}} \in \mathcal{A}_{\mathfrak{C}}$ to a probability distribution over next states;
- $\mathcal{R} := \{R_i | i \in \mathcal{I}\} = \{R_n | n \in \mathcal{N}_A\} \cup \{R_{\mathfrak{C}}\}$ is the set of the reward functions for the agents, where each $R_i : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ is the reward function of agent $i \in \mathcal{I}$;
- $\Omega := \{\Omega_i | i \in \mathcal{I}\} = \{\Omega_n | n \in \mathcal{N}_A\} \cup \{\Omega_{\mathfrak{C}}\}$, where Ω_i is the set of observations for agent $i \in \mathcal{I}$;
- $\mathcal{O} := \{O_i | i \in \mathcal{I}\} = \{O_n | n \in \mathcal{N}_A\} \cup \{O_{\mathfrak{C}}\}$ is the set of observation functions, where $O_n : \mathcal{S} \rightarrow \Omega_n$ is the observation function for node agent $n \in \mathcal{N}_A$ and $O_{\mathfrak{C}} : \mathcal{S} \times \mathcal{A}_{\mathcal{N}_A} \rightarrow \Omega_{\mathfrak{C}}$ is the observation function for the centralized agent.

Standard stochastic game formulations (Yang et al., 2025) typically assume that all agents act simultaneously. Conversely, our TPSG formulation is inherently sequential within a single learning step, being split into two phases. In the first phase, the node agents $\mathcal{N}_A$ simultaneously select and execute their actions $\mathbf{a}_{\mathcal{N}_A}$ based on their local observations. Subsequently, in the second phase, the centralized REC agent $\mathfrak{C}$ conditions its action $a_{\mathfrak{C}}$ on both the state and realization of $\mathbf{a}_{\mathcal{N}_A}$. Finally, rewards are distributed, and the system transitions are driven by the joint action pair $(\mathbf{a}_{\mathcal{N}_A}, a_{\mathfrak{C}})$.

11.2.2 Node Agent

At the node level, each *active* node is controlled by an independent agent that manages the BESS. Hereafter, we define observation and action spaces and the reward function of node agents.

Observation Space. The observation vector $o_{n,t} \in \Omega_n$ for a node agent $n \in \mathcal{N}_A$ at time t resembles the state space for the MG control problem, described in Section 8.3. Similarly, it is composed of local variables (internal battery state, energy demand, energy generation), market prices, and time encodings. However, unlike the MG setting, we

also include two additional variables related to community self-consumption. Formally, the observation at time t for a node n is the tuple:

$$o_{n,t} := \left(\sigma_{n,t}, \vartheta_{n,t}, \hat{d}_{n,t}, \hat{g}_{n,t}, c_{n,t}^{\text{buy}}, c_{n,t}^{\text{sell}}, p_{t-1}^+, p_{t-1}^-, t_{\text{day}}, t_{\text{year}} \right),$$

where the new features are $p_{t-1}^+ := \sum_{j \in \mathcal{N}} p_{j,t-1}^+$ and $p_{t-1}^- := \sum_{j \in \mathcal{N}} p_{j,t-1}^-$, i.e., the aggregate community injection and withdrawal, respectively. The assumptions about demand and generation estimates, market prices, and time encodings are the same as in Section 8.3.

Action Space. The action of a node agent $n \in \mathcal{N}_A$ is defined as a continuous scalar value $a_{n,t} \in \mathbb{R}$, representing the electrical current $i_{n,t}$ to be applied to the battery at time step t . This formulation marks a significant departure from the single MG setting presented in Section 8.3. There, the action was defined as a percentage of the instantaneous power surplus, inherently bounding the battery's operation to the availability of local renewable generation. In the REC context, we remove this constraint to allow for fully *grid-interactive* strategies. By defining the action as an absolute physical quantity (current), the agent is not limited to storing (injecting) excess (deficit) PV production but can also proactively charge (discharge) from (to) the grid when beneficial. This flexibility is essential for enabling *energy arbitrage*, buying energy during low-price periods regardless of generation, while simultaneously optimizing for the community's collective self-consumption objective.

If $a_{n,t} > 0$ the battery is charging; otherwise, with $a_{n,t} < 0$, it is discharging. As explained in Section 10.1.1, to avoid disruptive phenomena for the battery, the load should be maintained within the physical boundaries. Since here the action directly represents the current, these constraints can be expressed as simple bounds on the action values, defined by $i_{n,t}^{\min}$ and $i_{n,t}^{\max}$ such that:

$$i_{n,t}^{\min} := \frac{\sigma_n^{\min} - \sigma_{n,t}}{\Delta t} \cdot C_{n,t}^{\max}, \quad (11.10)$$

$$i_{n,t}^{\max} := \frac{\sigma_n^{\max} - \sigma_{n,t}}{\Delta t} \cdot C_{n,t}^{\max}, \quad (11.11)$$

where we recall that $C_{n,t}^{\max}$ is the maximum capacity available at node n . Then, from the selected current and the actual battery voltage, we compute the power $b_{n,t}$ used to determine the net power $p_{n,t}$.

Reward Function. The reward function $R_n(s_t, \mathbf{a}_t)$ of node agent $n \in \mathcal{N}_A$ at time t resembles the node-level objective described in Section 10.1 and it derives from the combination of market interactions, physical degradation, safety penalties, and the REC incentive. Formally, at time t the node n receives a reward signal defined as follows:

$$r_{n,t} := \underbrace{r_{n,t}^{\text{trad}} + r_{n,t}^{\text{deg}} + r_{n,t}^{\text{clip}}}_{=: r_{n,t}^{\text{loc}}} + r_{n,t}^{\text{inc}}, \quad (11.12)$$

where $r_{n,t}^{\text{loc}}$ is the local, self-interested reward component which depends only on the individual agent action $a_{n,t}$, while $r_{n,t}^{\text{inc}}$ captures the community-level incentive and is

influenced by the REC joint strategy. Notice that the local reward $r_{n,t}^{\text{loc}}$ is structurally similar to the reward function defined in Section 8.3, apart from the clipping penalty term $r_{n,t}^{\text{clip}}$, which needs to be rewritten due to the new action space definition. More specifically:

- $r_{n,t}^{\text{trad}}$ accounts for the monetary revenue (or cost) derived from energy transactions (*purchasing* and *selling*) with the main grid. The term is defined as in Equation (8.21);
- $r_{n,t}^{\text{deg}}$ represents the cost associated with BESS degradation. The term is defined as in Equation (8.23);
- $r_{n,t}^{\text{clip}}$ is a penalty term applied when the agent attempts infeasible actions, violating the physical bounds $i_{n,t}^{\min}$ and $i_{n,t}^{\max}$, defined in Equations (11.10) and (11.11) respectively. The term is formally defined as follows:

$$r_{n,t}^{\text{clip}} := \begin{cases} 0 & \text{if } i_{n,t}^{\min} \leq a_{n,t} \leq i_{n,t}^{\max} \\ - (i_{n,t}^{\min} - a_{n,t})^2 & \text{if } a_{n,t} < i_{n,t}^{\min} \\ - (a_{n,t} - i_{n,t}^{\max})^2 & \text{if } a_{n,t} > i_{n,t}^{\max} \end{cases}; \quad (11.13)$$

- $r_{n,t}^{\text{inc}}$ denotes the agent's share of incentives distributed by the REC agent. This component is formally defined in the following section.

11.2.3 REC Agent

At the community level, the centralized REC agent $\mathcal{C}$ aims to foster coordination among network members to maximize self-consumption. To achieve this goal, it employs a reward-shaping mechanism to align the individual interests of *active* nodes by distributing incentives. Indeed, the $r_{n,t}^{\text{inc}}$ reward component of node $n \in \mathcal{N}_A$ is ultimately decided by the REC agent, after observing, in the second phase of the TPSG, the state of the network and *active* nodes' actions.

Notably, as described in Section 10.1, REC members include both *active* and *passive* nodes. Hence, the incentive allocation policy must consider a portion of the incentives given to the *passive* households that cannot be controlled or influenced. We model this aspect by allocating to passive nodes the portion $V_t^{\mathcal{N}_P} := \frac{|\mathcal{N}_P|}{N} V_t$, which is divided uniformly among *passive* households, so that

$$\omega_{n,t} = \frac{V_t^{\mathcal{N}_P}}{|\mathcal{N}_P|}, \quad \forall n \in \mathcal{N}_P.$$

The remaining amount of incentives, i.e., $V_t^{\mathcal{N}_A} := \frac{|\mathcal{N}_A|}{N} V_t$, is distributed among the *active* nodes according to the actions of the REC agent. It is worth noting that while being a *passive* node might seem sufficient to receive incentives, this benefit is strictly contingent upon the presence and behavior of *active* members. Moreover, a passive approach still incurs the standard costs of purchasing energy from the grid, which cannot be mitigated by a local PV plant.

Observation Space. The observation of the REC agent $o_t^{\mathcal{C}}$ aggregates the status of all active nodes, community-level statistics, and time encodings. Formally:

$$o_t^{\mathcal{C}} := \left(\{d_{n,t}\}_{n \in \mathcal{N}_A}, \{g_{n,t}\}_{n \in \mathcal{N}_A}, \{b_{n,t}\}_{n \in \mathcal{N}_A}, p_t^+, p_t^-, \bar{d}_t, \bar{g}_t, \bar{b}_t, t_{\text{day}}, t_{\text{year}} \right),$$

where:

- $d_{n,t} \in \mathbb{R}$ is the current energy demand of *active* node n ;
- $g_{n,t} \in \mathbb{R}$ is the current energy generation of *active* node n ;
- $b_{n,t} \in \mathbb{R}$ is the current energy used to charge/discharge the battery of *active* node n ;
- $p_t^+ := \sum_{n \in \mathcal{N}} p_{n,t}^+$ and $p_t^- := \sum_{n \in \mathcal{N}} p_{n,t}^-$ are the two components of the REC self-consumption, namely the sum of the energy injections and the sum of the energy withdrawals, respectively;
- $\bar{d}_t := \frac{1}{N} \sum_{n \in \mathcal{N}} d_{n,t}$ and $\bar{g}_t := \frac{1}{N} \sum_{n \in \mathcal{N}} g_{n,t}$ are the mean demand and generation, respectively, among all nodes;
- $\bar{b}_t := \frac{1}{|\mathcal{N}_A|} \sum_{n \in \mathcal{N}_A} b_{n,t}$ is the mean energy for charging/discharging the batteries of *active* nodes.

Action Space. The action of the REC agent $\mathcal{A}_{\mathcal{C}}$ is the $(|\mathcal{N}_A| - 1)$ -dimensional simplex $\Delta^{|\mathcal{N}_A|-1}$, i.e., the set of all possible vectors of proportions specifying the share of incentives each node $n \in \mathcal{N}_A$ receives. Formally, $\mathbf{a}_t^{\mathcal{C}} \in \Delta^{|\mathcal{N}_A|-1}$. Consequently, the vector of the incentives allocated for the node agents is defined as:

$$\boldsymbol{\omega}_t^{\mathcal{N}_A} := \{\omega_{n,t}\}_{n \in \mathcal{N}_A} = V_t^{\mathcal{N}_A} \cdot \mathbf{a}_t^{\mathcal{C}}. \quad (11.14)$$

Hence, for each node agent $n \in \mathcal{N}_A$ the incentive component of the reward at time t corresponds to $r_{n,t}^{\text{inc}} = \omega_{n,t}$.

Reward Function. Since the REC agent's objective is strictly to maximize the community self-consumption, its reward function $R_{\mathcal{C}}(s_t, \mathbf{a}_t)$ coincides with the self-consumption, thus:

$$r_t^{\mathcal{C}} = U_t.$$

11.3 The Proposed Solution: INAlA

From the formulation of the TPSG, we derive that every round (or learning step) of the problem can be viewed as a general-sum game (Nash, 1951), in which agents can both cooperate and compete. General-sum games are notoriously difficult to solve, as the presence of multiple learning agents makes the environment non-stationary from each agent's perspective. Moreover, if agents do not explicitly account for the behavior of others, the learning process can easily converge to sub-optimal *tragedy of the commons* equilibria.

LOLA, already presented in Section 11.1.6, directly addresses this issue by allowing each agent to account for how its policy update will influence the behavior of other agents. However, in our scenario, applying LOLA in its standard *all-to-all* modeling

form poses significant challenges. First, it raises privacy concerns, as the standard all-to-all formulation requires each household to access the internal policy parameters of every other participant. Second, it suffers from scalability limitations, as the computational complexity of modeling the coupled learning processes of $\mathcal{N}_A$ agents grows quadratically with the network size. To overcome these barriers, we exploit the hierarchical structure of the REC, by restricting the opponent-awareness mechanism of LOLA solely to the centralized controller (the REC agent), while retaining standard independent learning for the node agents. This asymmetry preserves privacy, since nodes do not need to model each other and all the information is held by the centralized controller, and ensures linear scalability.

However, a fundamental challenge remains. Standard LOLA applications assumes that agents behaves like a standard RL agent, whose actions directly affect the environment dynamics, aiming to maximize their own expected discounted return. In contrast, our REC agent has no direct authority over physical state transitions (e.g., battery evolution), nor does its immediate reward depend on its own actions. Instead, its influence is strictly indirect: it just modifies the reward signals received by the node agents. Attempting to optimize a long-horizon discounted return through this indirect channel would exacerbate the *credit assignment problem*, introducing high variance into the gradient estimates and destabilizing the meta-learning process.

To address these constraints, we propose the *INcentive Allocation for Interest Alignment* (INAIA) algorithm, a tailored meta-learning procedure inspired by LOLA. Instead of optimizing a long-term discounted return, INAIA uses a myopic meta-objective for the centralized REC agent, aiming to maximize the instantaneous post-update self-consumption. By focusing on the immediate behavioral response induced by the incentives, we significantly reduce the variance of the meta-gradient, ensuring that the shaped rewards remain tightly coupled to the current system state. Moreover, to guarantee convergence stability, INAIA decouples the learning processes into two alternating phases. In the *REC meta-update phase*, the algorithm relies on simulated node updates. This allows the REC to estimate the sensitivity of the nodes' policies to the incentive allocation strategy in a controlled sandbox environment, without destabilizing actual node policies. This contrasts with standard approaches that might rely on potentially noisy real-time updates.

11.3.1 The INAIA Algorithm

Algorithm 4 details the INAIA procedure. Let $I \in \mathbb{N}$ denote the total number of learning iterations. Each iteration consists of two phases: (i) a meta-update of the REC-level policy $\pi_\phi^\mathcal{C}$ with parameters $\phi \in \Phi$ (Lines 3-12), and (ii) a simultaneous update of each node-level policy $\{\pi_{\theta_n}\}_{n \in \mathcal{N}_A}$ with parameters $\theta_n \in \Theta$ (Lines 13-16).¹ We denote by $E_\mathcal{C}$ and $E_{\mathcal{N}_A}$ the number of episodes per learning iteration for phase (i) and phase (ii), respectively. These hyperparameters govern the timescale separation between the leader (REC agent $\mathcal{C}$) and the followers (nodes $n \in \mathcal{N}_A$) in their learning processes.

During the REC meta-update phase (i), the algorithm anticipates the nodes' learning dynamics. For each episode up to $E_\mathcal{C}$, $\pi_\phi^\mathcal{C}$ is updated by simulating the behavioral response of the nodes through a sequence of $L \in \mathbb{N}$ *lookahead* gradient steps. These

¹Note that while all *active* nodes share the same policy architecture, they may maintain independent parameter sets θ_n to capture heterogeneous behaviors.

Algorithm 4 Pseudocode of INAIA algorithm.

Input: Iterations I , REC episodes $E_{\mathcal{C}}$, nodes episodes $E_{\mathcal{N}_A}$

- 1: Initialize REC parameters ϕ , node parameters $\{\theta_n\}_{n \in \mathcal{N}_A}$
- 2: **for** 1 to I **do**
- 3: **for** 1 to $E_{\mathcal{C}}$ **do**
- 4: $\{\hat{\theta}_n\} \leftarrow \{\theta_n\}$
- 5: **for** $l \in \{1, \dots, L\}$ **do**
- 6: Collect trajectory $\hat{\tau} \sim (\{\pi_{\hat{\theta}_n}\}, \pi_{\phi}^{\mathcal{C}})$
- 7: Simulate update on $\{\hat{\theta}_n\}$: $\hat{\theta}_n \leftarrow \hat{\theta}_n - \hat{\alpha}_n \nabla_{\hat{\theta}_n} \mathcal{L}_n(\hat{\tau}), \quad \forall n \in \mathcal{N}_A$
- 8: Collect new trajectory $\hat{\tau}' \sim (\{\pi_{\hat{\theta}_n}\}, \pi_{\phi}^{\mathcal{C}})$
- 9: Compute $J_{\mathcal{C}}$: $J_{\mathcal{C}}(\hat{\tau}') = \frac{1}{\hat{T}} \sum_{t=0}^{\hat{T}} R_{\mathcal{C}}(\hat{s}_t, \hat{\mathbf{a}}_t)$
- 10: Update ϕ : $\phi \leftarrow \phi + \alpha_{\mathcal{C}} \nabla_{\phi} J_{\mathcal{C}}(\hat{\tau}')$
- 11: **end for**
- 12: **end for**
- 13: **for** 1 to $E_{\mathcal{N}_A}$ **do**
- 14: Collect trajectory $\tau \sim (\{\pi_{\theta_n}\}, \pi_{\phi}^{\mathcal{C}})$
- 15: Update $\{\theta_n\}$: $\theta_n \leftarrow \theta_n - \alpha_n \nabla_{\theta} \mathcal{L}_n(\tau), \quad \forall n \in \mathcal{N}_A$
- 16: **end for**
- 17: **end for**

steps are applied to a temporary copy $\{\hat{\theta}_n\}$ of the current parameters $\{\theta_n\}$. Formally, within each inner step $l \in \{1, \dots, L\}$, we sample a trajectory $\hat{\tau}$ under the joint policy $(\{\pi_{\hat{\theta}_n}\}, \pi_{\phi}^{\mathcal{C}})$ (Line 6). We define a trajectory $\hat{\tau} := \{\hat{s}_0, \hat{\mathbf{a}}_0, \dots, \hat{s}_{\hat{T}}, \hat{\mathbf{a}}_{\hat{T}}\}$, where $\hat{T}$ represents the horizon of a simulated episode, and $\hat{s}_t$ and $\hat{\mathbf{a}}_t$ are the state and joint action at time t , where the node actions are sampled from the estimated policies $\{\pi_{\hat{\theta}_n}\}$. Hence, we perform a virtual gradient descent step for each node n :

$$\hat{\theta}_n \leftarrow \hat{\theta}_n - \hat{\alpha}_n \nabla_{\hat{\theta}_n} \mathcal{L}_n(\hat{\tau}),$$

where $\hat{\alpha}_n$ is the learning rate associated with the simulated updates and $\mathcal{L}_n$ is the loss function at the node level (Line 7). This loss depends on the RL algorithm adopted for the node agents, to which INAIA is agnostic. Crucially, the loss $\mathcal{L}_n$ is directly connected to the incentives generated by $\pi_{\phi}^{\mathcal{C}}$, establishing a functional dependence of $\{\hat{\theta}_n\}$ on ϕ . Subsequently, a second trajectory $\hat{\tau}'$ of length $\hat{T}$ is generated using the updated parameters $\{\hat{\theta}_n\}$ (Line 8), and the REC objective is evaluated (Line 9) as the mean community self-consumption over the trajectory:

$$J_{\mathcal{C}}(\hat{\tau}') = \frac{1}{\hat{T}} \sum_{t=0}^{\hat{T}} R_{\mathcal{C}}(\hat{s}_t, \hat{\mathbf{a}}_t).$$

The REC parameters ϕ are then updated to maximize this objective via gradient ascent with learning rate $\alpha_{\mathcal{C}}$ (Line 10). Crucially, this formulation establishes a coupled dependence between the REC parameters ϕ and the anticipated node parameters $\{\hat{\theta}_n\}$. To prevent optimization instability, we enforce the inner lookahead loop of L updates, at the beginning of which $\{\hat{\theta}_n\}$ are re-initialized to the current true parameters θ_n (Line 4). This resetting mechanism is essential to mitigate the compounding approximation error

inherent in long-horizon simulations, thereby ensuring the reliability of the meta-learning process.

After the REC agent learning phase, the algorithm proceeds to phase (ii) with the node policies' update. Over $E_{\mathcal{N}_A}$ episodes, the actual node parameters $\{\theta_n\}$ are updated through standard policy-gradient steps using trajectories $\tau := \{s_0, \mathbf{a}_0, \dots, s_T, \mathbf{a}_T\}$ sampled under $(\pi_{\theta_n}, \pi_\phi^{\mathbf{e}})$ from the real environment, where T is the horizon of the episode (Line 14). Hence, for each $n \in \mathcal{N}_A$, we perform a gradient descent step using the loss $\mathcal{L}_n$:

$$\theta_n \leftarrow \theta_n - \alpha_n \nabla_{\theta} \mathcal{L}_n(\tau),$$

with learning rate α_n associated with the true policy update (Line 15). A key property of INAIA is that the trajectories used to update ϕ in phase (i) are generated assuming that node agents act deterministically rather than through stochastic action sampling. This reduces gradient variance during the REC update and yields more stable optimization of the incentive allocation strategy.

CHAPTER 12

Experimental Evaluation

In the previous chapter, we formalized the problem of coordinating multiple households within an REC using a MARL framework. We also present our proposed solution, INAIA, a novel algorithm for training a community agent that effectively encourages cooperative behavior among self-interested household agents through incentives. In this chapter, we empirically evaluate the proposed approach through extensive experiments in a realistic simulation environment. Our evaluation focuses on two main aspects: (i) assessing the scalability and effectiveness of INAIA in various REC configurations, and (ii) demonstrating the benefits for individual households when participating in an REC managed by our method. Moreover, we compare two architectural variants of INAIA that differ in how the community agent processes information from individual households.

Chapter Outline. The chapter is organized as follows. In Section 12.1, we describe the experimental setup, including datasets, simulation environment, and baseline strategies. In Section 12.2, we present two case studies evaluating the performance of the proposed method in different scenarios. Finally, in Section 12.3, we conduct an ablation study to assess the impact of specific design choices on the overall performance.

12.1 Experimental Setup

To evaluate the proposed method under realistic conditions, we design a framework that integrates real-world data and the ERNESTO simulator. Specifically, as in the single MG case presented in Chapter 9, the time-series represent exogenous energy-related variables characterizing the evolution of the environment, while the simulator computes the dynamics of BESSs on active nodes. These components are described in

the following sections.

12.1.1 Datasets and Simulation Environment

We validate our approach by adopting the datasets described in Section 9.1, but using different generation profiles to have more variability among different nodes of the community. Specifically, renewable energy generation is addressed using synthetic data generated with the PVGIS tool (Huld et al., 2012), developed by the European Commission’s Joint Research Center. This tool takes into account the location and orientation of the solar panels, the losses in converting to electricity, as well as weather and other atmospheric data. This results in high-quality data that is very representative of real solar panels’ energy output. As for the other datasets, we always consider five years of data, from 2015 to 2019, using the first 4 years for training and the last one for testing.

The simulation environment is implemented using the JAX (Bradbury et al., 2018) library for efficient JIT-compiled computation and Flax (Heek et al., 2024) for neural network modeling. In this setup, ERNESTO simulates the behavior of each BESS in the active nodes. Each active node’s architecture mirrors that described in Section 9.1, with a PV system and a battery storage unit adopting the same technology and specifications. Hence, the configuration of ERNESTO is also similar to that presented in Section 9.1. Notably, in this setting, we need to run multiple instances of ERNESTO in parallel, one per active node, to simulate the behavior of the entire REC, and this required some modification of the simulator’s codebase to enable efficient parallel execution. More details are provided in Appendix D.

12.1.2 Baseline Strategies

To evaluate our method against other baselines, we first need to fix the behavior of agents at the node level. Building on results of Chapter 9, we train node agents with PPO (Schulman et al., 2017b) algorithm, keeping its hyperparameters fixed across all experiments and baselines to isolate the performance of the REC agent’s policy. For the REC agent, instead, we explore different training strategies. In particular, we compare INAIA against the following baselines:

- *PPO*: The REC agent is trained using PPO as for the node agents, but with the task of optimize the distribution of incentives to the REC participants.
- *Rule-based*: The REC agent rewards the households proportionally to their contributions to the current self-consumption. Hence, for each agent $n \in \mathcal{N}$, the component of the REC’s action becomes:

$$a_t^{\mathcal{C}} = \begin{cases} \frac{1}{N} & \text{if } p_t^+ = 0 \vee p_t^- = 0 \\ \frac{p_{n,t}^+}{p_t^+} & \text{if } p_t^+ \leq p_t^- \\ \frac{p_{n,t}^-}{p_t^-} & \text{if } p_t^+ > p_t^- \end{cases} \quad (12.1)$$

Notice that the first branch is needed only to avoid division by zero when the self-consumption is null. In this case, also the REC agent’s reward will be zero.

- **LOLA**: The REC agent is trained using LOLA (Foerster et al., 2018) to anticipate the learning step of node agents.

These baselines are compared with our proposed method. As a meta-learning algorithm, INAIA can incorporate different design choices and algorithmic structures to pursue varying objectives. In particular, in our experimental evaluation, we consider two INAIA-based implementations: INAIA-SHARED and INAIA-OPT, which differ in how the REC agent’s policy network processes input data from individual households. Specifically:

- **INAIA-SHARED**: The REC agent’s network uses a single, shared set of parameters to process all household data, i.e., $\hat{\theta}_n = \hat{\theta}$ for each $n \in \mathcal{N}_A$. This design maintains a constant parameter count regardless of community size, thus offering better scalability.
- **INAIA-OPT**: The REC agent assigns distinct parameters to each household, enabling greater modeling flexibility and potentially enhanced performance.

A more detailed explanation of these architectural choices for the REC agent, along with the policy network used for the node agents, is provided in Appendix D.3.

12.1.3 Evaluation Metrics

To evaluate the effectiveness of the proposed solutions, we rely primarily on two performance metrics: *self-consumption* in Equation (10.4), which quantifies the community-level utility, and *social welfare* from Equation (10.3), which indicates the aggregate financial benefit to individual REC members. In particular, in the following sections, we compare the best-performing models of the different strategies, evaluated at the training iteration with maximum social welfare. Social welfare was selected as the primary validation metric to choose the most performing models because the REC controller is designed to enforce a strategy that guarantees economic benefits for the nodes; without such guarantees, self-interested prosumers would not be incentivized to join the community.

While self-consumption and social welfare capture operational and economic efficiency, they overlook the distributional equity of the system. To address this, we introduce *fairness* as a supplementary evaluation criterion. In our context, fairness denotes the REC agent’s ability to equitably distribute financial incentives among the independent node agents (Reuel and Ma, 2024; Hughes et al., 2018; Jiang and Lu, 2019). In multi-agent environments, fairness typically emerges as a competing objective to global utility maximization. Indeed, purely performance-driven policies tend to learn unbalanced allocation strategies that maximize the global objective at the expense of individual equity.

Quantifying fairness is critical for real-world REC deployments, as perceived inequity directly influences a prosumer’s willingness to join or remain in the community. Although our proposed algorithm does not explicitly encode fairness constraints within its reward structure, it is essential to monitor its impact. To do so, we assess the fairness of the resulting policies using the Gini coefficient (Gini, 1912) and Jain’s index (Jain et al., 1984), computed over the cumulative annual incentives distributed to the agents.

12.2 Case Studies

In this section, we present the experiments conducted to validate the proposed approach. Specifically, we compare INAIA against a set of reference baselines. As said, the node agents are trained using PPO to ensure that performance differences are attributable solely to the behavior of the REC agent. We evaluate our method through two types of experiments that address different scenarios. The first investigates the system’s scalability as the number of active nodes increases. The second focuses on a representative heterogeneous REC configuration, allowing for a deeper analysis of both training dynamics and final performance. In this latter scenario, we also perform an ablation study.

12.2.1 First Scenario: Scaling Number of Node Agents

In the first experiment, we evaluate the scalability of the two INAIA variants and the baseline algorithms across increasingly complex network configurations. We consider an REC with $|\mathcal{N}_A| \in \{3, 5, 8\}$ active nodes and exclude passive nodes to isolate the impact of active control strategies during the performance evaluation. Assuming all REC members reside in the same geographic area, they share identical irradiance conditions and PV configurations (i.e., south-facing solar panels with a nominal peak power of 3 kW). Furthermore, all algorithms are evaluated using an identical set of randomly sampled energy demand profiles to guarantee a fair comparison.

In Figure 12.1, we report the learning process of the two INAIA versions and the baselines across the three configurations. The left column shows community self-consumption, which characterizes the learning progress of the REC agent’s policy, while the right column displays social welfare, the main metric for evaluating the joint satisfaction of the REC participants. The curves represent the mean performance over five runs, evaluated on seeded test profiles during the training phase. As observed, both INAIA methods demonstrate superior learning capabilities compared to the baselines, in both self-consumption and social welfare. Although their improvement in social welfare is relatively modest (at most +13.1% of INAIA-OPT in the setting with 8 agents with respect to the best baseline), they significantly outperform the baselines in self-consumption, which dictates the REC’s financial incentives, and it is the primary driver of cooperative benefits. Specifically, with respect to the best baseline, namely the *rule-based* strategy, in the setting with 3 agents INAIA-SHARED gets an average improvement of +10.63 €, which increases to +98.26 € with 8 agents. INAIA-OPT performs even better: the average improvement in self-consumption goes from +14.47 € with 3 agents to +128.74 € with 8 agents. Notably, the performance gap between the INAIA algorithms and the baselines significantly widens as the number of active nodes increases, demonstrating their aptitude to manage growing environmental complexity and scaling to larger REC configurations.

These results are quantitatively summarized in Table 12.1. The values denote the variation in yearly social welfare (left) and self-consumption (right) averaged over five years of testing, with respect to our INAIA-OPT algorithm, which demonstrated the best overall performance on average. It is worth noting that the variance present in these results stems primarily from the heterogeneous demand profiles, which inherently lead to significantly different self-consumption levels and, consequently, varying incentives

and welfare distributions. A detailed breakdown of energy consumption across the testing years is provided in Figure D.1 in Appendix D.2. Furthermore, to validate the robustness of the proposed approach, we conduct statistical significance testing across the experimental seeds. Our specific, a priori goal was to evaluate whether INAIA-OPT consistently outperformed the other methods. Hence, we conducted pairwise comparisons of its performance against the other strategies using one-tailed paired t-tests and applied the Bonferroni correction to correct for multiple hypotheses. With a standard significance level of $\alpha = 0.05$ and four planned comparisons, the corrected threshold for statistical significance was set strictly at $p < 0.0125$. Under these conditions, INAIA-OPT demonstrated statistically significant improvements over the other algorithms both in social welfare and self-consumption.

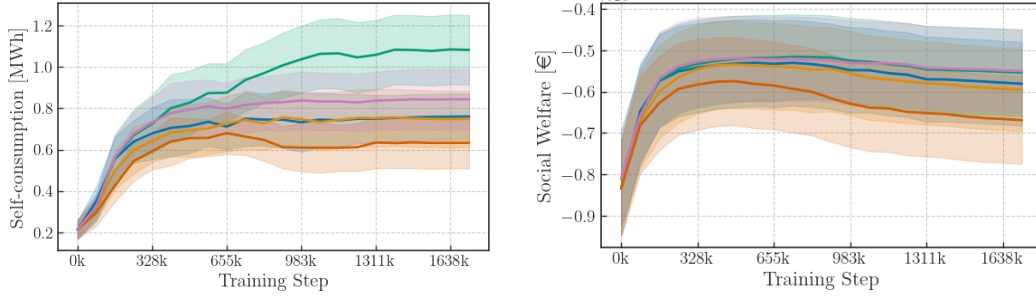
Comparing the two INAIA versions in Figure 12.1, INAIA-OPT achieves superior performance, as expected, due to its greater flexibility in allocating incentives to the node agents. However, INAIA-SHARED may represent a more practical solution in deployments involving a large number of households. In such cases, the performance gains offered by INAIA-OPT may not justify the increased computational and memory overhead associated with maintaining a separate set of parameters for each node agent. Among the baselines, the best strategy is the *rule-based* one, suggesting that a straightforward application of MARL without exploiting domain-specific structure may lead to suboptimal results. Indeed, both PPO and LOLA struggle to converge even at the proportional *rule-based* policy, since they cannot properly deal with the credit assignment problem.

Table 12.1: Comparison of the baselines with our proposed solution INAIA-OPT in the first scenario. We consider tasks with increasing complexity due to the growing number of agents. The values represent the yearly variation averaged over five runs of testing with respect to our method, along with relative standard deviation. For each comparison, we consider the best-performing trained models.

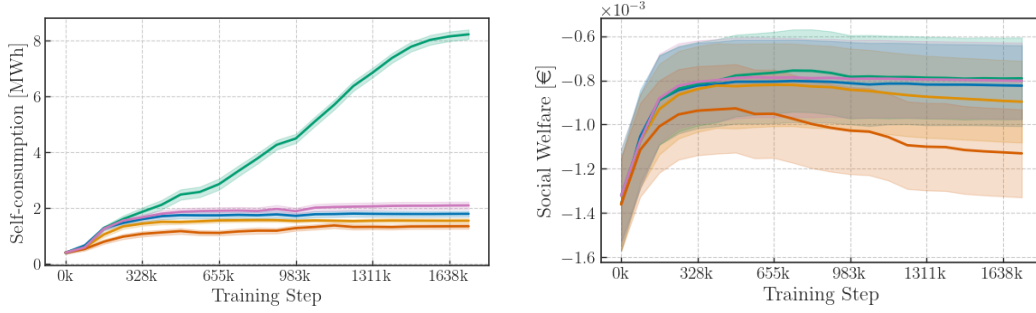
Algorithm	Social Welfare [€]			Self-Consumption [MWh]		
	3	5	8	3	5	8
RULE-BASED	-14.47 ± 3.64	-47.13 ± 3.43	-128.74 ± 31.59	-0.20 ± 0.05	-1.57 ± 0.12	-14.82 ± 0.42
PPO	-18.85 ± 3.99	-64.68 ± 4.97	-172.20 ± 36.03	-0.24 ± 0.05	-1.77 ± 0.15	-15.00 ± 0.52
LOLA	-59.16 ± 6.79	-171.28 ± 20.50	—*	-0.28 ± 0.08	-2.15 ± 0.19	—*
INAIA-SHARED	-3.84 ± 2.20	-30.48 ± 8.00	-88.96 ± 29.13	-0.14 ± 0.04	-1.43 ± 0.15	-7.56 ± 0.40

* Results are missing due to insufficient RAM availability, which prevents the completion of the model training.

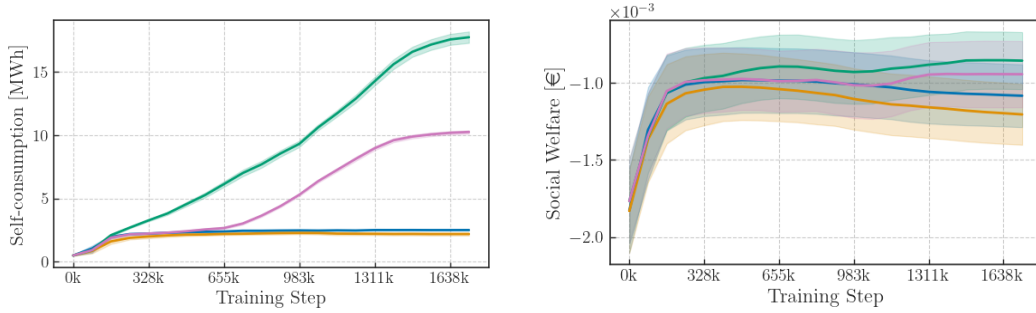
Finally, we evaluate the fairness of the considered strategies in Table 12.2. Overall, the *rule-based* approach naturally enforces the strictest equality, as it deterministically rewards agents proportionally to their contributions. Conversely, the learning-based approaches exhibit high variance and largely unpredictable incentive distributions. Without an explicit fairness constraint, these agents learn to greedily allocate incentives to whichever behaviors maximize global self-consumption. As demonstrated in Table 12.1, learning-based strategies yield greater efficiency for the overall community, but fundamentally rely on unequal resource distribution among members. Addressing this trade-off between global efficiency and individual fairness remains an open problem toward the design of sustainable, socially viable energy communities.



(a) Learning curves in the 3-agent configuration.



(b) Learning curves in the 5-agent configuration.



(c) Learning curves in the 8-agent configuration.

Figure 12.1: Comparison of learning curves for the two INAIA versions and the baselines in the three configurations with increasing number of agents. The left column shows the evolution of self-consumption, while the right column displays the social welfare. The curves represent the mean performance over five runs, with shaded areas indicating 95% confidence intervals.

12.2.2 Second Scenario: Demonstrating the Convenience of Joining the REC

In the second case study, we analyze the incentives for households to join an REC, acknowledging that participation may sometimes be perceived as unattractive due to the occasional need for altruistic behavior. In this case, we consider an REC composed of three active nodes and one passive node. Each active household is equipped with a 3 kW PV system, but the panels are oriented differently: one 60° East-facing (favoring morning

Table 12.2: Fairness evaluation of allocated incentives across 3, 5, and 8 agents. Gini Coefficient (lower is fairer) and Jain’s Index (higher is fairer) are reported as mean $\pm$ standard deviation across 5 seeds. Best results for each scale are highlighted in bold.

Algorithm	3 Agents		5 Agents		8 Agents	
	Gini ($\downarrow$)	Jain’s ($\uparrow$)	Gini ($\downarrow$)	Jain’s ($\uparrow$)	Gini ($\downarrow$)	Jain’s ($\uparrow$)
RULE-BASED	0.0802 $\pm$ 0.0166	0.9744 $\pm$ 0.0111	0.1425 $\pm$ 0.0553	0.9254 $\pm$ 0.0588	0.1541 $\pm$ 0.0394	0.9152 $\pm$ 0.0367
PPO	0.3632 $\pm$ 0.0190	0.6867 $\pm$ 0.0240	0.2954 $\pm$ 0.0267	0.7745 $\pm$ 0.0305	0.3250 $\pm$ 0.0050	0.7510 $\pm$ 0.0066
LOLA	0.2020 $\pm$ 0.0585	0.8622 $\pm$ 0.0725	0.4226 $\pm$ 0.0248	0.6114 $\pm$ 0.0372	–	–
INAIA-OPT	0.1135 $\pm$ 0.0682	0.9456 $\pm$ 0.0522	0.4004 $\pm$ 0.0455	0.6359 $\pm$ 0.0688	0.2111 $\pm$ 0.0162	0.8389 $\pm$ 0.0218
INAIA-SHARED	0.1369 $\pm$ 0.0674	0.9244 $\pm$ 0.0474	0.1247 $\pm$ 0.0394	0.9472 $\pm$ 0.0350	0.4325 $\pm$ 0.0304	0.5817 $\pm$ 0.0377

production), one South-facing (producing most around noon), and one 60° West-facing (favoring late afternoon production). The passive node has no energy generation systems. The difference in panel orientation leads to naturally shifted production peaks across the day, encouraging a form of temporal specialization in energy storage behavior.

To evaluate households’ convenience in joining an REC, we compare the agents’ performance with and without participation in the community. We denote as $r_{n,t}^{-\text{REC}}$ the instantaneous reward obtained by an independent node n not joining the REC at time t ($R_n^{-\text{REC}}$ for the relative return), while $r_{n,t}$ is used when the node is a regular community member (with return R_n). In Figure 12.2a, we plot the mean yearly returns $R_n^{-\text{REC}}$ during training (averaged over five years), obtained by the agents when the REC is not in place, i.e., when nodes receive no incentives to deviate from their local self-interested strategy. In this case, their only objective is to maximize their cumulative local reward, which we previously defined as $r_{n,t}^{\text{loc}}$. In Figure 12.2b, we illustrate for each active agent the point-wise variation of local return if they decide to join the REC, defined as $\Delta r_{n,t}^{\text{loc}} := r_{n,t}^{\text{loc}} - r_{n,t}^{-\text{REC}}$. This term represents the difference between the local reward obtained with and without joining the community (i.e., with respect to results in Figure 12.2a). The curves are mostly negative, indicating that agents sacrifice part of their local reward when participating in the REC. This is expected in our incentive-based setting where agents are encouraged to act cooperatively, even if it occasionally conflicts with their narrow self-interest.

Finally, in Figure 12.2c, we display the difference in global reward under the REC, with respect to the case in which the REC is not in place, and we denote this variation as $\Delta r_{n,t} := r_{n,t} - r_{n,t}^{-\text{REC}}$. Here, all curves are positive, indicating that the designed incentive strategy effectively aligns individual and collective interests, making REC membership beneficial for all agents. Notably, every agent has been tested on the same demand and generation profiles throughout the entire training, both in REC and non-REC cases, so the curves are nonetheless comparable. Furthermore, the setting with the REC in place trained with INAIA-OPT obtains an increment in self-consumption of +35.1% with respect to the case without REC and of +17.3% with respect to the case with the REC trained with the *rule-based* baseline.¹ At the same time, always against such baselines, the social welfare increases by +27.7% and +2.3%, respectively.

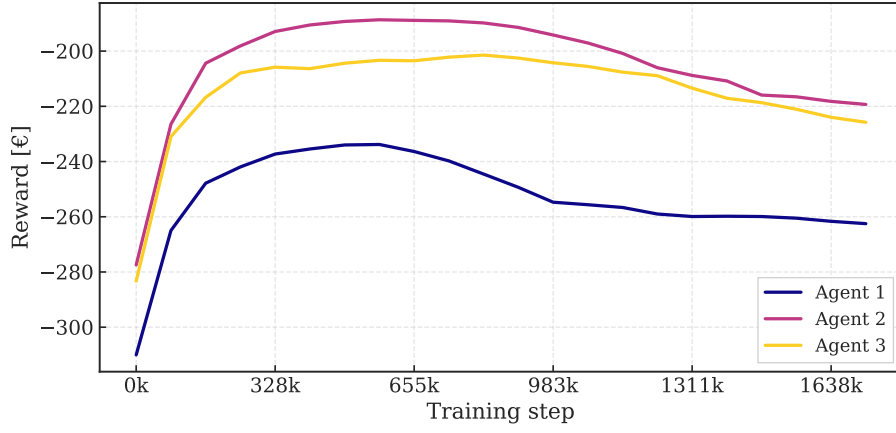
¹We used the *rule-based* strategy as a comparative benchmark in this setting, being the best performing baseline in the first scenario.

12.3 Ablation Study

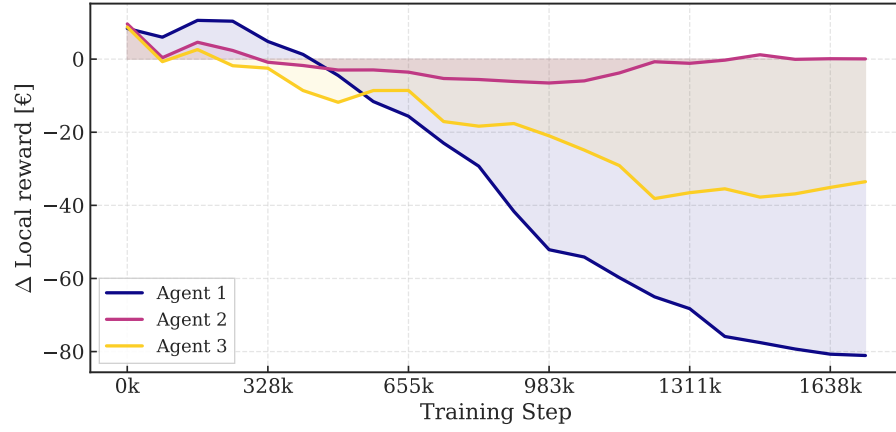
In the second scenario, we conduct an ablation study to further validate the effectiveness of our approach. Specifically, in our proposed method, node agents act deterministically during the REC training phase to reduce the variance in the gradient estimates for the REC agent (Lines 13-16). When this mechanism is ablated by sampling node agents' actions from distributions, the increased variability in the REC agent's gradient estimation negatively affects its learning. Specifically, we compare INAI-A-OPT with INAI-A-OPTND, its ablated version that excludes deterministic node actions during REC updates.

As shown in Figure 12.3a, the self-consumption performance during training for the ablated variant is consistently outperformed by INAI-A-OPT, with the *rule-based* strategy performing worst overall. The benefits of the proposed mechanism are further corroborated by Figure 12.3b, which illustrates the average daily self-consumption profile aggregated over five years of test data. As shown, our solution achieves higher hourly self-consumption than the other methods, demonstrating effective coordination capabilities among node agents.

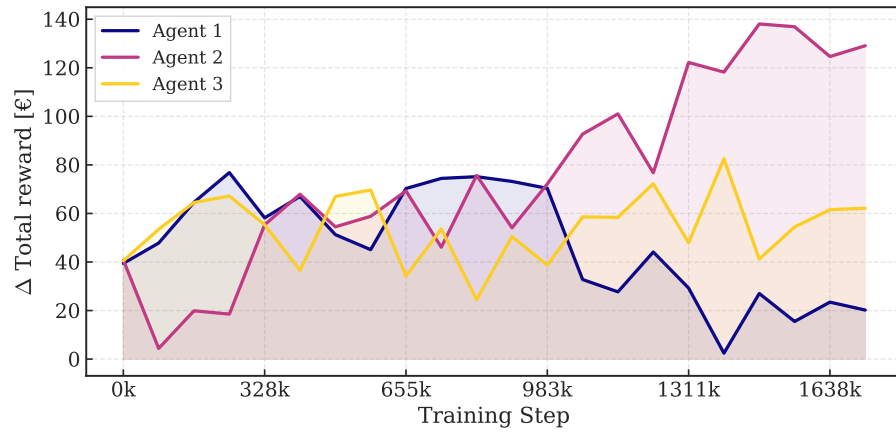
As shown in the above-mentioned plots, INAI-A-OPT improves training stability and enables faster gains in self-consumption than its ablated version. While both variants ultimately converge to similar long-term social welfare outcomes (yielding -2261€ for our method and -2274€ for INAI-A-OPTND aggregated over five years), our approach improves REC self-consumption by $+10.2\%$ and overall learning efficiency. Overall, these findings highlight the strengths of INAI-A: it not only improves individual agent returns through coordinated behavior but also enhances global efficiency by increasing self-consumption.



(a) Return R_n^{-REC} when agents do not join the REC.

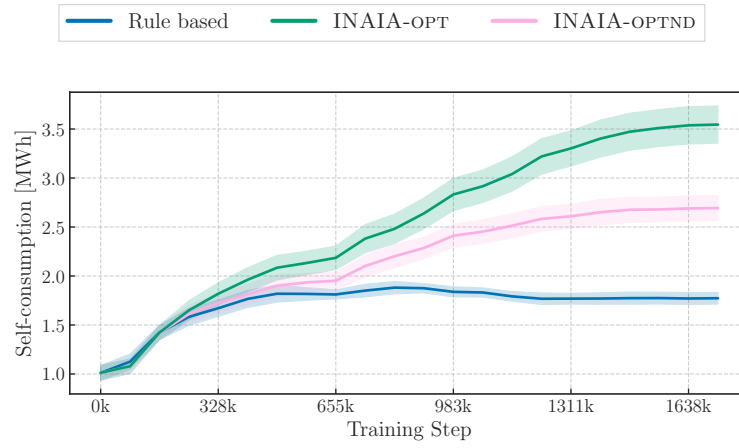


(b) Variation in local return ΔR_n^{loc} w.r.t. not joining the REC for every agent.

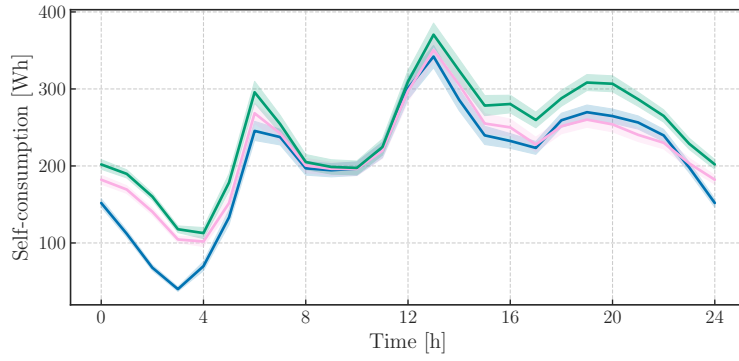


(c) Variation in total return ΔR_n w.r.t. not joining the REC for every agent.

Figure 12.2: Comparison of the yearly return of setting without and with REC agent. While Figure 12.2a shows the return of the agents that do not participate in the community, Figure 12.2b and Figure 12.2c show the impact of the REC agent on local and total rewards. Confidence intervals have been omitted since they would be very wide due to the high heterogeneity of the demand profiles.



(a) Learning curves during training with 95% confidence intervals.



(b) Daily self-consumption with 95% confidence intervals.

Figure 12.3: In the ablation study, we compare INAIA-OPT against INAIA-OPTND, its variant without deterministic node actions during REC updates. Figure 12.3a shows the self-consumption during training, while Figure 12.3b illustrates the average daily self-consumption profile over five years of test data.

CHAPTER 13

Discussion and Conclusion

In this dissertation, we introduced and explored the concept of digital twins (DTs) for battery management and energy systems, with a particular focus on the integration of machine learning (ML) techniques for real-time monitoring and control. The research was structured into four main parts, adopting a *micro-to-macro* perspective to systematically address the challenges and opportunities in this domain, starting from the battery cell level and extending to the broader context of microgrid (MGs) and renewable energy communities (RECs).

In Part I, we laid the foundational concepts and architecture of DTs, presenting a novel framework that constitutes the pillar of our subsequent research. This DT was rigorously validated against experimental data, demonstrating its capability to accurately replicate the physical system's behavior under various operating conditions.

In Part II, we tackled the challenge of enabling this framework for real-time system monitoring and adaptation to dynamic environments. We developed and evaluated ML integration strategies that allow the DT to adjust its parameters and predictions in response to shifting operational conditions, thereby enabling the lifetime tracking of battery state and performance.

Building upon this, Part III expanded the scope to MG networks. We adopted reinforcement learning (RL) to optimize energy management strategies and mitigate the uncertainties inherent in renewable energy sources and load demands. We provided a comprehensive analysis of the problem, formulated as a Markov Decision Process (MDP), and proposed a tailored algorithmic solution that leverages the simulation capabilities of the DT for training and evaluation.

Finally, in Part IV, we extended the RL framework to a multi-agent setting, addressing the challenges of coordination and cooperation among multiple actors within an REC. We developed a multi-agent reinforcement learning (MARL) algorithm that enables

individual agents to learn optimal, self-interested policies while simultaneously fostering a collaborative approach to energy management, ultimately leading to improved overall system performance and sustainability.

This research contributes to the growing body of knowledge on DTs and their applications in energy systems, particularly in the context of energy storage and management. In the following sections, we review the core contributions of this dissertation and discuss potential avenues for future research.

13.1 Design and Validation of the Digital Twin Framework

We design and validate the novel ERNESTO DT. Specifically, we develop a high-fidelity electro-thermal model of a lithium-ion battery cell, which serves as the core component of the DT, including aging effects and degradation mechanisms. We validate the DT against experimental data collected from a real-world battery cell employing the RSE testbed facility under various operating conditions. We demonstrate the DT's ability to accurately replicate the physical system's behavior, including voltage, temperature, and state of charge dynamics, as well as its capability to capture aging effects over time.

Future Work. While the DT framework is designed to be modular and extensible, allowing for the integration of additional components and functionalities, future works could focus on expanding the DT to include more complex battery chemistries, such as solid-state batteries or lithium-sulfur batteries, which may require different modeling approaches and parameter identification techniques. Moreover, the DT could integrate data-driven models, such as ML-based surrogate models, to enhance generalization capabilities to different battery technologies and operating conditions, while maintaining computational efficiency. Finally, the DT could be extended to include additional physical phenomena, such as mechanical stress and strain, which can impact battery performance and degradation, particularly in applications with high mechanical loads.

13.2 Machine Learning for Real-Time Monitoring and Adaptation

In Part II, we integrate ERNESTO with ML techniques to enable real-time monitoring and adaptation of the battery system. After introducing concepts related to drift detection and online learning, we propose and evaluate an evolving cluster-based strategy that enables battery state tracking and allows interpretability of the learned models. By doing this, we trade off a bit of responsiveness with interpretability, which is crucial for the adoption of such techniques in safety-critical applications like battery management. We demonstrate the effectiveness of such a method on both synthetic and real-world experiments, showing its ability to adapt to changing conditions and maintain accurate predictions of the battery state over time.

Future Work. A first research direction could point to improve and speed up the detection of abrupt drifts by integrating rapid change-point detection algorithms (e.g., CUSUM) to trigger an emergency adaptation mode for safety-critical shifts. Another future work could explore the integration of fault detection systems that, leveraging the provided approach, could identify and classify different types of faults in the battery system, such as internal short circuits, thermal runaway, or capacity fade. Additionally,

the introduction of isolation mechanisms can be investigated to localize the source of the detected drift, allowing for targeted interventions and maintenance actions. Moreover, the integration of predictive maintenance strategies could be explored, where the DT and the online learning approach are used to predict the remaining useful life of the battery and schedule maintenance activities accordingly, optimizing the overall system performance and reducing downtime.

13.3 Microgrid Energy Management through Reinforcement Learning

To expand the scope of our research, we investigate the energy management problem in MG networks. Leveraging the DT framework, we build a simulation environment that allows us to model the dynamics of an MG, including the interactions between different components such as renewable energy sources, energy storage systems, and loads. We formulate the energy management problem as an MDP and propose an RL algorithm that learns optimal energy management strategies through interaction with the simulated environment. We conducted an extensive evaluation of the proposed approach, demonstrating its ability to trade off between battery degradation and energy costs, in order to optimize the overall performance of the MG. We overall identified RL as a promising approach for MG energy management, particularly in the presence of uncertainties and dynamic conditions, which are common in real-world MG applications.

Future Work. Future research could explore the integration of other objectives into the RL framework, such as minimizing carbon emissions, as well as incorporating additional dispatchable loads, such as electric vehicles or flexible industrial processes which can provide additional flexibility to the energy management problem, and power line capacity constraints, integrating local network bottlenecks. This would complicate the problem, providing a more realistic scenario, but without changing the overall approach. Another direction is purely algorithmic, i.e., design RL methods that are inherently tailored to the specific characteristics of MG energy management, such as based on Exo-MDPs (Sinclair et al., 2023; Maran et al., 2026), addressing also non-stationarity and multi-objective optimization challenges.

13.4 Multi-Agent Control for Renewable Energy Communities

In the last part of this dissertation, we further expand our field of view to the broader context of RECs, where multiple agents, such as households or industries, interact and cooperate to manage their energy resources. After framing the problem as a variation of a Partially Observable Stochastic Game, we propose a MARL algorithm that leverages reward shaping coming from a central coordinator to foster cooperation among individual agents, while allowing them to learn optimal selfish policies. We evaluate the proposed approach employing real-world data and the DT framework to model the dynamics of single battery units, demonstrating its ability to improve overall system performance and sustainability through cooperative energy management strategies.

Future Work. While our approach is inherently designed to encourage REC participants to collaborate and specialize in complementary roles, it naturally results in varying incentive distributions across nodes. This behavior arises from the absence of

explicit constraints on allocation policies, allowing the algorithm to prioritize agents that demonstrate higher responsiveness to incentives. Although such flexibility enhances the system's overall efficiency, it may raise concerns regarding perceived fairness among community members. To address this, future works could investigate the integration of *fairness* mechanisms aimed at promoting a more balanced and equitable distribution of incentives, aligning technical performance with social acceptability. Furthermore, future research might investigate more complex and heterogeneous scenarios, encompassing prosumers of varying scales and energy profiles, and considering different geographical contexts. Future works should seek to establish rigorous theoretical guarantees regarding the structural capabilities of RECs. Specifically, it is crucial to investigate the boundary conditions of this decentralized architecture, such as identifying the critical threshold (or minimum ratio) of active to passive prosumers necessary to yield a net positive benefit for the community. Insights derived from these analyses could inspire the design of novel organizational frameworks that share the overarching sustainability and efficiency goals of current REC structures. Finally, a particularly interesting analysis is the validation of the learned strategies by deploying them within a controlled physical testbed.

Bibliography

- S. N. Adasah, Z. Wang, S. Hu, S. Capezza, J. Shao, and M.-Y. Chow. Review of fault diagnosis based protection mechanisms for battery energy storage systems. In *2024 IEEE 33rd International Symposium on Industrial Electronics (ISIE)*, pages 1–6. IEEE, 2024.
- A. Agarwal, N. Jiang, and S. M. Kakade. Reinforcement learning: Theory and algorithms. 2019.
- S. Ahleroff, X. Xu, R. Y. Zhong, and Y. Lu. Digital twin as a service (dtaas) in industry 4.0: An architecture reference model. *Advanced Engineering Informatics*, 47:101225, 2021. ISSN 1474-0346.
- S. Aittahar, A. Bolland, G. Derval, and D. Ernst. Optimal Control of Renewable Energy Communities subject to Network Peak Fees with Model Predictive Control and Reinforcement Learning Algorithms, Feb. 2024.
- F. S. Al-Ismael. Dc microgrid planning, operation, and control: A comprehensive review. *IEEE Access*, 9:36154–36172, 2021.
- S. V. Albrecht, F. Christianos, and L. Schäfer. *Multi-agent reinforcement learning: Foundations and modern approaches*. 2024.
- C. Alippi, M. Roveri, and F. Trovò. A self-building and cluster-based cognitive fault diagnosis system for sensor networks. *IEEE Transactions on Neural Networks and Learning Systems*, 25(6):1021–1032, 2014.
- N. Amruthnath and T. Gupta. A research study on unsupervised machine learning algorithms for early fault detection in predictive maintenance. In *2018 5th international conference on industrial engineering and applications (ICIEA)*, pages 355–361. IEEE, 2018.
- K. Baker, J. Guo, G. Hug, and X. Li. Distributed mpc for efficient coordination of storage and renewable energy sources across control areas. *IEEE Transactions on Smart Grid*, 7(2):992–1001, 2016.
- M. Barbieri, M. Ceraolo, G. Lutzemberger, C. Scarpelli, T. Pessa, and M. Giovannucci. Simplified electro-thermal model for lithium cells based on experimental tests. In *2020 AEIT International Conference of Electrical and Electronic Technologies for Automotive (AEIT AUTOMOTIVE)*, pages 1–6, 2020.
- M. Barbieri, M. Ceraolo, G. Lutzemberger, and C. Scarpelli. An electro-thermal model for lfp cells: Calibration procedure and validation. *Energies*, 15(7), 2022. ISSN 1996-1073.
- S. Barcellona, S. Colnago, G. Dotelli, S. Latorrata, and L. Piegari. Aging effect on the variation of li-ion battery resistance as function of temperature and state of charge. *Journal of Energy Storage*, 50:104658, 2022a. ISSN 2352-152X.
- S. Barcellona, S. Colnago, P. Montrasio, and L. Piegari. Integrated electro-thermal model for li-ion battery packs. *Electronics*, 11(10), 2022b. ISSN 2079-9292.

Bibliography

- S. Barcellona, L. Codecasa, S. Colnago, and L. Piegari. Cycle aging effect on the open circuit voltage of lithium-ion battery. In *2023 IEEE International Conference on Electrical Systems for Aircraft, Railway, Ship Propulsion and Road Vehicles and International Transportation Electrification Conference (ESARS-ITEC)*, pages 1–6, 2023.
- A. Barré, B. Deguilhem, S. Grolleau, M. Gérard, F. Suard, and D. Riu. A review on lithium-ion battery ageing mechanisms and estimations for automotive applications. *Journal of power sources*, 241:680–689, 2013.
- B. R. Barricelli, E. Casiraghi, and D. Fogli. A survey on digital twin: Definitions, characteristics, applications, and design implications. *IEEE Access*, 7:167653–167671, 2019.
- M. A. Basit, S. Dilshad, R. Badar, and S. M. Sami ur Rehman. Limitations, challenges, and solution approaches in grid-connected renewable energy systems. *International Journal of Energy Research*, 44(6):4132–4162, 2020.
- J. Baxter and P. L. Bartlett. Infinite-horizon policy-gradient estimation. *Journal of Artificial Intelligence Research*, 15:319–350, Nov. 2001. ISSN 1076-9757.
- H. Berenji. Fuzzy q-learning: a new approach for fuzzy dynamic programming. In *Proceedings of 1994 IEEE 3rd International Fuzzy Systems Conference*, pages 486–491 vol.1, 1994.
- C. Berner, G. Brockman, B. Chan, V. Cheung, P. Dębiak, C. Dennison, D. Farhi, Q. Fischer, S. Hashme, C. Hesse, et al. Dota 2 with large scale deep reinforcement learning. *arXiv preprint arXiv:1912.06680*, 2019.
- A. Bidram and A. Davoudi. Hierarchical structure of microgrids control system. *IEEE Transactions on Smart Grid*, 3(4):1963–1976, 2012.
- A. Bifet and R. Gavalda. Learning from time-changing data with adaptive windowing. In *Proceedings of the 2007 ACM Symposium on Applied Computing*, pages 1206–1210. ACM, 2007.
- C. M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer, 1 edition, 2007. ISBN 0387310738.
- B. Bole, C. Kulkarni, and K. Goebel. Randomized battery usage data set. <https://data.nasa.gov/>, 2022.
- F. Bonassi, A. La Bella, R. Lazzari, C. Sandroni, and R. Scattolini. Supervised control of hybrid ac-dc grids for power balance restoration. *Electric Power Systems Research*, 196:107107, 2021.
- J. Bradbury, R. Frostig, P. Hawkins, M. J. Johnson, C. Leary, D. Maclaurin, G. Necula, A. Paszke, J. VanderPlas, S. Wanderman-Milne, and Q. Zhang. JAX: composable transformations of Python+NumPy programs, 2018. URL <http://github.com/jax-ml/jax>.
- F. Cadini, C. Sbarufatti, F. Cancelliere, and M. Giglio. State-of-life prognosis and diagnosis of lithium-ion batteries by data-driven particle filters. *Applied Energy*, 235:661–672, 2019. ISSN 0306-2619.
- S. Cai, Y. Xie, Q. Wu, and Z. Xiang. Robust mpc-based microgrid scheduling for resilience enhancement of distribution system. *International Journal of Electrical Power & Energy Systems*, 121:106068, 2020.
- Center for Advanced Life Cycle Engineering (CALCE). Battery research data. <https://calce.umd.edu/battery-data>, 2010–2024.
- V. Chandola, A. Banerjee, and V. Kumar. Anomaly detection: A survey. *ACM computing surveys (CSUR)*, 41(3): 1–58, 2009.
- W.-Y. Chang. The state of charge estimating methods for battery: A review. *International Scholarly Research Notices*, 2013(1):953792, 2013.
- B. Chen, J. Wang, X. Lu, C. Chen, and S. Zhao. Networked microgrids for grid resilience, robustness, and efficiency: A review. *IEEE Transactions on Smart Grid*, 12(1):18–32, 2020.
- T. Chen and S. Bu. Realistic peer-to-peer energy trading model for microgrids using deep reinforcement learning. In *2019 IEEE PES Innovative Smart Grid Technologies Europe (ISGT-Europe)*, pages 1–5, 2019.
- B. Chreim, M. Esseghir, and L. Merghem-Boulahia. Energy management in residential communities with shared storage based on multi-agent systems: Application to smart grids. *Engineering Applications of Artificial Intelligence*, 126:106886, Nov. 2023. ISSN 0952-1976.

- W. C. Clarke, C. Manzie, and M. J. Brear. An economic mpc approach to microgrid control. In *2016 Australian Control Conference (AuCC)*, pages 276–281. IEEE, 2016.
- W. C. Clarke, C. Manzie, and M. J. Brear. Hierarchical economic mpc for systems with storage states. *Automatica*, 94:138–150, 2018.
- C. Claus and C. Boutilier. The dynamics of reinforcement learning in cooperative multiagent systems. *AAAI/IAAI*, 1998(746-752):2, 1998.
- X. Cui, J. Zeng, H. Zhang, J. Yang, J. Qiao, J. Li, and W. Li. Optimization of the lumped parameter thermal model for hard-cased li-ion batteries. *Journal of Energy Storage*, 32:101758, 2020a. ISSN 2352-152X.
- X. Cui, J. Zeng, H. Zhang, J. Yang, J. Qiao, J. Li, and W. Li. Optimization of the lumped parameter thermal model for hard-cased li-ion batteries. *Journal of Energy Storage*, 32:101758, 2020b. ISSN 2352-152X.
- O. Da Silva, F. Caio, A. Dabiri, and B. De Schutter. Integrating reinforcement learning and model predictive control for mixed-logical dynamical systems. *IEEE Open Journal of Control Systems*, 4:316–331, 2025.
- O. Das, M. H. Zafar, F. Sanfilippo, S. Rudra, and M. L. Kolhe. Advancements in digital twin technology and machine learning for energy systems: A comprehensive review of applications in smart grids, renewable energy, and electric vehicle optimisation. *Energy Conversion and Management: X*, 24:100715, 2024.
- C. S. de Witt, T. Gupta, D. Makoviichuk, V. Makoviychuk, P. H. S. Torr, M. Sun, and S. Whiteson. Is Independent Learning All You Need in the StarCraft Multi-Agent Challenge?, Nov. 2020.
- G. Ditzler, M. Roveri, C. Alippi, and R. Polikar. Learning in nonstationary environments: A survey. *IEEE Computational intelligence magazine*, 10(4):12–25, 2015.
- D. Domínguez-Barbero, J. García-González, M. A. Sanz-Bobi, and E. F. Sánchez-Úbeda. Optimising a microgrid system by deep reinforcement learning techniques. *Energies*, 13(11), 2020. ISSN 1996-1073.
- Y. Du and F. Li. Intelligent multi-microgrid energy management based on deep neural network and model-free reinforcement learning. *IEEE Transactions on Smart Grid*, 11(2):1066–1076, 2020.
- O. Ellabban, H. Abu-Rub, and F. Blaabjerg. Renewable energy resources: Current status, future prospects and their enabling technology. *Renewable and sustainable energy reviews*, 39:748–764, 2014.
- J. A. Erkoyuncu, I. F. del Amo, D. Ariensyah, D. Bulka, R. Roy, et al. A design framework for adaptive digital twins. *CIRP annals*, 69(1):145–148, 2020.
- D. Ernst, P. Geurts, and L. Wehenkel. Tree-based batch mode reinforcement learning. *Journal of Machine Learning Research*, 6(18):503–556, 2005.
- I. Errandonea, S. Beltrán, and S. Arrizabalaga. Digital twin for maintenance: A literature review. *Computers in Industry*, 123:103316, 2020. ISSN 0166-3615.
- A. B. Esan and H. Shareef. Augmented deep reinforcement learning for the energy management of microgrids considering renewable stochastic parameters. *Engineering Applications of Artificial Intelligence*, 162:112785, 2025. ISSN 0952-1976.
- M. Ester, H.-P. Kriegel, J. Sander, X. Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *kdd*, volume 96, pages 226–231, 1996.
- European Commission - Joint Research Centre. Photovoltaic geographical information system, 2022. URL https://re.jrc.ec.europa.eu/pvg_tools/en/.
- European Parliament and European Council. Directive (eu) 2018/2001 of the european parliament and of the council of 11 december 2018 on the promotion of the use of energy from renewable sources, 2018. <https://eur-lex.europa.eu/eli/dir/2018/2001/oj/eng>.
- D. Fioriti, L. Pellegrino, G. Lutzemberger, E. Micolano, and D. Poli. Optimal sizing of residential battery systems with multi-year dynamics and a novel rainflow-based model of storage degradation: An extensive italian case study. *Electric Power Systems Research*, 203, 2022. ISSN 0378-7796.

Bibliography

- J. N. Foerster, R. Y. Chen, M. Al-Shedivat, S. Whiteson, P. Abbeel, and I. Mordatch. Learning with Opponent-Learning Awareness, Sept. 2018.
- E. Foruzan, L.-K. Soh, and S. Asgarpour. Reinforcement learning approach for optimal distributed energy management in a microgrid. *IEEE Transactions on Power Systems*, 33(5):5749–5758, 2018.
- K. Fu, T. Hamacher, and V. S. Perić. Development of self-adaptive digital twin for battery monitoring and management system. *Electric Power Systems Research*, 234:110698, 2024. ISSN 0378-7796.
- T. Gabor, L. Belzner, M. Kiermeier, M. T. Beck, and A. Neitz. A simulation-based architecture for smart cyber-physical systems. In *2016 IEEE International Conference on Autonomic Computing (ICAC)*, pages 374–379, 2016.
- J. Gama, P. Medas, G. Castillo, and P. Rodrigues. Learning with drift detection. In *Brazilian symposium on artificial intelligence*, pages 286–295. Springer, 2004.
- J. Gama, I. Žliobaitė, A. Bifet, M. Pechenizkiy, and A. Bouchachia. A survey on concept drift adaptation. *ACM computing surveys (CSUR)*, 46(4):1–37, 2014.
- A. H. Gandomi, X.-S. Yang, and A. H. Alavi. Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems. *Engineering with computers*, 29(1):17–35, 2013.
- J. Gertler. *Fault detection and diagnosis in engineering systems*. CRC Press, London, England, Sept. 2019.
- Gestore dei Mercati Energetici (GME). Historical data mgp, 2015-2020. Data retrieved from GME: <https://www.mercatoelettrico.org/it/download/DatiStorici.aspx>.
- Z. Ghahramani. *Unsupervised Learning*, pages 72–112. Springer Berlin Heidelberg, Berlin, Heidelberg, 2004. ISBN 978-3-540-28650-9.
- C. Gini. *Variabilità e mutabilità: contributo allo studio delle distribuzioni e delle relazioni statistiche. [Fasc. I.]*. Tipogr. di P. Cuppini, 1912.
- E. Glaessgen and D. Stargel. *The Digital Twin Paradigm for Future NASA and U.S. Air Force Vehicles*. 2012.
- J. Gomez, R. Nelson, E. E. Kalu, M. H. Weatherspoon, and J. P. Zheng. Equivalent circuit model parameters of a high-power li-ion battery: Thermal and state of charge effects. *Journal of Power Sources*, 196(10):4826–4831, 2011. ISSN 0378-7753.
- M. Grieves. *Virtually Perfect: Driving Innovative and Lean Products through Product Lifecycle Management*. 11 2011. ISBN 0982138008.
- M. Grieves. Digital twin: manufacturing excellence through virtual factory replication. *White paper*, 1(2014):1–7, 2014.
- S. Gronauer and K. Diepold. Multi-agent deep reinforcement learning: a survey. *Artificial Intelligence Review*, 55(2): 895–943, 2022.
- L. Guiducci, G. Palma, M. Stentati, A. Rizzo, and S. Paoletti. A Reinforcement Learning approach to the management of Renewable Energy Communities. In *2023 12th Mediterranean Conference on Embedded Computing (MECO)*, pages 1–8, June 2023. ISSN: 2637-9511.
- C. Guo, X. Wang, Y. Zheng, and F. Zhang. Real-time optimal energy management of microgrid with uncertainties based on deep reinforcement learning. *Energy*, 238:121873, 2022. ISSN 0360-5442.
- J. Guo, N. Zhao, L. Sun, and S. Zhang. Modular based flexible digital twin for factory design. *Journal of Ambient Intelligence and Humanized Computing*, 10(3):1189–1200, 2019.
- T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor, 2018.
- C. A. Hans, P. Sopasakis, A. Bemporad, J. Raisch, and C. Reincke-Collon. Scenario-based model predictive operation control of islanded microgrids. In *2015 54th IEEE conference on decision and control (CDC)*, pages 3272–3277. IEEE, 2015.

- E. A. Hansen, D. S. Bernstein, and S. Zilberstein. Dynamic programming for partially observable stochastic games. In *Proceedings of the 19th National Conference on Artificial Intelligence (AAAI)*, volume 4, pages 709–715, 2004.
- W. He, J. Mao, K. Song, Z. Li, Y. Su, Y. Wang, and X. Pan. Structural performance prediction based on the digital twin model: A battery bracket example. *Reliability Engineering & System Safety*, 229:108874, 2023a. ISSN 0951-8320.
- Y. He, Z. Li, J. Zhang, G. Shi, and W. Cao. Day-ahead and intraday multi-time scale microgrid scheduling based on light robustness and mpc. *International Journal of Electrical Power & Energy Systems*, 144:108546, 2023b. ISSN 0142-0615.
- J. Heek, A. Levskaya, A. Oliver, M. Ritter, B. Rondepierre, A. Steiner, and M. van Zee. Flax: A neural network library and ecosystem for JAX, 2024. URL <http://github.com/google/flax>.
- J. Hong, F. Liang, J. Yang, and S. Du. An exhaustive review of battery faults and diagnostic techniques for real-world electric vehicle safety. *Journal of Energy Storage*, 99:113234, 2024. ISSN 2352-152X.
- M. Hossain, S. Saha, M. E. Haque, M. T. Arif, and A. M. T. Oo. A parameter extraction method for the thevenin equivalent circuit model of li-ion batteries. In *2019 IEEE industry applications society annual meeting*, pages 1–7. IEEE, 2019.
- J. Hu and M. P. Wellman. Nash q-learning for general-sum stochastic games. *Journal of machine learning research*, 4(Nov):1039–1069, 2003.
- J. Hu, Y. Shan, Y. Yang, A. Parisio, Y. Li, N. Amjady, S. Islam, K. W. Cheng, J. M. Guerrero, and J. Rodríguez. Economic model predictive control for microgrid optimization: A review. *IEEE Transactions on Smart Grid*, 15(1):472–484, 2023.
- S. Huang, R. F. J. Dossa, A. Raffin, A. Kanervisto, and W. Wang. The 37 implementation details of proximal policy optimization. In *ICLR Blog Track*, 2022.
- E. Hughes, J. Z. Leibo, M. Phillips, K. Tuyls, E. Dueñez-Guzman, A. García Castañeda, I. Dunning, T. Zhu, K. McKee, R. Koster, et al. Inequity aversion improves cooperation in intertemporal social dilemmas. *Advances in neural information processing systems*, 31, 2018.
- T. Huld, R. Müller, and A. Gambardella. A new solar radiation database for estimating PV performance in Europe and Africa. *Solar Energy*, 86(6), June 2012. ISSN 0038-092X.
- R. Isermann. *Fault-diagnosis systems: an introduction from fault detection to fault tolerance*. Springer Science & Business Media, 2005.
- Italian Government. Testo coordinato del decreto-legge 30 dicembre 2019, n. 162, 2019. www.gazzettaufficiale.it/eli/id/2020/02/29/20A01353/sg.
- Italian Government. Decreto 16 settembre 2020, 2020. <https://www.gazzettaufficiale.it/eli/id/2020/11/16/20A06224/sg>.
- Italian Parliament. Legge 28 febbraio 2020, n. 8, 2020. www.gazzettaufficiale.it/eli/id/2020/02/29/20G00021/sg.
- R. K. Jain, D.-M. W. Chiu, W. R. Hawe, et al. A quantitative measure of fairness and discrimination. *Eastern Research Laboratory, Digital Equipment Corporation, Hudson, MA*, 21(1):2022–2023, 1984.
- P. Janiszewski. Rainflow. <https://github.com/iamlikeme/rainflow>, 2022.
- W. Jia, W. Wang, and Z. Zhang. From simple digital twin to complex digital twin part i: A novel modeling method for multi-scale and multi-scenario digital twin. *Advanced Engineering Informatics*, 53:101706, 2022. ISSN 1474-0346.
- J. Jiang and Z. Lu. Learning fairness in multi-agent systems. *Advances in Neural Information Processing Systems*, 32, 2019.
- C. Ju, P. Wang, L. Goel, and Y. Xu. A two-layer energy management system for microgrids with hybrid energy storage considering degradation costs. *IEEE Transactions on Smart Grid*, 9(6):6047–6057, 2017.

Bibliography

- N. Kharlamova, C. Træholt, and S. Hashemi. A digital twin of battery energy storage systems providing frequency regulation. In *2022 IEEE International Systems Conference (SysCon)*, pages 1–7, 2022.
- P. Khumprom and N. Yodo. A data-driven predictive prognostic model for lithium-ion batteries based on a deep learning algorithm. *Energies*, 12(4):660, 2019.
- D. P. Kingma and J. Ba. Adam: A method for stochastic optimization, 2017.
- W. Kolodziejczyk, I. Zoltowska, and P. Cichosz. Real-time energy purchase optimization for a storage-integrated photovoltaic system by deep reinforcement learning. *Control Engineering Practice*, 106:104598, 2021. ISSN 0967-0661.
- W. Kong, Z. Y. Dong, Y. Jia, D. J. Hill, Y. Xu, and Y. Zhang. Short-term residential load forecasting based on lstm recurrent neural network. *IEEE Trans. on Smart Grid*, 10(1):841–851, 2019.
- J. Kosowatz. Energy storage smooths the duck curve. *Mechanical Engineering*, 140(06):30–35, 2018.
- P. Kou, Y. Feng, D. Liang, and L. Gao. A model predictive control approach for matching uncertain wind generation with pev charging demand in a microgrid. *International Journal of Electrical Power & Energy Systems*, 105: 488–499, 2019.
- M. Krarti and M. Aldubyan. Review analysis of covid-19 impact on electricity demand for residential buildings. *Renewable and Sustainable Energy Reviews*, 143:110888, 2021. ISSN 1364-0321.
- W. Kritzinger, M. Karner, G. Traar, J. Henjes, and W. Sihn. Digital twin in manufacturing: A categorical literature review and classification. *16th IFAC Symposium on Information Control Problems in Manufacturing*, 51(11): 1016–1022, 2018. ISSN 2405-8963.
- E. Kuznetsova, Y.-F. Li, C. Ruiz, E. Zio, G. Ault, and K. Bell. Reinforcement learning for microgrid energy management. *Energy*, 59:133–146, 2013. ISSN 0360-5442.
- B.-C. Lai, W.-Y. Chiu, and Y.-P. Tsai. Multiagent Reinforcement Learning for Community Energy Management to Mitigate Peak Rebounds Under Renewable Energy Uncertainty. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 6(3):568–579, June 2022. ISSN 2471-285X.
- R. Lazzari and A. L. Bella. Optimal distributed control of hybrid ac/dc grids. In *2023 International Conference on Clean Electrical Power (ICCEP)*, pages 571–578, 2023.
- I. Legrene, T. Wong, and L.-A. Dessaint. Deep reinforcement learning approach for hybrid renewable energy systems optimization. *Engineering Applications of Artificial Intelligence*, 159:111650, 2025. ISSN 0952-1976.
- F. Leng, C. M. Tan, and M. Pecht. Effect of temperature on the aging rate of li ion battery operating above room temperature. *Scientific reports*, 5(1):12967, 2015.
- H. Li, M. Bin Kaleem, I.-J. Chiu, D. Gao, and J. Peng. A digital twin model for the battery management systems of electric vehicles. In *2021 IEEE 23rd Int Conf on High Performance Computing & Communications; 7th Int Conf on Data Science & Systems; 19th Int Conf on Smart City; 7th Int Conf on Dependability in Sensor, Cloud & Big Data Systems & Application (HPCC/DSS/SmartCity/DependSys)*, pages 1100–1107, 2021.
- H. Li, Z. Liu, M. Bin Kaleem, L. Duan, S. Ruan, and W. Liu. Fault detection for lithium-ion batteries of electric vehicles with spatio-temporal autoencoder. *Applied Energy*, 392:125933, 2025. ISSN 0306-2619.
- W. Li, M. Rentemeister, J. Badedda, D. Jöst, D. Schulte, and D. U. Sauer. Digital twin for battery systems: Cloud battery management system with online state-of-charge and state-of-health estimation. *Journal of energy storage*, 30:101557, 2020.
- W. Li, J. Chen, K. Quade, D. Luder, J. Gong, and D. U. Sauer. Battery degradation diagnosis with field data, impedance-based modeling and artificial intelligence. *Energy Storage Materials*, 53:391–403, 2022. ISSN 2405-8297.
- T. Lillicrap. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.
- K. Y. H. Lim, P. Zheng, and C.-H. Chen. A state-of-the-art survey of digital twin: techniques, engineering product lifecycle management and business innovation perspectives. *Journal of intelligent manufacturing*, 31(6):1313–1337, 2020.

- M. L. Littman. Markov games as a framework for multi-agent reinforcement learning. In *Machine learning proceedings 1994*, pages 157–163. Elsevier, 1994.
- L. Liu, J. Zhu, J. Chen, and H. Ye. Deep reinforcement learning for stochastic dynamic microgrid energy management. In *2021 IEEE 4th International Electrical and Energy Conference (CIEEC)*, pages 1–6, 2021.
- X. Liu, P. Wang, and P. C. Loh. A hybrid ac/dc microgrid and its coordination control. *IEEE Transactions on Smart Grid*, 2(2):278–286, 2011.
- X. Liu, M. Wang, R. Cao, M. Lyu, C. Zhang, S. Li, B. Guo, L. Zhang, Z. Zhang, X. Gao, H. Cheng, B. Ma, and S. Yang. Review of abnormality detection and fault diagnosis methods for lithium-ion batteries. *Automotive Innovation*, 6, 03 2023.
- C. Lo, C. Chen, and R. Y. Zhong. A review of digital twin in product design and development. *Advanced Engineering Informatics*, 48:101297, 2021. ISSN 1474-0346.
- J. P. Lopes, C. L. Moreira, and A. Madureira. Defining control strategies for microgrids islanded operation. *IEEE Transactions on power systems*, 21(2):916–924, 2006.
- J. P. Lopes, N. Hatziaargyriou, J. Mutale, P. Djapic, and N. Jenkins. Integrating distributed generation into electric power systems: A review of drivers, challenges and opportunities. *Electric power systems research*, 77(9): 1189–1203, 2007.
- R. Lowe, Y. I. Wu, A. Tamar, J. Harb, O. Pieter Abbeel, and I. Mordatch. Multi-agent actor-critic for mixed cooperative-competitive environments. *Advances in neural information processing systems*, 30, 2017.
- J. Lu, A. Liu, F. Dong, F. Gu, J. Gama, and G. Zhang. Learning under concept drift: A review. *IEEE transactions on knowledge and data engineering*, 31(12):2346–2363, 2018.
- Z. Lyu, Y. Tang, Z. Wu, L. Wu, and X. Qiang. Online state of health estimation for li-ion batteries in evs through a data-fusion-model method. *Journal of Energy Storage*, 100:113588, 2024. ISSN 2352-152X.
- K. Ma, Z. Yun, X. Zhang, and F. Duan. The calculation method of optimal thevenin equivalent parameters based on the fitting of steady-state response characteristics of nodes. In *2020 IEEE/IAS Industrial and Commercial Power System Asia (I&CPS Asia)*, pages 675–680. IEEE, 2020.
- D. Maran, D. Salaorni, and M. Restelli. Learning in markov decision processes with exogenous dynamics. *arXiv preprint arXiv:2603.02862*, 2026.
- L. Mariam, M. Basu, and M. F. Conlon. Microgrid: Architecture, policy and future trends. *Renewable and Sustainable Energy Reviews*, 64:477–489, 2016.
- F. J. Massey Jr. The kolmogorov-smirnov test for goodness of fit. *Journal of the American statistical Association*, 46 (253):68–78, 1951.
- M. Matsuishi and T. Endo. Fatigue of metals subjected to varying stress. *Japan society of mechanical engineers, Fukuoka, Japan*, 68(2):37–40, 1968.
- D. Q. Mayne. Model predictive control: Recent developments and future promise. *Automatica*, 50(12):2967–2986, 2014.
- G. J. McLachlan. Mahalanobis distance. *Resonance*, 4(6):20–26, 1999.
- L. Meng, Q. Shafiee, G. F. Trecate, H. Karimi, D. Fulwani, X. Lu, and J. M. Guerrero. Review on control of dc microgrids and multiple microgrid clusters. *IEEE journal of emerging and selected topics in power electronics*, 5 (3):928–948, 2017.
- T. M. Mitchell. *Machine learning*, volume 1. McGraw-hill New York, 1997.
- V. Mnih. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.
- V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, Feb. 2015.

Bibliography

- A. Mohammed, S. S. Refaat, S. Bayhan, and H. Abu-Rub. Ac microgrid control and management strategies: Evaluation and review. *IEEE Power Electronics Magazine*, 6(2):18–31, 2019.
- J. G. Moreno-Torres, T. Raeder, R. Alaiz-Rodríguez, N. V. Chawla, and F. Herrera. A unifying view on dataset shift in classification. *Pattern recognition*, 45(1):521–530, 2012.
- F. Moret and P. Pinson. Energy collectives: A community and fairness based approach to future electricity markets. *IEEE Transactions on Power Systems*, 34(5):3994–4004, 2019.
- A. Murillo, R. Taormina, N. O. Tippenhauer, D. Salaorni, R. van Dijk, L. Jonker, S. Vos, M. Weyns, and S. Galelli. High-fidelity cyber and physical simulation of water distribution systems. i: Models and data. *Journal of Water Resources Planning and Management*, 149(5):04023009, 2023.
- M. Mussi, L. Pellegrino, M. Restelli, and F. Trovò. A voltage dynamic-based state of charge estimation method for batteries storage systems. *Journal of Energy Storage*, 44:103309, 2021. ISSN 2352-152X.
- M. Mussi, L. Pellegrino, O. F. Pindaro, M. Restelli, and F. Trovò. A reinforcement learning controller optimizing costs and battery state of health in smart grids. *Journal of Energy Storage*, 82:110572, 2024. ISSN 2352-152X.
- A. D. Mustika, R. Rigo-Mariani, V. Debusschere, and A. Pachurka. A two-stage management strategy for the optimal operation and billing in an energy community with collective self-consumption. *Applied Energy*, 310:118484, Mar. 2022. ISSN 0306-2619.
- P. Nahata, A. La Bella, R. Scattolini, and G. Ferrari-Trecate. Hierarchical control in islanded dc microgrids with flexible structures. *IEEE transactions on control systems technology*, 29(6):2379–2392, 2020.
- U. R. Nair and R. Costa-Castelló. A model predictive control-based energy management scheme for hybrid storage system in islanded microgrids. *IEEE access*, 8:97809–97822, 2020.
- T. A. Nakabi and P. Toivanen. Deep reinforcement learning for energy management in a microgrid with flexible demand. *Sustainable Energy, Grids and Networks*, 25:100413, 2021. ISSN 2352-4677.
- J. Nash. Non-cooperative games. *Annals of Mathematics*, 54(2):286–295, 1951. ISSN 0003486X, 19398980.
- J. F. Nash. The bargaining problem. *Econometrica*, 18(2):155–162, 1950. ISSN 00129682, 14680262.
- K. S. Ng, C.-S. Moo, Y.-P. Chen, and Y.-C. Hsieh. Enhanced coulomb counting method for estimating state-of-charge and state-of-health of lithium-ion batteries. *Applied Energy*, 86(9):1506–1511, 2009. ISSN 0306-2619.
- T. A. Nguyen and M. L. Crow. Stochastic optimization of renewable-based microgrid operation incorporating battery operating cost. *IEEE Transactions on Power Systems*, 31(3):2289–2296, 2015.
- W. Nie, Z. Deng, J. Li, K. Zhang, J. Zhou, and F. Xiang. An early fault detection method of series battery packs based on multi-feature clustering and unsupervised scoring. *Energy*, 323:135754, 2025. ISSN 0360-5442.
- F. A. Oliehoek, C. Amato, et al. *A concise introduction to decentralized POMDPs*, volume 1. Springer, 2016.
- D. E. Olivares, A. Mehrizi-Sani, A. H. Etemadi, C. A. Cañizares, R. Iravani, M. Kazerani, A. H. Hajimiragha, O. Gomis-Bellmunt, M. Saeedifard, R. Palma-Behnke, et al. Trends in microgrid control. *IEEE Transactions on smart grid*, 5(4):1905–1919, 2014.
- A. Oroojlooy and D. Hajinezhad. A review of cooperative multi-agent deep reinforcement learning. *Applied Intelligence*, 53(11):13677–13722, 2023.
- G. Owen. *Game theory*. Emerald Group Publishing, 2013.
- N. Padmawansa, K. Gunawardane, S. Madanian, and A. M. Than Oo. Battery energy storage capacity estimation for microgrids using digital twin concept. *Energies*, 16(12), 2023. ISSN 1996-1073.
- E. S. Page. Continuous inspection schemes. *Biometrika*, 41(1/2):100–115, 1954.
- G. Palma, L. Guiducci, M. Stentati, A. Rizzo, and S. Paoletti. Reinforcement Learning for Energy Community Management: A European-Scale Study. *Energies*, 17(5):1249, Jan. 2024. ISSN 1996-1073.
- R. Palma-Behnke, C. Benavides, F. Lanás, B. Severino, L. Reyes, J. Llanos, and D. Sáez. A microgrid energy management system based on the rolling horizon strategy. *IEEE Transactions on smart grid*, 4(2):996–1006, 2013.

- A. Parisio, E. Rikos, and L. Glielmo. A model predictive control approach to microgrid operation optimization. *IEEE Transactions on Control Systems Technology*, 22(5):1813–1827, 2014.
- A. Parisio, E. Rikos, and L. Glielmo. Stochastic model predictive control for economic/environmental operation management of microgrids: An experimental case study. *Journal of Process Control*, 43:24–37, 2016.
- A. Parisio, C. Wiezorek, T. Kytäjä, J. Elo, K. Strunz, and K. H. Johansson. Cooperative mpc-based energy management for networked microgrids. *IEEE Transactions on Smart Grid*, 8(6):3066–3074, 2017.
- H.-A. Park, G. Byeon, W. Son, H.-C. Jo, J. Kim, and S. Kim. Digital twin for operation of microgrid: Optimal scheduling in virtual space of digital twin. *Energies*, 13(20), 2020. ISSN 1996-1073.
- M. Pasqui, A. Felice, M. Messagie, T. Coosemans, T. T. Bastianello, D. Baldi, P. Lubello, and C. Carcasci. A new smart batteries management for Renewable Energy Communities. *Sustainable Energy, Grids and Networks*, 34: 101043, June 2023. ISSN 2352-4677.
- D. Patra and S. Pradhan. Adaptive control strategies for enhancing the integration and stability of renewable energy sources in smart microgrids. *International Journal OF Electrical Engineering and Technology*, 16:1–23, 08 2025.
- W. Peng, Z. Yang, C. Liu, J. Xiu, and Z. Zhang. An improved pso algorithm for battery parameters identification optimization based on thevenin battery model. In *2018 5th IEEE International Conference on Cloud Computing and Intelligence Systems (CCIS)*, pages 295–298. IEEE, 2018.
- X. Peng, S. Duan, C. Sankvaram, and X. Jin. Robust physics guided data-driven fleet battery pack fault detection under dynamic operating conditions. In *2022 IEEE 18th International Conference on Automation Science and Engineering (CASE)*, pages 971–978. IEEE, 2022.
- X. Peng, S. Duan, C. Sankavaram, and X. Jin. Unsupervised adaptive fleet battery pack fault detection with concept drift under evolving environment. *IEEE Transactions on Automation Science and Engineering*, 21(3):2276–2288, 2024.
- X. B. Peng, E. Coumans, T. Zhang, T.-W. Lee, J. Tan, and S. Levine. Learning agile robotic locomotion skills by imitating animals. *arXiv preprint arXiv:2004.00784*, 2020.
- Y. Peng, X. Zhang, Y. Song, and D. Liu. A low cost flexible digital twin platform for spacecraft lithium-ion battery pack degradation assessment. In *2019 IEEE International Instrumentation and measurement technology conference (I2MTC)*, pages 1–6. IEEE, 2019.
- S. Pfenninger and I. Staffell. Long-term patterns of european pv output using 30 years of validated hourly reanalysis and satellite data. *Energy*, 114:1251–1265, 2016. ISSN 0360-5442.
- M. A. Pimentel, D. A. Clifton, L. Clifton, and L. Tarassenko. A review of novelty detection. *Signal Processing*, 99: 215–249, 2014. ISSN 0165-1684.
- T. Pippia, J. Sijs, and B. De Schutter. A single-level rule-based model predictive control approach for energy management of grid-connected microgrids. *IEEE Transactions on Control Systems Technology*, 28(6):2364–2376, 2019.
- G. L. Plett. Extended kalman filtering for battery management systems of lipb-based hev battery packs: Part 3. state and parameter estimation. *Journal of Power Sources*, 134(2):277–292, 2004. ISSN 0378-7753.
- S. Polimeni, L. Meraldi, L. Moretti, S. Leva, and G. Manzolini. Development and experimental validation of hierarchical energy management system based on stochastic model predictive control for off-grid microgrids. *Advances in Applied Energy*, 2:100028, 2021.
- K. Ponse, F. Kleuker, M. Fejër, A. Serra-Gòmez, A. Plaat, and T. Moerland. Reinforcement learning for sustainable energy: A survey, 2024.
- A. Prasad and I. Dusparic. Multi-agent Deep Reinforcement Learning for Zero Energy Communities. In *2019 IEEE PES Innovative Smart Grid Technologies Europe (ISGT-Europe)*, pages 1–5, Sept. 2019.
- M. L. Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.
- Q. Qi and F. Tao. Digital twin and big data towards smart manufacturing and industry 4.0: 360 degree comparison. *IEEE Access*, 6:3585–3593, 2018.

Bibliography

- X. Qu, Y. Song, D. Liu, X. Cui, and Y. Peng. Lithium-ion battery performance degradation evaluation in dynamic operating conditions based on a digital twin model. *Microelectronics Reliability*, 114:113857, 2020. ISSN 0026-2714. 31st European Symposium on Reliability of Electron Devices, Failure Physics and Analysis, ESREF 2020.
- A. Raffin, A. Hill, A. Gleave, A. Kanervisto, M. Ernestus, and N. Dormann. Stable-baselines3: Reliable reinforcement learning implementations. *Journal of Machine Learning Research*, 22(268), 2021.
- L. Ran, L. Sibó, and Z. Yongqin. Fault diagnosis of lithium battery based on fuzzy bayesian network. *International Journal of Performability Engineering*, 14(10):2302, 2018.
- A. Rasheed, O. San, and T. Kvamsdal. Digital twin: Values, challenges and enablers from a modeling perspective. *IEEE Access*, 8:21980–22012, 2020.
- T. Rashid, M. Samvelyan, C. S. De Witt, G. Farquhar, J. Foerster, and S. Whiteson. Monotonic value function factorisation for deep multi-agent reinforcement learning. *Journal of Machine Learning Research*, 21(178):1–51, 2020.
- S. T. Rastkar, S. Jamili, E. De Santis, and A. Rizzi. Improving prediction performances by integrating second derivative in microgrids energy load forecasting. In *2024 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, 2024.
- J. M. Reniers and D. A. Howey. Digital twin of a mwh-scale grid battery system for efficiency and degradation analysis. *Applied Energy*, 336:120774, 2023. ISSN 0306-2619.
- A. Reuel and D. Ma. Fairness in reinforcement learning: A survey. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, volume 7, pages 1218–1230, 2024.
- J. Richalet. Algorithmic control of industrial processes. *Proc. of the 4th IFAC Sympo. on Identification and System Parameter Estimation*, pages 1119–1167, 1976.
- R. Rosen, G. von Wichert, G. Lo, and K. D. Bettenhausen. About the importance of autonomy and digital twins for the future of manufacturing. *15th IFAC Symposium on Information Control Problems in Manufacturing*, 48(3): 567–572, 2015. ISSN 2405-8963.
- P. J. Rousseeuw and K. V. Driessen. A fast algorithm for the minimum covariance determinant estimator. *Technometrics*, 41(3):212–223, 1999.
- J. Sachs and O. Sawodny. A two-stage model predictive control strategy for economic diesel-pv-battery island microgrid operation in rural areas. *IEEE Transactions on Sustainable Energy*, 7(3):903–913, 2016.
- B. Saha and K. Goebel. Battery data set. NASA Ames Prognostics Data Repository, 2010. Available at: <https://ti.arc.nasa.gov/tech/dash/groups/pcoe/prognostic-data-repository/>.
- R. Sakile and U. K. Sinha. Estimation of lithium-ion battery state of charge for electric vehicles using a nonlinear state observer. *Energy Storage*, 4(2):e290, 2022.
- D. Salaorni. Dropflow, Oct. 2024. <https://github.com/Daveonwave/dropflow>.
- D. Salaorni, F. Bianchi, S. Colnago, A. Barisione, F. Trovò, and M. Restelli. A novel digital twin for battery energy storage systems in micro-grids. *Journal of Energy Storage*, 132:117745, 2025a. ISSN 2352-152X.
- D. Salaorni, F. Bianchi, M. Restelli, and F. Trovò. A reinforcement learning approach for optimal control in microgrids. In *2025 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, 2025b.
- C. Sandroni, M. Verga, R. Lazzari, M. Fantini, M. Sacchi, and V. Prandoni. Rse’s microgrid: A facility for research, development and testing of future distributed generation and microgrid technologies. In *2016 AEIT International Annual Conference (AEIT)*, pages 1–6, 2016.
- J. Schulman, S. Levine, P. Moritz, M. I. Jordan, and P. Abbeel. Trust region policy optimization, 2017a.
- J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal policy optimization algorithms. *arXiv*, 2017b.

- J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal Policy Optimization Algorithms, Aug. 2017c.
- C. Semeraro, H. Aljaghoub, M. A. Abdelkareem, A. H. Alami, and A. Olabi. Digital twin in battery energy storage systems: Trends and gaps detection through association rule mining. *Energy*, 273:127086, 2023a. ISSN 0360-5442.
- C. Semeraro, M. Lezoche, H. Panetto, and M. Dassisti. Data-driven invariant modelling patterns for digital twin design. *Journal of Industrial Information Integration*, 31:100424, 2023b. ISSN 2452-414X.
- C. Semeraro, A. Olabi, H. Aljaghoub, A. H. Alami, M. Al Radi, M. Dassisti, and M. A. Abdelkareem. Digital twin application in energy storage: Trends and challenges. *Journal of Energy Storage*, 58:106347, 2023c. ISSN 2352-152X.
- K. A. Severson, P. M. Attia, N. Jin, N. Perkins, B. Jiang, Z. Yang, M. H. Chen, M. Aykol, P. K. Herring, D. Fraggedakis, et al. Data-driven prediction of battery cycle life before capacity degradation. *Nature Energy*, 4(5):383–391, 2019.
- M. B. Shams, H. Budman, and T. Duever. Fault detection, identification and diagnosis using cusum based pca. *Chemical Engineering Science*, 66(20):4488–4498, 2011.
- Y. Shang, S. Wang, N. Tang, Y. Fu, and K. Wang. Research progress in fault detection of battery systems: A review. *Journal of Energy Storage*, 98:113079, 2024. ISSN 2352-152X.
- L. S. Shapley. Stochastic games*. *Proceedings of the National Academy of Sciences*, 39(10):1095–1100, 1953.
- A. Shojaeighadikolaie, A. Ghasemi, A. G. Bardas, R. Ahmadi, and M. Hashemi. Weather-aware data-driven microgrid energy management using deep reinforcement learning. In *2021 North American Power Symposium (NAPS)*, pages 1–6, 2021.
- S. R. Sinclair, F. V. Frujeri, C.-A. Cheng, L. Marshall, H. D. O. Barbalho, J. Li, J. Neville, I. Menache, and A. Swaminathan. Hindsight learning for mdps with exogenous inputs. In *International Conference on Machine Learning*, pages 31877–31914. PMLR, 2023.
- B. F. Skinner. *Science and human behavior*. Simon and Schuster, 1953.
- I. Staffell, S. Pfenninger, and N. Johnson. A global model of hourly space heating and cooling demand at multiple spatial scales. *Nature Energy*, 8, 09 2023.
- Y. Sui and S. Song. A multi-agent reinforcement learning framework for lithium-ion battery scheduling problems. *Energies*, 13(8), 2020. ISSN 1996-1073.
- P. Sunehag, G. Lever, A. Gruslys, W. M. Czarnecki, V. Zambaldi, M. Jaderberg, M. Lanctot, N. Sonnerat, J. Z. Leibo, K. Tuyls, and T. Graepel. Value-decomposition networks for cooperative multi-agent learning, 2017.
- R. S. Sutton. Learning to predict by the methods of temporal differences. *Machine learning*, 3:9–44, 1988.
- R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. The MIT Press, second edition, 2018.
- R. Swarnkar, H. Ramachandran, S. H. M. Ali, and R. Jabbar. A systematic literature review of state of health and state of charge estimation methods for batteries used in electric vehicle applications. *World Electric Vehicle Journal*, 14(9), 2023. ISSN 2032-6653.
- C. Szepesvari. Algorithms for reinforcement learning. In *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 2010.
- M. Tan. Multi-agent reinforcement learning: Independent versus cooperative agents. In *International Conference on Machine Learning*, 1997.
- R. Tan, W. Hong, J. Tang, X. Lu, R. Ma, X. Zheng, J. Li, J. Huang, and T.-Y. Zhang. BatteryLife: A comprehensive dataset and benchmark for battery life prediction. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pages 5789–5800, 2025.
- X. Tang, Y. Sun, Y. Zhao, W. Pei, N. Li, and L. Kong. Digital twin based bess state estimation and operating optimization. In *2021 IEEE 5th Conference on Energy Internet and Energy System Integration (EI2)*, pages 3402–3405, 2021.

Bibliography

- F. Tao, W. Liu, J. Liu, X. Liu, Q. Liu, T. Qu, T. Hu, Z. Zhang, F. Xiang, W. Xu, J. Wang, Y. Zhang, Z. Liu, H. Li, J. Cheng, Q. Qi, M. Zhang, H. Zhang, F. Sui, L. He, W. Yi, and H. Cheng. Digital twin and its potential application exploration. *Jisuanji Jicheng Zhizao Xitong/Computer Integrated Manufacturing Systems, CIMS*, 24(1):1–18, Jan. 2018. ISSN 1006-5911.
- F. Tao, F. Sui, A. Liu, Q. Qi, M. Zhang, B. Song, Z. Guo, S. C.-Y. Lu, and A. Y. C. Nee. Digital twin-driven product design framework. *International Journal of Production Research*, 57(12):3935–3953, 2019a.
- F. Tao, H. Zhang, A. Liu, and A. Y. C. Nee. Digital twin in industry: State-of-the-art. *IEEE Transactions on Industrial Informatics*, 15(4):2405–2415, 2019b.
- F. Tao, B. Xiao, Q. Qi, J. Cheng, and P. Ji. Digital twin modeling. *Journal of Manufacturing Systems*, 64:372–389, 2022. ISSN 0278-6125.
- T. Tjaden, J. Bergner, J. Weniger, and Quaschnig. Representative electrical load profiles of residential buildings in germany with a temporal resolution of one second, 2015.
- M. Towers, A. Kwiatkowski, J. Terry, J. U. Balis, G. De Cola, T. Deleu, M. Goulão, A. Kallinteris, M. Krimmel, A. KG, et al. Gymnasium: A standard interface for reinforcement learning environments. *arXiv preprint arXiv:2407.17032*, 2024.
- M. Towers, A. Kwiatkowski, J. Terry, J. U. Balis, G. D. Cola, T. Deleu, M. Goulão, A. Kallinteris, M. Krimmel, A. KG, R. Perez-Vicente, A. Pierré, S. Schulhoff, J. J. Tai, H. Tan, and O. G. Younis. Gymnasium: A standard interface for reinforcement learning environments, 2025.
- M.-K. Tran, M. Mathew, S. Janhunnen, S. Panchal, K. Raahemifar, R. Fraser, and M. Fowler. A comprehensive equivalent circuit model for lithium-ion batteries, incorporating the effects of state of health, state of charge, and temperature on model parameters. *Journal of Energy Storage*, 43:103252, 2021. ISSN 2352-152X.
- V. Venkatasubramanian, R. Rengaswamy, S. N. Kavuri, and K. Yin. A review of process fault detection and diagnosis: Part iii: Process history based methods. *Computers & Chemical Engineering*, 27(3):327–346, 2003. ISSN 0098-1354.
- R. Vignali, A. Falsone, F. Ruiz, and G. Gruosso. Towards a comprehensive framework for V2G optimal operation in presence of uncertainty. *Sustainable Energy, Grids and Networks*, 31:100740, 2022.
- C. Wang, M. Xu, Q. Zhang, J. Feng, R. Jiang, Y. Wei, and Y. Liu. Parameters identification of thevenin model for lithium-ion batteries using self-adaptive particle swarm optimization differential evolution algorithm to estimate state of charge. *Journal of Energy Storage*, 44:103244, 2021. ISSN 2352-152X.
- Y. Wang, J. Tian, Z. Sun, L. Wang, R. Xu, M. Li, and Z. Chen. A comprehensive review of battery modeling and state estimation approaches for advanced battery management systems. *Renewable and Sustainable Energy Reviews*, 131:110015, 2020. ISSN 1364-0321.
- C. J. Watkins and P. Dayan. Q-learning. *Machine learning*, 8:279–292, 1992.
- S. Weyer, T. Meyer, M. Ohmer, D. Gorecky, and D. Zühlke. Future modeling and simulation of cps-based factories: an example from the automotive industry. *12th IFAC Workshop on Intelligent Manufacturing Systems IMS 2016*, 49(31):97–102, 2016. ISSN 2405-8963.
- R. J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8:229–256, 2004.
- C. Wu, C. Zhu, and Y. Ge. A new fault diagnosis and prognosis technology for high-power lithium-ion battery. *IEEE Transactions on Plasma Science*, 45(7):1533–1538, 2017.
- K. Xia, C. Sacco, M. Kirkpatrick, C. Saidy, L. Nguyen, A. Kircaliali, and R. Harik. A digital twin to train deep reinforcement learning agent for smart manufacturing plants: Environment, interfaces and intelligence. *Journal of Manufacturing Systems*, 58:210–230, 2021.
- B. Xu, A. Oudalov, A. Ulbig, G. Andersson, and D. S. Kirschen. Modeling of lithium-ion battery degradation for cell life assessment. *IEEE Transactions on Smart Grid*, 9(2):1131–1140, 2018.
- Z. Xu, J. Xu, Z. Guo, H. Wang, Z. Sun, and X. Mei. Design and optimization of a novel microchannel battery thermal management system based on digital twin. *Energies*, 15(4), 2022. ISSN 1996-1073.

- R. R. Yager and D. P. Filev. Approximate clustering via the mountain method. *IEEE Transactions on systems, man, and Cybernetics*, 24(8):1279–1284, 1994.
- D. Yang, Y. Cui, Q. Xia, F. Jiang, Y. Ren, B. Sun, Q. Feng, Z. Wang, and C. Yang. A digital twin-driven life prediction method of lithium-ion batteries based on adaptive model evolution. *Materials*, 15(9), 2022. ISSN 1996-1944.
- Y. Yang, S. Bremner, C. Menictas, and M. Kay. Battery energy storage system size determination in renewable energy systems: A review. *Renewable and Sustainable Energy Reviews*, 91:109–125, 2018. ISSN 1364-0321.
- Y. Yang, C. Ma, Z. Ding, S. McAleer, C. Jin, J. Wang, and T. Sandholm. Game-Theoretic Multiagent Reinforcement Learning, Aug. 2025.
- Z. Yun, X. Cui, and K. Ma. Online thevenin equivalent parameter identification method of large power grids using lu factorization. *IEEE Transactions on Power Systems*, 34(6):4464–4475, 2019.
- H. Zang and J. Kim. Reinforcement Learning Based Peer-to-Peer Energy Trade Management Using Community Energy Storage in Local Energy Market. *Energies*, 14(14):4131, Jan. 2021. ISSN 1996-1073.
- K. Zhang, Z. Yang, and T. Başar. Multi-agent reinforcement learning: A selective overview of theories and algorithms. *Handbook of reinforcement learning and control*, pages 321–384, 2021.
- R. Zhang, F. Wang, J. Cai, Y. Wang, H. Guo, and J. Zheng. Digital twin and its applications: A survey. *The International Journal of Advanced Manufacturing Technology*, 123(11):4123–4136, 2022.
- W. Zhang, X. Li, and X. Li. Deep learning-based prognostic approach for lithium-ion batteries with adaptive time-series prediction and on-line validation. *Measurement*, 164:108052, 2020. ISSN 0263-2241.
- K. Zhao, Y. Liu, W. Ming, Y. Zhou, and J. Wu. Digital twin-driven estimation of state of charge for li-ion battery. In *2022 IEEE 7th International Energy Conference (ENERGYCON)*, pages 1–6, 2022a.
- Y. Zhao, P. Liu, Z. Wang, L. Zhang, and J. Hong. Fault and defect diagnosis of battery for electric vehicles based on big data analysis methods. *Applied Energy*, 207:354–362, 2017. ISSN 0306-2619.
- Z. Zhao, J. Guo, X. Luo, C. S. Lai, P. Yang, L. L. Lai, P. Li, J. M. Guerrero, and M. Shahidehpour. Distributed robust model predictive control-based energy management strategy for islanded multi-microgrids considering uncertainty. *IEEE Transactions on Smart Grid*, 13(3):2107–2120, 2022b.
- K. Zhou, K. Zhou, and S. Yang. Reinforcement learning-based scheduling strategy for energy storage in microgrid. *Journal of Energy Storage*, 51:104379, 2022. ISSN 2352-152X.
- S. Zhou, Z. Hu, W. Gu, M. Jiang, and X.-P. Zhang. Artificial intelligence based smart energy community management: A reinforcement learning approach. *CSEE Journal of Power and Energy Systems*, 5(1):1–10, Mar. 2019. ISSN 2096-0042.
- C. Zhu, R. H. Byrd, P. Lu, and J. Nocedal. Algorithm 778: L-bfgs-b: Fortran subroutines for large-scale bound-constrained optimization. *ACM Transactions on mathematical software (TOMS)*, 23(4):550–560, 1997.
- J. Zhu, M. Knapp, D. R. Sørensen, M. Heere, M. S. Darma, M. Müller, L. Mereacre, H. Dai, A. Senyshyn, X. Wei, et al. Investigation of capacity fade for 18650-type lithium-ion batteries cycled in different state of charge (soc) ranges. *Journal of Power Sources*, 489:229422, 2021.

Appendices

Digital Twin Validation: Additional Details

In this appendix chapter, we provide additional details on the desing and validation of ERNESTO DT presented in Part I. Specifically, we include the image of the lab setup used to collect model parameters in Figure A.1. In addition, we report the characteristics of the Li-ion cell in Table A.1 used for the DT validation and the complete set of parameters used in the different experiments in Table A.2. All simulations were performed on a MacBook Pro 2023 equipped with an Apple M2 Pro CPU and 16 GB of RAM.

Table A.1: *EIG ePLB-C020 cell characteristics.*

Parameter	Value
Nominal capacity (C_N)	20 Ah
Maximum voltage (V_{max})	4.2 V
Minimum voltage (V_{min})	3 V
Dimensions $L \times W \times H$	$196 \times 127 \times 7$ mm
Weight (M)	480 g

A.0.1 DROPFLOW Performance Note

To compute the stress cycles for the aging model in real-time, we implemented the DROPFLOW algorithm as a computationally efficient alternative to the standard Rainflow-counting method. Benchmarking on a stream of 35 000 samples revealed that while the iterative Rainflow algorithm requires approximately 241.8 s, the DROPFLOW implementation completes the task in just 4.26 s, achieving a computational reduction of $\approx 98\%$. This efficiency is critical for enabling the long-term simulations and real-time applications envisioned for the ERNESTO framework.

Appendix A. Digital Twin Validation: Additional Details

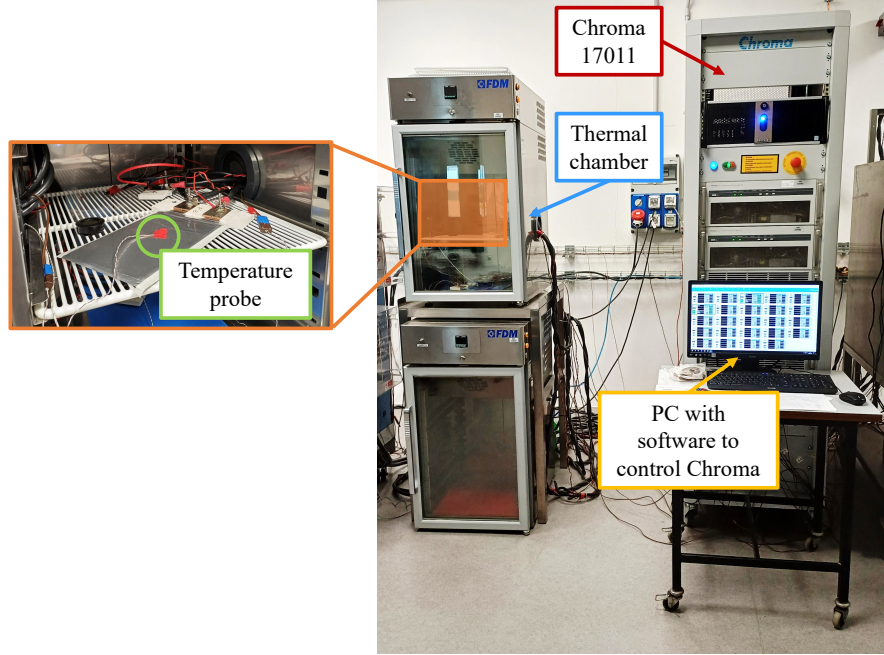


Figure A.1: RSE laboratory experimental setup: thermal chamber, Chroma 17011 cycler, temperature probe and data acquisition system.

Table A.2: Parameters used in the different experiments.

Parameters	Experiments			
	PV	DER-TF	TJ	Aging
Profile type	CP	CP	CP	CC
C_N [Ah]	20	20	20	20
$V(t_0)$ [V]	3.58	3.72	4.12	3.43
$SOC(t_0)$	0.18	0.76	0.94	0.36
$SOH(t_0)$	1	1	1	1
α_{SEI}	—	—	—	0
β_{SEI}	—	—	—	121
k_t	—	—	—	$4.14e-10$
k_{SOC}	—	—	—	1.04
k_{temp}	—	—	—	$6.93e-2$
k_{δ_1}	—	—	—	$1.4e5$
k_{δ_2}	—	—	—	$-5.01e-1$
k_{δ_3}	—	—	—	$-1.23e5$

Real-Time Monitoring and Adaptation: Additional Details

In this appendix chapter, we provide additional details on the online learning routine for real-time monitoring and adaptation presented in Part II. In particular, Figure B.1, Figure B.2 and Figure B.3 show a zoomed-in view of the voltage signal in the three scenarios, to better visualize how the different strategies adapt to the evolution of the physical battery.

In addition, Figure B.4, Figure B.5 and Figure B.6 show the evolution of the parameters in the three scenarios. Then, in Table B.1 and Table B.2, we report the hyperparameters employed for the synthetic and real-world scenarios, respectively. In particular, we want to articulate the rationale behind our choice of hyperparameters. While the optimizer parameters have been decided after a preliminary grid search procedure, the ones used in the adaptation phase undergo a more careful tuning process. For instance, the choice of having 30 as the minimum number of outliers to trigger the adaptation phase is motivated by the need to balance sensitivity and robustness, and gather enough samples to perform the aggregation in a new cluster. The value for the λ parameter of the spatio-temporal norm has been chosen to give more predominance to the spatial component, since the passing of time is somehow described also by the system degradation. Instead, for r , η , and support fraction we adopted the default parameters presented in the original paper by (Alippi et al., 2014).

Finally, in Figure B.7, Figure B.8 and Figure B.9 we show a 3D visualization in the parameter space of the evolving clustering mechanism, to give a spatial intuition of the adaptation process of the algorithm in the three scenarios.

All the experiments were performed on a MacBook Pro 2023 equipped with an Apple M2 Pro CPU and 16 GB of RAM.

Appendix B. Real-Time Monitoring and Adaptation: Additional Details

Table B.1: Configuration parameters for real-time monitoring on the synthetic scenarios

Parameter	Cluster Shift	Memoryless Adaptation	Evolving Clustering
Algorithm	L-BFGS-B	L-BFGS-B	L-BFGS-B
Max Iterations	100	100	100
Tolerance ($ftol$)	1×10^{-6}	1×10^{-6}	1×10^{-6}
Gradient Tolerance ($gtol$)	1×10^{-4}	1×10^{-4}	1×10^{-4}
Batch Size	3000	3000	3000
N Jobs	4	4	4
N Guesses	4	4	4
Regularization Weight (ξ)	0	0	0
r_0	[0.001, 0.01]	[0.001, 0.01]	[0.001, 0.01]
r_1	[0.001, 0.01]	[0.001, 0.01]	[0.001, 0.01]
c_1	[5000, 50000]	[5000, 50000]	[5000, 50000]
Scale Factors	[1e3, 1e3, 2e-4]	[1e3, 1e3, 2e-4]	[1e3, 1e3, 2e-4]
Min Outliers	–	–	30
KS Test Alpha	–	–	2e-2
Spatio-temporal norm λ	–	–	0.95
Mountain Method Radius r	–	–	0.1
Mountain Method η	–	–	$1e-6$
MCD Max Iterations	–	–	10
MCD Support Fraction	–	–	0.75

Table B.2: Configuration parameters for real-time monitoring on the real-world scenario

Parameter	Cluster Shift	Memoryless Adaptation	Evolving Clustering
Algorithm	L-BFGS-B	L-BFGS-B	L-BFGS-B
Max Iterations	100	100	100
Tolerance ($ftol$)	1×10^{-6}	1×10^{-6}	1×10^{-6}
Gradient Tolerance ($gtol$)	1×10^{-4}	1×10^{-4}	1×10^{-4}
Batch Size	2700	2700	2700
N Jobs	4	4	4
N Guesses	4	4	4
Regularization Weight (ξ)	0.5	0.5	0.5
r_0	[0.001, 0.01]	[0.001, 0.01]	[0.001, 0.01]
r_1	[0.001, 0.01]	[0.001, 0.01]	[0.001, 0.01]
c_1	[5000, 50000]	[5000, 50000]	[5000, 50000]
Scale Factors	[1e3, 1e3, 2e-4]	[1e3, 1e3, 2e-4]	[1e3, 1e3, 2e-4]
Min Outliers	–	–	30
KS Test Alpha	–	–	2e-2
Spatio-temporal norm λ	–	–	0.95
Mountain Method Radius r	–	–	0.1
Mountain Method η	–	–	$1e-6$
MCD Max Iter Factor	–	–	10
MCD Support Fraction	–	–	0.75

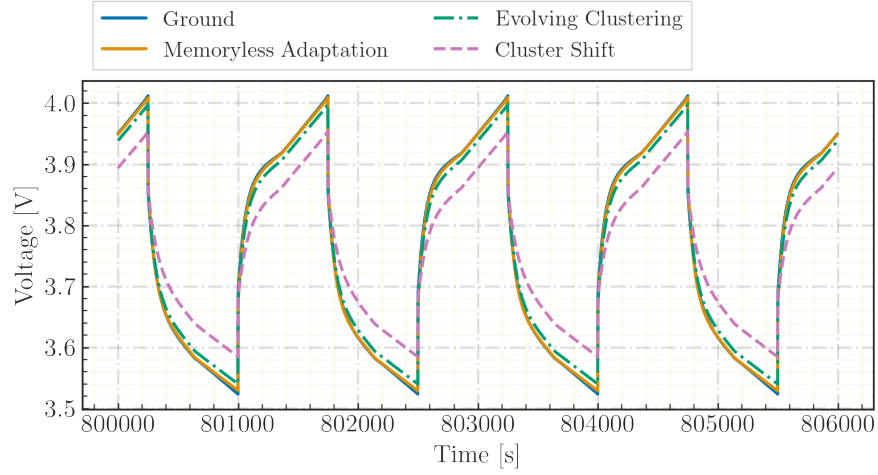


Figure B.1: Zoomed-in view of the voltage signal in the single thermal jump scenario. The plot represents the voltage signal of the system after the thermal jump, highlighting how the different strategies adapt to the new regime.

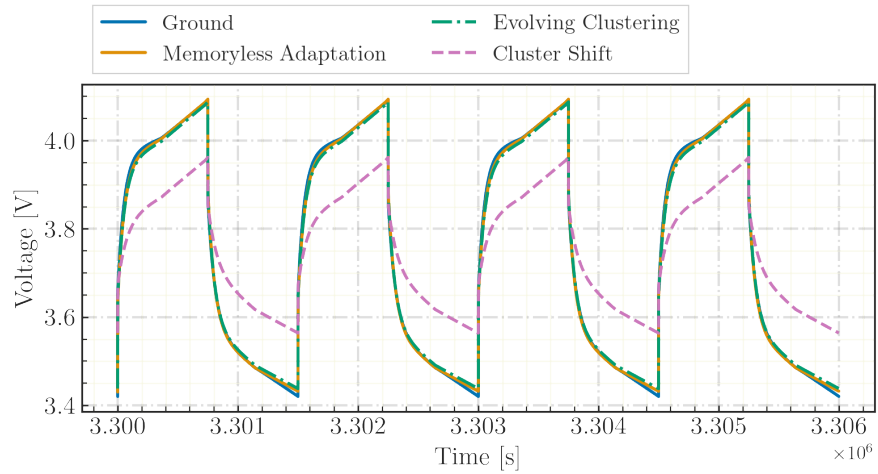


Figure B.2: Zoomed-in view of the voltage signal in the multiple thermal jumps scenario. The plot represents the voltage signal after the third thermal jump, to compare how the different strategies adapt to the new regime.

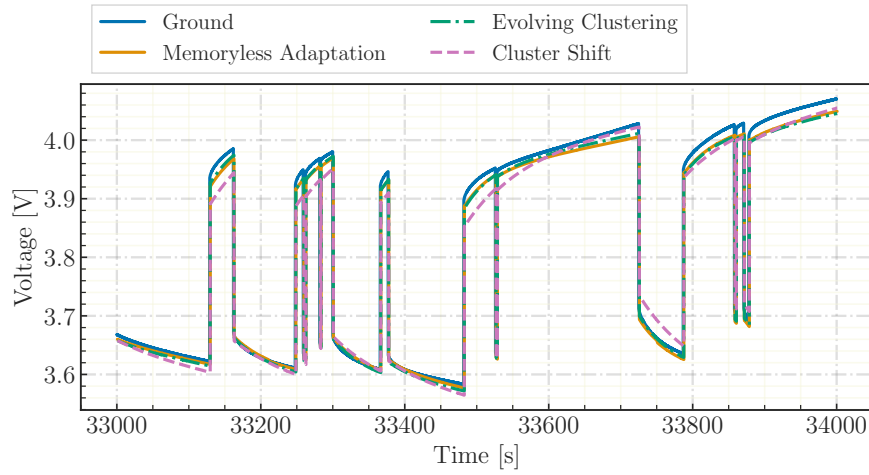
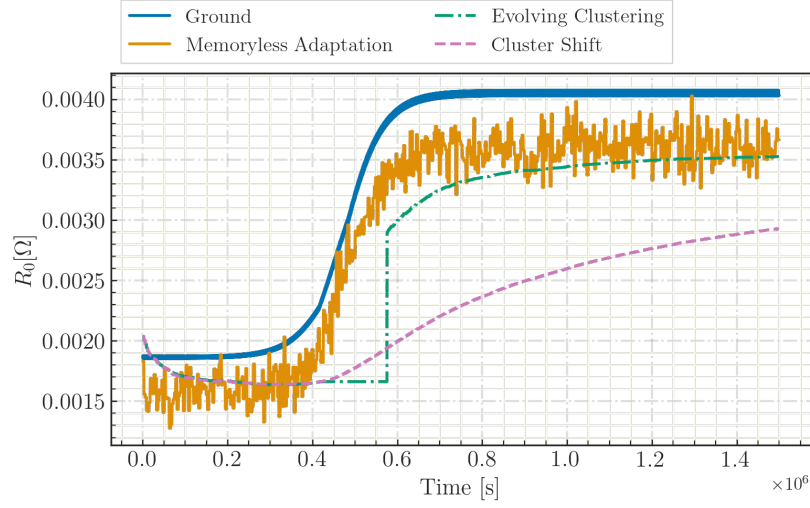
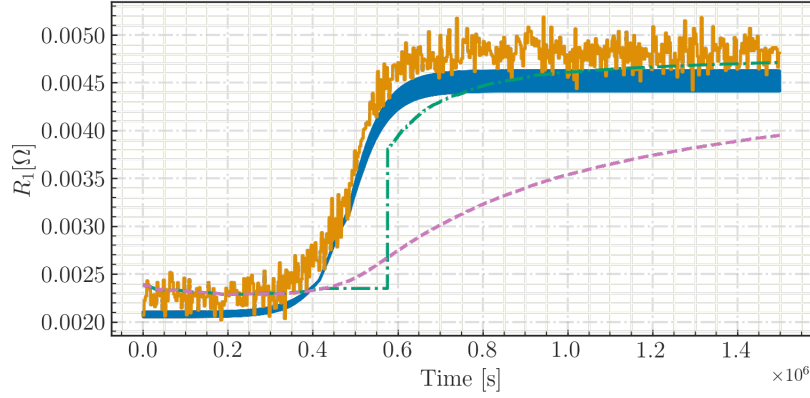


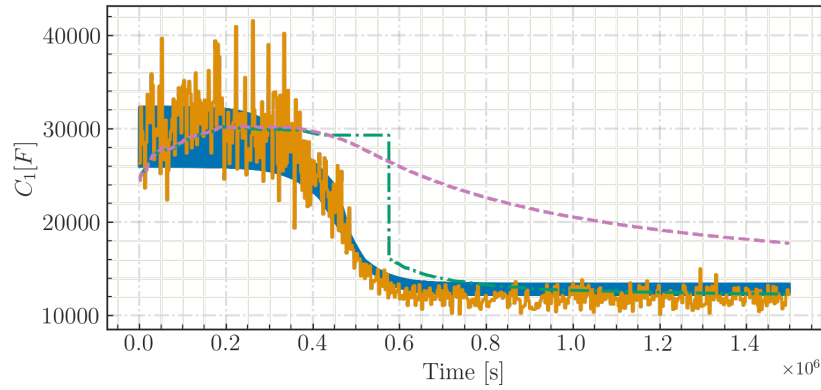
Figure B.3: Zoomed-in view of the voltage signal in the real-world scenario. The plot represents the voltage signal before the thermal jump, but after the first alignment phase due to the unknown initial conditions, to compare how the different strategies adapt to the new regime.



(a) Evolution of the R_0 parameter.

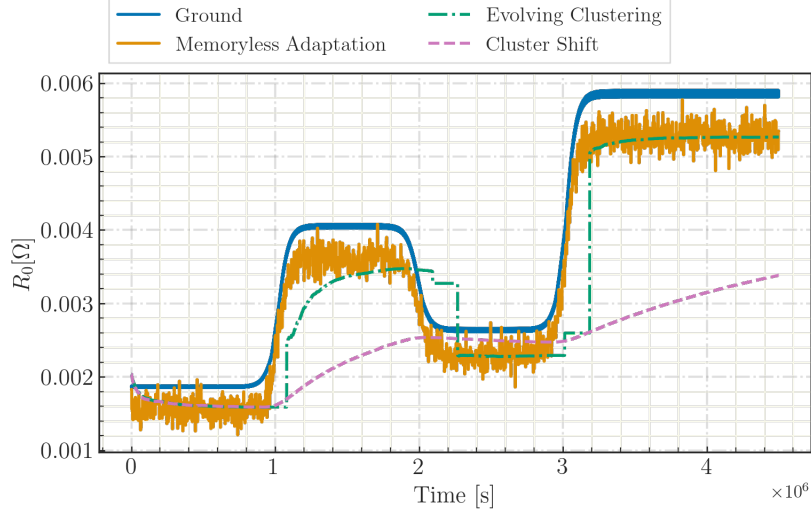


(b) Evolution of the R_1 parameter.

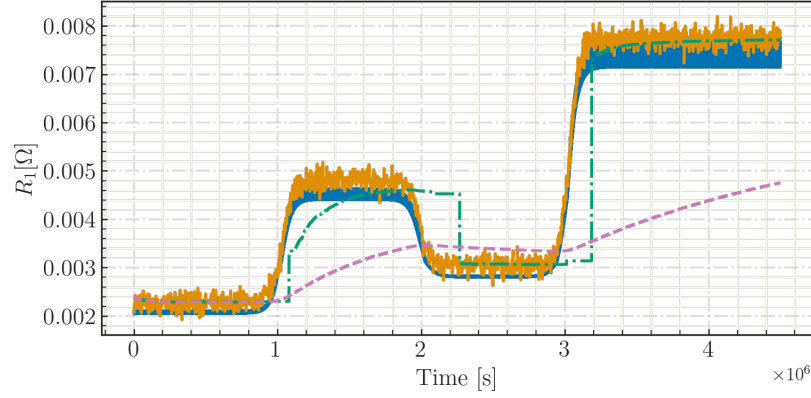


(c) Evolution of the C_1 parameter.

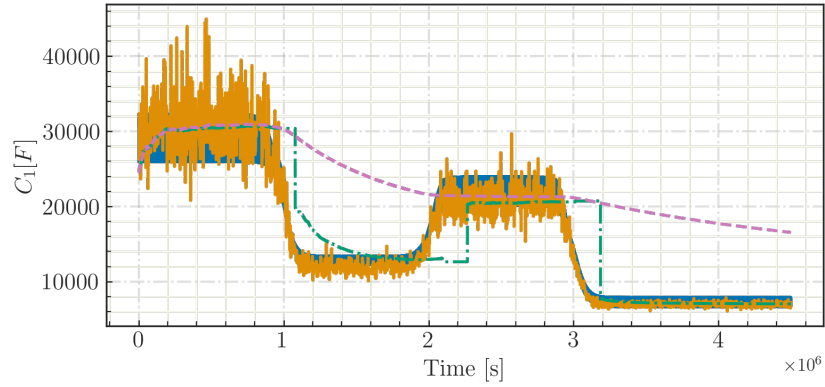
Figure B.4: Evolution of the R_0 , R_1 , and C_1 parameters in the single thermal jump scenario.



(a) Evolution of the R_0 parameter.

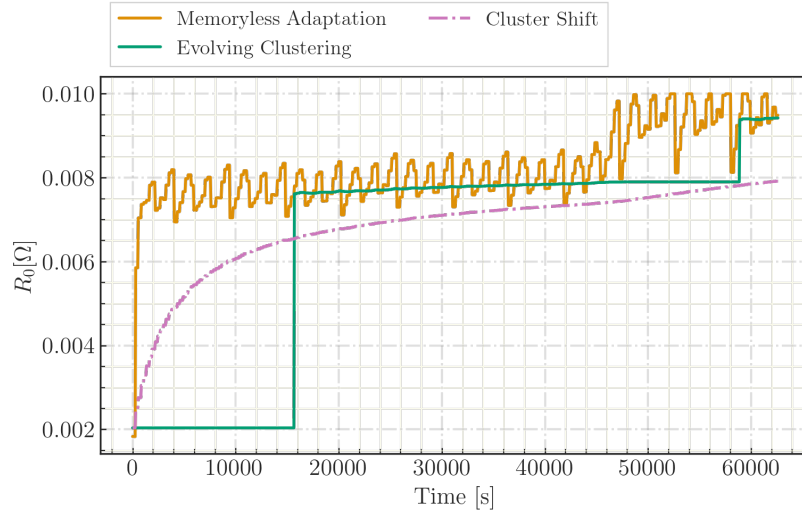


(b) Evolution of the R_1 parameter.

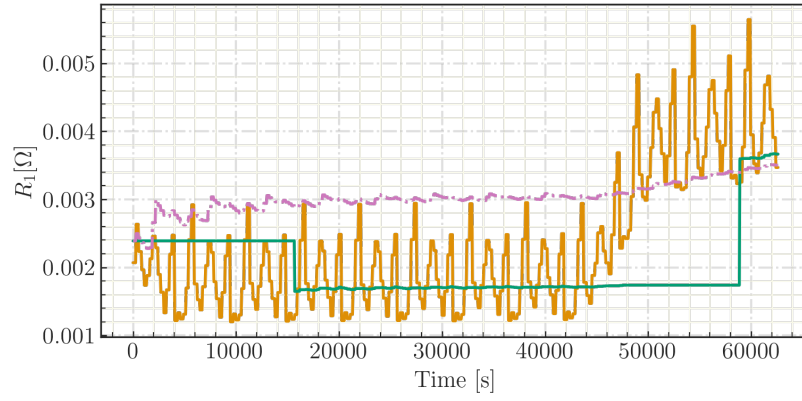


(c) Evolution of the C_1 parameter.

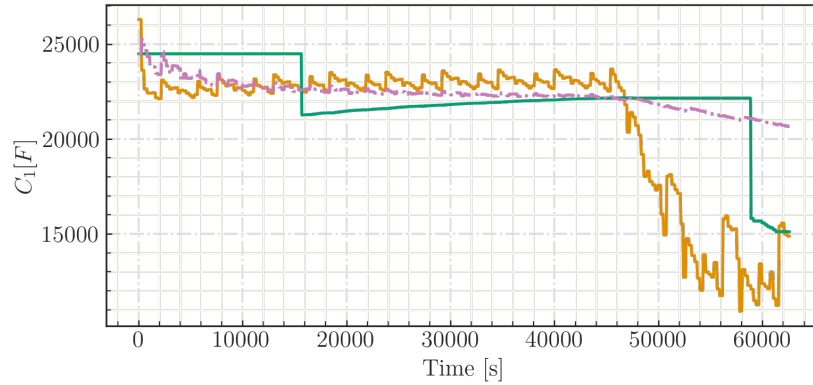
Figure B.5: Evolution of the R_0 , R_1 , and C_1 parameters in the multiple thermal jumps scenario.



(a) Evolution of the R_0 parameter.



(b) Evolution of the R_1 parameter.



(c) Evolution of the C_1 parameter.

Figure B.6: Evolution of the R_0 , R_1 , and C_1 parameters in the real-world scenario.

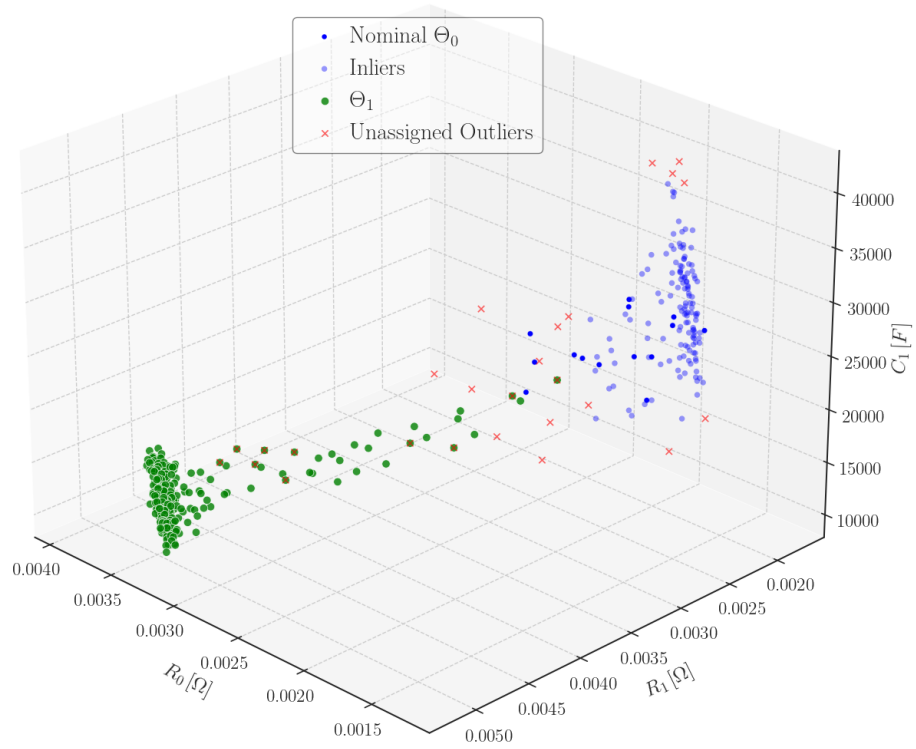


Figure B.7: 3D visualization of the evolving clustering mechanism in the single thermal jump scenario. The plot illustrates how the model identifies and adapts to the new regime after the thermal jump, with the formation of a new cluster representing the new nominal parameters of the system.

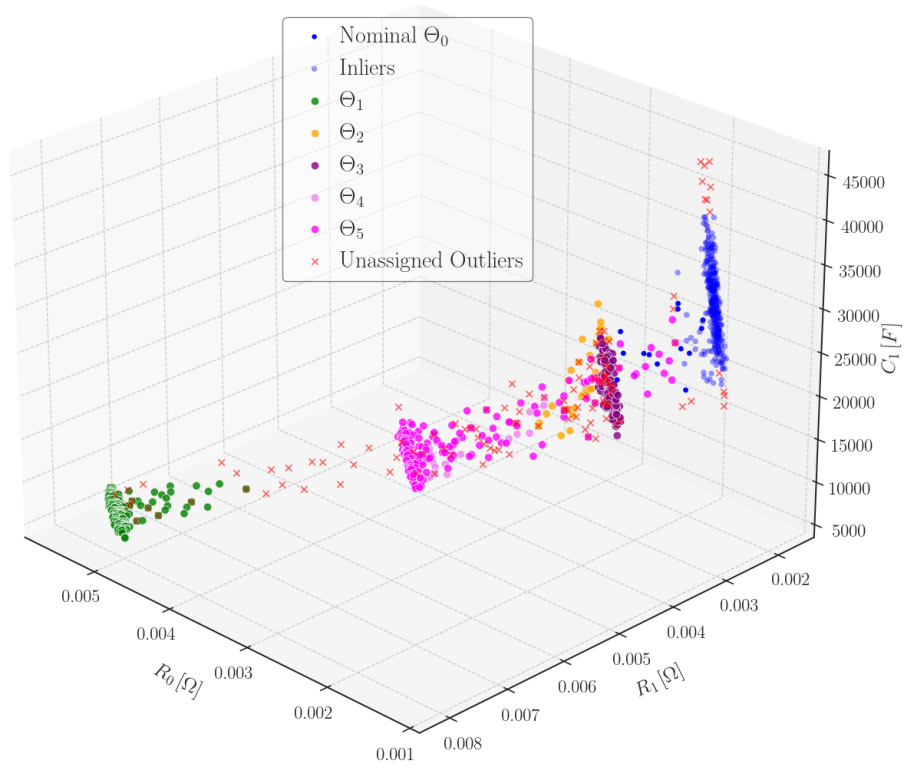


Figure B.8: 3D visualization of the evolving clustering mechanism in the multiple thermal jumps scenario. The plot illustrates how the model identifies and adapts to the new regime after each thermal jump, with the formation of a new cluster representing the new nominal parameters of the system.

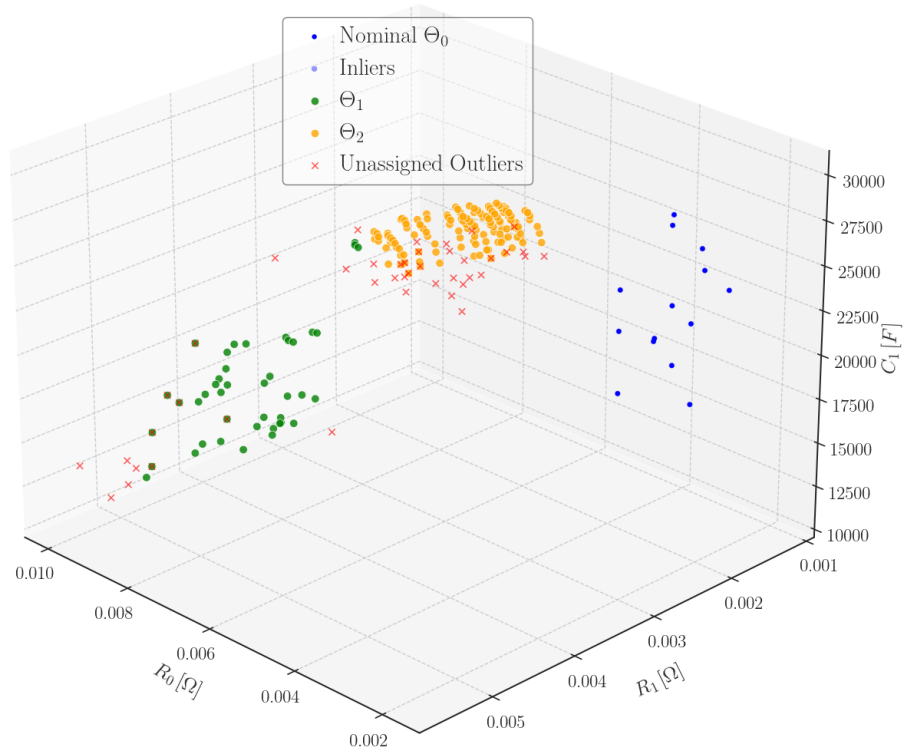


Figure B.9: 3D visualization of the evolving clustering mechanism in the real-world scenario. The plot illustrates how the model identifies and adapts to the new regime after each thermal jump, with the formation of a new cluster representing the new nominal parameters of the system.

Microgrid Energy Management: Additional Details

In this appendix chapter, we provide additional details on the microgrid energy management system presented in Part III. Specifically, we include the battery pack configuration and the complete set of parameters used in the different experiments in Table C.1.

The RL agent is trained on 4 parallel environments leveraging the PPO algorithm (Raffin et al., 2021). The training phase lasts $n_{\text{ep}} = 100$ episodes and has a decision step of $\Delta t = 3600 \text{ s}$ (or equivalently a frequency of 0.0003 Hz). For the networks of PPO’s actor and critic has been chosen a feedforward architecture with 2 hidden layers of 64 neurons each.

The presented experiments were conducted on a MacBook Pro, 2023, with an Apple M2 Pro CPU, 16 GB of RAM, and MacOS Sequoia 15.

Table C.1: *Battery pack parameters.*

Parameter	Symbol	Value
Voltage rated	v_r	350.4 V
Minimum voltage	$v^{\min}$	288 V
Maximum voltage	$v^{\max}$	398.4 V
Nominal capacity	C^{nom}	60 Ah
Replacement cost	c^{bat}	3000 €

Table C.2: *PPO parameter configuration.*

Parameter	Value
# Episodes	100
# Envs	8
Policy Network Size	[64, 32]
Gamma	0.99
Learning Rate	$5e-5$
Batch size	512
# Epochs	10
Rollouts	8912
Initial log. std.	−1
Normalize obs.	True
Seed	42

Renewable Energy Communities Management: Additional Details

In this appendix chapter, we provide additional details on the renewable energy communities management system presented in Part IV. Specifically, we include additional information about implementation and hardware details in Chapter 12.

For the whole framework and the experimental campaign, we adopt a Python implementation, leveraging the JAX (Bradbury et al., 2018) and Flax (Heek et al., 2024) libraries – the former for efficient JIT-compiled computation, and the latter for neural network modeling. The server used to run the experiments is equipped with an AMD EPYC 7453 28-Core Processor, 24 GB of RAM, and an Nvidia L4 with 24 GB of vRAM.

D.1 Incentives Structure

In the main paper we showed that the financial incentives given to the whole REC are:

$$V_t := U_t \cdot (\alpha + \beta + \varphi_t).$$

Specifically, the constants utilized in this equation have the following meanings:

- α is a cost refund term, that can slightly change from year to year, and has a value of about 8–10 €/MWh. In our experiments we used 8 €/MWh;
- β is the constant term of the incentives. Its base value is 80 €/MWh, but depending on what region the REC is located in, 4 €/MWh (in center regions) or 10 €/MWh (in northern regions) are added to it. We used its base value;
- φ_t is the variable part of the incentives, that changes depending on the market energy price. It has value between 0 and 40 €/MWh.

D.2 Additional Experimental Details

In this section, we report additional details about the experimental procedure. In particular, we show the results obtained in the first scenario in Table D.1. The large variability of such results is due to the variation of demand profiles, which are summarized in Figure D.1.

Table D.1: Comparison of the algorithms in the first scenario, comprising tasks with increasing complexity due to the growing numbers of agents. The values (best in bold) represent the average yearly performance with standard deviation over five runs of testing (of one year of simulation each).

Algorithm	Social Welfare [€]			Self-Consumption [MWh]		
	3	5	8	3	5	8
Rule-based	-530 ± 120	-803 ± 214	-982 ± 242	0.73 ± 0.13	1.76 ± 0.12	2.33 ± 0.73
PPO	-535 ± 120	-821 ± 214	-1025 ± 247	0.69 ± 0.15	1.56 ± 0.08	2.14 ± 0.13
LOLA	-575 ± 121	-927 ± 228	—*	0.66 ± 0.11	1.18 ± 0.12	—*
INAIA-SHARED	-520 ± 118	-787 ± 207	-942 ± 239	0.79 ± 0.16	1.90 ± 0.14	9.59 ± 0.18
INAIA-OPT	-516 ± 117	-756 ± 213	-853 ± 212	0.94 ± 0.17	3.33 ± 0.22	17.15 ± 0.48

* Results are missing due to insufficient RAM availability which prevents the completion of the experiment.

D.3 Network architectures and hyperparameters

In this section, we provide additional details about the architectures and hyperparameters used for both node agents and the REC agent in all experiments presented in Part IV.

D.3.1 Node agents

In all experiments, both considering INAIA and the different baselines, the node agents are trained in the same way, i.e., with PPO. The PPO architecture comprises an actor and a critic network, both consisting of two hidden layers of 64 and 32 neurons, respectively. In Table D.2, we reported the main hyperparameters for the training of the node agents.

Table D.2: Node agents training hyperparameters. The true value parameters are those utilized in phase (ii) of the algorithm by node agents. Instead, simulated value parameters are employed by the REC agent to perform virtual updates in phase (i).

Hyperparameter	True Value	Simulated Value
Initial learning rate (α_n)	$5 \cdot 10^{-5}$	$1 \cdot 10^{-2}$
Final learning rate	$1 \cdot 10^{-7}$	-
Learning rate schedule	cosine	constant
Number of steps	8192	256
Number of minibatches	32	2
Number of epochs	10	3
Discount factor	0.99	0.99
GAE lambda	0.98	0.98

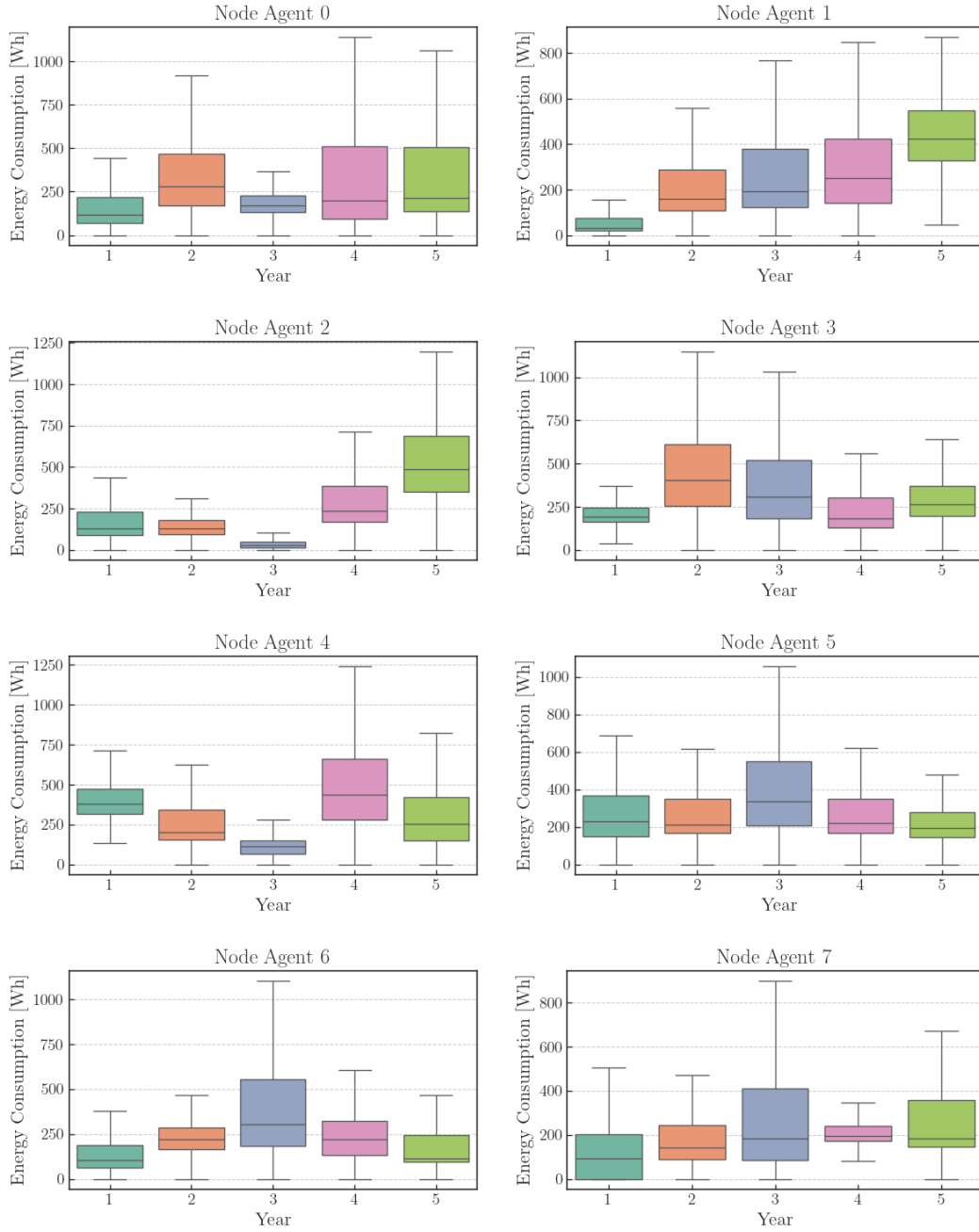


Figure D.1: Energy consumption yearly profiles for the first experiment. Each boxplot represents the statistics of energy consumption for each year of testing. The large variability both across different years and agents explain the high variance of the reported results.

D.3.2 REC agent

The REC agent, in the case of PPO and LOLA, includes both an actor and a critic network. These networks have a multi-branch architecture: each branch is a small feedforward network with two hidden layers of 64 and 32 neurons. Each branch takes as input the data of a single node agent $n \in \mathcal{N}_A$ (namely, $d_{n,t}$, $g_{n,t}$, and $b_{n,t}$) along with aggregated REC-level data (p_t^+ , p_t^- , $\bar{d}_t$, $\bar{g}_t$, $\bar{b}_t$), and time encodings. The scalar outputs of all branches are then concatenated. In the actor network, this concatenated vector is passed through a softplus layer, and serves as the parameter of the Dirichlet distribution used to sample actions in PPO¹. In the critic network, the branch outputs are instead combined via a weighted sum using a learnable parameter vector.

In the case of INAIA, no critic network is required, and the policy is deterministic. Here, the concatenated outputs of the branches are passed through a softmax layer to produce the action.

The distinction between INAIA-OPT and INAIA-SHARED lies in the parameterization of the branches. In INAIA-OPT, each branch has its own set of learnable parameters, allowing the REC agent to process each household differently. This is also the case for PPO and LOLA. In INAIA-SHARED, by contrast, the branches share the same parameters: a single set of weights is applied to the data from all households, effectively processing them in parallel through a shared function.

The main hyperparameters used for training the REC agent are listed in Table D.3.

Table D.3: REC agent training hyperparameters.

Hyperparameter	INAIA-SHARED	INAIA-OPT	PPO	LOLA
Initial learning rate ($\alpha_{\mathcal{E}}$)	$3 \cdot 10^{-4}$	$8 \cdot 10^{-4}$	$4 \cdot 10^{-4}$	0.1
Final learning rate	$1 \cdot 10^{-6}$	$1 \cdot 10^{-6}$	$1 \cdot 10^{-6}$	$1 \cdot 10^{-6}$
Learning rate schedule	cosine	cosine	cosine	cosine
Number of steps	256	256	8192	8192
Number of minibatches	-	-	64	-
Number of epochs	-	-	10	-
Discount factor	-	-	0.99	0.99
Number of simulated steps	32	32	-	-
Episodes Phase (i) ($E_{\mathcal{E}}$)	3	3	1	-
Episodes Phase (ii) ($E_{\mathcal{N}_A}$)	1	1	1	-

¹The softplus function is $f(x) := \ln(1 + e^x)$. It ensures that the output is positive, which is a requirement for the concentration parameters of a Dirichlet distribution.