Doctoral theses at NTNU, 2024:288

Wenqi Cai

Engineering Applications of Model Predictive Control-based Reinforcement Learning

Doctoral thesis

NTNU
Norwegian University of Science and Technology
Thesis for the Degree of
Philosophiae Doctor
Faculty of Information Technology and Electrical Engineering
Department of Engineering Cybernetics

NTNU
Norwegian University of
Science and Technology

Wenqi Cai

Engineering Applications of Model Predictive Control-based Reinforcement Learning

Thesis for the Degree of Philosophiae Doctor

Trondheim, September 2024

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Engineering Cybernetics

NTNU
Norwegian University of
Science and Technology

NTNU
Norwegian University of Science and Technology

Thesis for the Degree of Philosophiae Doctor

Faculty of Information Technology and Electrical Engineering
Department of Engineering Cybernetics

© Wenqi Cai

ISBN 978-82-326-8168-6 (printed ver.)
ISBN 978-82-326-8167-9 (electronic ver.)
ISSN 1503-8181 (printed ver.)
ISSN 2703-8084 (online ver.)

Doctoral theses at NTNU, 2024:288

Printed by NTNU Grafisk senter

Summary

Model Predictive Control (MPC) has emerged as a highly influential control strategy, leveraging models of real system dynamics to generate input-state sequences that minimize costs under certain constraints. However, building an accurate MPC model, especially for stochastic systems, remains a significant challenge, leading to potential performance degradation. The integration of Machine Learning (ML) in Data-driven Model Predictive Control (DMPC) aimed to alleviate this issue has brought its own problems. Specifically, modeling in DMPC is often disconnected from the control objectives because ML-based models focus on predictions rather than MPC performance, which can lead to significantly suboptimal policies.

Reinforcement Learning (RL), a model-free approach, has arisen as a promising tool, with a core advantage in learning policies through interaction with the environment. Although does not rely on system models, conventional RL methods are known to suffer from extensive data requirements, a lack of formal tools to satisfy system constraints, and challenges related to the parameterization of Deep Neural Network (DNN).

This thesis focuses on an innovative Model Predictive Control-based Reinforcement Learning (MPC-based RL) method that amalgamates the strengths of MPC and RL, compensating for the shortcomings of both. The approach focuses on parameterizing the MPC model, cost, and constraints, applying RL to tune these parameters to minimize the closed-loop performance. This fusion leads to a method that not only

takes advantage of prior knowledge but also considers system constraints, analyzes stability, overcomes uncertainties, and deals with long-term or even infinite-horizon problems. The applicability and effectiveness of the MPC-based RL approach are demonstrated through three engineering applications, each characterized by no exact model, high uncertainty, or economic cost function.

- Autonomous Surface Vehicle (ASV): By applying the MPC-RL method to ASVs, the thesis demonstrates its efficacy in optimizing a simplified freight mission with constraints like collision-free path tracking and autonomous docking. Simulations showed an improvement in closed-loop performance.
- Energy Management in Residential Microgrids: In a more complex scenario involving fluctuating spot-market prices and uncertainties, the MPC-based RL approach effectively optimized benefits for residential microgrid systems, greatly cutting economic costs while ensuring user comfort. The application also introduced the Shapley value method for equitable bill distribution among residents.
- Home Energy Management System (HEMS): The third application tackled a real-world problem in HEMS, dealing with discrepancies due to model mismatch and uncertainties in various parameters. The MPC-based RL approach was shown to deliver policies satisfying thermal comfort and economic costs, even with inaccurate models derived from model fitting.

In summary, the thesis contributes a nuanced understanding of the potential synergies between MPC and RL, unveiling an approach that transcends the boundaries of conventional methods. By applying and slightly modifying or improving the MPC-based RL method to formulate different algorithms across the three different applications, the research verifies its theoretical merits, proposes new solutions to challenging engineering problems, and identifies potential methodological issues.

Preface

This thesis is submitted in partial fulfillment of the requirements for the Degree of Philosophiae Doctor (Ph.D.) at the Norwegian University of Science and Technology (NTNU), Trondheim.
The research expounded within this document was executed within the purview of the Department of Engineering Cybernetics at NTNU. The supervision of this project was under the auspices of Professor Sebastien Gros, a distinguished academic from the Department of Engineering Cybernetics, whilst co-supervision was provided by Professor Pedro Crespo del Granado of the Department of Industrial Economics and Technology Management. The work was supported by the Research Council of Norway (RCN) (grant no. NFR 300172) project “Safe Reinforcement Learning using Model Predictive Control" (SARLEM) (project no. UV988962100) at NTNU.

Acknowledgements

First and foremost I am sincerely grateful to my project supervisor, Professor Sebastien Gros, from the Department of Engineering Cybernetics. His profound knowledge, guidance, and constructive criticism have been invaluable for the progress and successful completion of this research project.

My earnest appreciation extends to my collaborators and colleagues, Arash Bahari Kordabad, Hossein Nejatbakhsh Esfahani, Shambhuraj

Sawant, and Dirk Reinhardt. Their intellectual input, teamwork, and unwavering support have been essential in navigating the complexities of this project. Our collective discussions, shared ideas, and constructive critiques have greatly enhanced the quality of this research. Besides, to Kai, Jiayu, Kang, Kiet, Akhil, Aurora, Mika, Oliver, Trym, Erlend, Daniel, Mihir, and Nikhil, thank you for the countless moments of laughter and respite you provided amidst the intense periods of research.

I would also like to acknowledge the Norwegian University of Science and Technology (NTNU), for providing an intellectually stimulating environment that nurtured my academic growth. My heartfelt thanks to the Department of Engineering Cybernetics for all the resources, support, and motivation that enabled me to achieve my research goals.

In the end, I wish to express my gratitude to my family. Their unwavering faith, constant encouragement, and understanding during this demanding phase of my academic career have been a source of immense strength.

This journey has indeed been a collaborative endeavor, and I extend my sincere thanks to all those who have played a part in helping me reach this significant milestone.

June 2024, Trondheim

Wenqi Cai

Contents

Abbreviations

ASV Autonomous Surface Vehicle . . . . . iv, 3, 5, 27, 28, 153

DMPC Data-driven Model Predictive Control . . . iii, 1, 4, 5, 13, 14

DNN Deep Neural Network . . . . . iii, 2, 27, 28

DPG Deterministic Policy Gradient 23, 24

DRL Deep Reinforcement Learning 17, 18

EnNMPC Ensemble NMPC 111

HEMS Home Energy Management System . . . . . iv, 4, 5, 27, 153

LS Least Square 16, 25

LSTD Least-Squares Temporal-Difference 25, 26

MDP Markov Decision Process 2, 3, 15, 22

ML Machine Learning . . . . . iii, 1, 3, 13, 14

MPC Model Predictive Control iii, iv, 1–5, 11–14, 17–25, 27–29, 153, 154

MPC-based RL Model Predictive Control-based Reinforcement Learning . . . . . iii, iv, 2–5, 21, 23, 27–29, 153–155

NMPC Nonlinear Model Predictive Control 107

Abbreviations

NN Neural Network 18, 23

RERs Renewable Energy Resources 3

RL Reinforcement Learning . . . . . iii, iv, 2–5, 15–23, 27–29, 154

RMPC Robust MPC 12

SMPC Stochastic MPC 12

1 Introduction

This chapter briefly describes the motivation and objectives of this thesis, its outline and main contributions, and concludes with an overview of published papers.

1.1 Motivation and Objectives

Model Predictive Control (MPC) utilizes predictive models to minimize costs and manage constraints and has been widely adopted across various industries [1]. However, the construction of an accurate MPC model, especially for real systems with inherent stochasticity, remains a significant challenge. Inaccuracies in modeling can severely affect the performance of the MPC scheme, especially when the objective is not just to attain a specific state but to minimize a generic economic cost [2].

The challenges associated with building accurate MPC models have led to an exploration of integrating Machine Learning (ML) techniques to improve model precision. The classical approach has been to use ML to develop more accurate Data-driven Model Predictive Control (DMPC) models [3, 4]. This paradigm seeks to directly address the issues related to model inaccuracies, which become highly problematic when dealing with stochastic systems and economic objectives. However, the integration of ML into the MPC framework introduces new complexities. Increasing model accuracy typically demands greater model complexity, which can lead to more intricate MPC schemes. Consequently, DMPC

finds itself entwined with its own complexity, limiting the performance enhancement it could provide [5]. Moreover, the objective of the ML-based model, which is designed to deliver the best possible predictions, may not align with the actual control goals of the MPC scheme. Even model-free MPC techniques that aim to tailor predictions to MPC objectives have limitations, such as producing significantly suboptimal MPC policies, and often lack formal support, making them vulnerable to different regularization [6].

Reinforcement Learning (RL) offers an intriguing alternative to these challenges by focusing on policy learning through environment interaction. Unlike traditional methods that rely heavily on system models, RL can leverage data to adapt to uncertainties and disturbances [7]. However, conventional RL methodologies are not without their challenges. Firstly, for complex or highly uncertain systems, RL requires an enormous amount of data to learn the policy from scratch, making it less efficient [8]. Secondly, the frequent reliance on Deep Neural Network (DNN) of RL poses problems, as DNN-based RL lacks formal tools to handle system constraints and evaluate closed-loop stability. Moreover, there is no systematic or physically meaningful way to configure the intricate parameters of a DNN network, such as the initial values, number of hidden layers, and number of hidden units [9, 10]. Additionally, the incorporation of time-series predictive information in RL can result in the curse of dimensionality, exacerbating the complexities in model formulation and execution.

Considering these limitations, the authors in [11] first proposed an Model Predictive Control-based Reinforcement Learning (MPC-based RL) approach in combining the structure and constraints awareness of MPC with the adaptiveness and data-driven nature of RL. Instead of using DNN, this fusion applies a parametrized MPC as a function approximator for a specific Markov Decision Process (MDP) within the context of RL. It is proved that under a mild condition, by extensively parameterizing the MPC model, costs, and constraints, the parametrized MPC scheme could capture the optimal policy and value functions even in scenarios with inaccurate modeling or system uncertainties. This is achieved through the application of RL techniques such as Q-learning and policy

gradient, which are harnessed to adjust the MPC parameters to achieve optimized closed-loop performance. This approach bears significant merits, including the ability to leverage prior knowledge, explicitly consider system constraints, analyze stability, tackle long-term problems, and most notably, align the control strategy directly with performance objectives. However, the MPC-based RL method is in its infancy, and many practical applications remain unexplored.

Therefore, this thesis aims to take the nascent theory into the realm of practical engineering, focusing on applications marked by high uncertainty, noise, inaccurate models, economic cost functions, and complex challenges resistant to pure ML solutions.

- Autonomous Surface Vehicle (ASV): The development of a control strategy that ensures collision-free path tracking and autonomous docking amid time-varying disturbances presents a significant challenge, as the complex dynamics of the marine environment, coupled with intricate control requirements, lead to difficulties in model accuracy, stringent safety, and real-time responsiveness. MPC type methods struggle to encapsulate complex dynamics, including time-dependent perturbations, often leading to conservative solutions, while RL-type approaches face difficulties in terms of massive data requirements and ensuring safety within complex MDP [12].

- Energy Management in Residential Microgrids: The objective of Energy Management is to optimize energy dispatch, considering factors such as operating costs, power demand, and consumer preferences. However, the complex interplay between the volatile nature of Renewable Energy Resources (RERs) and customer load demand uncertainties presents significant challenges. MPC strategies suffer from problems related to model accuracy, adaptability to changing system characteristics, and difficulty in considering long-term objectives. Conversely, RL-based methods grapple with issues such as extensive data requirements and high-dimensional spaces due to time-series information [13, 14].

- Home Energy Management System (HEMS): Designing a HEMS strategy is also a complex optimization challenge. First, the thermodynamic models of buildings are inherently intricate and typically reduced to simplified forms that overlook factors like air permeability, furniture thermal mass, and occupancy changes. Second, the system must contend with myriad uncertainties such as renewable energy generation fluctuations, erratic household load demand, electricity price volatility, and weather forecast inconsistencies. MPC grapples with the complexities and uncertainties of real system dynamics, DMPC is heavily dependent on accurate data representation, and RL is hindered by data requirements, training time, and high-dimension issues [15, 16].

In the context of the three application problems delineated above, conventional methodologies such as MPC, DMPC, and RL have been extensively applied. However, as presented earlier, they inherently grapple with specific limitations that constrain their efficacy in these complex scenarios. Recognizing these challenges, the focus of this research is to meticulously design and implement MPC-based RL algorithms tailored to each individual problem. The MPC-based RL approaches synergize the strengths of both MPC and RL, enabling a more nuanced handling of system constraints, stability analysis, uncertainties, and long-term optimization, and could intrinsically enhance the closed-loop performance. The objective is to transcend the limitations of traditional approaches, aiming for a more nuanced and comprehensive solution to those issues. The proposed MPC-based RL solutions will be subjected to rigorous simulation, serving to validate their effectiveness and underscore their potential contributions to the field.

1.2 Outline and Contributions

The thesis is structured as follows. Chapter 1 sets the stage, elucidating the motivation, objectives, seminal contributions, and associated publications of this research. Chapter 2 delves into the foundational

underpinnings, introducing the rudiments of MPC, DMPC, and RL. It further offers a comprehensive elucidation of the core idea and theoretical clarification of the MPC-based RL approach, which is the underlying methodology utilized in this work. With a foundational understanding established, Chapter 3 to Chapter 5 shift focus to practical applications: Chapter 3 expounds on deploying the MPC-based RL method to address the ASV freight transportation conundrum; Chapter 4 harnesses the same methodology for the residential microgrid energy management challenge; while Chapter 5 is dedicated to its application in resolving the HEMS issue. Finally, Chapter 6 offers a comprehensive synthesis of the research insights and casts a vision for prospective explorations in the field of MPC&RL.

The main contribution of this thesis is that, in the scholarly realm of MPC-based RL, this thesis pioneers its application to tangible real-world contexts. Our deliberate adaptation of the MPC-based RL framework across three distinct applications not only validates the theory at its core but also introduces innovative engineering solutions. In doing so, we also highlight areas where the approach may falter, providing a foundation for subsequent methodological improvements and refinements. For more details, we further elaborate on the respective contributions in three application chapters, see subsection 3.1.3, subsection 4.1.4, and subsection 5.1.5.

1.3 Publications

Throughout the Ph.D. journey, concerted efforts in the field have resulted in the creation of 9 academic papers, with 4 of these attributing the candidate as the lead contributor. This compilation comprises 5 papers presented at conferences (with the candidate being the lead author in 2 instances), 3 articles prepared for scholarly journals (where the candidate has been acknowledged as the main author in 2 instances), and 1 paper written by invitation for an edited book. By the time this thesis was composed, 5 conference papers and 3 journal articles have successfully been published. The remaining works are in the peer review process.

Conference publications

i **Wenqi Cai**, Arash Bahari Kordabad, Hossein Nejatbakhsh Esfahani, Anastasios M. Lekkas, and Sebastien Gros. "MPC-based Reinforcement Learning for a Simplified Freight Mission of Autonomous Surface Vehicles". In: *2021 60th IEEE Conference on Decision and Control (CDC)* (2021), pp.2990-2995.

ii **Wenqi Cai**, Hossein Nejatbakhsh Esfahani, Arash Bahari Kordabad, and Sebastien Gros. "Optimal Management of the Peak Power Penalty for Smart Grids Using MPC-based Reinforcement Learning". In: *2021 60th IEEE Conference on Decision and Control (CDC)* (2021), pp.6365-6370.

iii Arash Bahari Kordabad, **Wenqi Cai**, and Sebastien Gros. "MPC-based reinforcement learning for economic problems with application to battery storage". In: *2021 European Control Conference (ECC)* (2021), pp.2573-2578.

iv Arash Bahari Kordabad, **Wenqi Cai**, and Sebastien Gros. "Multi-agent Battery Storage Management using MPC-based Reinforcement Learning". In: *2021 IEEE Conference on Control Technology and Applications (CCTA)* (2021), pp.57-62.

v Arash Bahari Kordabad, Hossein Nejatbakhsh Esfahani, **Wenqi Cai**, and Sebastien Gros. "Quasi-Newton Iteration in Deterministic Policy Gradient". In: *2022 American Control Conference (ACC)* (2022), pp.2124-2129.

Journal publications

xiv **Wenqi Cai**, Arash Bahari Kordabad, and Sebastien Gros. "Energy Management in Residential Microgrid Using Model Predictive Control-based Reinforcement Learning and Shapley Value". In: *Engineering Applications of Artificial Intelligence*, vol.119, (2023), pp.105793.

[xv] **Wenqi Cai**, Shambhuraj Sawant, Dirk Reinhardt, Soroush Rastegarpour, and Sebastien Gros. "A Learning-Based Model Predictive Control Strategy for Home Energy Management Systems". In: *IEEE access*, vol.11, (2023), pp.145264.

[xvi] Hossein Nejatbakhsh Esfahani, Arash Bahari Kordabad, **Wenqi Cai**, and Sebastien Gros. "Learning-based State Estimation and Control using MHE and MPC Schemes with Imperfect Models". In: *European Journal of Control*, (2023), pp.100880.

Book chapters

[xv] Dirk Reinhardt, **Wenqi Cai**, and Sebastien Gros. "Data-Driven Domestic Flexible Demand: observations from experiments in cold climate". In: *Towards Energy Systems Integration for Multi-Energy Systems: From Operation to Planning in the Green Energy Context*, (Expected to be published by Springer in the third quarter of 2024), (Under review).

References

[1] James Blake Rawlings and David Q Mayne. "Model predictive control: theory and design, Nob Hill Pub". In: *Madison, Wisconsin* 825 (2009).

[2] Manfred Morari and Jay H Lee. "Model predictive control: past, present and future". In: *Computers & chemical engineering* 23.4-5 (1999), pp. 667–682.

[3] Juš Kocijan, Roderick Murray-Smith, Carl Edward Rasmussen, and Bojan Likar. "Predictive control with Gaussian process models". In: *The IEEE Region 8 EUROCON 2003. Computer as a Tool.* Vol. 1. IEEE. 2003, pp. 352–356.

[4] Eduardo F Camacho, Carlos Bordons, Eduardo F Camacho, and Carlos Bordons. *Model predictive controllers*. Springer, 2007.

[5] Ugo Rosolia and Francesco Borrelli. “Learning model predictive control for iterative tasks. a data-driven control framework”. In: *IEEE Transactions on Automatic Control* 63.7 (2017), pp. 1883–1896.

[6] Florian Dörfler, Jeremy Coulson, and Ivan Markovsky. “Bridging direct and indirect data-driven control formulations via regularizations and relaxations”. In: *IEEE Transactions on Automatic Control* 68.2 (2022), pp. 883–897.

[7] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[8] Kai Arulkumaran, Marc Peter Deisenroth, Miles Brundage, and Anil Anthony Bharath. “A brief survey of deep reinforcement learning”. In: *arXiv preprint arXiv:1708.05866* (2017).

[9] Sebastien Gros and Mario Zanon. “Economic MPC of Markov Decision Processes: Dissipativity in undiscounted infinite-horizon optimal control”. In: *Automatica* 146 (2022), p. 110602.

[10] Sebastien Gros and Mario Zanon. “Reinforcement Learning for Mixed-Integer Problems Based on MPC”. In: *arXiv preprint arXiv:2004.01430* (2020).

[11] Sébastien Gros and Mario Zanon. “Data-driven economic nmpc using reinforcement learning”. In: *IEEE Transactions on Automatic Control* 65.2 (2019), pp. 636–648.

[12] Zhixiang Liu, Youmin Zhang, Xiang Yu, and Chi Yuan. “Unmanned surface vehicles: An overview of developments and challenges”. In: *Annual Reviews in Control* 41 (2016), pp. 71–93.

[13] Muhammad Fahad Zia, Elhoussin Elbouchikhi, and Mohamed Benbouzid. “Microgrids energy management systems: A critical review on methods, solutions, and prospects”. In: *Applied energy* 222 (2018), pp. 1033–1055.

[14] Yimy E García Vera, Rodolfo Dufo-López, and José L Bernal-Agustín. “Energy management in microgrids with renewable energy sources: A literature review”. In: *Applied Sciences* 9.18 (2019), p. 3854.

[15] Marc Beaudin and Hamidreza Zareipour. “Home energy management systems: A review of modeling and complexity”. In: *Renewable and sustainable energy reviews* 45 (2015), pp. 318–335.

[16] Hussain Shareef, Maytham S Ahmed, Azah Mohamed, and Eslam Al Hassan. “Review on home energy management system considering demand responses, smart technologies, and intelligent controllers”. In: *Ieee Access* 6 (2018), pp. 24498–24509.

2 | Background

2.1 Model Predictive Control

MPC has emerged as a preferred approach to optimal control, particularly valued for its adeptness at addressing both input and state constraints [1]. The essence of MPC lies in its iterative procedure: at every instance, it evaluates the control and state sequence that minimizes a specified cost function over a defined prediction horizon, all while respecting system constraints.

Formally, given a system state $\mathbf{s}$, the MPC methodology is rooted in the continuous resolution of the optimal control problem, articulated as:

$$\min_{\mathbf{x},\mathbf{u}} \quad T\left(\mathbf{x}_N\right) + \sum_{k=0}^{N-1} L\left(\mathbf{x}_k, \mathbf{u}_k\right) \tag{2.1a}$$

$$\text{s.t.} \quad \forall\ k = 0, \ldots, N-1$$

$$\mathbf{x}_{k+1} = \mathbf{f}\left(\mathbf{x}_k, \mathbf{u}_k\right), \tag{2.1b}$$

$$\mathbf{h}\left(\mathbf{x}_k, \mathbf{u}_k\right) \leq 0, \quad \mathbf{h}^f\left(\mathbf{x}_N\right) \leq 0, \tag{2.1c}$$

$$\mathbf{x}_0 = \mathbf{s}. \tag{2.1d}$$

This optimization yields an optimal control input sequence $\mathbf{u}^\star$, and the associated state predictions $\mathbf{x}^\star$. Conventionally, only the first control action $\mathbf{u}_0^\star$ is implemented on the system. As the system advances to the subsequent sampling time, the optimization is solved again over the shifted horizon.

MPC has a solid theoretical foundation and has been widely and successfully applied. However, it has some limitations:

- Model Dependence: If the model does not capture the true dynamics or uncertainties of the system, the performance of MPC can degrade, leading to suboptimal or even unstable control actions. And for many complex systems in the real world, obtaining accurate system models is difficult or even impossible [2].

- Computational Intensity: When dealing with intricate problems, even when an accurate representation of a system is obtainable, the resultant model can be inherently complicated. Such complexity invariably escalates the computational demands of MPC, especially when solving the optimization problem online at each sampling instant [3].

- Cost Function Design: The design of the cost function can be intricate and sometimes more of an art than a science [4]. An improperly designed cost function can lead to undesirable control behaviors or convergence to non-optimal solutions.

- Robustness Issues: While conventional MPC may struggle with guaranteeing robustness amid model uncertainties or disturbances, there are specialized variants such as Stochastic MPC (SMPC) and Robust MPC (RMPC) aimed at addressing these challenges. SMPC, tailored to grapple with uncertainties by accounting for stochastic models, often entails increased computational demands and can be complex to implement due to the intricacy of probabilistic constraints and the necessity of scenario generation or stochastic simulations [5, 6]. On the other hand, RMPC, designed to remain feasible under a range of disturbance or model error scenarios, tends to yield over-conservative control policies. Such conservatism can limit the system's operational efficiency and can be overly restrictive in scenarios where a more aggressive control strategy might be beneficial [7].

2.2 Data-driven Model Predictive Control

DMPC, a.k.a learning-based MPC, represents a paradigm shift from traditional model-based MPC towards harnessing the power of ML techniques to either construct the predictive models directly from data or to utilize MPC for the generation of training datasets for ML controllers [8]. Mathematically, considering the first variant, assuming we have a dataset $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{u}_i, \mathbf{x}_{i+1})\}_{i=1}^{N}$ comprising of state-input-next state triples. A predictive model $\mathbf{f}_\theta$ is learned, parameterized by $\boldsymbol{\theta}$, such that:

$$\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x}_i, \mathbf{u}_i) \approx \mathbf{x}_{i+1}. \tag{2.2}$$

This learned model can be integrated into the MPC optimization framework replacing traditional dynamics. For the second variant, MPC can be utilized to generate state-input trajectories that are subsequently used to train an ML-based controller, denoted as $\pi_{\boldsymbol{\phi}}(\mathbf{x})$, parameterized by $\boldsymbol{\phi}$.

DMPC offers several advantages over traditional MPC approaches. Primarily, DMPC inherently adapts to changing dynamics by refining its predictive models using observational data, negating the need for models based on first principles (deriving models from underlying physics), which might not always be accurate or available [8]. This adaptability also enables DMPC to effectively manage non-linear systems without requiring linear approximations, a common constraint in traditional MPC, as ML techniques are inherently adept at capturing non-linear relationships [9]. Moreover, data-based models can reflect the stochastic nature of the system and thus can improve the robustness of the control strategy to some extent [10]. Finally, DMPC that leveraging pre-trained ML controllers can reduce computational costs during real-time decision-making compared to traditional MPC's on-the-fly optimization.

However, DMPC still suffers from several limitations:

- Potential for Overfitting in Model Training: For DMPC variants directly training predictive models from data, there is an inherent risk of overfitting, especially when the training data isn't representative of all possible operating conditions [11]. Overfitting implies

that while the model might work impeccably on the seen data, its performance can drastically degrade when exposed to unseen conditions, leading to unpredictable and potentially unsafe control actions.

- Model Explainability and Trustworthiness: For DMPC variants using MPC-generated data to train ML controllers, the resulting controllers might lack explainability. Machine learning models, especially complex ones like deep neural networks, are notoriously difficult to interpret. This black-box nature can be problematic in critical systems where understanding control decisions is crucial [12].

- Data Quality and Coverage Dependency: Both types of DMPC methodologies intrinsically depend on the quality and coverage of the data. If the dataset harbors biases, is noisy, or lacks samples from critical regions of the state space, the performance of the DMPC can be severely compromised. Ensuring a comprehensive and high-quality data collection can be resource-intensive and not always feasible in practice [13].

- Consistency with Physical Reality: Learned models, especially those trained on limited datasets, might not capture the true underlying physics of the system. This could lead to scenarios where the model suggests control actions that, while optimal according to its training, are misaligned with the actual dynamics of the system, potentially compromising safety and efficiency.

- Model Objectives Misalignment: A core issue with DMPC (those using ML-based models) resides in the dichotomy between modeling objectives and control goals [14]. Traditional ML frameworks tailored for DMPC primarily focus on ensuring the accuracy of system predictions. Optimizing prediction accuracy, while invaluable, does not always guarantee optimal control performance in a real-time scenario. This arises from the fact that the model is trained to minimize the prediction error, rather than directly optimize a control-relevant objective. As demonstrated in some

studies, there exist scenarios in which a highly accurate predictive model results in a significantly suboptimal control policy.

2.3 Reinforcement Learning

In the realm of RL, the fundamental framework revolves around the MDP, which serves as the cornerstone for modeling sequential decision-making problems and obeys the Markov property: Transitions only depend on the most recent state and action, and no prior history. RL considers real-world systems as instances of MDP, where the state transitions are characterized by the conditional probability density $P[\mathbf{s}'|\mathbf{s}, \mathbf{a}]$. Here, $\mathbf{s}$ represents the current state, $\mathbf{a}$ denotes the chosen action, and $\mathbf{s}'$ indicates the subsequent state.

Central to RL is the concept of the Bellman Equation, which is the bedrock of deriving optimal policies. The Bellman Equation mathematically encapsulates the relationship between the value of a state and the expected cumulative rewards attainable from that state onwards. Formally, the optimum value function $V^\star(\mathbf{s})$ for a given state $\mathbf{s}$ is determined as the minimum of the optimal action-value function $Q^\star(\mathbf{s}, \mathbf{a})$ over all possible actions:

$$V^\star(\mathbf{s}) = \min_{\mathbf{a}} Q^\star(\mathbf{s}, \mathbf{a}). \tag{2.3}$$

The action-value function $Q^\star(\mathbf{s}, \mathbf{a})$ integrates the immediate reward, characterized by the baseline stage cost $L(\mathbf{s}, \mathbf{a})$, with the discounted expected value of future states that emanate from taking action $\mathbf{a}$ in state $\mathbf{s}$:

$$Q^\star(\mathbf{s}, \mathbf{a}) = L(\mathbf{s}, \mathbf{a}) + \gamma \mathbb{E}[V^\star(\mathbf{s}')|\mathbf{s}, \mathbf{a}]. \tag{2.4}$$

Here, $\gamma \in (0, 1]$ represents the MDP discount factor, influencing the balance between immediate rewards and future values. Then, each state's associated optimal action is the one minimizing the corresponding optimal action-value function:

$$\mathbf{a}^\star(\mathbf{s}) = \arg\min_{\mathbf{a}} Q^\star(\mathbf{s}, \mathbf{a}). \tag{2.5}$$

The objective in RL is to determine a policy that minimizes the expected cumulative stage cost when the agent follows it, which is quantified as follows:

$$J(\boldsymbol{\pi}) = \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k L\left(\mathbf{s}_k, \mathbf{a}_k\right) \,\middle|\, \mathbf{a}_k = \boldsymbol{\pi}\left(\mathbf{s}_k\right)\right]. \tag{2.6}$$

Consequently, RL seeks to find an optimal policy $\boldsymbol{\pi}^\star$ that minimizes $J(\boldsymbol{\pi})$:

$$\boldsymbol{\pi}^\star = \arg\min_{\boldsymbol{\pi}} J(\boldsymbol{\pi}). \tag{2.7}$$

Two fundamental types of RL algorithms are Q-learning and Policy Gradient methods.

- Q-learning: Methods in this family learn an approximator $Q_{\boldsymbol{\theta}}(\mathbf{s}, \mathbf{a})$ for the optimal action-value function $Q^\star(\mathbf{s}, \mathbf{a})$. The update rule lies in minimizing the following Least Square (LS) problem:

  $$\min_{\boldsymbol{\theta}} \mathbb{E}\left[\left(Q_{\boldsymbol{\theta}}(\mathbf{s}, \mathbf{a}) - Q^\star(\mathbf{s}, \mathbf{a})\right)^2\right]. \tag{2.8}$$

  The iterative parameter update process is driven by the Bellman equation, wherein parameters $\boldsymbol{\theta}$ are updated as follows:

  $$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \alpha[L(\mathbf{s}, \mathbf{a}) + \gamma \min_{\mathbf{a}'} Q_{\boldsymbol{\theta}}(\mathbf{s}', \mathbf{a}') - Q_{\boldsymbol{\theta}}(\mathbf{s}, \mathbf{a})]\nabla_{\boldsymbol{\theta}} Q_{\boldsymbol{\theta}}(\mathbf{s}, \mathbf{a}), \tag{2.9}$$

  where $\alpha > 0$ is the learning rate. Then the policy is obtained via the connection between $Q^\star$ and $\pi^\star$:

  $$\pi^\star(\mathbf{s}) = \arg\min_{\mathbf{a}} Q_{\theta^\star}(\mathbf{s}, \mathbf{a}). \tag{2.10}$$

- Policy Gradient: Methods in this family explicitly approximate the optimal policy $\pi^\star(\mathbf{a}|\mathbf{s})$ by $\pi_{\boldsymbol{\theta}}(\mathbf{a}|\mathbf{s})$. The parameters $\boldsymbol{\theta}$ are updated either directly by gradient descent on the performance objective $J(\boldsymbol{\pi}_{\boldsymbol{\theta}})$, or indirectly, by maximizing local approximations of $J(\boldsymbol{\pi}_{\boldsymbol{\theta}})$:

  $$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \alpha \nabla_{\boldsymbol{\theta}} J(\boldsymbol{\pi}_{\boldsymbol{\theta}}). \tag{2.11}$$

RL offers a transformative approach to control systems design, its most significant advantage being its model-free nature. Traditional control methods often necessitate precise models of system dynamics, which can be arduous to derive for complex real-world systems. RL, however, allows control design without exhaustive knowledge of system dynamics [15]. Moreover, RL's adaptive nature makes it well-suited to deal with uncertainties, non-stationarities, and evolving system dynamics [16]. And They learn through interaction with their environment, thereby continually refining their policies as new data become available. Additionally, the integration of deep learning with RL, known as Deep Reinforcement Learning (DRL), extends the methodology's capability to manage high-dimensional state and action spaces effectively. Such capability is crucial in modern control challenges, such as robotics, autonomous vehicles, and intricate industrial processes where controllers must process rich sensor data, and this integration allows RL algorithms to automatically identify relevant features from raw data [17].

Despite its advantages, RL also carries with it inherent limitations :

- Sample Inefficiency: One of the primary criticisms of RL is its sample inefficiency. Many RL techniques, especially in their nascent learning stages, demand vast amounts of data through exploration to converge to optimal or near-optimal strategies. The authors in [18] emphasize this challenge, highlighting how RL's data-hungry nature can be problematic, especially in environments where collecting data is expensive, time-consuming, or risky.

- Safety Concerns and Constraint Handling: The explorative nature of RL, vital for its learning process, can inadvertently result in the selection of unsafe policies. Particularly during initial training phases, RL agents might pursue actions that are detrimental or suboptimal, aiming to thoroughly explore their environment [19]. This poses significant challenges in domains where safety is paramount, such as automotive control systems. Furthermore, integrating constraints directly within RL is not straightforward. Unlike MPC where safety constraints are explicitly handled, RL typically addresses these constraints by penalizing their violations

within the reward function. This indirect approach introduces potential risks, as constraints might not be rigorously adhered to, and the severity of violations might not be sufficiently captured by the penalty in the reward function.

- Interpretability and Transparency: RL often uses Neural Network (NN) as function approximations. These "black-boxes" DRL models offer little transparency in the decision-making processes. This can be particularly concerning in control applications where understanding why certain actions are taken is as crucial as the actions themselves. The interpretability challenge extends not just to end-users but also to researchers and developers who might find debugging or refining these models a daunting task due to the layers of abstraction [20].

- Theoretical Underpinnings and Guarantees: MPC is grounded in well-established mathematical principles that provide concrete guarantees on system behavior, stability, and convergence. In contrast, the theoretical framework of RL, especially when considering real-world, non-episodic tasks, is less established. Questions about convergence, optimality, and stability, while addressed in academic literature, don't always have clear-cut answers in the context of RL [21].

- Dependency on Reward Design: Crafting an appropriate reward function for RL can be intricate. An inadequately designed reward might lead an agent to undesirable behavior or local optima [22].

2.4 Model Predictive Control-based Reinforcement Learning

The combination of MPC and RL is promising and has been explored in various recent research efforts. The integration seeks to harness the model-based predictive capability (as well as its ability to handle

constraints and its solid theoretical foundation) of MPC and the adaptive, data-driven learning capabilities of RL.

2.4.1 Other approaches

To the best of our knowledge, there are roughly four mainstream approaches to combine MPC and RL.
(1) MPC as a Guiding Policy for RL training [23, 24]:

- Principle: Use MPC as a "teacher" to guide the learning of an RL agent to acquire good initial policies quickly. Essentially, the MPC, with its model-based foresight, provides expert trajectories or actions, and the RL agent learns from these.
- Advantage: This can significantly speed up the learning process, especially in the early stages, because the RL agent gets guidance from the model-based expertise of MPC.
- Limitations: This approach requires a certain level of model accuracy within the MPC. Besides, relying solely on MPC for initial training can lead to sample inefficiency since MPC itself requires multiple roll-outs to compute control actions. For some complex problems, the iterative nature of MPC might introduce overhead during the learning phase.

(2) Learning the Dynamics for MPC using RL [23, 25]:

- Principle: In environments where the system dynamics are uncertain or non-stationary, RL (specifically model-based RL) can be used to learn or refine these dynamics. This learned model is then used in the MPC framework for control.
- Advantage: MPC can handle systems where the precise dynamics are initially unknown.

- Limitations: Learned dynamics might not perfectly represent the real-world system, and/or the model is trained to minimize prediction error rather than directly optimize the control objective.

(3) RL for Cost Function Design in MPC [26, 27]:

- Principle: Often, designing an appropriate cost function for MPC (that truly represents desired behavior) can be challenging. RL can be utilized to learn or adapt this cost function based on feedback from the environment.

- Advantage: This ensures that the MPC's objective aligns well with the desired outcomes in the real-world system.

- Limitations: If the system model is poorly known or the uncertainties are high, modifying the cost function alone may be far from sufficient to eliminate the effects of model errors.

(4) Nested MPC-RL [28, 29]:

- Principle: An MPC scheme operates at a higher level, making decisions based on a model, while an RL agent operates at a lower level, making real-time decisions. The MPC provides setpoints or reference trajectories for the RL agent, which then takes these setpoints as goals and determines how to achieve them in real-time, typically handling finer-grained decisions or adapting to unmodeled disturbances.

- Advantage: Combines the foresight and constraint-handling abilities of MPC with the adaptability of RL.

- Limitations: Having a two-level decision-making process might introduce latency, especially in fast-changing environments. And properly integrating the feedback from RL into the MPC and vice-versa can be non-trivial, requiring careful design.

In the presented review, while we strive for thoroughness, constraints of scope and depth might lead to an incomplete coverage of all MPC&RL methodologies. Nonetheless, our summary underscores that existing approaches often retain some inherent limitations of both MPC and RL, without achieving a seamless integration. This observation motivated us to adopt our MPC-based RL methodology, which aims to integrate the two paradigms in a more harmonized manner.

2.4.2 Our approach

(1) Core Theory

The MPC-based RL method employed in this thesis was first proposed by Sebastien Gros and Mario Zanon in 2019 [14] and centers on the following theorem.

Theorem 1. *Consider a fully parameterized MPC:*

$$\min_{\mathbf{x},\mathbf{u},\boldsymbol{\sigma}} \quad T_{\boldsymbol{\theta}}\left(\mathbf{x}_N\right) + \boldsymbol{\omega}_f^\top \boldsymbol{\sigma}_N + \sum_{k=0}^{N-1} \left(L_{\boldsymbol{\theta}}\left(\mathbf{x}_k, \mathbf{u}_k\right) + \boldsymbol{\omega}^\top \boldsymbol{\sigma}_k\right), \tag{2.12a}$$

$$\text{s.t.} \quad \forall\ k = 0, \ldots, N-1$$

$$\mathbf{x}_{k+1} = \mathbf{f}_{\boldsymbol{\theta}}\left(\mathbf{x}_k, \mathbf{u}_k\right), \tag{2.12b}$$

$$\mathbf{g}(\mathbf{u}_k) \leq 0, \tag{2.12c}$$

$$\mathbf{h}_{\boldsymbol{\theta}}\left(\mathbf{x}_k, \mathbf{u}_k\right) \leq \boldsymbol{\sigma}_k, \quad \mathbf{h}_{\boldsymbol{\theta}}^f\left(\mathbf{x}_N\right) \leq \boldsymbol{\sigma}_N, \tag{2.12d}$$

$$\boldsymbol{\sigma}_k \geq 0, \quad \boldsymbol{\sigma}_N \geq 0 \tag{2.12e}$$

$$\mathbf{x}_0 = \mathbf{s}. \tag{2.12f}$$

Given the MPC scheme as described in (2.12), where the parameterization is "rich" enough such that the stage cost $L_{\boldsymbol{\theta}}(\cdot)$, terminal cost $T_{\boldsymbol{\theta}}(\cdot)$, model $\mathbf{f}_{\boldsymbol{\theta}}(\cdot)$, and constraints $\mathbf{h}_{\boldsymbol{\theta}}(\cdot)$ are universal function approximators, and exact relaxations $\boldsymbol{\sigma}$ (slack variables) are ensured with sufficiently large $\boldsymbol{\omega}$ and $\boldsymbol{\omega}_f$, then there exist parameters $\boldsymbol{\theta}^\star$ such that the following identities hold on $\forall \mathbf{s} \in S$:

1. $V_{\boldsymbol{\theta}^\star}(\mathbf{s}) = V^\star(\mathbf{s})$,
2. $\boldsymbol{\pi}_{\boldsymbol{\theta}^\star}(\mathbf{s}) = \boldsymbol{\pi}^\star(\mathbf{s})$,
3. $Q_{\boldsymbol{\theta}^\star}(\mathbf{s}, \mathbf{a}) = Q^\star(\mathbf{s}, \mathbf{a})$ *for the inputs* $\mathbf{a}$ *such that* $|\mathbb{E}[V^\star(\mathbf{s}')|\mathbf{s}, \mathbf{a}]| < \infty]$,

if the set

$$\mathcal{S} =: \left\{ \mathbf{s} \in S \;\middle|\; \left| \mathbb{E}\left[V^\star(\mathbf{s}_k^{\boldsymbol{\pi}^\star}) \right] \right| < \infty \right\} \tag{2.13}$$

is non-empty.

See [14, 30] for a rigorous proof of the theorem.

Theorem 1 presents a pivotal connection between MPC and the principles of MDP in the realm of RL. It suggests that, by intensively parameterizing aspects such as the system model, costs, and constraints, an MPC framework is capable of capturing the optimal value functions and policies inherent to a specific MDP, even in situations marred by model inaccuracies or system uncertainties. The integration of RL methodologies, notably Q-learning and policy gradient, facilitates the tuning of these MPC parameters, pushing the system towards optimal closed-loop performance. In simpler terms, provided the conditions are met, (2.12) can effectively serve as a surrogate, delivering the optimal policy of the real original MDP, regardless of potential model inaccuracies and uncertainties. Therefore, an obvious conclusion is that one can use a fully parameterized MPC as function approximators of the value function or the policy, and then use RL to tune the parameters based on the principle of minimizing the closed-loop performance.

The condition presented in (2.13) can be viewed as a type of stability criterion for $\mathbf{f}_{\boldsymbol{\theta}^\star}$ following the optimal trajectory. In essence, this requirement mandates that there exists a set S wherein the optimal value function $V^\star$ of the predicted optimal trajectory is finite with a unitary probability for every initial state emanating from that set. This assumption limits the indiscriminate use of any model within the MPC

framework although this stability criterion is less stringent. In practice, it is not feasible to confirm this assumption directly. However, we can avoid explicitly verifying this assumption in the MPC-based RL approach. By integrating RL and MPC in a fully parameterized form, the stability question can be approached as a fairly simple a priori design requirement rather than a complex a posteriori verification. To this end, we can modify the stage cost function with a specific condition ($L_{\boldsymbol{\theta}}(\mathbf{s}, \mathbf{a}) \geq \alpha(||\mathbf{s} - \bar{\mathbf{s}}_{\boldsymbol{\theta}}||)$, where $\bar{\mathbf{s}}_{\boldsymbol{\theta}}$ is the steady state) and integrate a storage function into the MPC cost.

Earlier in the discourse, we highlighted two predominant methodologies for parameter updating: Q-learning and the policy gradient method. This thesis predominantly harnesses the policy gradient approach, and for good reason. The policy gradient method boasts a suite of advantages over Q-learning, including its inherent capacity for direct policy optimization and reduced overestimation bias. Crucially, when integrated within the MPC-based RL framework, policy gradient emerges superior in efficiency: Q-learning requires that the MPC optimization problem be solved twice per iterative update [31], resulting in substantial computational overhead, while the policy gradient method only needs to solve it once. Specifically, this thesis adopts the Deterministic Policy Gradient (DPG) approach, as the problems addressed herein require deterministic policies.

(2) Core Formulas: DPG for MPC-based RL

Due to the use of parameterized MPCs instead of NNs for function approximation in RL, the DPG update equation deviates somewhat from its conventional form, where the primary distinction arises in the computation of the gradient. This section details how the DPG method adjusts the parameters $\boldsymbol{\theta}$ of the MPC scheme (2.12). The DPG method [32], as a direct RL approach, directly optimizes the policy parameters $\boldsymbol{\theta}$ via gradient descent steps on the performance function J, As we showed earlier in (2.11), i.e.,

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \boldsymbol{\eta} \odot \nabla_{\boldsymbol{\theta}} J(\boldsymbol{\pi}_{\boldsymbol{\theta}}), \tag{2.14}$$

where $\boldsymbol{\eta} > \mathbf{0}$ is the learning step-size vector and "$\odot$" represents the element-wise product. Applying the DPG method developed by [33], the gradient of J with respect to parameters $\boldsymbol{\theta}$ is obtained as

$$\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\pi}_{\boldsymbol{\theta}}) = \mathbb{E}_{\boldsymbol{\pi}_{\boldsymbol{\theta}}} \left[\nabla_{\boldsymbol{\theta}} \boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathbf{s}) \nabla_{\mathbf{a}} Q_{\boldsymbol{\pi}_{\boldsymbol{\theta}}}(\mathbf{s},\, \mathbf{a})|_{\mathbf{a}=\boldsymbol{\pi}_{\boldsymbol{\theta}}} \right], \tag{2.15}$$

where $Q_{\boldsymbol{\pi}_{\boldsymbol{\theta}}}$ and its inner function $V_{\boldsymbol{\pi}_{\boldsymbol{\theta}}}$ are the action-value function and value function associated to the policy $\boldsymbol{\pi}_{\boldsymbol{\theta}}$, respectively, defined as follows

$$Q_{\boldsymbol{\pi}_{\boldsymbol{\theta}}}(\mathbf{s},\, \mathbf{a}) = L(\mathbf{s},\, \mathbf{a}) + \gamma \mathbb{E}_{\boldsymbol{\pi}_{\boldsymbol{\theta}}} \left[V_{\boldsymbol{\pi}_{\boldsymbol{\theta}}}\left(\mathbf{s}^{+} | \mathbf{s},\, \mathbf{a}\right) \right], \tag{2.16a}$$

$$V_{\boldsymbol{\pi}_{\boldsymbol{\theta}}}(\mathbf{s}) = Q_{\boldsymbol{\pi}_{\boldsymbol{\theta}}}(\mathbf{s},\, \boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathbf{s})), \tag{2.16b}$$

where $\mathbf{s}^{+}$ is the subsequent state of the state-input pair $(\mathbf{s},\, \mathbf{a})$. The calculations of $\nabla_{\boldsymbol{\theta}} \boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathbf{s})$ and $\nabla_{\mathbf{a}} Q_{\boldsymbol{\pi}_{\boldsymbol{\theta}}}(\mathbf{s},\, \mathbf{a})$ in (2.15) are discussed in the following.

(a) $\nabla_{\boldsymbol{\theta}} \boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathbf{s})$

The Karush Kuhn Tucker (KKT) condition used in the primal-dual interior-point method underlying the MPC scheme (2.12) is written as

$$\mathbf{R} = \begin{bmatrix} \nabla_{\boldsymbol{\zeta}} \mathcal{L}_{\boldsymbol{\theta}}{}^{\top} & \mathbf{G}_{\boldsymbol{\theta}}{}^{\top} & \operatorname{diag}(\boldsymbol{\mu})\, \mathbf{H}_{\boldsymbol{\theta}} + \boldsymbol{\tau} \end{bmatrix}^{\top}, \tag{2.17}$$

where $\boldsymbol{\zeta} = \{\mathbf{x}, \mathbf{u}, \boldsymbol{\sigma}\}$ is the primal decision variable of the MPC and $\mathcal{L}_{\boldsymbol{\theta}}$ is the associated Lagrange function, written as

$$\mathcal{L}_{\boldsymbol{\theta}}(\mathbf{y}) = \Omega_{\boldsymbol{\theta}} + \boldsymbol{\lambda}^{\top} \mathbf{G}_{\boldsymbol{\theta}} + \boldsymbol{\mu}^{\top} \mathbf{H}_{\boldsymbol{\theta}}, \tag{2.18}$$

where $\Omega_{\boldsymbol{\theta}}$ is the MPC cost (2.12a), vector $\mathbf{G}_{\boldsymbol{\theta}}$ gathers the equality constraints and $\mathbf{H}_{\boldsymbol{\theta}}$ collects the inequality constraints of the MPC (2.12). Vectors $\boldsymbol{\lambda}, \boldsymbol{\mu}$ are the associated dual variables. Argument $\mathbf{y}$ is read as $\mathbf{y} = \{\boldsymbol{\zeta},\, \boldsymbol{\lambda},\, \boldsymbol{\mu}\}$ and $\mathbf{y}^{\star}$ refers to the solution of the MPC. Consequently, the policy sensitivity $\nabla_{\boldsymbol{\theta}} \boldsymbol{\pi}_{\boldsymbol{\theta}}$ required in (2.15) can be calculated as [14]

$$\nabla_{\boldsymbol{\theta}} \boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathbf{s}) = -\nabla_{\boldsymbol{\theta}} \mathbf{R}(\mathbf{y}^{\star},\, \mathbf{s},\, \boldsymbol{\theta})\, \nabla_{\mathbf{y}} \mathbf{R}(\mathbf{y}^{\star},\, \mathbf{s},\, \boldsymbol{\theta})^{-1} \frac{\partial \mathbf{y}}{\partial \mathbf{u}_0}, \tag{2.19}$$

where $\mathbf{u}_0$ is the first element of the MPC input solution.

(b) $\nabla_{\mathbf{a}} Q_{\boldsymbol{\pi}_{\boldsymbol{\theta}}}(\mathbf{s}, \mathbf{a})$

Under some conditions[1], the action-value function $Q_{\boldsymbol{\pi}_{\boldsymbol{\theta}}}$ can be replaced by an approximator $Q_{\mathbf{w}}$, i.e. $Q_{\mathbf{w}} \approx Q_{\boldsymbol{\pi}_{\boldsymbol{\theta}}}$, without affecting the policy gradient. Such an approximation is labeled *compatible* and, in this thesis, takes the following form

$$Q_{\mathbf{w}}(\mathbf{s},\, \mathbf{a}) = \underbrace{\left(\mathbf{a} - \boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathbf{s})\right)^{\top} \nabla_{\boldsymbol{\theta}} \boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathbf{s})^{\top}}_{\boldsymbol{\Psi}^{\top}(\mathbf{s},\, \mathbf{a})} \mathbf{w} + V_{\mathbf{v}}(\mathbf{s}), \tag{2.20}$$

where $\boldsymbol{\Psi}(\mathbf{s},\, \mathbf{a})$ is the state-action feature vector and $\mathbf{w}$ is the parameters vector. Component $V_{\mathbf{v}} \approx V_{\boldsymbol{\pi}_{\boldsymbol{\theta}}}$ is the parameterized baseline function approximating the true value function. It can take a linear form as

$$V_{\mathbf{v}}(\mathbf{s}) = \boldsymbol{\Phi}(\mathbf{s})^{\top} \mathbf{v}, \tag{2.21}$$

where $\boldsymbol{\Phi}(\mathbf{s})$ is the state feature vector and $\mathbf{v}$ is the corresponding parameters vector. The state feature vector $\boldsymbol{\Phi}(\mathbf{s})$ should be designed on a problem-by-problem basis, which may require some expert knowledge related to the system of the problem. Altogether, by adopting the approximation function, we will have

$$\nabla_{\mathbf{a}} Q_{\boldsymbol{\pi}_{\boldsymbol{\theta}}}(\mathbf{s},\, \mathbf{a}) \approx \nabla_{\mathbf{a}} Q_{\mathbf{w}}(\mathbf{s},\, \mathbf{a}) = \nabla_{\boldsymbol{\theta}} \boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathbf{s})^{\top} \mathbf{w}, \tag{2.22}$$

where the parameters $\mathbf{w}$ and $\mathbf{v}$ of the action-value function approximation (2.20) can be obtained by solving the Least Squares (LS) problem

$$\min_{\mathbf{w},\, \mathbf{v}} \mathbb{E}_{\boldsymbol{\pi}_{\boldsymbol{\theta}}} \left[\left(Q_{\boldsymbol{\pi}_{\boldsymbol{\theta}}}(\mathbf{s},\, \mathbf{a}) - Q_{\mathbf{w}}(\mathbf{s},\, \mathbf{a}) \right)^2 \right]. \tag{2.23}$$

In this thesis, we use two approaches to solve the above LS problem, the Least-Squares Temporal-Difference (LSTD) approach and the gradient Q-learning approach.

[1] See *Theorem* 3 in [33] and *Assumption* 1 in [34].

LSTD [35] seeks to find the best fitting value function and action-value function using an on-policy manner, and is therefore more stable compared to the gradient Q-learning approach. The LSTD update formulas are shown in below

$$\mathbf{v} = \mathbb{E}_m\Bigg\{\Big[\sum\Big[\mathbf{\Phi}(\mathbf{s})\big(\mathbf{\Phi}(\mathbf{s}) - \gamma\mathbf{\Phi}(\mathbf{s}^+)\big)^\top\Big]\Big]^{-1}\sum\Big[\mathbf{\Phi}(\mathbf{s})L(\mathbf{s},\,\mathbf{a})\Big]\Bigg\}, \tag{2.24a}$$

$$\mathbf{w} = \mathbb{E}_m\Bigg\{\Big[\sum\Big[\mathbf{\Psi}(\mathbf{s},\,\mathbf{a})\mathbf{\Psi}(\mathbf{s},\,\mathbf{a})^\top\Big]\Big]^{-1} \sum\Big[\big(L(\mathbf{s},\,\mathbf{a}) + \gamma V_{\mathbf{v}}\left(\mathbf{s}^+\right) - V_{\mathbf{v}}\left(\mathbf{s}\right)\big)\,\mathbf{\Psi}(\mathbf{s},\,\mathbf{a})\Big]\Bigg\}, \tag{2.24b}$$

where the summation is taken over the whole episode, and the values are then averaged by taking expectation ($\mathbb{E}_m$) over m episodes.

Gradient Q-learning approach [36], on the other hand, uses off-policy Q-learning to update, which significantly increases the data efficiency. Besides, while a potential risk is that off-policy Q-learning may diverge with the linear function approximation, the gradient Q-learning technique would ensure that the parameters are updated towards the true gradient descent and get converged eventually [15]. The update equations are given as follows

$$\mathbf{w} \leftarrow \mathbf{w} + \alpha_w \frac{1}{|B|}\sum_B \left[\delta\mathbf{\Psi}(\mathbf{s},\mathbf{a}) - \gamma\mathbf{\Psi}(\mathbf{s}^+, \boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathbf{s}^+))(\mathbf{\Psi}(\mathbf{s},\mathbf{a})^\top\boldsymbol{\nu})\right], \tag{2.25a}$$

$$\mathbf{v} \leftarrow \mathbf{v} + \alpha_v \frac{1}{|B|}\sum_B \left[\delta\mathbf{\Phi}(\mathbf{s}) - \gamma\mathbf{\Phi}(\mathbf{s}^+)(\mathbf{\Psi}(\mathbf{s},\mathbf{a})^\top\boldsymbol{\nu})\right], \tag{2.25b}$$

$$\boldsymbol{\nu} \leftarrow \boldsymbol{\nu} + \alpha_\nu \frac{1}{|B|}\sum_B \left[(\delta - \mathbf{\Psi}(\mathbf{s},\mathbf{a})^\top\boldsymbol{\nu})\mathbf{\Psi}(\mathbf{s},\mathbf{a})\right], \tag{2.25c}$$

where B is the batch size, δ is the temporal difference, $\boldsymbol{\nu}$ is an extra parameter variable required for the gradient Q-learning approach, and $\alpha_w, \alpha_v, \alpha_\nu$ are the corresponding learning rates for each parameter.

Finally, equation (2.14), for the MPC parameters updating, can be rewritten as

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \boldsymbol{\eta} \odot \mathbb{E}_m \left\{ \sum_{k=1}^{K} \left[\nabla_{\boldsymbol{\theta}} \boldsymbol{\pi}_{\boldsymbol{\theta}} \left(\mathbf{s}_k \right) \nabla_{\boldsymbol{\theta}} \boldsymbol{\pi}_{\boldsymbol{\theta}} (\mathbf{s}_k)^{\top} \mathbf{w} \right] \right\}, \tag{2.26}$$

where the summation is taken over the whole episode (from $k = 1$ to K), and the values are then averaged by taking expectation ($\mathbb{E}_m$) over m episodes.

(3) Advantages of Our MPC-based RL Approach

Our proposed MPC-based RL approach, as applied to ASV navigation, energy management in microgrids, and smart HEMSs, offers a range of advantages that address the limitations of traditional MPC and RL methods as well as the other mentioned MPC-based RL methods. These advantages can be summarized as follows:

- Integration of MPC and RL: The MPC-based RL framework leverages the structured, constraint-aware nature of MPC and the adaptive, performance-oriented capabilities of RL, resulting in a robust and efficient control strategy.
- Enhanced Policy Approximation and System Performance: In this approach, a parameterized MPC scheme is employed as a functional approximation for the optimal policy, superseding the use of DNNs. This allows for the tuning of MPC parameters via RL to enhance long-term performance, addressing the suboptimality often inherent in traditional MPC applications. This dual adjustment of MPC model parameters and the parameters within the MPC cost and constraints enables the derivation of an optimal policy even with an inaccurate model, as shown in theoretical frameworks and practical applications.
- Utilization of Prior Knowledge and Data Efficiency: The MPC-based RL approach can capitalize on existing model knowledge,

starting with a suboptimal but reasonable policy. This use of prior knowledge, particularly in complex systems like smart grids or home energy management, enables more efficient learning from smaller datasets compared to approaches that learn from scratch.

- Explicit Consideration of System Constraints: One of the significant advantages of this approach is its inherent ability to explicitly consider system constraints (e.g., battery capacity, safety requirements in ASV). This feature is critical in real-world applications where adherence to operational and safety constraints is non-negotiable.

- Stability and Feasibility Analysis: The rich theoretical underpinnings of MPC allow for rigorous analysis of system stability and solution feasibility. This aspect is crucial for ensuring the reliability and predictability of the control systems in practical applications.

- Handling System Uncertainties and Nonlinearities: The incorporation of RL enables the approach to effectively handle system uncertainties and nonlinearities, such as those present in home energy systems. This adaptability is essential for managing unpredictable elements like renewable energy sources, varying loads, and weather conditions.

- Performance-Driven Approach: Unlike methods that focus primarily on model fidelity, the MPC-based RL approach is driven by the objective of enhancing closed-loop performance. This focus on performance, such as reducing power costs or improving energy efficiency, aligns well with the practical objectives of the systems it is applied to.

- Interpretability and Adaptability: The parameterized nature of the MPC in this approach offers interpretability, a critical aspect often lacking in DNNs. Additionally, the adaptability of the approach makes it suitable for long-term or even infinite-horizon problems, a key consideration in sustainable system management.

In summary, the MPC-based RL approach presents a novel, robust, and efficient strategy for control system optimization. Its ability to combine the predictive power of MPC with the adaptive learning capabilities of RL, while addressing their respective limitations, makes it a promising solution for a variety of complex and dynamic systems.

References

[1] James Blake Rawlings, David Q Mayne, and Moritz Diehl. *Model predictive control: theory, computation, and design*. Vol. 2. Nob Hill Publishing Madison, WI, 2017.

[2] Eduardo F Camacho, Carlos Bordons, Eduardo F Camacho, and Carlos Bordons. *Model predictive controllers*. Springer, 2007.

[3] S Joe Qin and Thomas A Badgwell. "A survey of industrial model predictive control technology". In: *Control engineering practice* 11.7 (2003), pp. 733–764.

[4] J Rawlings and D Mayne. "Postface to model predictive control: Theory and design". In: *Nob Hill Pub* 5 (2012), pp. 155–158.

[5] David Q Mayne, James B Rawlings, Christopher V Rao, and Pierre OM Scokaert. "Constrained model predictive control: Stability and optimality". In: *Automatica* 36.6 (2000), pp. 789–814.

[6] Ali Mesbah. "Stochastic model predictive control: An overview and perspectives for future research". In: *IEEE Control Systems Magazine* 36.6 (2016), pp. 30–44.

[7] Graham C Goodwin, He Kong, Galina Mirzaeva, and María M Seron. "Robust model predictive control: reflections and opportunities". In: *Journal of Control and Decision* 1.2 (2014), pp. 115–148.

[8] Lukas Hewing, Kim P Wabersich, Marcel Menner, and Melanie N Zeilinger. "Learning-based model predictive control: Toward safe learning in control". In: *Annual Review of Control, Robotics, and Autonomous Systems* 3 (2020), pp. 269–296.

[9] Frank Allgöwer and Alex Zheng. *Nonlinear model predictive control*. Vol. 26. Birkhäuser, 2012.

[10] Felix Berkenkamp, Matteo Turchetta, Angela Schoellig, and Andreas Krause. "Safe model-based reinforcement learning with stability guarantees". In: *Advances in neural information processing systems* 30 (2017).

[11] Mohammed S Alhajeri, Fahim Abdullah, Zhe Wu, and Panagiotis D Christofides. "Physics-informed machine learning modeling for predictive control using noisy data". In: *Chemical Engineering Research and Design* 186 (2022), pp. 34–49.

[12] Cynthia Rudin. "Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead". In: *Nature machine intelligence* 1.5 (2019), pp. 206–215.

[13] Ugo Rosolia, Xiaojing Zhang, and Francesco Borrelli. "Data-driven predictive control for autonomous systems". In: *Annual Review of Control, Robotics, and Autonomous Systems* 1 (2018), pp. 259–286.

[14] Sébastien Gros and Mario Zanon. "Data-driven economic nmpc using reinforcement learning". In: *IEEE Transactions on Automatic Control* 65.2 (2019), pp. 636–648.

[15] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[16] Dimitri Bertsekas. *Dynamic programming and optimal control: Volume I*. Vol. 1. Athena scientific, 2012.

[17] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

[18] Kai Arulkumaran, Marc Peter Deisenroth, Miles Brundage, and Anil Anthony Bharath. "A brief survey of deep reinforcement learning". In: *arXiv preprint arXiv:1708.05866* (2017).

[19] Javier Garcıa and Fernando Fernández. "A comprehensive survey on safe reinforcement learning". In: *Journal of Machine Learning Research* 16.1 (2015), pp. 1437–1480.

[20] Lei Lei et al. “Deep reinforcement learning for autonomous internet of things: Model, applications and challenges”. In: *IEEE Communications Surveys & Tutorials* 22.3 (2020), pp. 1722–1760.

[21] Lucian Buşoniu, Tim de Bruin, Domagoj Tolić, Jens Kober, and Ivana Palunko. “Reinforcement learning for control: Performance, stability, and deep approximators”. In: *Annual Reviews in Control* 46 (2018), pp. 8–28.

[22] Jonas Eschmann. “Reward function design in reinforcement learning”. In: *Reinforcement Learning Algorithms: Analysis and Applications* (2021), pp. 25–33.

[23] Anusha Nagabandi, Gregory Kahn, Ronald S Fearing, and Sergey Levine. “Neural network dynamics for model-based deep reinforcement learning with model-free fine-tuning”. In: *2018 IEEE international conference on robotics and automation (ICRA)*. IEEE. 2018, pp. 7559–7566.

[24] Gregory Kahn, Tianhao Zhang, Sergey Levine, and Pieter Abbeel. “Plato: Policy learning using adaptive trajectory optimization”. In: *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2017, pp. 3342–3349.

[25] Kurtland Chua, Roberto Calandra, Rowan McAllister, and Sergey Levine. “Deep reinforcement learning in a handful of trials using probabilistic dynamics models”. In: *Advances in neural information processing systems* 31 (2018).

[26] Ilgin Dogan, Zuo-Jun Max Shen, and Anil Aswani. “Regret analysis of learning-based mpc with partially-unknown cost function”. In: *arXiv preprint arXiv:2108.02307* (2021).

[27] Baha Zarrouki et al. “Weights-varying mpc for autonomous vehicle guidance: a deep reinforcement learning approach”. In: *2021 European Control Conference (ECC)*. IEEE. 2021, pp. 119–125.

[28] Rohan Chitnis et al. “IQL-TD-MPC: Implicit Q-Learning for Hierarchical Model Predictive Control”. In: *arXiv preprint arXiv:2306.00867* (2023).

[29] Flavia Sofia Acerbo, Jan Swevers, Tinne Tuytelaars, and Tong Duy Son. “Evaluation of MPC-based Imitation Learning for Human-like Autonomous Driving”. In: *arXiv preprint arXiv:2211.12111* (2022).

[30] Arash Bahari Kordabad, Mario Zanon, and Sébastien Gros. “Equivalence of Optimality Criteria for Markov Decision Process and Model Predictive Control”. In: *arXiv preprint arXiv:2210.04302* (2022).

[31] Arash Bahari Kordabad, Hossein Nejatbakhsh Esfahani, Anastasios M. Lekkas, and Sébastien Gros. “Reinforcement Learning based on Scenario-tree MPC for ASVs”. In: *2021 American Control Conference (ACC)*. IEEE. 2021.

[32] Richard S Sutton, David A McAllester, Satinder P Singh, and Yishay Mansour. “Policy gradient methods for reinforcement learning with function approximation”. In: *Advances in neural information processing systems*. 2000, pp. 1057–1063.

[33] David Silver et al. “Deterministic Policy Gradient Algorithms”. In: *Proceedings of the 31st International Conference on International Conference on Machine Learning*. JMLR.org, 2014, I–387–I–395.

[34] Sébastien Gros and Mario Zanon. “Bias Correction in Reinforcement Learning via the Deterministic Policy Gradient Method for MPC-Based Policies”. In: *2021 American Control Conference (ACC)*. IEEE. 2021, pp. 2543–2548.

[35] Michail G Lagoudakis and Ronald Parr. “Least-squares policy iteration”. In: *Journal of machine learning research* 4 (2003), pp. 1107–1149.

[36] Hamid Reza Maei, Csaba Szepesvári, Shalabh Bhatnagar, and Richard S Sutton. “Toward off-policy learning control with function approximation”. In: *ICML*. 2010.

3 | MPC-based RL for Autonomous Surface Vehicles

In this chapter, we propose a Model Predictive Control (MPC)-based Reinforcement Learning (RL) method for Autonomous Surface Vehicles (ASVs). The objective is to find an optimal policy that minimizes the closed-loop performance of a simplified freight mission, including collision-free path tracking, autonomous docking, and a skillful transition between them. We use a parametrized MPC-scheme to approximate the optimal policy, which considers tracking/docking costs and states (position, velocity)/inputs (thruster force, angle) constraints. A Least Squares Temporal Difference (LSTD)-based Deterministic Policy Gradient (DPG) method is applied to update the action-value function parameters and policy parameters. Our simulation results demonstrate that the proposed MPC-LSTD-based DPG method could improve the closed-loop performance during learning for the freight mission problem of ASV.

3.1 Introduction

Autonomous Surface Vehicles (ASVs) are widely applied for many fields, such as freight transportation, oceanographic, military, search and rescue [1–3], and therefore attract broad attention for scientific and industrial researches. For example, the project, Yara Birkeland [4], devoted to realizing fully autonomous container vessels, exhibits many positive

sides for bringing autonomy into the maritime sector.

3.1.1 Literature Review

Various methods have been proposed to solve the problem of operating and automating the ASV, including path tracking (path following) [5, 6], collision avoidance [7, 8], and autonomous docking [9, 10]. However, designing a control strategy that could realize both collision-free path tracking and docking in a freight mission with time-varying disturbances is still an intractable topic. With the development of Machine Learning (ML), Reinforcement Learning (RL)-based control strategies are getting noticed by people, as they can make good use of real data to reduce the impact of disturbances [11, 12].

RL is a technique for solving problems involving Markov Decision Processes (MDP) without prior knowledge about the model dynamics (state transition probabilities). RL exploits samples (state-action pairs) and rewards to seek an optimal feedback policy that renders the best closed-loop performance [13]. Deterministic Policy Gradient (DPG), as the direct RL method, estimates the optimal policy by a parameterized function approximator, and optimizes the policy parameters directly via gradient descent steps of the performance [14, 15]. Deep Neural Networks (DNNs) are very commonly used function approximators in RL [16]. However, DNN-based RL lacks the abilities concerning the closed-loop stability analysis, state/input constraints satisfaction, and meaningful weights initialization [17].

3.1.2 MPC-based RL Approach

To address these problems, the perspective of using Model Predictive Control (MPC)-based RL has been proposed and justified in [18], i.e. it suggests using MPC as the function approximation for the optimal policy in RL (Note that MPC can also be used as the function approximators for the value function and action-value function). MPC is a well-known

model-based control strategy that produces an input sequence over a finite receding prediction horizon such that the resulting predicted state trajectory minimizes a given cost function while respecting the constraints imposed on the system [19]. Unlike DNNs, MPC-based policies satisfy the state/input constraints and safety requirements by construction, and its well-structured property enables the stability analysis of the system.

However, for computational reasons, simple models are usually preferred in the MPC-scheme. Hence, the MPC model often does not have the required structure to correctly capture the real system dynamics and stochasticity. As a result, MPC can deliver a reasonable approximation of the optimal policy, but it is usually suboptimal [19]. Besides, choosing the model parameters that best fit the MPC model to the real system does not necessarily yield the MPC policy that achieves the best closed-loop performance [18]. Therefore, choosing appropriate MPC parameters to achieve the best closed-loop performance is extremely challenging. Nevertheless, it is shown in [17, 18] that in principle, by adjusting not only the MPC model parameters but also the parameters in the MPC cost and constraints, the MPC scheme can generate the optimal closed-loop policy, even if the MPC model is inaccurate. It is also shown that RL is a suitable candidate to perform that adjustment in practice. Recent researches focused on the MPC-based RL have further developed this approach [20–23].

3.1.3 Contributions

In this chapter, we use the above-mentioned MPC-based RL method for ASV to solve a freight mission problem with external disturbances. A parametrized MPC-scheme is used to approximate the optimal policy, which considers tracking/docking costs and states (position, velocity)/inputs (thruster force, angle) constraints. Then the Least Squares Temporal Difference (LSTD)-based DPG would tune the parameters inside the MPC model, cost, and constraints, such that the MPC scheme controlling the ship achieves a close-to-optimal policy to accomplish

collision-free path tracking, docking, and the transition between them.

- From the theoretical point of view: based on the section IV-D of reference [24], we elaborate the proposed MPC-based RL method in the ASV problem framework, as well as formulate an algorithm for the MPC-LSTD-based DPG method.

- From the application point of view: ASVs exhibit many positive sides for bringing autonomy into the maritime sector. For example, the project, Yara Birkeland [25], is devoted to realizing fully autonomous container vessels. Our work, in this context, provides a promising approach for a complicated ASV freight mission problem. We propose a strategy that solves obstacle avoidance, path following, and autonomous docking simultaneously, in a stochastic environment.

The rest of the chapter is structured as follows. Section 3.2 provides the ASV model that consists of vessel dynamics and the thruster allocation. Section 3.3 details the simplified freight mission problem: collision-free path tracking, docking, and the objective function of the problem. The proposed MPC-based RL method is elaborated in Section 3.4, which first formulates the parametrized MPC-based policy approximation and then explains the LSTD-based DPG method. Section 3.5 presents the simulations and Section 3.6 delivers the conclusion.

3.2 ASV Model

The 3-Degree of Freedom (3-DOF) position of the vessel can be represented by a pose vector $\boldsymbol{\eta} = [x, y, \psi]^T \in \mathbb{R}^3$ in the North-East-Down (NED) frame, where x is the North position, y is the East position, and ψ is the heading angle (see Fig. 3.1). The velocity vector $\boldsymbol{\nu} = [u, v, r]^T \in \mathbb{R}^3$, including the surge velocity u, sway velocity v, and yaw rate r, is decomposed in the body-fixed frame.

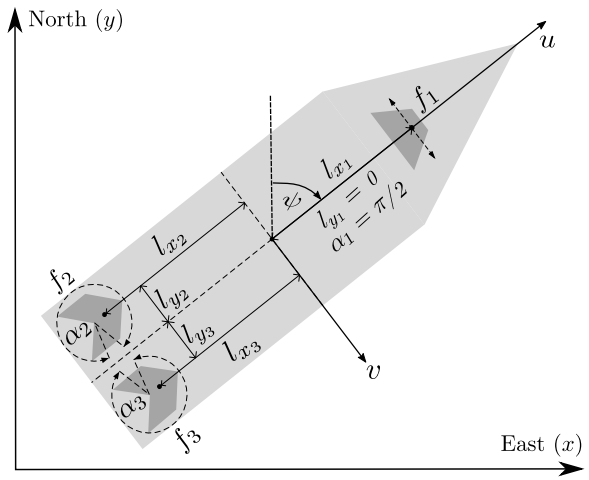

Figure 3.1: The 3-DOF ASV model in the NED frame.

3.2.1 3-DOF Model

The nonlinear dynamics can be written as follows [26]

$$\dot{\boldsymbol{\eta}} = \mathbf{J}(\psi)\boldsymbol{\nu} \tag{3.1a}$$

$$\mathbf{M}\dot{\boldsymbol{\nu}} + \mathbf{D}\boldsymbol{\nu} = \boldsymbol{\tau} + \boldsymbol{\tau}_a, \tag{3.1b}$$

where $\mathbf{J}(\psi) \in \mathbb{R}^{3\times3}$ is the rotation matrix, given by

$$\mathbf{J}(\psi) = \begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{3.2}$$

$\mathbf{M} \in \mathbb{R}^{3\times3}$ is the mass matrix and $\mathbf{D} \in \mathbb{R}^{3\times3}$ is the damping matrix (see [10] for their specific physical meanings and values). $\boldsymbol{\tau} = [X, Y, N]^\top \in \mathbb{R}^3$ is the external control forces (X, Y) and moment (N) vector empowered by the thrusters. Vector $\boldsymbol{\tau}_a \in \mathbb{R}^3$ is the additional forces rendered from disturbances, e.g., wind, ocean waves and etc.

3.2.2 Thruster Allocation

The thrust configuration is illustrated in Fig. 3.1. The vector $\boldsymbol{\tau}$ could be specifically written as $\boldsymbol{\tau} = \mathbf{T}(\boldsymbol{\alpha})\,\mathbf{f}$, where $\mathbf{f} = [f_1, f_2, f_3]^\top \in \mathbb{R}^3$ is the thruster forces vector as we consider one tunnel thruster f_1 and two azimuth thrusters f_2, f_3. They are subjected to the bounds

$$f_{p_{\min}} \leq f_p \leq f_{p_{\max}}, \quad p = 1, 2, 3. \tag{3.3}$$

Matrix $\mathbf{T}(\boldsymbol{\alpha}) \in \mathbb{R}^{3\times 3}$ presents the thruster configuration, written as

$$\mathbf{T}(\boldsymbol{\alpha}) = \begin{bmatrix} 0 & \cos(\alpha_2) & \cos(\alpha_3) \\ 1 & \sin(\alpha_2) & \sin(\alpha_3) \\ l_{x1} & T_{32} & T_{33} \end{bmatrix}, \tag{3.4}$$

where elements $T_{32} = l_{x2}\sin(\alpha_2) - l_{y_2}\cos(\alpha_2)$, and $T_{33} = l_{x3}\sin(\alpha_3) - l_{y_3}\cos(\alpha_3)$. Constants l_{xi} and l_{y_i} with $i = 1, 2, 3$ are the distances between each thruster and the cross line of the ship's center. Term $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \alpha_3]^\top \in \mathbb{R}^3$ is the corresponding orientation vector. The angle α_1 is fixed $(\pi/2)$, while α_2 and α_3, associated to the two azimuth thrusters, are restricted in the range

$$|\alpha_2 + \pi/2| \leq \alpha_{\max}, \quad |\alpha_3 - \pi/2| \leq \alpha_{\max}. \tag{3.5}$$

A maximum angle of $\alpha_{\max}$ with a forbidden sector is considered in this work to avoid thrusters 2 and 3 directly work against each other, as shown in Fig. 3.1. With a sampling time of dt, we discretize the ship system (3.1) as

$$\mathbf{s}_{k+1} = F(\mathbf{s}_k, \mathbf{a}_k, \boldsymbol{\tau}_a), \tag{3.6}$$

where $\mathbf{s}_k = \left[\boldsymbol{\eta}_k^\top, \boldsymbol{\nu}_k^\top\right]^\top$ and $\mathbf{a}_k = \left[\mathbf{f}_k^\top, \boldsymbol{\alpha}_k^\top\right]^\top$ are system state and input vectors, respectively. Subscript k denotes the physical time and $F(\cdot)$ is the discretized real system.

3.3 Problem Formulation – Simplified Freight Mission

In this work, we consider a simplified freight mission problem: the ASV starts from an origin $\mathbf{A}$ to the end $\mathbf{B}$, which is supposed to follow a designed collision-free course and finally dock at the wharf autonomously. Note that the transition from path following to docking is a notable point of this problem.

3.3.1 Collision-Free Path Following

Given a reference path P_{ref}. At time instance k, $\mathbf{P}_k^{\text{ref}} = [x_k^{\text{ref}}, y_k^{\text{ref}}]^\top$. Then path following could be thought as minimizing the error $l(\boldsymbol{\eta}_k)$

$$l(\boldsymbol{\eta}_k) = \left\|\boldsymbol{\eta}_k^p - \mathbf{P}_k^{\text{ref}}\right\|_2^2 = (x_k - x_k^{\text{ref}})^2 + (y_k - y_k^{\text{ref}})^2, \tag{3.7}$$

where $\boldsymbol{\eta}_k^p = [x_k, y_k]^\top$ contains the first two elements of $\boldsymbol{\eta}_k$. Besides, we assume obstacles of round shape. To avoid these obstacles, the following term $g_n(\boldsymbol{\eta}_k)$, representing the position of the ship relative to the n^{th} obstacle, should satisfy

$$(x_k - o_{x,n})^2 + (y_k - o_{y,n})^2 \geq (r_n + r_o)^2, \tag{3.8}$$

i.e.,

$$\underbrace{1 - \left((x_k - o_{x,n})^2 + (y_k - o_{y,n})^2\right) \Big/ (r_n + r_o)^2}_{g_n(\boldsymbol{\eta}_k)} \leq 0, \tag{3.9}$$

where $(o_{x,n}, o_{y,n})$ and r_n are the center and radius of the n^{th} circular obstacle ($n = 1, \ldots, N_o$), respectively. Constant r_o is the radius of the vessel and N_o is the number of obstacles.

3.3.2 Autonomous Docking

Docking refers to stopping the vessel exactly at the endpoint $\mathbf{B}$ as well as avoiding collisions between any part of the vessel and the quay [9]. The "accurate stop" requires not only an accurate docking position but also zero-valued velocities and thruster forces at the final time, i.e., we ought to minimize

$$h\left(\boldsymbol{\eta}_k, \boldsymbol{\nu}_k, \mathbf{f}_k\right) = \|\boldsymbol{\eta}_k - \boldsymbol{\eta}_d\|_2^2 + \|\boldsymbol{\nu}_k\|_2^2 + \|\mathbf{f}_k\|_2^2, \tag{3.10}$$

where $\boldsymbol{\eta}_d = (x_d, y_d, \psi_d)$ is the desired docking position. Successfully docking requires $h\left(\boldsymbol{\eta}_K, \boldsymbol{\nu}_K, \mathbf{f}_K\right) \approx 0$, where subscript K denotes the terminal time step of the freight mission. As for "collision avoidance", we define a safety operation region $\mathbb{S}$ as the spatial constraints for the vessel. The operation region is chosen as the largest convex region that encompasses the docking point but not intersecting with the land. Thus, as long as the vessel is within the region $\mathbb{S}$, no collision will occur during docking, i.e. the following condition should hold

$$\boldsymbol{\eta}_k^p \in \mathbb{S}, \quad \mathbb{S} = \{\mathbf{x} | \mathbf{A}\mathbf{x} < \mathbf{b}\}, \tag{3.11}$$

where $\boldsymbol{\eta}_k^p = [x_k, y_k]^\top$ describes the position of the vessel. The matrix $\mathbf{A}$ and the vector $\mathbf{b}$ are determined by the shape of the quay and together define the convex region $\mathbb{S}$.

3.3.3 Objective Function

In the context of RL, we seek a control policy $\boldsymbol{\pi}$ that minimizes the following closed-loop performance J

$$J(\boldsymbol{\pi}) = \mathbb{E}_{\boldsymbol{\pi}}\left[\sum_{k=0}^{K} \gamma^k L\left(\mathbf{s}_k, \mathbf{a}_k\right) \middle| \mathbf{a}_k = \boldsymbol{\pi}(\mathbf{s}_k)\right], \tag{3.12}$$

where $\gamma \in (0, 1]$ is the discount factor. Expectation $\mathbb{E}_{\boldsymbol{\pi}}$ is taken over the distribution of the Markov chain in the closed-loop under policy $\boldsymbol{\pi}$.

The RL-stage cost $L(\mathbf{s}_k, \mathbf{a}_k)$, in this problem, is defined as a piecewise function

$$L = \begin{cases} l(\boldsymbol{\eta}_k) + O(\boldsymbol{\eta}_k) + \xi(\boldsymbol{\alpha}_k) & \|\boldsymbol{\eta}_k - \boldsymbol{\eta}_d\|_2^2 > d \\ h(\boldsymbol{\eta}_k, \boldsymbol{\nu}_k, \mathbf{f}_k) + \Gamma(\boldsymbol{\eta}_k) + \xi(\boldsymbol{\alpha}_k) & \|\boldsymbol{\eta}_k - \boldsymbol{\eta}_d\|_2^2 \leq d, \end{cases} \tag{3.13}$$

where $O(\boldsymbol{\eta}_k)$ is the obstacle penalty for path following

$$O(\boldsymbol{\eta}_k) = \sum_{n=1}^{N_o} c_n \cdot \max(0, g_n(\boldsymbol{\eta}_k) + d_s), \tag{3.14}$$

where $c_n > 0$ is the penalty weight, constant $d_s > 0$ is the desired safe distance between vessel and obstacles. Therefore, once the ship breaks the safe distance, i.e. $g_n(\boldsymbol{\eta}_k) + d_s > 0$, a positive penalty will be introduced to the objective function. Function $\Gamma(\boldsymbol{\eta}_k)$ is the collision penalty for docking

$$\Gamma(\boldsymbol{\eta}_k) = \kappa \cdot (1 - \mathbf{1}_{\mathbb{S}}(\boldsymbol{\eta}_k^p)), \tag{3.15}$$

where $\kappa > 0$ is the penalty weight and $\mathbf{1}_{\mathbb{S}}(\cdot)$ is the indicator function. When the ship is out of the safe region, i.e. $\boldsymbol{\eta}_k^p \notin \mathbb{S}$, a positive penalty will be imposed in the objective function. Function $\xi(\boldsymbol{\alpha}_k)$ is the singular configuration penalty, aiming to avoid the thruster configuration matrix $\mathbf{T}(\boldsymbol{\alpha}_k)$ in (3.4) being singular [27]

$$\xi(\boldsymbol{\alpha}_k) = \frac{\rho}{\varepsilon + \det\left(\mathbf{T}(\boldsymbol{\alpha}_k)\mathbf{W}^{-1}\mathbf{T}^\top(\boldsymbol{\alpha}_k)\right)}, \tag{3.16}$$

where "$\det$" stands for the determinant of the matrix. Constant $\varepsilon > 0$ is a small number to avoid division by zero, $\rho > 0$ is the weighting of maneuverability, and $\mathbf{W}$ is a diagonal weighting matrix. Constant $d > 0$ is designed to substitute the stage cost from path following to docking at $\|\boldsymbol{\eta}_k - \boldsymbol{\eta}_d\|_2^2 = d$, which means that our target transits from path-following to docking when the ship approaches the destination.

3.4 MPC-based RL

The core idea of our proposed approach is to use a parameterized MPC-scheme as the policy approximation function, and apply the LSTD-based

DPG method to update the parameters so as to improve the closed-loop performance.

3.4.1 MPC-based Policy Approximation

Consider the following MPC-scheme parameterized by $\boldsymbol{\theta}$

$$\min_{\hat{\boldsymbol{\eta}},\hat{\boldsymbol{\nu}},\hat{\mathbf{f}},\hat{\boldsymbol{\alpha}},\boldsymbol{\sigma}} \frac{\theta_d}{\|\hat{\boldsymbol{\eta}}_N - \boldsymbol{\eta}_d\|_2^2 + \delta} \cdot \Big(h_{\boldsymbol{\theta}}\left(\hat{\boldsymbol{\eta}}_N, \hat{\boldsymbol{\nu}}_N\right) + \Gamma_\theta\left(\hat{\boldsymbol{\eta}}_N\right) \Big) + \boldsymbol{\omega}_f^\top \boldsymbol{\sigma}_N + \sum_{i=0}^{N-1} \gamma^i \Big(l_{\boldsymbol{\theta}}\left(\hat{\boldsymbol{\eta}}_i\right) + \xi\left(\hat{\boldsymbol{\alpha}}_i\right) + \boldsymbol{\omega}^\top \boldsymbol{\sigma}_i \Big) \quad (3.17a)$$

$$\text{s.t.} \quad \forall i = 0, \ldots, N-1,\ n = 1, \ldots, N_o$$

$$\left[\hat{\boldsymbol{\eta}}^\top{}_{i+1}, \hat{\boldsymbol{\nu}}^\top{}_{i+1}\right]^\top = F_{\boldsymbol{\theta}}(\hat{\boldsymbol{\eta}}_i, \hat{\boldsymbol{\nu}}_i, \hat{\mathbf{f}}_i, \hat{\boldsymbol{\alpha}}_i, \boldsymbol{\theta}_a) \quad (3.17b)$$

$$f_{p_{\min}} \le \hat{f}_{p,i} \le f_{p_{\max}}, \quad p = 1, 2, 3 \quad (3.17c)$$

$$|\hat{\alpha}_{2,i} + \pi/2| \le \alpha_{\max},\ \ |\hat{\alpha}_{3,i} - \pi/2| \le \alpha_{\max}, \quad (3.17d)$$

$$g_n\left(\hat{\boldsymbol{\eta}}_i\right) + \theta_g \le \sigma_{n,i},\ \ g_n\left(\hat{\boldsymbol{\eta}}_N\right) + \theta_g \le \sigma_{n,N}, \quad (3.17e)$$

$$\boldsymbol{\sigma}_i \ge 0,\ \ \boldsymbol{\sigma}_N \ge 0, \quad (3.17f)$$

$$\hat{\boldsymbol{\eta}}_0 = \boldsymbol{\eta}_k,\ \ \hat{\boldsymbol{\nu}}_0 = \boldsymbol{\nu}_k, \quad (3.17g)$$

where N is the prediction horizon. Arguments $\hat{\boldsymbol{\eta}} = \{\hat{\boldsymbol{\eta}}_0, \ldots, \hat{\boldsymbol{\eta}}_N\}$, $\hat{\boldsymbol{\nu}} = \{\hat{\boldsymbol{\nu}}_0, \ldots, \hat{\boldsymbol{\nu}}_N\}$, $\hat{\mathbf{f}} = \{\hat{\mathbf{f}}_0, \ldots, \hat{\mathbf{f}}_{N-1}\}$, $\hat{\boldsymbol{\alpha}} = \{\hat{\boldsymbol{\alpha}}_0, \ldots, \hat{\boldsymbol{\alpha}}_{N-1}\}$, and $\boldsymbol{\sigma} = \{\boldsymbol{\sigma}_0, \ldots, \boldsymbol{\sigma}_N\}$ are the primal decision variables. The term $\frac{\theta_d}{\|\hat{\boldsymbol{\eta}}_N - \boldsymbol{\eta}_d\|_2^2 + \delta} \cdot \left(h_{\boldsymbol{\theta}}\left(\cdot\right) + \Gamma_{\boldsymbol{\theta}}\left(\cdot\right)\right)$ introduces a gradually increasing terminal cost as the ship approaches the endpoint, where $\delta > 0$ is a small constant to avoid division by zero. The weighting parameter θ_d, designed to balance the priority of path following and docking, is tuned by RL. Note that θ_d is chosen to minimize the closed-loop performance that considering both path following and docking, although it may be suboptimal for either single problem. Parameter θ_g is the tightening variable used to adjust the strength of the collision avoidance constraints. If the value of θ_g (positive) is larger, it means that the constraints are tighter and the ship is supposed to be farther away from the obstacles. It is important to use RL to pick an appropriate θ_g, since when θ_g is too large, although

we ensure that the ship safely avoids obstacles, the path following error is increased. Conversely, a smaller θ_g reduces the following error, but we may gain more penalty when the vessel breaks the safe distance, as described in (3.14). Note that the obstacle penalties are considered directly as constraints (3.17e) in the MPC rather than as penalties in the MPC cost, because (3.9) is a conservative model of the obstacle penalty (3.14). Variables $\boldsymbol{\sigma}_i$ $\left(\boldsymbol{\sigma}_i = \{\sigma_{1,i}, \ldots, \sigma_{N_0,i}\}\right)$ and $\boldsymbol{\sigma}_N$ $\left(\boldsymbol{\sigma}_N = \{\sigma_{1,N}, \ldots, \sigma_{N_0,N}\}\right)$ are slacks for the relaxation of the state constraints, weighted by the positive vectors $\boldsymbol{\omega}$ and $\boldsymbol{\omega}_f$. The relaxation prevents the infeasibility of the MPC in the presence of some hard constraints.

The parameterized stage cost $l_{\boldsymbol{\theta}}(\cdot)$, terminal cost $h_{\boldsymbol{\theta}}(\cdot)$, and docking collision penalty $\Gamma_\theta(\cdot)$ in the MPC cost (3.17a) are designed as follows

$$l_{\boldsymbol{\theta}} = \left\|\hat{\boldsymbol{\eta}}_i^p - \mathbf{P}_i^{\text{ref}}\right\|_{\boldsymbol{\Theta}_l}^2 \tag{3.18a}$$

$$h_{\boldsymbol{\theta}} = \|\hat{\boldsymbol{\eta}}_N - \boldsymbol{\eta}_d\|_{\boldsymbol{\Theta}_\eta}^2 + \|\hat{\boldsymbol{\nu}}_N\|_{\boldsymbol{\Theta}_\nu}^2 \tag{3.18b}$$

$$\Gamma_\theta = \theta_\kappa \cdot (1 - \mathbf{1}_{\mathbb{S}}(\hat{\boldsymbol{\eta}}_N^p)), \tag{3.18c}$$

where $\boldsymbol{\Theta}_l, \boldsymbol{\Theta}_\eta, \boldsymbol{\Theta}_\nu \in \mathbb{R}^{3\times 3}$ are the weighing matrices that are symmetric semi-positive definite. They are expressed as $\boldsymbol{\Theta}_l = (\text{diag}(\boldsymbol{\theta}_l))^2$, $\boldsymbol{\Theta}_\eta = (\text{diag}(\boldsymbol{\theta}_\eta))^2$, $\boldsymbol{\Theta}_\nu = (\text{diag}(\boldsymbol{\theta}_\nu))^2$. Operator "diag" assigns the vector elements onto the diagonal elements of a square matrix. Parameter θ_κ is treated as a degree of freedom for the docking collision penalty. The real model is (3.6) and we assume the disturbance $\boldsymbol{\tau}_a$ follows a Gaussian distribution. To address the disturbance without using a complex stochastic model in the MPC scheme, one measure is to use a parameter vector $\boldsymbol{\theta}_a \in \mathbb{R}^3$ to parameterize the model as $F_{\boldsymbol{\theta}}(\hat{\mathbf{s}}_i, \hat{\mathbf{a}}_i, \boldsymbol{\theta}_a)$. As detailed in [18], the full adaptation of the parametrized MPC scheme (model, costs, constraints) can compensate for that unmodelled disturbance. Overall, the adjustable parameters vector $\boldsymbol{\theta}$ consists of

$$\boldsymbol{\theta} = \{\boldsymbol{\theta}_l, \boldsymbol{\theta}_\eta, \boldsymbol{\theta}_\nu, \boldsymbol{\theta}_a, \theta_\kappa, \theta_d, \theta_g\}. \tag{3.19}$$

And $\boldsymbol{\theta}$ will be adjusted by RL according to the principle of "improving the closed-loop performance". Note that: 1. the span of the RL ($K \approx 550$) is much longer than the horizon of the MPC ($N = 60$); 2. the RL cost (3.13) is a "switching" function, while the MPC cost (3.17a)

contains simultaneously the path following and docking cost to avoid the mixed-integer treatment of the problem; 3. the MPC model does not perfectly match the real system. For the above reasons, having different cost functions in the MPC scheme and RL is rational [18]. Therefore, in order to improve the closed-loop performance of the MPC scheme as assessed by the RL cost, it can be beneficial to parameterize the MPC cost functions, model, and constraints. RL then adjusts these parameters according to the principle of "improving the closed-loop performance". From *Theorem 1* and *Corollary 2* in [18], we know that, theoretically, under some assumptions, if the parametrization is rich enough, the MPC scheme is capable of capturing the optimal policy $\pi^\star$ in presence of model uncertainties and disturbances.

Importantly, the deterministic policy $\boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathbf{s})$ can be obtained as

$$\boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathbf{s}) = \mathbf{u}_0^\star(\mathbf{s}, \boldsymbol{\theta}), \tag{3.20}$$

where $\mathbf{u}_0^\star(\mathbf{s}, \boldsymbol{\theta})$ is the first element of $\mathbf{u}^\star$, which is the input solution of the MPC scheme (3.17).

3.4.2 LSTD-based Deterministic Policy Gradient

For this problem, we use the LSTD-based DPG method elaborated in Section 2.4.2: *"Core Formulas: DPG for MPC-based RL"* to update the MPC parameters. The general update rule is

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \alpha \nabla_{\boldsymbol{\theta}} J(\boldsymbol{\pi}_{\boldsymbol{\theta}}), \tag{3.21}$$

where $\alpha > 0$ is the step size and the gradient $\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\pi}_{\boldsymbol{\theta}})$ is computed using the same formulas as in Section 2.4.2, except that the variables are replaced with those defined in this ASV problem. Specifically, $\boldsymbol{\zeta}$ in (2.17) has the form $\boldsymbol{\zeta} = \{\hat{\boldsymbol{\eta}}, \hat{\boldsymbol{\nu}}, \hat{\mathbf{f}}, \hat{\boldsymbol{\alpha}}, \boldsymbol{\sigma}\}$, which is the primal decision variable of the MPC (3.17). And in (2.18), $\Omega_{\boldsymbol{\theta}}$ now represents the MPC cost (3.17a), $\mathbf{G}_{\boldsymbol{\theta}}$ gathers the equality constraints and $\mathbf{H}_{\boldsymbol{\theta}}$ collects the inequality constraints of the MPC (3.17). In addition, the first element of the input $\mathbf{u}_0$ in (2.19) is represented in this case as

$$\mathbf{u}_0 = \left[\hat{\mathbf{f}}_0^\top, \hat{\boldsymbol{\alpha}}_0^\top\right]^\top. \tag{3.22}$$

The state feature vector $\mathbf{\Phi}(\mathbf{s})$ in (2.21) is designed to constitute all monomials of the state with degrees less than or equal to 2.

Finally, equation (3.21) can be rewritten as

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \alpha \mathbb{E}_m \left\{ \sum_{k=1}^{K} \left[\nabla_{\boldsymbol{\theta}} \boldsymbol{\pi}_{\boldsymbol{\theta}} \left(\mathbf{s}_k \right) \nabla_{\boldsymbol{\theta}} \boldsymbol{\pi}_{\boldsymbol{\theta}} (\mathbf{s}_k)^{\top} \mathbf{w} \right] \right\}, \tag{3.23}$$

where the summation is taken over the whole episode, which terminates at K when the ship reaches the destination (i.e. $\|\boldsymbol{\eta}_K - \boldsymbol{\eta}_d\|_2^2 \leq d_{\text{error}}$) and the values will be then averaged by taking expectation ($\mathbb{E}_m$) over m episodes. The proposed MPC-LSTD-based DPG method is summarized in Algorithm 1.

Algorithm 1: MPC-LSTD-based DPG method

Input: vessel model, objective function, initial parameters $\boldsymbol{\theta}_0$
Output: locally optimal policy $\boldsymbol{\pi}_{\boldsymbol{\theta}^\star}$

```
repeat
    for each episode in m episodes do
        initialize η_0, ν_0;
        while ||η_k − η_d||_2^2 ≤ d_error do
            solve the MPC (3.17) and get y*;
            calculate and record the RL stage cost L(s_k, a_k)
              according to (3.13) and the sensitivity ∇_θ π_θ(s_k)
              according to (2.19);
        end
    end
    calculate v according to (2.24a);
    calculate w according to (2.24b);
    update θ according to (3.23);
until convergence;
```

3.5 Simulation and Discussion

In the simulation, we choose the initial parameters vector as $\boldsymbol{\theta}_0 = \{\mathbf{0.55}, \mathbf{3}, \mathbf{3}, \mathbf{1e{-}7}, 60, 35, 0.5\}$, where the bold numbers represent constant vectors with suitable dimension. Other parameter values used in the simulation are given in Table 3.1.

Table 3.1: Parameters values.

Symbol	Value	Symbol	Value
γ, N, dt	$1, 60, 0.5$	$\tau_a, \alpha_{\max}$	$\mathcal{N}(0, 1\text{e}{-3}), \frac{17\pi}{18}$
$f_{1_{\min,\max}}$	$-100, 100$	$f_{2,3_{\min,\max}}$	$0, 200$
$\rho, \varepsilon, \delta$	$1, 0.001, 0.001$	$\mathbf{W}$	$\text{diag}([1, 1, 1])$
$\boldsymbol{\omega}, \boldsymbol{\omega}_f$	$[1, 5, 5]^\top$	$c_{1,2,3}$	$5, 8, 8$
d, d_s, d_{error}	$42.5, 1, 0.5$	N_o, m	$3, 10$
r_0, r_1, r_2, r_3	$1, 1.4, 1.7, 1.9$	$\boldsymbol{\eta}_d$	$[21.3, 23.3, 8.4]^\top$
$\boldsymbol{\eta}_0$	$[0, 0, \frac{\pi}{4}]^\top$	$\boldsymbol{\nu}_0$	$[0.4, 0, 0]^\top$

Figure 3.2 shows the prescribed reference path and the thirteen shipping paths updated after each learning step. The last path P_{13} is obtained under the final learned policy $\boldsymbol{\pi}_{\boldsymbol{\theta}^\star}$ with an episode length of $K = 550$. It is worth noting that, although we say that if the parametrization is rich enough, the MPC scheme can generate the optimal policy, this is a theoretical result. In practice, the assumption of a “rich enough" parametrization is typically not satisfied. Other practical issues can come in the way of optimality such as, e.g., the local convergence of the RL algorithm and of the solver treating the MPC scheme. Addressing these potential issues typically requires good initial guesses. Although these are often available in the MPC context, we can only claim that the final learned policy $\boldsymbol{\pi}_{\boldsymbol{\theta}^\star}$ obtained from the converged parameters $\boldsymbol{\theta}^\star$ is locally optimal. This observation applies to most RL techniques. Following the reference path P_{ref} defined from the origin A to the point q, the vessel departs from A and passes through three obstacles to reach q. At the

point q, where $\|\boldsymbol{\eta}_k - \boldsymbol{\eta}_d\|_2^2 = d$, the vessel transits from path following to docking. The vessel eventually stops at the end B with zero velocities and thruster forces, and has no collision with the quay (within the safety operation region $\mathbb{S}$) during the docking process. It can be seen that in the first few paths (P_1-P_4), the ship does not follow P_{ref} precisely, and is relatively far away from the three obstacles when it bypasses them. After learning, such as in the P_{13}, the ship closely follows the reference route, and the distance when avoiding obstacles is also reduced.

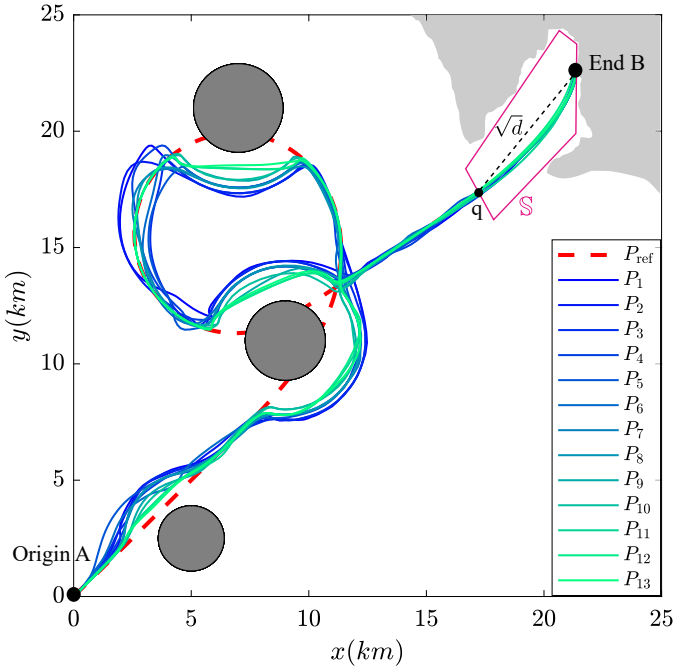

Figure 3.2: Freight shipping paths from A to B. P_{ref}: the reference path. P_1-P_{13}: the renewed path after each learning step.

Figure 3.3 shows the convergences of the MPC parameters $\boldsymbol{\theta}$ over learning steps ($\boldsymbol{\theta}^\star$ represents the converged parameters). Note that θ_l^1 is the first element of $\boldsymbol{\theta}_l$, and the same fashion for others. It can be seen that the initial value of $\boldsymbol{\theta}_l$ is relatively small, and the initial values of $\boldsymbol{\theta}_\eta, \boldsymbol{\theta}_\nu, \theta_\kappa, \theta_d$ are relatively large. Therefore, in the MPC cost (3.17a), the terminal cost weighs more than the stage cost, i.e., docking is regarded

as more important than path following. Consequently, the path following performance is relatively poor in the initial episodes, and then gets improved as $\boldsymbol{\theta}_l$ increases and $\boldsymbol{\theta}_\eta, \boldsymbol{\theta}_\nu, \theta_\kappa, \theta_d$ decrease. In addition, the initial value of θ_g is large, which means that the ship must be very far away from the obstacles. However, this is unnecessary under the premise of ensuring the safe distance d_s. To reduce the cost, RL gradually reduces θ_g, and therefore results in what we have in Fig. 3.2: the distance for avoiding obstacles tends to decrease over learning.

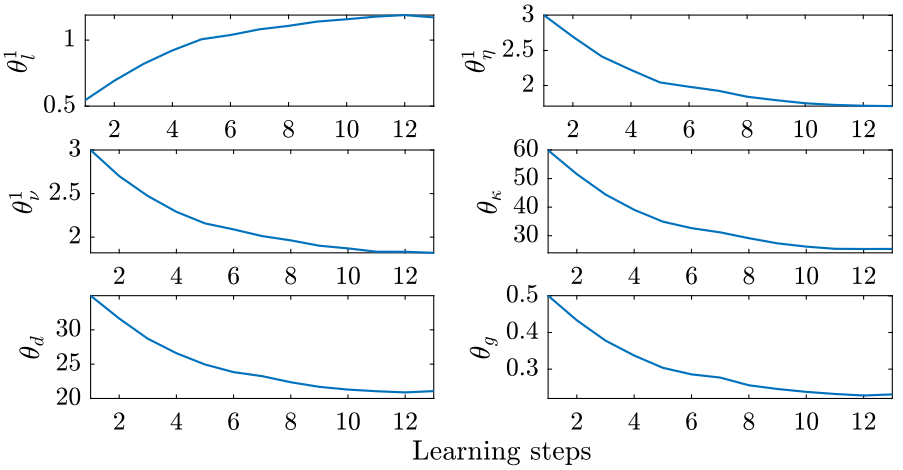

Figure 3.3: Variations of some selected MPC parameters $\{\theta_l^1, \theta_\eta^1, \theta_\nu^1, \theta_\kappa, \theta_d, \theta_g\}$ over learning steps.

The Variations of the normed policy gradient $\|\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\pi}_{\boldsymbol{\theta}})\|_2$ and the closed-loop performance $J(\boldsymbol{\pi}_{\boldsymbol{\theta}})$ are displayed in Fig. 3.4. As can be seen, the policy gradient converges to near zero with learning as the parameters approach to their optimal points, and the performance is improved significantly over the learning. Besides, since the value of policy gradient $\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\pi}_{\boldsymbol{\theta}})$ is relatively large within the first 5 steps, the performance J drops faster in this range.

Figure 3.5 illustrates the variations of error between the ship pose state $\boldsymbol{\eta}$ and the desired docking state $\boldsymbol{\eta}_d$ under the learned optimal policy $\boldsymbol{\pi}(\boldsymbol{\theta}^\star)$. Fig. 3.6 presents the variations of the vessel velocity $\boldsymbol{\nu}$ with time under

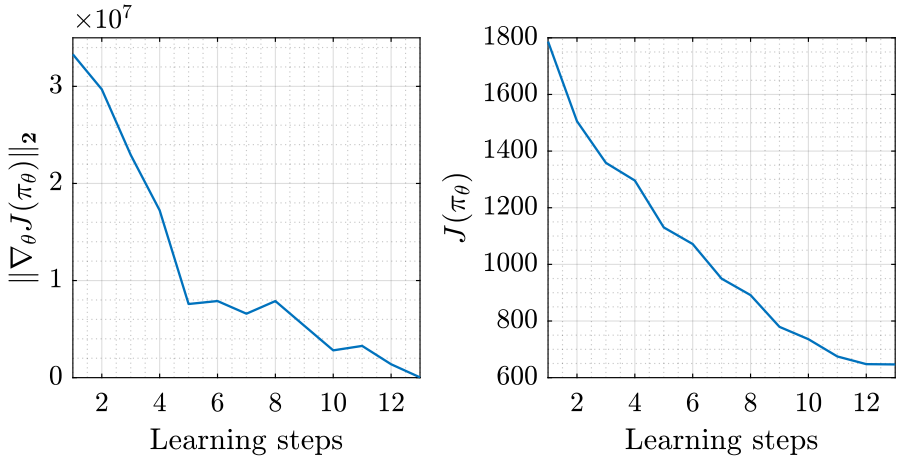

Figure 3.4: Variations of the normed policy gradient $\|\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\pi_\theta})\|_2$ and the closed-loop performance $J(\boldsymbol{\pi_\theta})$ over learning steps.

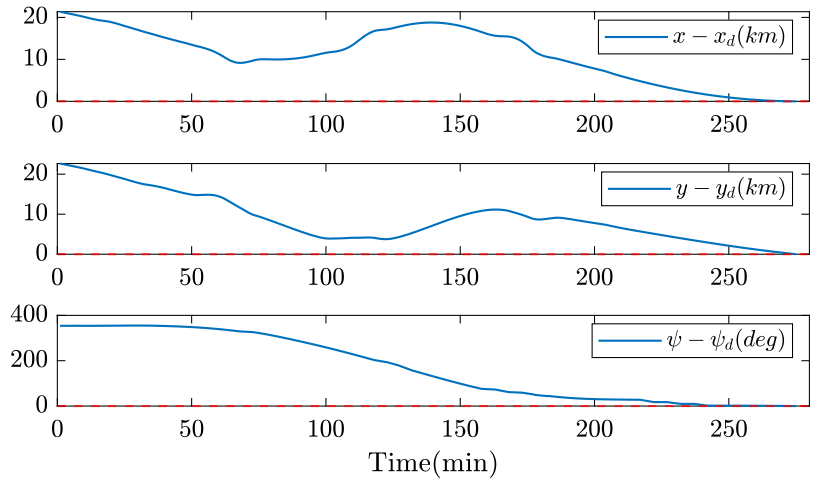

Figure 3.5: Variations of the error $\boldsymbol{\eta} - \boldsymbol{\eta}_d$ with time under the learned policy $\boldsymbol{\pi}_{\boldsymbol{\theta}^\star}$. Red line: the desired value.

the learned optimal policy $\boldsymbol{\pi}(\boldsymbol{\theta}^{\star})$. The red dash lines in these two figures represent the zero-valued reference lines. It can be seen that both the pose error and velocity converge to the red dash lines, which signifies a satisfactory docking. The variations of the vessel's thruster force $\mathbf{f}$

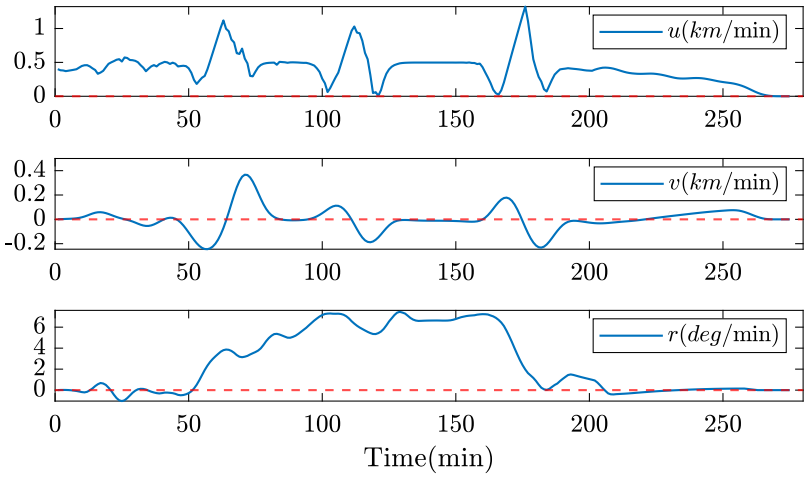

Figure 3.6: Variations of the vessel velocity $\boldsymbol{\nu}$ with time under the learned policy $\boldsymbol{\pi}_{\boldsymbol{\theta}^{\star}}$. Red line: the desired value.

and thruster angle $\boldsymbol{\alpha}$ under policy $\boldsymbol{\pi}(\boldsymbol{\theta}^{\star})$ are exhibited in Fig. 3.7. The red dash lines stand for the desired values as before, and the green lines stand for the constraint values. Note that, since α_1 of the tunnel thruster is fixed as $\pi/2$, its force f_1 is a vector that restricted in $[-100, 100]\mathrm{KN}$, while f_2, f_3 are scalars within $[0, 200]\mathrm{KN}$. For thruster angles, $\alpha_{q_{\max}}$ in (3.5) is chosen as 170°. Both the forces and the angles obey their constraints, and when approaching the endpoint, the forces decline to zero and the angles remain constant.

3.6 Conclusion

This chapter presents an MPC-LSTD-based DPG method for the ASV to accomplish a freight mission, which includes collision-free path tracking,

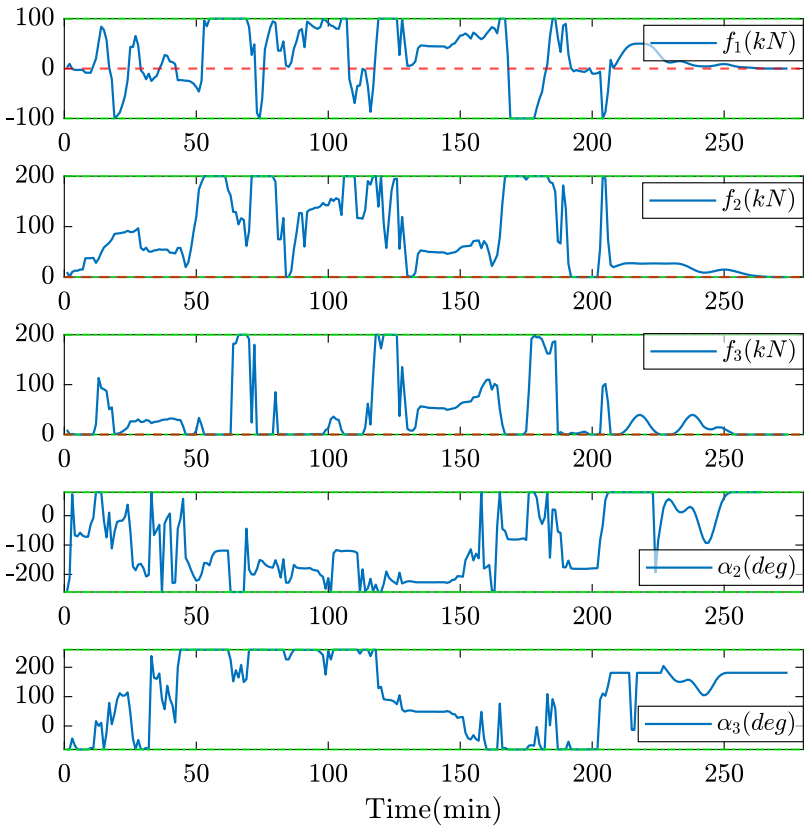

Figure 3.7: Variations of the thruster force $\mathbf{f}$ and thruster angle $\boldsymbol{\alpha}$ with time under the learned policy $\boldsymbol{\pi}_{\boldsymbol{\theta}^\star}$. Green line: the constraint value.

autonomous docking, and an ingenious transition between them. We use a parameterized MPC-scheme as the policy approximation function, and adopt the LSTD-based DPG method to update the parameters such that the closed-loop performance is minimized after learning. For future works, we will further validate our proposed method by realizing the experimental implementations.

References

[1] Justin E Manley. "Unmanned surface vehicles, 15 years of development". In: *OCEANS 2008*. IEEE. 2008, pp. 1–4.

[2] Scott Savitz et al. *US Navy employment options for unmanned surface vehicles (USVs)*. Tech. rep. RAND National Defense Research Institute Santa Monica, CA, 2013.

[3] Byung-Cheol Kum et al. "Monitoring applications for multifunctional unmanned surface vehicles in marine coastal environments". In: *Journal of Coastal Research* 85 (2018), pp. 1381–1385.

[4] Marte Hvarnes Evensen. "Safety and security of autonomous vessels. Based on the Yara Birkeland project". MA thesis. The University of Bergen, 2020.

[5] Nan Gu, Zhouhua Peng, Dan Wang, Yang Shi, and Tianlin Wang. "Antidisturbance coordinated path following control of robotic autonomous surface vehicles: Theory and experiment". In: *IEEE/ASME transactions on mechatronics* 24.5 (2019), pp. 2386–2396.

[6] Lu Liu, Dan Wang, Zhouhua Peng, Tieshan Li, and CL Philip Chen. "Cooperative path following ring-networked under-actuated autonomous surface vehicles: Algorithms and experimental results". In: *IEEE transactions on cybernetics* 50.4 (2018), pp. 1519–1529.

[7] Edmund Førland Brekke et al. “The Autosea project: Developing closed-loop target tracking and collision avoidance systems”. In: *journal of physics: conference series*. Vol. 1357. 1. IOP Publishing. 2019, pp. 012–020.

[8] Glenn Bitar, Bjørn-Olav H Eriksen, Anastasios M Lekkas, and Morten Breivik. “Energy-optimized hybrid collision avoidance for ASVs”. In: *2019 18th European Control Conference (ECC)*. IEEE. 2019, pp. 2522–2529.

[9] Andreas B Martinsen, Glenn Bitar, Anastasios M Lekkas, and Sébastien Gros. “Optimization-Based Automatic Docking and Berthing of ASVs Using Exteroceptive Sensors: Theory and Experiments”. In: *IEEE Access* 8 (2020), pp. 204974–204986.

[10] Andreas B Martinsen, Anastasios M Lekkas, and Sebastien Gros. “Autonomous docking using direct optimal control”. In: *IFAC-PapersOnLine* 52.21 (2019), pp. 97–102.

[11] Arash Bahari Kordabad, Hossein Nejatbakhsh Esfahani, Anastasios M. Lekkas, and Sébastien Gros. “Reinforcement Learning based on Scenario-tree MPC for ASVs”. In: *2021 American Control Conference (ACC)*. IEEE. 2021.

[12] Qingrui Zhang, Wei Pan, and Vasso Reppa. “Model-reference reinforcement learning for collision-free tracking control of autonomous surface vehicles”. In: *arXiv preprint arXiv:2008.07240* (2020).

[13] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[14] Richard S Sutton, David A McAllester, Satinder P Singh, and Yishay Mansour. “Policy gradient methods for reinforcement learning with function approximation”. In: *Advances in neural information processing systems*. 2000, pp. 1057–1063.

[15] Csaba Szepesvári. “Algorithms for reinforcement learning”. In: *Synthesis lectures on artificial intelligence and machine learning* 4.1 (2010), pp. 1–103.

[16] Lucian Buşoniu, Tim de Bruin, Domagoj Tolić, Jens Kober, and Ivana Palunko. "Reinforcement learning for control: Performance, stability, and deep approximators". In: *Annual Reviews in Control* 46 (2018), pp. 8–28.

[17] Mario Zanon and Sébastien Gros. "Safe reinforcement learning using robust MPC". In: *IEEE Transactions on Automatic Control* (2020).

[18] Sébastien Gros and Mario Zanon. "Data-driven economic nmpc using reinforcement learning". In: *IEEE Transactions on Automatic Control* 65.2 (2019), pp. 636–648.

[19] Eduardo F Camacho and Carlos Bordons Alba. *Model predictive control*. Springer science & business media, 2013.

[20] Sébastien Gros and Mario Zanon. "Reinforcement Learning for mixed-integer problems based on MPC". In: *ArXiv Preprint:2004.01430* (2020).

[21] Torsten Koller, Felix Berkenkamp, Matteo Turchetta, and Andreas Krause. "Learning-based model predictive control for safe exploration". In: *2018 IEEE Conference on Decision and Control (CDC)*. IEEE. 2018, pp. 6059–6066.

[22] Mohak Bhardwaj, Sanjiban Choudhury, and Byron Boots. "Blending MPC & Value Function Approximation for Efficient Reinforcement Learning". In: *ArXiv Preprint:2012.05909* (2020).

[23] Mario Zanon, Sébastien Gros, and Alberto Bemporad. "Practical reinforcement learning of stabilizing economic MPC". In: *2019 18th European Control Conference (ECC)*. IEEE. 2019, pp. 2258–2263.

[24] Arash Bahari Kordabad, Hossein Nejatbakhsh Esfahani, Anastasios M Lekkas, and Sebastien Gros. "Reinforcement learning based on scenario-tree MPC for ASVs". In: *2021 American Control Conference (ACC)* (2021), pp. 1985–1990. DOI: 10.23919/ACC50511.2021.9483100.

[25] Yara International ASA. *Yara Birkeland Project*. https://www.yara.com/news-and-media/media-library/press-kits/yara-birkeland-press-kit/. 2023.

[26] Roger Skjetne, Øyvind Smogeli, and Thor I Fossen. "Modeling, identification, and adaptive maneuvering of Cybership II: A complete design with experiments". In: *IFAC Proceedings Volumes* 37.10 (2004), pp. 203–208.

[27] Tor Arne Johansen, Thor I Fossen, and Stig P Berge. "Constrained nonlinear control allocation with singularity avoidance using sequential quadratic programming". In: *IEEE Transactions on Control Systems Technology* 12.1 (2004), pp. 211–216.

4 | MPC-based RL for Residential Microgrids

This chapter presents an Energy Management (EM) strategy for residential microgrid systems using Model Predictive Control (MPC)-based Reinforcement Learning (RL) and Shapley value. We construct a typical residential microgrid system that considers fluctuating spot-market prices, highly uncertain user demand and renewable generation, and collective peak power penalties. To optimize the benefits for all residential prosumers, the EM problem is formulated as a Cooperative Coalition Game (CCG). The objective is to first find an energy trading policy that reduces the collective economic cost (including spot-market cost and peak-power cost) of the residential coalition, and then to distribute the profits obtained through cooperation to all residents. An MPC-based RL approach, which compensates for the shortcomings of MPC and RL and benefits from the advantages of both, is proposed to reduce the monthly collective cost despite the system uncertainties. To determine the amount of monthly electricity bill each resident should pay, we transfer the cost distribution problem into a profit distribution problem. Then, the Shapley value approach is applied to equitably distribute the profits (i.e., cost savings) gained through cooperation to all residents based on the weighted average of their respective marginal contributions. Finally, simulations are performed on a three-household microgrid system located in Oslo, Norway, to validate the proposed strategy, where a real-world dataset of April 2020 is used. Simulation results show that the proposed MPC-based RL approach could effectively reduce the long-term economic cost

by about 17.5%, and the Shapley value method provides a solution for allocating the collective bills fairly.

4.1 Introduction

With the recent development of smart grids and smart cities, residential communities have become essential stakeholders in future power grid transactions [1, 2]. The aggregation of Distributed Generations (DGs), Renewable Energy Resources (RERs), Energy Storage Systems (ESSs), and controllable loads of consumers has led to the concept of residential microgrids. To coordinate various DGs, RERs, ESSs, and loads to achieve an optimal energy dispatch strategy considering operating costs, power demand, and consumer preferences, the concept of microgrid Energy Management (EM) has been proposed [3]. Its purpose is to effectively coordinate the energy sharing/trading between each microgrid and the main grid, and to achieve the optimal allocation of energy sharing through rational strategies, thus improving the stability, reliability, and energy efficiency of the overall power system. However, the volatile and intermittent characteristics of RERs (e.g., Wind Turbines (WTs), Photo-Voltaic (PV)) and the uncertainties of customer load demand pose serious challenges to the energy management problem [4]. There have been many research works for the EM problem. We mainly focus on Model Predictive Control (MPC)-based approaches and Reinforcement Learning (RL)-based approaches.

4.1.1 MPC-based Approaches

For EM problems, MPC is a widely considered and well-studied control strategy. It leverages a known and explicit model to describe the dynamics of residential microgrids, and can handle system constraints explicitly. The authors in [5] applied a standard MPC approach to solve the EM problem while satisfying time-varying requirements and operational constraints. Considering the uncertainties associated with fluctuations in

demand and RERs production, a two-layer stochastic MPC approach was proposed in [6]. Besides, uncertainties can also arise from the energy storage units and demand-side management technologies in microgrids. To accommodate these combined uncertainties, a two-stage robust MPC-based optimization method was introduced [7]. In [8], the authors further considered fluctuations in stochastic RERs as well as weather conditions, and presented an innovative control strategy-based coordinated MPC approach for networked smart greenhouse integrated microgrids. The work in [9] presented a hierarchical distributed MPC mechanism to tackle the EM problem with multi-time frames and multi-layer optimization, dedicated to large-scale system management. The author of [10] proposed a novel EM framework based on tube-based MPCs that made energy trading strategies robust to system uncertainties, reducing the loss of economic performance and computational efficiency.

Despite the advantages of MPC over other classical control strategies and its successful application in the aforementioned works, it has three known drawbacks [11, 12]: 1. A sufficiently accurate model is required, which is often difficult or even impossible for complex or highly uncertain microgrid systems (e.g., where the RERs production, user demand, and electricity prices are highly volatile). 2. Even if the current model is accurate, in the long run it may not represent the characteristics of the real system. For example, the structure, size, and capacity of the microgrid may change over time, as may the uncertain distribution of RERs and demand, which will result in suboptimal operation of the MPC. And if the controller is redesigned according to system modifications, additional development and maintenance costs will be required. 3. Due to the finite-horizon formulation of MPC, it is challenging to fully consider long-term objectives and constraints (e.g., seasonal storage and monthly peak power penalties).

4.1.2 RL-based Approaches

RL, in contrast, does not rely on system models. RL learns a policy by interacting with the environment and can make good use of data to reduce

the effects of uncertainties and disturbances [13]. Many researchers have deployed RL methods to solve the EM problem. The authors in [14] proposed a two-step RL algorithm to plan battery scheduling for microgrid EM. A deep RL-based adaptive EM strategy was proposed in [15] for a microgrid with flexible demand. In [16], the authors presented an RL algorithm based on a multi-agent system to develop an optimal strategy for distributed EM. Similarly, a fuzzy Q-Learning for multi-agent decentralized EM was presented in [17]. The authors in [18] proposed an online optimization algorithm with Q-learning techniques to address economic dispatch and unit commitment in smart grids. In [19], the authors proposed a joint load scheduling strategy for household EM that aims at reducing residential energy costs while maintaining thermal comfort. The proposed method was also compared with a deep Q-network-based approach and an MPC-based approach. For more recent advances in RL-based approaches to residential microgrid EM, see [20–24].

However, for the EM problem, those using conventional RL approaches tend to suffer from the following three difficulties [12]: 1. Since it is a model-free approach, for complex or highly uncertain microgrid systems, RL requires a huge amount of data to learn the policy from scratch. 2. The forecasts of RERs and demand in the form of time-series in the decision policy create high-dimensional information spaces that are detrimental to efficient learning. 3. RL often relies on Deep Neural Networks (DNNs) as function approximators [25], yet DNN-based RL does not offer formal tools to satisfy system constraints and evaluate closed-loop stability. Furthermore, there is no systematic or physically meaningful way to choose the initial values, number of hidden layers, number of hidden units, etc. of a DNN network [26].

For a comprehensive comparative study of MPC and RL, see [12]. There, the two methods are benchmarked simultaneously in the context of a multi-energy management system, and their advantages and disadvantages are elaborated.

4.1.3 Combining MPC and RL

To compensate for the shortcomings of MPC and RL and benefit from the advantages of both, we propose to combine these two strategies to develop an MPC-based RL approach. Instead of DNN, a parameterized MPC scheme is used as a function approximation of the optimal policy, and RL helps to tune the parameters to reduce the long-term performance (objective function). The idea of the MPC-based RL was first formally proposed in [27], along with a complete analysis and a formal argument. It states that, theoretically, under a mild condition[1], if the MPC scheme is parametrized "richly" enough, MPC can deliver the optimal policy for the real system even if the MPC model is inaccurate or the system has disturbances and uncertainties (see *Theorem 1* and *Corollary 2* in [27] for rigorous statements and explanations). Besides, this approach has been further developed [28–30] and successfully applied to some practical problems [31–33].

The MPC-based RL approach has the following merits that motivate us to use it to address the EM problem:

- Rather than learning from scratch, it could use prior knowledge of the model to start learning with a suboptimal but reasonable policy. (Thanks to MPC.)
- System constraints (e.g., battery capacity) can be explicitly taken into account, whereas satisfying constraints can be challenging in pure RL methods. (Thanks to MPC.)
- The rich theoretical framework behind MPC makes it possible to analyze the stability of the system and the feasibility of the solution. (Thanks to MPC.)
- System uncertainties (e.g., uncertainties in user demand, RERs, etc.) can be overcome to some extent while obtaining optimized closed-loop performance, whereas pure robust/stochastic MPCs

[1] See equation (9) in [27].

tend to deliver conservative solutions that usually result in some performance loss. (Thanks to RL.)

- It can deal with long-term or even infinite-horizon problems (e.g., consider the long-term operating costs of the system). (Thanks to RL.)

- It is a performance-driven approach rather than a model-driven approach, i.e., the MPC parameters are tuned to improve the closed-loop performance of the MPC scheme (e.g., reducing power costs in our case) rather than to replicate the dynamics of the real system as faithfully as possible.

4.1.4 Contributions

This chapter is an extension of [34] with some significant new contributions (marked in parentheses in the bullet points below) and presents a more comprehensive discussion. The novelty of this chapter is that we propose a novel and complete solution for microgrid energy management. To the best of our knowledge, the MPC-based RL approach, proposed in [27], has never been deployed to solve the optimization problem of EM. And the Shapley value method, although a known method for profit distribution problems, is used to solve the cost distribution problem of EM with our modifications (by appropriate definitions of profits and utility values). The main contributions of this article are summarized as follows:

(i) We construct a typical residential microgrid EM problem, which considers fluctuating spot-market prices, highly uncertain user demand and RERs, and peak power penalties. Accordingly, the problem is formulated as a Cooperative Coalition Game (CCG) and a two-step strategy is proposed to first find an energy trading policy that reduces the collective monthly cost of residents and then distribute the cost fairly. (New contribution.)

(ii) Given the difficulties posed by highly uncertain user demand and RERs (i.e., stochastic local power consumption-production), we innovatively apply an MPC-based RL approach to find the energy trading policy that could reduce the collective cost (including spot-market cost and peak-power cost) of the CCG.

(iii) A feasible solution for distributing the collective cost is proposed. The energy management problem in this work requires the distribution of collective cost rather than collective profit, so one contribution is to transform the cost distribution problem into a profit distribution problem. Specifically, in the context of the EM problem, we define properly the individual and coalitional profits and utility values. Then, the collective bill is equitably distributed to all residents based on the weighted average of their respective marginal contributions, using the Shapley value mechanism. (New contribution.)

(iv) A complete policy learning and profit distribution algorithm is presented for the residential microgrid EM problem. (New contribution.)

(v) The performance of the proposed algorithm is validated by conducting multiple stochastic experiments using a real-world dataset. The results show that the monthly collective cost is reduced by about 17.5% by using the MPC-based RL approach, and the optimized collective cost can be fairly distributed to users by the Shapley value approach. (Greatly improved.)

The remainder of this chapter is organized as follows. The problem background and formulation are given in Section 4.2. Section 4.3 develops the MPC-based RL and Shapley value approaches for the residential EM. A case study is presented in Section 4.4. Finally, Section 4.5 draws the conclusions, limitations, and future works.

4.2 Problem Formulation – Microgrid Energy Management

4.2.1 Background

Consider a typical residential microgrid system, as shown in Fig. 4.1. The main grid provides power to a group of households through a local transformer. Each household is a prosumer whose electricity profile has three components: production, consumption, and storage. Home-scale renewable energy sources, like PVs, are used to generate electricity. Power consumption comes from heating systems, appliances, Electrical Vehicle (EV) chargers, etc. The battery is used to store energy generated by RERs or purchased from the main grid, and its State-Of-Charge (SOC) indicates the level of charge relative to the battery capacity.

The spot-market prices are hourly prices, announced publicly one day in advance (e.g., prices for the next 24 hours are announced at around $13{:}00$ each day). Therefore, all residents know the electricity prices for at least the next 12 hours. This operating mode incentivizes residential prosumers to consume and/or store energy at low prices, and to reduce consumption and/or use the stored energy to meet demand when prices are high [35]. As a result, each household could decide on an energy trading policy (buying or selling amount) based on its power demand and spot-market prices to reduce power costs. We refer to this cost incurred by trading energy with the grid as the *spot-market cost* throughout the text. However, finding such an energy trading policy is quite tricky due to the high uncertainties in load demand and RERs (in local power consumption and production).

Furthermore, the spot market may exacerbate power peaks. If all houses in a neighborhood tend to exchange power with the grid at the same time, the local power infrastructures (e.g., transformers), to ensure safe operation, must be sized accordingly. Such sizing incurs high investment costs for the distribution system. Utilities are thus attempting to reduce the peak power that prosumers impose on the system via some economic

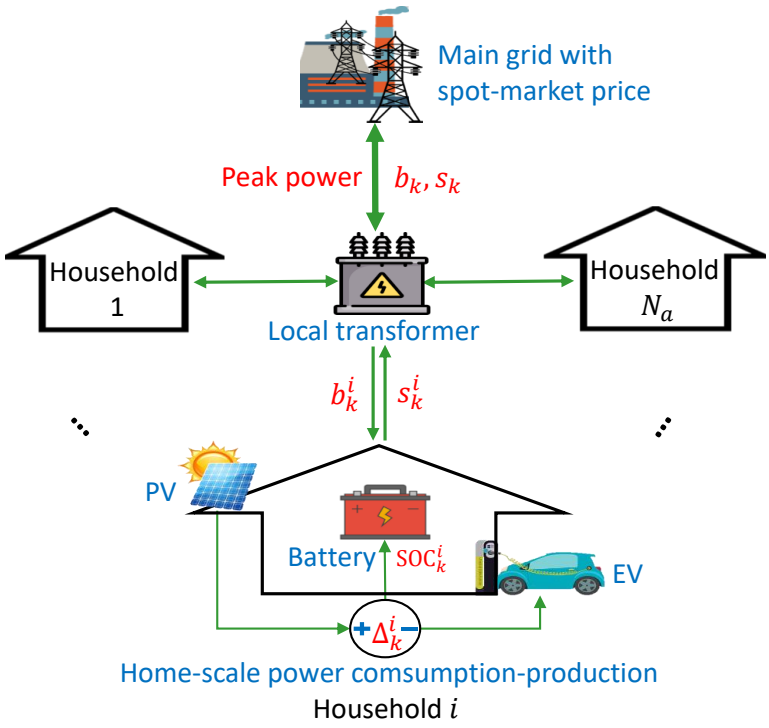

Figure 4.1: Illustration of a typical N_a-agent residential microgrid system.

incentives. An ideal structure of the economic incentive is to invoice the prosumers behind a given transformer collectively. This collective bill includes the *peak-power cost* caused jointly by all users, as well as the sum of their individual spot-market costs. However, implementing such management is not trivial because of two reasons:

- Reducing peak-power cost and spot-market cost are competing goals. To avoid high peak-power penalties, or to make more peak profits (pay less for peak costs), some households may need to make some “compromises", such as deviating from their preferred consumption patterns and/or buying or selling less energy than ideally desired. This leads to lower profits from the power exchange. Therefore, the balance between peak-power cost and spot-market cost is delicate.

- Since the peak profits are earned at the expense of users’ spot-market profits, and the final collective profit is obtained by offsetting the two, this collective profit should be distributed based on the principle that prosumers who contribute more to the coalition (“sacrifice" more) receive a larger share of the profit. Nevertheless, it is challenging to properly assess the contribution of each individual house and distribute the profit in a fair manner.

Given all this background, we consider the problem from a coalition perspective. The goal is to first find a global energy trading policy that reduces the collective cost (including spot-market cost and peak-power cost) for all residents over the course of a month; and then to find a reasonable distribution strategy that allocates the collective bill fairly, i.e., determines the electricity fee each prosumer should pay. The difficulties lie in the need to deal with high uncertainties of the system when finding the energy trading policy, and the need to equitably measure user contributions when allocating the collective bill.

4.2.2 System Model

Considering Fig. 4.1, the stochastic dynamics of the energy profile for each household can be simply modeled as follows

$$\mathrm{soc}_{k+1}^{i} = \mathrm{soc}_{k}^{i} + \alpha^{i}\left(b_{k}^{i} - s_{k}^{i} - \Delta_{k}^{i}\right) \tag{4.1a}$$

$$\Delta_{k}^{i} = (\beta_{\Delta})_{k}^{i} + \delta_{k}^{i}, \tag{4.1b}$$

where superscript $i = 1, \ldots, N_a$ denotes the index of the i^{th} user among the total N_a users. The sampling time is $1\mathrm{h}$ as the spot-market price is hourly and the subscript $k = 0,\ 1,\ 2\ldots$ is the physical time index. Equation (4.1a) describes the energy flow of the user from the battery perspective. State soc_{k}^{i} is a dimensionless variable corresponding to the SOC level of battery i at time t_k, which satisfies

$$0 \leq \mathrm{soc}_{k}^{i} \leq 1. \tag{4.2}$$

The interval $[0,\ 1]$ represents the SOC level considered as non-damaging for the battery (typically $20\% - 80\%$ range of the physical SOC). Constant $\alpha^{i}\left[\mathrm{kWh}^{-1}\right] > 0$ reflects the battery size. Input b_{k}^{i} (resp. s_{k}^{i}) $[\mathrm{kWh}]$ is the energy bought (resp. sold) from (resp. to) the main grid in time interval $[t_k,\ t_{k+1}]$, bounded as

$$0 \leq b_{k}^{i} \leq \bar{U}^{i}, \qquad 0 \leq s_{k}^{i} \leq \bar{U}^{i}, \tag{4.3}$$

where $\bar{U}^{i}$ is the maximum allowed buying (resp. selling) amount of the i^{th} user. The state soc_{k}^{i} is determined by the amount of electricity bought b_{k}^{i} and sold s_{k}^{i} and the power consumption-production difference Δ_{k}^{i} within time interval $[t_k,\ t_{k+1}]$. The dynamics of the state Δ_{k}^{i} $[\mathrm{kWh}]$ is provided in (4.1b), where $(\beta_{\Delta})_{k}^{i}$ is the forecast of the consumption-production difference at time t_k and δ_{k}^{i} is the corresponding forecast error. Note that the forecast β_{Δ} can be easily obtained since the profile of user demand and the generation pattern of RERs are usually known. However, such forecasts are inaccurate and stochastic, as the production of RERs may be affected by factors such as weather, and the load demand may change significantly due to sudden changes in lifestyle habits. For simplicity, we consider a normally distributed forecast error

$$\delta_{k}^{i} \sim \mathcal{N}\left(0,\ {\sigma^{i}}^{2}\right). \tag{4.4}$$

Consequently, for the whole system which is a set $\mathcal{M} = \{1,\, 2,\, \ldots,\, N_a\}$ of N_a households, we denote the system state $\mathbf{s}_k$ and input[2] $\mathbf{a}_k$, respectively, as

$$\mathbf{s}_k = \{\mathrm{soc}_k^1,\, \Delta_k^1,\, \ldots,\, \mathrm{soc}_k^{N_a},\, \Delta_k^{N_a}\}, \tag{4.5a}$$

$$\mathbf{a}_k = \{b_k^1,\, s_k^1,\, \ldots,\, b_k^{N_a},\, s_k^{N_a}\}. \tag{4.5b}$$

Besides, it is worth mentioning that the battery model we adopted is ideal because we ignore the effects of charge/discharge efficiency, energy leakage, and battery usage on battery life. However, these factors can be easily incorporated into the battery model without substantially affecting our methodology and analysis.

4.2.3 Cooperative Coalition Game

In this section, we formulate the EM problem as a Cooperative Coalition Game (CCG). CCG refers to the formation of coalitions or binding agreements between game players in order to work together to achieve a common goal. CCGs emphasize collective rationality, efficiency, fairness, and equality, rather than individual rationality and personal optimal decisions. This means that the participants are no longer in complete competition with each other, avoiding the overall inefficiencies caused by non-collaborative behavior among participants [36]. In this particular energy trading CCG, individual prosumers are supposed to cooperate together to reduce the collective cost including spot-market cost and peak-power cost. This CCG is formulated as an *episodic task* in the RL framework, since the electricity bill is paid on a monthly basis.

Collective spot-market cost

The spot-market cost of each agent at time t_k is modeled as a linear function based on the difference between the profit made by selling

[2]The *input* in the control community corresponds to the *action* in the RL context.

electricity to the power grid, and the losses incurred from buying it [37], i.e.,

$$L_{\mathrm{S}}^{i}(b_k^i,\, s_k^i) = \phi_k^b b_k^i - \phi_k^s s_k^i, \tag{4.6}$$

where ϕ_k^b and ϕ_k^s are the spot-market buying and selling prices at time instance t_k. The *spot-market stage cost* $L_{\mathrm{S}}(\mathbf{s}_k,\, \mathbf{a}_k)$ is defined as the sum of the individual spot-market costs for all prosumers

$$L_{\mathrm{S}}(\mathbf{s}_k,\, \mathbf{a}_k) = \sum_{i=1}^{N_a} L_{\mathrm{S}}^{i}(b_k^i,\, s_k^i). \tag{4.7}$$

Then, the collective spot-market cost $\mathcal{S}$ of the whole system in the entire interval $[t_0,\, t_K]$ is expressed as

$$\mathcal{S} := \sum_{k=0}^{K} L_{\mathrm{S}}(\mathbf{s}_k,\, \mathbf{a}_k) = \sum_{k=0}^{K} \sum_{i=1}^{N_a} L_{\mathrm{S}}^{i}(b_k^i,\, s_k^i), \tag{4.8}$$

where K is the index of t_K, the terminal time of the episodic task, which is assumed to be one month in this work with $t_K = 720\mathrm{h}$.

Collective peak-power cost

To formulate the collective peak-power cost, we first introduce the following monotonically non-decreasing variable P_k^{peak}

$$P_{k+1}^{\mathrm{peak}} = \max\left(P_k^{\mathrm{peak}},\, \sum_{i=1}^{N_a} b_k^i,\, \sum_{i=1}^{N_a} s_k^i\right), \tag{4.9a}$$

$$P_0^{\mathrm{peak}} = 0. \tag{4.9b}$$

Variable P_{k+1}^{peak} describes the peak power up to time t_k, and thus P_{K+1}^{peak} is the monthly peak power for the entire interval $[t_0,\, t_K]$. The collective peak-power cost $\mathcal{P}$ can then be expressed as

$$\mathcal{P} := \lambda P_{K+1}^{\mathrm{peak}}, \tag{4.10}$$

where $\lambda > 0$ is a constant coefficient, determined by the bearing capacity of the local transformer.

Objective function in RL

Given the collective spot-market cost $\mathcal{S}$ and the collective peak-power cost $\mathcal{P}$, the objective is to find a control policy $\boldsymbol{\pi}(\mathbf{s})$ that reduces the monthly cost $J(\boldsymbol{\pi})$ (i.e., the closed-loop performance) for the multi-agent system, expressed as

$$\begin{aligned} J(\boldsymbol{\pi}) =& \mathbb{E}_{\boldsymbol{\pi}}\left[\mathcal{P}+\mathcal{S}\,\middle|\,\mathbf{a}_k=\boldsymbol{\pi}(\mathbf{s}_k)\right] \\ =& E_{\boldsymbol{\pi}}\left[\lambda P_{K+1}^{\text{peak}}+\sum_{k=0}^{K} L_{\text{S}}(\mathbf{s}_k,\mathbf{a}_k)\,\middle|\,\mathbf{a}_k=\boldsymbol{\pi}(\mathbf{s}_k)\right], \end{aligned} \tag{4.11}$$

where the expectation $\mathbb{E}_{\boldsymbol{\pi}}$ is taken over the distributions of the system uncertainties.

To deploy a classical RL method, we recast the monthly collective peak-power cost (4.10) as a sum of *peak-power stage cost* $L_P(\mathbf{s}_k,\,\mathbf{a}_k)$, which is defined as

$$L_P(\mathbf{s}_k,\mathbf{a}_k)=\lambda(P_{k+1}^{\text{peak}}-P_k^{\text{peak}}). \tag{4.12}$$

Obviously, (4.10) equals to the sum of (4.12) over $[t_0,\,t_K]$, because

$$\begin{aligned} \sum_{k=0}^{K} L_P(\mathbf{s}_k,\,\mathbf{a}_k) &= \sum_{k=0}^{K}\lambda(P_{k+1}^{\text{peak}}-P_k^{\text{peak}}) \\ &= \lambda(P_{K+1}^{\text{peak}}-P_0^{\text{peak}})=\lambda P_{K+1}^{\text{peak}}. \end{aligned} \tag{4.13}$$

Then, the stage cost function $L(\mathbf{s}_k,\,\mathbf{a}_k)$ of this problem can be defined as consisting of the peak-power stage cost L_P and the spot-market stage cost L_S, i.e.,

$$L(\mathbf{s}_k,\mathbf{a}_k)=\underbrace{L_P(\mathbf{s}_k,\,\mathbf{a}_k)}_{\text{Peak-power stage cost}}+\underbrace{L_{\text{S}}(\mathbf{s}_k,\,\mathbf{a}_k)}_{\text{Spot-market stage cost}}. \tag{4.14}$$

Consequently, the objective function (4.11) in RL can be equivalently expressed as the expected sum of the stage costs (4.14) over the interval

$[t_0,\, t_K]$, i.e.,

$$J(\boldsymbol{\pi}) = \mathbb{E}_{\boldsymbol{\pi}}\left[\sum_{k=0}^{K} L(\mathbf{s}_k,\, \mathbf{a}_k)\middle|\mathbf{a}_k = \boldsymbol{\pi}(\mathbf{s}_k)\right]. \tag{4.15}$$

Note that no discount factor is used here as we consider an episodic scenario.

Finally, the CCG problem can be formulated as

$$\begin{aligned}\min_{\boldsymbol{\pi}} J(\boldsymbol{\pi}) &= \mathbb{E}_{\boldsymbol{\pi}}\left[\sum_{k=0}^{K} L(\mathbf{s}_k,\, \mathbf{a}_k)\middle|\mathbf{a}_k = \boldsymbol{\pi}(\mathbf{s}_k)\right] \\ \text{s.t.}\quad & \forall i = 1,\, \ldots,\, N_a,\ \forall k = 0,\, \ldots,\, K \\ & (4.1),\ (4.2),\ (4.3),\ (4.9),\end{aligned} \tag{4.16}$$

where all prosumers jointly decide the control policy $\boldsymbol{\pi}(\mathbf{s})$ to reduce the monthly collective cost $J(\boldsymbol{\pi})$.

4.3 MPC-based RL and Shapley Value Methods

As the core of this chapter, this section presents an EM solution to the CCG problem in two steps: policy learning and profit distribution. The objective of policy learning is to train a policy that reduces the monthly collective cost (i.e., solve (4.16)) by the MPC-based RL approach. We use a parameterized MPC as the optimal policy function approximation whose parameters are learned using a Least Square (LS)-based Deterministic Policy Gradient (DPG) optimizer. The profit distribution discusses how the profits obtained in the previous step are fairly distributed to all prosumers through the Shapley value mechanism, and yields the electricity bill that each household should pay. The complete policy learning and profit distribution EM algorithm is developed at the end.

4.3.1 MPC as An Optimal Policy Approximation

One technical point of the MPC-based RL approach is to use MPCs as function approximators in RL. In this EM problem, the MPC is employed as a function approximation for the optimal policy $\boldsymbol{\pi}^\star(\mathbf{s})$. However, there are two difficulties here:

1. The horizons of MPC and RL are significantly inconsistent. The objective of RL is to reduce the monthly cost $J(\boldsymbol{\pi})$, whose horizon is $720\mathrm{h}$. However, since prosumers only know the power prices for $12\mathrm{h}$ in advance, the prediction horizon of MPC is only 12, which is much shorter than the horizon of RL.

2. The MPC scheme should be fairly simple since it is beneficial to use a cheap computing architecture for investment cost reasons. However, the power consumption-production difference Δ_k^i is stochastic, i.e., there exist uncertainties in the real system. Hence, a simple deterministic MPC cannot capture the optimal policy, and in fact, even a stochastic MPC may not capture the policy well as we typically just make a number of approximations (typ. scenario trees) when doing so.

For the above two reasons, the policy delivered using a simple MPC scheme can be significantly suboptimal. Therefore, it is very reasonable to perform MPC adaptation. In this chapter, we parameterize the MPC model and cost functions (one could also parameterize the MPC constraints). Then, RL is applied to adjust these parameters according to the principle of improving the closed-loop performance evaluated by (4.15). We describe this method in detail next.

Consider the following MPC-scheme parametrized with $\boldsymbol{\theta}$

$$\min_{\widehat{\mathbf{soc}},\hat{\boldsymbol{\Delta}},\hat{\mathbf{b}},\hat{\mathbf{s}}} \ \theta_\lambda \hat{P}_N^{\mathrm{peak}} + \sum_{i=1}^{N_a} \Bigg(T(\widehat{\mathrm{soc}}_N^i,\, \theta_T^i,\, \theta_\varphi^i) + \sum_{j=0}^{N-1} \Big(L_S^i(\hat{b}_j^i,\, \hat{s}_j^i) + \psi(\widehat{\mathrm{soc}}_j^i,\, \theta_\psi^i,\, \theta_\varphi^i) \Big) \Bigg) \tag{4.17a}$$

$$
\begin{aligned}
\text{s.t.} \quad & \forall i = 1, \dots, N_a, \quad \forall j = 0, \dots, N-1 \\
& \widehat{\text{soc}}^i_{j+1} = \widehat{\text{soc}}^i_j + \theta^i_\alpha(\hat{b}^i_j - \hat{s}^i_j - \hat{\Delta}^i_j), && \text{(4.17b)} \\
& \hat{\Delta}^i_j = \theta^i_\beta(\beta_\Delta)^i_j + \theta^i_\delta, && \text{(4.17c)} \\
& 0 \le \widehat{\text{soc}}^i_j \le 1, \qquad 0 \le \widehat{\text{soc}}^i_N \le 1, && \text{(4.17d)} \\
& 0 \le \hat{b}^i_j \le \bar{U}^i, \qquad 0 \le \hat{s}^i_j \le \bar{U}^i, && \text{(4.17e)} \\
& \hat{P}^{\text{peak}}_{j+1} = \max\left(\hat{P}^{\text{peak}}_j, \sum_{i=1}^{N_a} \hat{b}^i_j, \sum_{i=1}^{N_a} \hat{s}^i_j\right), && \text{(4.17f)} \\
& \widehat{\text{soc}}^i_0 = \text{soc}^i_k, \;\; \hat{\Delta}^i_0 = \Delta^i_k, \;\; \hat{P}^{\text{peak}}_0 = 0, && \text{(4.17g)}
\end{aligned}
$$

where N is the prediction horizon. Arguments $\widehat{\mathbf{soc}} = \{\widehat{\text{soc}}^{1,\dots,N_a}_{0,\dots,N}\}$, $\hat{\mathbf{\Delta}} = \{\hat{\Delta}^{1,\dots,N_a}_{0,\dots,N}\}$, $\hat{\mathbf{b}} = \{\hat{b}^{1,\dots,N_a}_{0,\dots,N-1}\}$, and $\hat{\mathbf{s}} = \{\hat{s}^{1,\dots,N_a}_{0,\dots,N-1}\}$ are the primal decision variables. The parameterization is implemented as follows. For difficulty 1), since the horizons of MPC and RL do not coincide, the value function of MPC can not represent the one for RL. To compensate for this inconsistency, we add additional parameterized stage cost $\psi(\widehat{\text{soc}}^i_j, \theta^i_\psi, \theta^i_\varphi)$ and terminal cost $T(\widehat{\text{soc}}^i_N, \theta^i_T, \theta^i_\varphi)$ as cost modifiers in (4.17a). Likewise, the peak power of the MPC at its terminal time is not the actual monthly peak power in RL. Therefore, the peak-power cost is parameterized as $\theta_\lambda \hat{P}^{\text{peak}}_N$, such that the discrepancy caused by the limited view of MPC would be compensated by an appropriate θ_λ. As for difficulty 2), we parameterize the original model with θ^i_α, θ^i_β, θ^i_δ to form a simple deterministic MPC which, with appropriate parameter values, is able to deliver the policy for the real stochastic system. It is worth mentioning that, as detailed in [27], it is actually the full parameterization of the MPC model, cost functions, and constraints that compensates for the system stochasticity and model error.

We now summarize (4.17) as follows:

- Cost (4.17a) includes the spot-market stage cost $L^i_S(\cdot)$, parameterized additional stage cost $\psi(\cdot)$ and terminal cost $T(\cdot)$, as well as the parameterized peak-power cost $\theta_\lambda \hat{P}^{\text{peak}}_N$.
- Equality constraints (4.17b) and (4.17c) represent the parameterized deterministic model of the real stochastic system (4.1).

- Inequality constraints (4.17d) and (4.17e) are the state and input constraints for each prosumer, respectively.
- Equality constraint (4.17f) indicates the dynamics of the peak power.
- Equality constraints (4.17g) handle the MPC initialization.

The additional stage cost $\psi(\cdot)$ and terminal $T(\cdot)$ cost in (4.17a) are designed as

$$\psi(\widehat{\mathrm{soc}}_j^i,\, \theta_\psi^i,\, \theta_\varphi^i) = \theta_\psi^i(\widehat{\mathrm{soc}}_j^i - \theta_\varphi^i)^2, \tag{4.18a}$$

$$T(\widehat{\mathrm{soc}}_N^i,\, \theta_T^i,\, \theta_\varphi^i) = \theta_T^i(\widehat{\mathrm{soc}}_N^i - \theta_\varphi^i)^2, \tag{4.18b}$$

which are quadratic functions of SOC with adjustable parameter θ_φ^i as their setpoint, and positive factors θ_T^i and θ_ψ^i as the coefficients. Overall, the adjustable parameters vector $\boldsymbol{\theta}$ is composed of

$$\boldsymbol{\theta} := \{\theta_\alpha^i,\, \theta_\beta^i,\, \theta_\delta^i,\, \theta_\psi^i,\, \theta_T^i,\, \theta_\varphi^i,\, \theta_\lambda\}, \quad i = 1, \ldots, N_a \tag{4.19}$$

in which θ_α^i, θ_β^i and θ_δ^i play a major role in compensating for the system stochasticity. Parameters θ_ψ^i with θ_φ^i and θ_T^i with θ_φ^i modify the MPC stage and terminal costs, respectively. Besides, the parameter θ_φ^i allows RL to assign some preferred SOC levels in the MPC scheme in view of improving its long-term performance. Parameter θ_λ weighs the peak-power cost against the spot-market cost. A larger θ_λ implies a greater focus on the peak-power cost in the total cost of the MPC scheme. Again, although the objective (4.15) involves a billing period of one month, which is much longer than the short MPC horizon, those parameters with suitable values can compensate for that discrepancy. Certainly, manually tuning these parameters to optimal is extremely difficult, but RL is well suited for this task, and all these parameters are tuned by RL toward reducing the long-term collective cost (4.15). It is worth mentioning that this choice of parameterization is arbitrary and different options are possible. From *Theorem 1* and *Corollary 2* in [27], we know that, theoretically, as long as the parametrization is "rich" enough, the MPC scheme is capable of capturing the optimal policy $\pi^\star(\mathrm{s})$.

The parameterized policy for agent i at time k is the first input of the input sequence delivered by the MPC scheme (4.17), i.e.,

$$\boldsymbol{\pi}^i_{\boldsymbol{\theta}}(\mathbf{s}_k) = \left[\hat{b}^{i\star}_0(\mathbf{s}_k, \boldsymbol{\theta}),\ \hat{s}^{i\star}_0(\mathbf{s}_k, \boldsymbol{\theta})\right]^\top, \tag{4.20}$$

where $\hat{b}^{i\star}_0$ and $\hat{s}^{i\star}_0$ are the first elements of $\hat{\mathbf{b}}^{i\star}$ and $\hat{\mathbf{s}}^{i\star}$, which are the solutions of the MPC scheme (4.17) associated to the decision variables $\hat{\mathbf{b}}^i$ and $\hat{\mathbf{s}}^i$. Consequently, the global parametric policy of the EM problem can be written as

$$\boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathbf{s}_k) = \left[\pi^{1\top}_{\boldsymbol{\theta}}, \ldots, \pi^{N_a\top}_{\boldsymbol{\theta}}\right]^\top. \tag{4.21}$$

The actual action performed in learning is obtained by adding a small Gaussian exploration $\boldsymbol{\varrho}$ to the policy, i.e.,

$$\mathbf{a}(\mathbf{s}_k) = \boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathbf{s}_k) + \boldsymbol{\varrho}. \tag{4.22}$$

4.3.2 LSTD-based Deterministic Policy Gradient

To adjust the parameters (4.19) of the MPC scheme (4.17), we use the LSTD-based DPG method elaborated in Section 2.4.2: *"Core Formulas: DPG for MPC-based RL"*. The general update rule is

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \boldsymbol{\eta} \odot \nabla_{\boldsymbol{\theta}} J(\boldsymbol{\pi}_{\boldsymbol{\theta}}), \tag{4.23}$$

where $\boldsymbol{\eta} > \mathbf{0}$ is the learning step-size vector and the gradient $\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\pi}_{\boldsymbol{\theta}})$ is computed using the same formulas as in Section 2.4.2, except that the variables are replaced with those defined in this microgrid energy management problem. Specifically, $\boldsymbol{\zeta}$ in (2.17) has the form $\boldsymbol{\zeta} = \{\hat{\mathbf{soc}}, \hat{\boldsymbol{\Delta}}, \hat{\mathbf{b}}, \hat{\mathbf{s}}\}$, which is the primal decision variable of the MPC (4.17). And in (2.18), $\Omega_{\boldsymbol{\theta}}$ now represents the MPC cost (4.17a), $\mathbf{G}_{\boldsymbol{\theta}}$ gathers the equality constraints and $\mathbf{H}_{\boldsymbol{\theta}}$ collects the inequality constraints of the MPC (4.17). In addition, the first element of the input $\mathbf{u}_0$ in (2.19) is represented in this case as

$$\mathbf{u}_0 = \left[\hat{\mathbf{b}}^\top_0, \hat{\mathbf{s}}^\top_0\right]^\top. \tag{4.24}$$

The state feature vector $\mathbf{\Phi}(\mathbf{s})$ in (2.21) is designed as

$$\mathbf{\Phi}(\mathbf{s}) = \Big[(\mathrm{soc}^1 - 0.5)^2, \ldots, (\mathrm{soc}^{N_a} - 0.5)^2, (\mathrm{soc}^1 - 0.5), \ldots, (\mathrm{soc}^{N_a} - 0.5), \Delta^1, \ldots, \Delta^{N_a}, 1\Big]. \quad (4.25)$$

Finally, equation (4.23) can be rewritten as

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \boldsymbol{\eta} \odot \mathbb{E}_m \left\{ \sum_{k=1}^{K} \Big[\nabla_{\boldsymbol{\theta}} \boldsymbol{\pi}_{\boldsymbol{\theta}} (\mathbf{s}_k) \, \nabla_{\boldsymbol{\theta}} \boldsymbol{\pi}_{\boldsymbol{\theta}} (\mathbf{s}_k)^\top \mathbf{w} \Big] \right\}, \quad (4.26)$$

where the summation is taken over the whole episode, and the values are then averaged by taking expectation ($\mathbb{E}_m$) over m episodes.

4.3.3 Profit Distribution

By applying the proposed MPC-based RL approach, we are able to learn an energy trading policy that reduces the monthly collective cost $J(\boldsymbol{\pi})$. The next critical issue is to provide a feasible solution for fairly distributing this collective cost. Note, however, that the EM problem requires the distribution of collective cost rather than collective profit. Thus, in our case, *profit* is defined as the cost savings gained through cooperation, which relates cost and profit, and helps transform the cost distribution problem into a profit distribution problem. In this CCG, profit distribution is of great importance as a reasonable distribution can facilitate the prosumers to strengthen the focus on collective benefits and reduce the natural tendency to protect themselves at the expense of the community. That is, a decent profit distribution scheme should ensure that every prosumer receives a fair and more substantial benefit, which discourages them from leaving the coalition. There are several profit distribution methods for CCG, such as nucleolus sets, bargaining sets, stable sets, and Shapley value [38]. Among them, the Shapley value method, which represents the marginal contribution of each stakeholder to the alliance, is a popular method due to its simple logic (based on fairly intuitive reasoning) and the uniqueness of the solution [38]. The

"fairness" and "uniqueness" of the Shapley value method have been mathematically proven, see [39].

Recall that we have a set $\mathcal{M} = \{1, 2, \ldots, N_a\}$ of N_a households. We define a *coalition* $\mathcal{I} \subseteq \mathcal{M}$ as a non-empty subset of $\mathcal{M}$. There are $2^{N_a} - 1$ possible coalitions (excluding the empty set). The *coalitional structure* is formed by all possible combinations of coalitions that do not overlap and their union contains all players. We further provide an example of the coalition and coalitional structure in Table 4.1 for a tri-player CCG [40]. The *utility values*, including coalitional utility

Table 4.1: Coalition $\mathcal{I}$ and coalitional structures of a tri-player CCG.

Coalition $\mathcal{I}$	Coalitional structures
$\{1\}$	$\{\{1\},\{2\},\{3\}\}$
$\{2\}$	$\{\{1,2\},\{3\}\}$
$\{3\}$	$\{\{1,3\},\{2\}\}$
$\{1,2\}$	$\{\{2,3\},\{1\}\}$
$\{1,3\}$	$\{\{1,2,3\}\}$
$\{2,3\}$	
$\{1,2,3\}$	

value, individual utility value and independent utility value, are defined from different perspectives of a game. We denote $\mathcal{V}(\mathcal{I})$ as the *coalitional utility value* that maps between coalition $\mathcal{I}$ to its collective profit. For each player, $\mathcal{X}_i$ is denoted as the *individual utility value* of prosumer $i = 1, 2, \ldots, N_a$ in coalition $\mathcal{I}$, where the solution of $\mathcal{X}_i$ is the Shapley value to be calculated. And we denote $\mathcal{U}_i$ as the *independent utility value* of player i without joining any coalition. In a CCG game, the profit distribution should meet the following three conditions, otherwise, player i will refuse to join the alliance [41]:

$$\text{Individual rationality:} \quad \mathcal{X}_i \geq \mathcal{U}_i. \tag{4.27a}$$

$$\text{Collective rationality:} \quad \mathcal{V}(\mathcal{I}) = \sum\nolimits_{i \in \mathcal{I}} \mathcal{X}_i. \tag{4.27b}$$

$$\text{Superadditivity property:} \tag{4.27c}$$
$$\mathcal{V}(\mathcal{D} \cup \mathcal{B}) \geq \mathcal{V}(\mathcal{D}) + \mathcal{V}(\mathcal{B}), \quad \forall\, \mathcal{D}, \mathcal{B} \subseteq \mathcal{M},\, \mathcal{D} \cap \mathcal{B} = \varnothing.$$

From (4.27a) and (4.27b), it can be inferred that $\mathcal{V}(\mathcal{I}) \geq \sum_{i\in\mathcal{I}} \mathcal{U}_i$, which implies that in a CCG, the collective profit of a coalition is greater than the sum of profits made by all players independently. This is an incentive for players not to leave the alliance. Furthermore, (4.27c) corroborates that, the grand coalition $\mathcal{M}$, the case considered in this chapter, is the optimal coalitional structure that yields the highest profit.

The *Shapley value* Π_i, as a unique solution of $\mathcal{X}_i$, is the expected profit allocated to player i. The expression is

$$\Pi_i = \sum_{\mathcal{I}\subseteq\mathcal{M}\setminus i} \left(\frac{|\mathcal{I}|!\,(|\mathcal{M}| - |\mathcal{I}| - 1)!}{|\mathcal{M}|!} \right) \Big[\mathcal{V}\left(\mathcal{I}\cup\{i\}\right) - \mathcal{V}\left(\mathcal{I}\right) \Big], \quad (4.28)$$

where $\frac{|\mathcal{I}|!(|\mathcal{M}|-|\mathcal{I}|-1)!}{|\mathcal{M}|!}$ is the probability of prosumer i joining the coalition $\mathcal{I}$, and $\mathcal{V}\left(\mathcal{I}\cup\{i\}\right) - \mathcal{V}\left(\mathcal{I}\right)$ indicates the marginal contribution (marginal profit) that prosumer i brings to the alliance $\mathcal{I}$.

For this EM problem, the coalitional utility value $\mathcal{V}(\mathcal{I})$, i.e., the collective profit of coalition $\mathcal{I}$, is defined as the cost savings obtained by the members of $\mathcal{I}$ through cooperation. Specifically, it is calculated by subtracting the collective cost $J_\mathcal{I}$ from the sum of the individual costs J_i of all members, i.e.,

$$\mathcal{V}(\mathcal{I}) = \left(\sum\nolimits_{i\in\mathcal{I}} J_i \right) - J_\mathcal{I}, \quad (4.29)$$

where the collective cost $J_\mathcal{I}$ is calculated by solving the following optimization problem

$$\min_{\boldsymbol{\pi}_\mathcal{I}} J_\mathcal{I}(\boldsymbol{\pi}_\mathcal{I}) = \mathbb{E}_{\boldsymbol{\pi}} \left[\sum_{k=0}^{K} L_\mathcal{I}(\bar{\mathbf{s}}_k,\, \bar{\mathbf{a}}_k) \,\middle|\, \bar{\mathbf{a}}_k = \boldsymbol{\pi}_\mathcal{I}(\bar{\mathbf{s}}_k) \right] \quad (4.30a)$$

$$\text{s.t.} \quad \forall i \in \mathcal{I}, \quad \forall k = 0, \ldots, K$$

$$(4.1),\ (4.2),\ (4.3), \quad (4.30b)$$

$$P^{\text{peak}}_{\mathcal{I},\,k+1} = \max\left(P^{\text{peak}}_{\mathcal{I},\,k},\, \sum_{i\in\mathcal{I}} b^i_k,\, \sum_{i\in\mathcal{I}} s^i_k \right), \quad (4.30c)$$

$$P^{\text{peak}}_{\mathcal{I},\,0} = 0, \quad (4.30d)$$

with,

$$\bar{\mathbf{s}}_k = \{\mathrm{soc}_k^i,\ \Delta_k^i \quad \forall i \in \mathcal{I}\}, \tag{4.31a}$$
$$\bar{\mathbf{a}}_k = \{b_k^i,\ s_k^i \quad \forall i \in \mathcal{I}\}, \tag{4.31b}$$
$$L_{\mathcal{I}}(\bar{\mathbf{s}}_k,\ \bar{\mathbf{a}}_k) = L_{\mathcal{I},P}(\bar{\mathbf{s}}_k,\ \bar{\mathbf{a}}_k) + L_{\mathcal{I},\mathrm{S}}(\bar{\mathbf{s}}_k,\ \bar{\mathbf{a}}_k), \tag{4.31c}$$
$$L_{\mathcal{I},P}(\bar{\mathbf{s}}_k,\ \bar{\mathbf{a}}_k) = \lambda(P_{\mathcal{I},k+1}^{\mathrm{peak}} - P_{\mathcal{I},k}^{\mathrm{peak}}), \tag{4.31d}$$
$$L_{\mathcal{I},\mathrm{S}}(\bar{\mathbf{s}}_k,\ \bar{\mathbf{a}}_k) = \sum\nolimits_{i\in\mathcal{I}} L_{\mathrm{S}}^i(b_k^i,\ s_k^i). \tag{4.31e}$$

Especially, we have $J_{\mathcal{I}} = J_i$ when $\mathcal{I}$ contains only one prosumer i, i.e., $\mathcal{I} = \{i\}$; and from (4.16) and (4.30), we have $J_{\mathcal{I}} = J_{\mathcal{M}} = J$ when $\mathcal{I}$ contains all prosumers, i.e., $\mathcal{I} = \mathcal{M}$. Therefore, (4.30) is actually a general version of (4.16), and solving (4.30) is the same as solving (4.16), except that it involves some specific prosumers rather than all of them.

The profit $\mathcal{V}(\mathcal{M})$ gained through the grand coalition should be fairly distributed to all prosumers according to (4.28), i.e., it obeys

$$\mathcal{V}(\mathcal{M}) = \sum\nolimits_{i\in\mathcal{M}} \Pi_i. \tag{4.32}$$

Consequently, each prosumer $i \in \mathcal{M}$ is supposed to pay $\mathcal{B}_i$ for his/her monthly electricity bill, computed as

$$\mathcal{B}_i = J_i - \Pi_i. \tag{4.33}$$

This satisfies that the sum of individual payments equals the collective cost (collective bill), i.e., $\sum_{i\in\mathcal{M}} \mathcal{B}_i = J$, because

$$\begin{aligned}\sum\nolimits_{i\in\mathcal{M}} \mathcal{B}_i &= \sum\nolimits_{i\in\mathcal{M}} J_i - \sum\nolimits_{i\in\mathcal{M}} \Pi_i \\ &= \sum\nolimits_{i\in\mathcal{M}} J_i - \mathcal{V}(\mathcal{M}) \\ &= J_{\mathcal{M}} = J.\end{aligned} \tag{4.34}$$

It is worth highlighting that although the Shapley method ensures a fair distribution, it is computationally intractable due to the combinatorial nature of its computation. For this N_a-player game, we need to solve

$2^{N_a}-1$ optimization problems, which is unrealistic. To address this issue, numerous approaches have been proposed to estimate the Shapley value instead of calculating it explicitly. Other than sampling-based approximations [42, 43], other techniques like multi-linear extension methods [44, 45], permutation methods [46, 47], and linear regression-based methods [48, 49] provide applicable solutions considering different metrics, such as computational complexity and estimation error. Therefore, although computing the Shapley value is an NP-complete problem, estimating the Shapley value can be efficiently calculated in polynomial time [42]. However, this is beyond the scope of this chapter and will not be discussed intensively. For this chapter, the contribution is to properly define the individual and coalitional profits and utility values for this EM problem, and to give a logic about how the collective profit should be distributed. In practice, by using the estimate of the Shapley value (4.28) (e.g., through Monte Carlo sampling), and formulas (4.29), (4.30) and (4.33), it is feasible to allocate the collective profit to all prosumers.

4.3.4 Policy Learning and Profit Distribution Algorithm

We present the proposed algorithm in this section. The single-step computation procedure of MPC (4.17) is given in Algorithm 2, which is invoked in Algorithm 3. Algorithm 3 presents the complete algorithm for residential microgrid energy management, including policy learning and profit distribution. For further illustration, we add a flowchart of the EM framework corresponding to Algorithm 3. As shown in Fig. 4.2, the energy trading policy is generated by the parameterized MPC. At the beginning of each month, based on the previous optimal MPC parameters, the policy is supposed to be retrained using historical data and the latest data from the previous month (including the predicted and actual values of household consumption and production, as well as the spot-market prices). This process uses the proposed MPC-based RL approach and can be trained offline. Once the MPC parameters converge, the trained policy can be used for the online application. Residents in the cooperative coalition will trade energy according to the well-trained policy

that reduces the collective cost of the coalition. To fairly allocate the collective cost, the Shapley value technique is employed to calculate the contribution of each user in the coalition and, from there, the electricity bill payable by each user. Besides, the real-time data collected from the current month are uploaded to the cloud buffer for the next update of the policy.

Algorithm 2: Single-step MPC computation.

Input: State $\mathbf{s}_k$, electricity prices ϕ_k^b and ϕ_k^s, parameters $\boldsymbol{\theta}$

1 Solve the MPC scheme (4.17) and get the solution $\mathbf{y}^\star$;
2 Store the current state and action in the replay memory $\mathbb{D}$;
3 Calculate the RL stage cost $L(\mathbf{s}_k,\ \mathbf{a}_k)$ by (4.14) and add it to $\mathbb{D}$;
4 Obtain the policy $\pi_{\boldsymbol{\theta}}(\mathbf{s}_k)$ by (4.21) and add it to $\mathbb{D}$;
5 Calculate the sensitivity $\nabla_{\boldsymbol{\theta}}\pi_{\boldsymbol{\theta}}(\mathbf{s}_k)$ by (2.19) and add it to $\mathbb{D}$;
6 Observe the next state $\mathbf{s}_{k+1}$ from the real system (4.1).

4.4 Simulation and Discussion

This section provides simulation results of the proposed algorithm in a three-agent residential microgrid system, including an analysis of policy learning and profit distribution.

4.4.1 Case Configuration

We chose three houses located in Oslo, Norway, with increasing house size and peak power of the PV system, as shown in Table 4.2. A real-world household consumption-production dataset $(\beta_\Delta)^{1,2,3}$ is adopted in the simulation. Considering the balance between power consumption and production, the dataset of April 2020 is used because heating is still needed and the sunlight is relatively abundant during this period. The spot-market prices are collected from the

Algorithm 3: Policy learning and profit distribution for residential microgrid energy management.

repeat
  (1) Policy learning;
  Retrieve the MPC parameters $\boldsymbol{\theta}$ from the cloud;
  while *not converge to* $\boldsymbol{\theta}^{\star}$ **do**
    Initialize state $\mathbf{s}_{0,1}$;
    for *episode* $l = 1$ **to** m **do**
      for $k = 0$ **to** K **do**
        Confirm $\mathbf{s}_{k,l}$, $\phi^{b}_{k,l}$, $\phi^{s}_{k,l}$;
        Single-step MPC computation (Algorithm 2);
      end
    end
    Retrieve data from the replay memory $\mathbb{D}$;
    Calculate $\mathbf{v}$ by (2.24a) and calculate $\mathbf{w}$ by (2.24b);
    Update the MPC parameters $\boldsymbol{\theta}$ by (4.26).
  end
  ——————————————————————————————;
  Online application: run the trained MPC to obtain the minimized monthly collective cost $J(\boldsymbol{\pi}_{\theta^{\star}})$, and upload $\boldsymbol{\theta}^{\star}$ and the real-time data collected online to the cloud;
  (2) Profit distribution;
  for *each* $\mathcal{I} \in \mathcal{M}$ **do**
    Calculate $J_{\mathcal{I}}$ by solving (4.30);
  end
  Calculate the coalitional utility value $\mathcal{V}(\mathcal{I})$ by (4.29);
  Calculate the Shapley value Π_i for all prosumers by (4.28);
  Calculate the individual payment $\mathcal{B}_i$ for each prosumer by (4.33), i.e., distribute the collective cost $J(\boldsymbol{\pi}_{\theta^{\star}})$.
until $t \to \infty$;

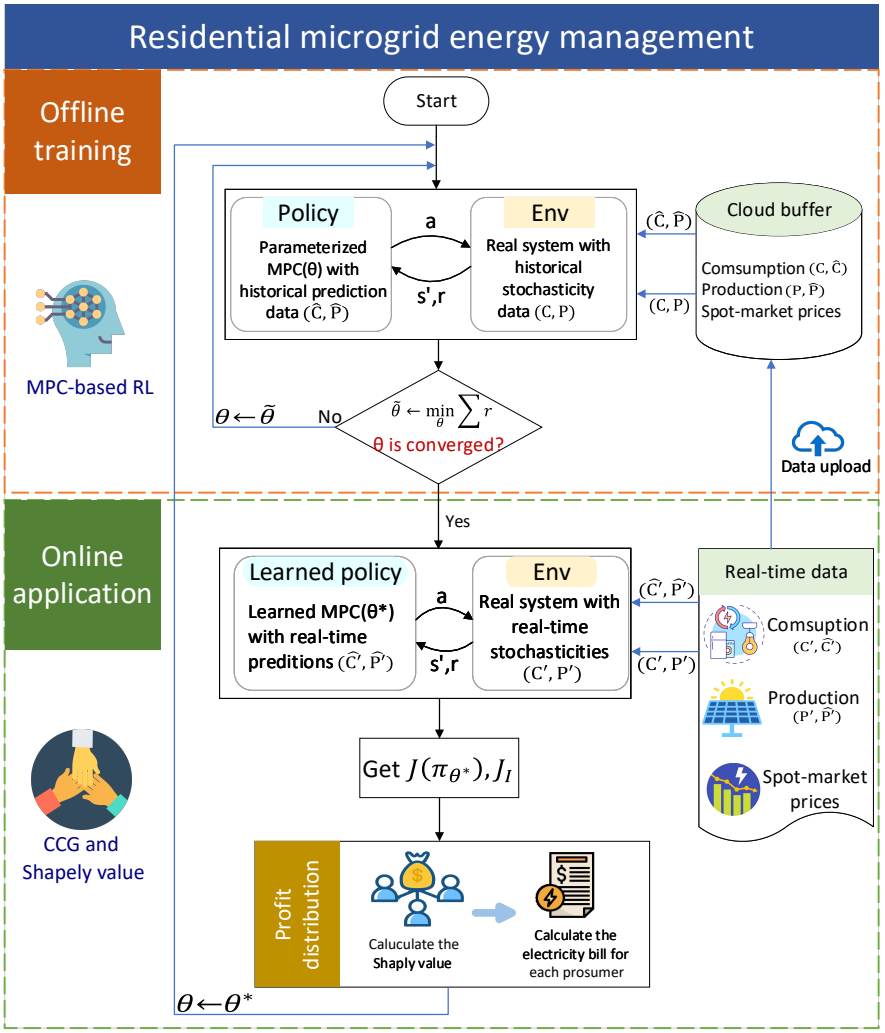

Figure 4.2: Illustration of the residential microgrid energy management framework.

Nord Pool European Power Exchange [50]. The selling price is the spot-market value, while the buying price is $\phi^b = 1.2\phi^s$, accounting for a 20% VAT. The initial values of the parameters $\boldsymbol{\theta}$ are set as $\boldsymbol{\theta}_0 = \{[1/50,\ 1/55,\ 1/60]^\top,\ \mathbf{1},\ \mathbf{0},\ \mathbf{0.1},\ \mathbf{0.1},\ \mathbf{0.5},\ 0.32\}$, where the bold numbers represent constant vectors of dimension three for the three prosumers. The two main hyperparameters involved in this work are the learning step size ($\boldsymbol{\eta} \in \mathbb{R}^{19}$) and the exploration rate ($\boldsymbol{\varrho} \in \mathbb{R}^6$). They are initialized according to the magnitude of the initial sensitivities and the range of actions, respectively. Then, the two hyperparameters are coarsely tuned according to the performance variations, and are continuously fine-tuned in turn until the best performance is obtained. Finally, the exploration rate $\boldsymbol{\varrho}$ is chosen as $\boldsymbol{\varrho} \sim \mathcal{N}(\mathbf{0},\ \mathbf{0.1^2})$, which is a vector of dimension six, as we have in total six actions in the three-agent system. The step size vector $\boldsymbol{\eta}$ corresponding to $\{\theta_\alpha^{1,2,3},\ \theta_\beta^{1,2,3},\ \theta_\delta^{1,2,3},\ \theta_\psi^{1,2,3},\ \theta_T^{1,2,3},\ \theta_\varphi^{1,2,3},\ \theta_\lambda\}$ is chosen as $\boldsymbol{\eta} = \{\mathbf{5 \times 10^{-9}},\ \mathbf{5 \times 10^{-7}},\ \mathbf{5 \times 10^{-6}},\ \mathbf{5 \times 10^{-7}},\ \mathbf{5 \times 10^{-4}},\ \mathbf{5 \times 10^{-8}},\ 7.5 \times 10^{-7}\}$, with the bold numbers represent constant vectors of dimension three. Other parameter values used in the simulation are listed in Table 4.3. In addition, considering the presence of stochastic prediction errors in power consumption-production and stochastic explorations in learning, we repeat the learning five times independently on the real dataset to provide more solid simulation results.

Table 4.2: Specifications of the three houses.

	Size	PV info
House 1	127 m^2	20 panels, 5.94 kWp
House 2	160 m^2	25 panels, 8 kWp
House 3	192 m^2	28 panels, 9.72 kWp

4.4.2 Analysis of the Policy Learning

The policy learning applies the proposed MPC-based RL approach to seek an energy trading policy that reduces the monthly collective cost.

Table 4.3: Parameter values.

symbol	value	symbol	value
Sampling time	1h	λ	15
$\alpha^{1,2,3}$	1/50, 1/55, 1/60	N_a, N	3, 12
$\sigma^{1,2,3}$	0.24, 0.28, 0.32	$\bar{U}^i$	20
K	720	$\boldsymbol{\varrho}$	$\mathcal{N}(\mathbf{0}, \mathbf{0.1^2})$
$\boldsymbol{\theta}_0$	$\{[1/50, 1/55, 1/60]^\top, \mathbf{1}, \mathbf{0}, \mathbf{0.1}, \mathbf{0.1}, \mathbf{0.5}, 0.32\}$		
$\boldsymbol{\eta}$	$\{\mathbf{5 \times 10^{-9}}, \mathbf{5 \times 10^{-7}}, \mathbf{5 \times 10^{-6}}, \mathbf{5 \times 10^{-7}}, \mathbf{5 \times 10^{-4}}, \mathbf{5 \times 10^{-8}}, 7.5 \times 10^{-7}\}$		

The results are presented in Fig. 4.3 - Fig. 4.8. Figure 4.3 shows the parameter variations of the three prosumers, where the shaded areas represent the 95% confidence intervals of the five independent experiments. The initial values of the parameters $\theta_\alpha^{1,2,3}$ are consistent with those of the actual system. Parameters $\theta_\beta^{1,2,3}$ and $\theta_\delta^{1,2,3}$ are initialized as $\mathbf{1}$ and $\mathbf{0}$, respectively, to replicate the values of $(\beta_\Delta)^{1,2,3}$. The setpoint parameters $\theta_\varphi^{1,2,3}$ are initialized as 0.5, based on the sense that picking the reference SOC at around half promotes the SOC to be in the middle of the feasible domain, which provides more freedom for the MPC scheme to decide to either store or release energy. However, the actual preferred values should be determined by RL. It can be seen that as learning proceeds, all the parameters approach the convergence values, denoted as $\boldsymbol{\theta}^\star$.

Figure 4.4 presents the variation of the parameter θ_λ, which is the most crucial one among the total 19 parameters. It can be seen that the value of θ_λ increases from 0.32 to about 0.52, which means that the concern for peak-power cost increases gradually in the MPC cost (4.17a). Therefore, the peak power should be decreased to reduce the peak-power cost. As expected, we show in the right plot that the peak power continues to decrease as the learning proceeds. Besides, it is worth mentioning that the convergence value of θ_λ is 0.52 instead of 15 (the true penalty coefficient λ in (4.10)) because, as we mentioned before, the actual RL peak-power cost is computed over a one-month period, while MPC

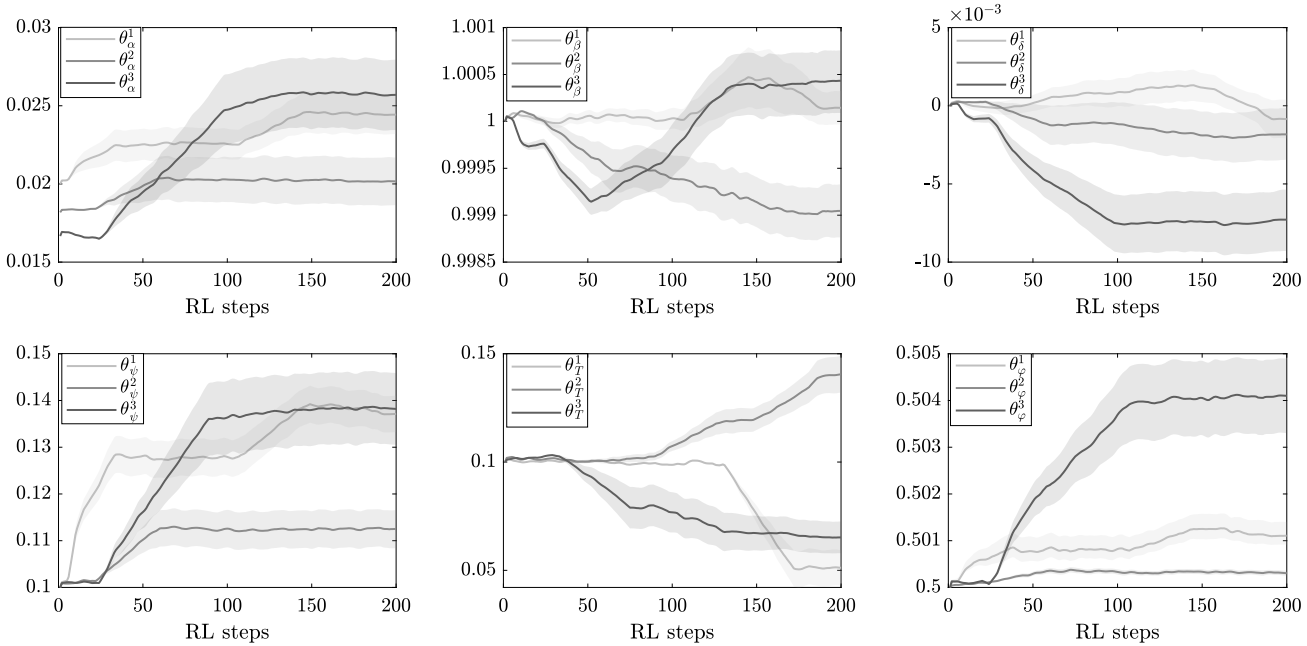

Figure 4.3: The parameters $\{\theta^i_\alpha,\, \theta^i_\beta,\, \theta^i_\delta,\, \theta^i_\psi,\, \theta^i_T,\, \theta^i_\varphi\}$ of prosumers $i = 1,\, 2,\, 3$ over learning steps. The shaded areas represent the 95% confidence intervals of the five independent experiments.

considers a much shorter interval of 12h, so it is reasonable that these two values are not identical.

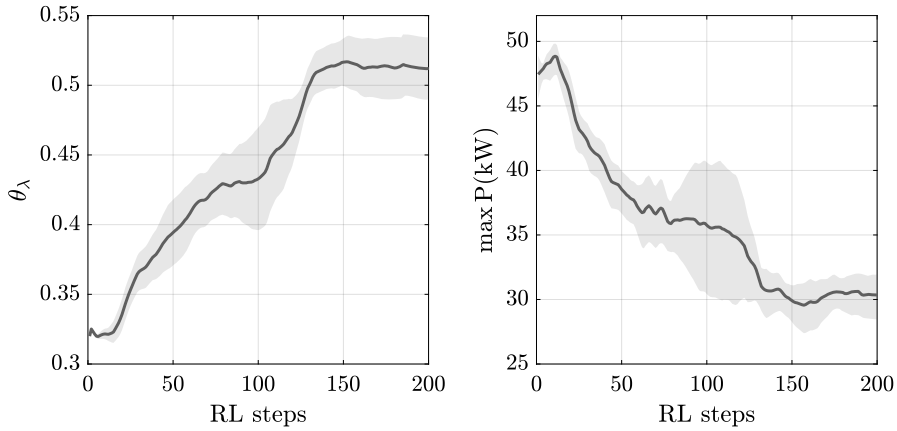

Figure 4.4: The parameter θ_λ and the peak power over learning steps. The shaded areas represent the 95% confidence intervals of the five independent experiments.

Figure 4.5 and figure 4.6 show the amount of electricity bought (b_k^i) and sold (s_k^i) by the three prosumers during the episodes of five sampled RL steps: 1^{st}, 50^{th}, 100^{th}, 150^{th}, 200^{th}. The gray curves represent the electricity buying prices ϕ_k^b and selling prices ϕ_k^s, respectively. It can be observed that purchases occur mainly when power prices are low, while when prices are high, prosumers tend to use the energy stored in the battery and sell the excess if there is any surplus. This energy trading strategy ensures that the costs incurred due to the spot market remain low. Moreover, during the learning process, the volume of buying/selling transactions of each resident is significantly reduced due to the increase in θ_λ. In other words, during price troughs or peaks, the buying/selling strategy gradually tends to become more conservative compared to the initial one, hence avoiding generating large peak power and therefore reducing the peak-power cost in (4.15).

The battery SOC of each prosumer at different learning steps is presented in Fig. 4.7. It can be seen that the variances of $\text{soc}^{1,2,3}$ decrease grad-

Figure 4.5: The buying amounts of the three prosumers during episodes of the sampled RL steps: 1^{st}, 50^{th}, 100^{th}, 150^{th}, 200^{th}.

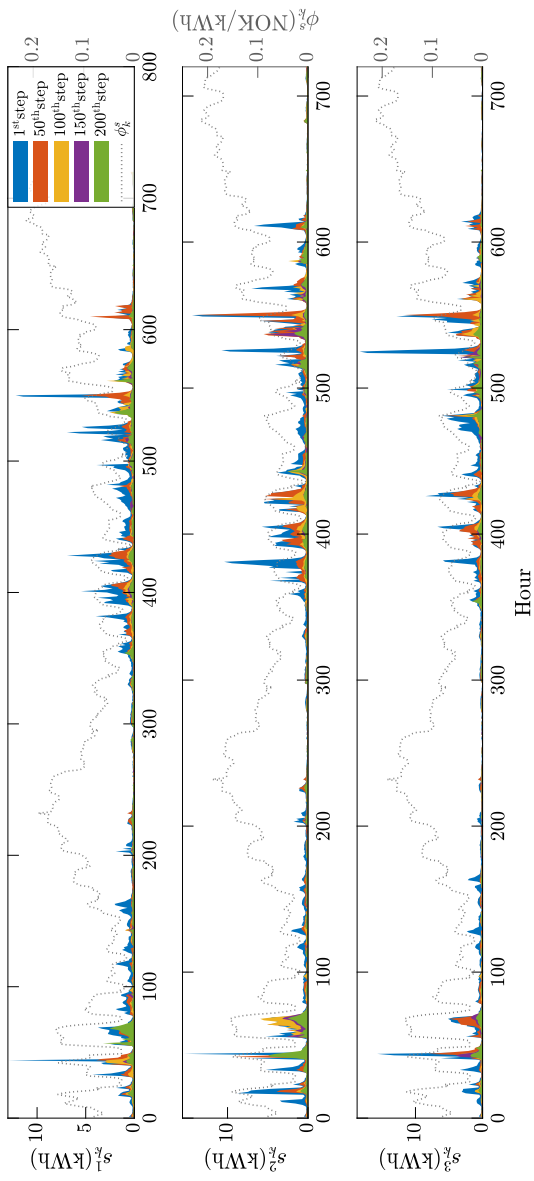

Figure 4.6: The selling amounts of the three prosumers during episodes of the sampled RL steps: 1^{st}, 50^{th}, 100^{th}, 150^{th}, 200^{th}.

ually with learning, which further corroborates the conclusions drawn from Fig. 4.5 and Fig. 4.6. Finally, the variations of the closed-loop performance $J(\pi_{\theta})$ and the normalized policy gradient $\|\nabla_{\theta} J(\pi_{\theta})\|_2$ are given in Fig. 4.8. The policy gradient $\nabla_{\theta} J(\pi_{\theta})$ is decreasing to zero. Correspondingly, the closed-loop performance (4.15) decreases gradually and eventually converges, i.e. we have $J(\pi_{\theta^\star}) = 844\text{NOK}$[3]. It can be calculated that by using the MPC-based RL method, the monthly collective cost J is reduced by about 17.5%, which is a considerable improvement.

It is worth noting that although we mentioned that the MPC scheme can yield the optimal policy if the parametrization is rich enough, this is a theoretical result. In practice, the assumption of a "rich enough" parametrization is usually not satisfied. And it is also possible that the very simple choices of the value function and terminal cost entail that some opportunities in the MPC tuning are "missed". All these reasons lead to the MPC policy remaining suboptimal. Besides, other practical issues can come in the way of optimality, such as the local convergence of the RL algorithm, and of the solver treating the MPC scheme. Addressing these potential issues typically requires good initial guesses, and this observation applies to most RL-based techniques. However, it is a token that the MPC can, in principle, converge to optimality as its parametrization becomes finer and finer. For this specific EM problem, the final learned policy $\pi_{\theta^\star}$ obtained from the converged parameters $\theta^\star$ may be locally optimal. However, the theoretical optimality of the MPC-based RL method cannot be ignored, and the proposed method does improve the closed-loop performance significantly.

Lastly, to understand which MPC parameters are more critical for the learning performance, we conduct an ablation study. Specifically, each kind of parameter of the MPC is ablated separately, i.e., the ablated parameters cannot be learned. The performance $J(\pi_{\theta^\star_{\text{abl}}})$ obtained after training is observed and the loss is calculated by comparing $J(\pi_{\theta^\star_{\text{abl}}})$ with the performance obtained in the fully parameterized case $J(\pi_{\theta^\star})$. Then, the Average Degradation Level (ADL) of the parameters is defined

[3]Norwegian kroner (NOK) is used as the unit of measure.

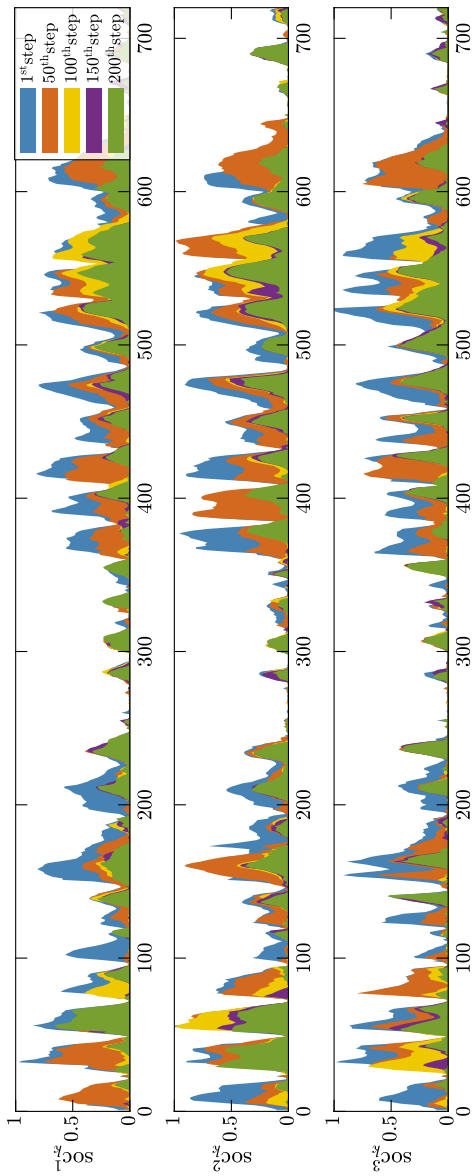

Figure 4.7: The soc_k^t of the three prosumers during episodes of the sampled RL steps: 1^{st}, 50^{th}, 100^{th}, 150^{th}, 200^{th}.

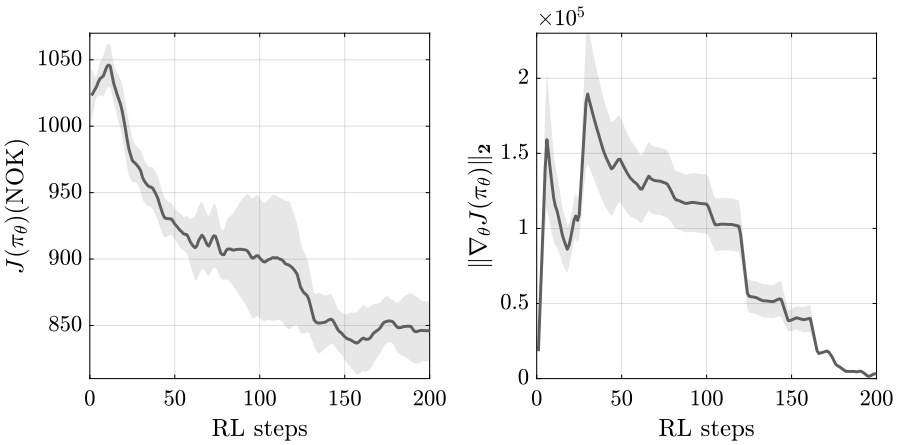

Figure 4.8: The closed-loop performance $J(\boldsymbol{\pi_\theta})$ and the normalized policy gradient $\|\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\pi_\theta})\|_2$ over learning steps. The shaded areas represent the 95% confidence intervals of the five independent experiments.

as

$$\text{ADL} = \mathbb{E}\left[\frac{J(\boldsymbol{\pi}_{\boldsymbol{\theta}^\star_{\text{abl}}}) - J(\boldsymbol{\pi}_{\boldsymbol{\theta}^\star})}{J(\boldsymbol{\pi}_{\boldsymbol{\theta}_0}) - J(\boldsymbol{\pi}_{\boldsymbol{\theta}^\star})}\right], \tag{4.35}$$

which is the average of the loss due to ablation, normalized by the value obtained in the fully parameterized case. Therefore, a higher ADL means that ablating this parameter can be more detrimental to the performance, and hence implies a greater importance of this parameter. We ablate similar parameters of the three prosumers as a whole, e.g., ablate $\theta_\alpha^{1,2,3}$ simultaneously, and compute the ADL of these parameters. The complete ADL results for all parameters are shown in Table 4.4. An observation is that θ_λ has a dominant effect on the performance, followed by $\theta_\psi^{1,2,3}$, $\theta_\varphi^{1,2,3}$, and $\theta_\alpha^{1,2,3}$, while others have the least effect.

4.4.3 Analysis of the Profit Distribution

The profit distribution is dedicated to allocating the optimized collective cost $J(\boldsymbol{\pi}_{\boldsymbol{\theta}^\star}) = 844\text{NOK}$ to each household in a rational way. The

Table 4.4: Average degradation level of the MPC parameters.

MPC parameters	$\theta_{\alpha}^{1,2,3}$	$\theta_{\beta}^{1,2,3}$	$\theta_{\delta}^{1,2,3}$	$\theta_{\psi}^{1,2,3}$
Average degradation level	13.13%	4.07%	0.15%	15.66%
MPC parameters	$\theta_{T}^{1,2,3}$	$\theta_{\varphi}^{1,2,3}$	θ_{λ}	
Average degradation level	1.42%	13.38%	60.09%	

results are shown in Fig. 4.9 - Fig. 4.10 and Table 4.5. In Fig. 4.9, we illustrate the costs $J_{\mathcal{I}}$ and cost savings $\mathcal{V}_{\mathcal{I}}$ (i.e. coalitional utility values) for all seven coalitions of the tri-player CCG. As can be seen, the cost savings are all zero for singleton coalitions $\{1\}$, $\{2\}$, $\{3\}$. And the cost savings increase with the number of participants in the coalition, which is consistent with the characteristics of CCG. In addition, since the consumption-production levels of the three players in the simulation are incremental, their contributions to the coalition are incremental as well. That is, prosumers with higher consumption-production levels will be more dominant in the alliance. Under the grand coalition $\{1,\ 2,\ 3\}$, the collective cost is 844NOK and the cost savings obtained through cooperation is 310NOK. Therefore, the essence of the Shapley value approach in this problem is to allocate the 310NOK cost savings.

The Shapley values Π_i and individual payments $\mathcal{B}_i$ for each prosumer under the grand coalition are summarized in Table 4.5. The corresponding pie chart is depicted in Fig. 4.10 for a better understanding of the relationship between those values. Briefly, the sum of the Shapley values Π_i equals the cost savings (310NOK) under the grand coalition; the sum of the individual payments $\mathcal{B}_i$ equals the collective cost (844NOK); and the cost of a prosumer in a singleton coalition would be the sum of its payment and Shapley value in the grand coalition, i.e., $J_i = \mathcal{B}_i + \pi_i$. Furthermore, the magnitude of each prosumer's marginal contribution to the coalition can be seen in the ratio term, where the third prosumer is the most influential member in the alliance with a percentage of 42.54%. Ultimately, the resulting individual payment $\mathcal{B}_i$ is the electricity bill that each prosumer should pay in this EM problem.

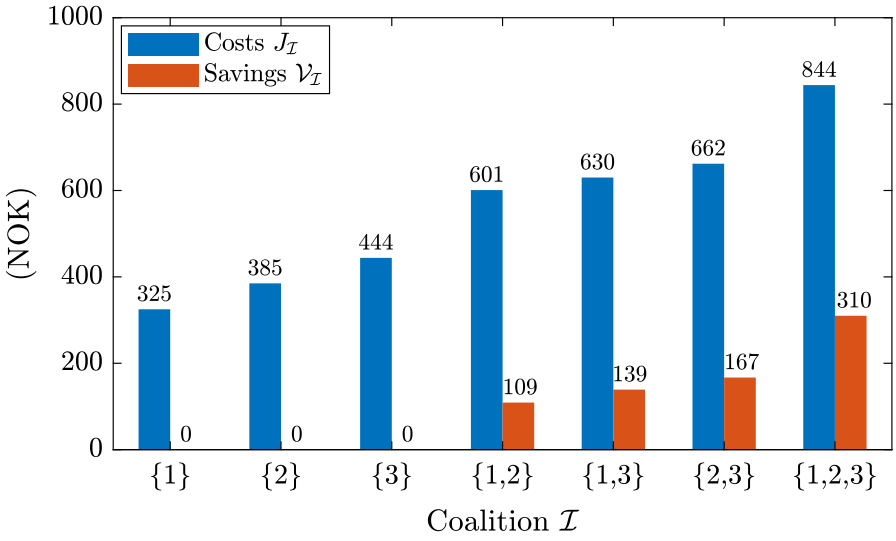

Figure 4.9: The costs and cost savings of all coalitions of the tri-player CCG.

Table 4.5: Shapley value and profit distribution of the grand coalition.

Prosumer i	1	2	3	Total
Cost J_i(NOK)	325	385	444	1154
Shapley value Π_i(NOK)	77.87	100.25	131.88	310
Ratio	25.12%	32.34%	42.54%	100%
Payment $\mathcal{B}_i$(NOK)	247.13	284.75	312.12	844

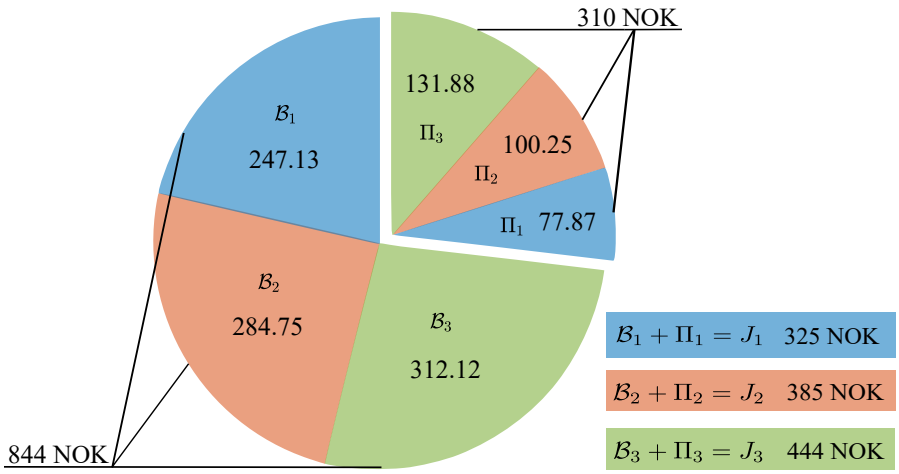

Figure 4.10: Pie chart of the Shapley value and profit distribution of the grand coalition.

4.5 Conclusion

In this chapter, we propose a complete two-step strategy to solve the energy management problem of a residential microgrid system, which is formulated as a CCG problem. The proposed MPC-based RL approach compensates for the drawbacks of MPC and RL and combines the advantages of both, where the parameterized MPC-scheme is served as a function approximator of the optimal policy and the parameters are adjusted by RL to optimize the closed-loop performance. Even with stochastic local power consumption-production, this approach could reduce the monthly collective cost by a significant amount of about 17.5%, as demonstrated in the simulations. Besides, we show that the Shapley value is an applicable solution to fairly distribute the collective bill based on the marginal contribution of each prosumer. Last but not least, some may be concerned about the computation of the Shapley value. However, as we mentioned in Section 4.3.3, many methods have been proposed to estimate the Shapley value instead of computing it explicitly, ensuring to efficiently compute the Shapley value in polynomial time. And for future work, it would be interesting to consider more efficient ways of

estimating Shapley values. For example, the estimation function of the Shapley value for each user could be trained simultaneously during the policy training.

References

[1] Rémy Vincent, Mourad Ait-Ahmed, Azeddine Houari, and Mohamed Fouad Benkhoris. "Residential microgrid energy management considering flexibility services opportunities and forecast uncertainties". In: *International Journal of Electrical Power & Energy Systems* 120 (2020), p. 105981.

[2] Amjad Anvari-Moghaddam, Josep M Guerrero, Juan C Vasquez, Hassan Monsef, and Ashkan Rahimi-Kian. "Efficient energy management for a grid-tied residential microgrid". In: *IET Generation, Transmission & Distribution* 11.11 (2017), pp. 2752–2761.

[3] Wencong Su and Jianhui Wang. "Energy management systems in microgrid operations". In: *The Electricity Journal* 25.8 (2012), pp. 45–60.

[4] Hualei Zou et al. "A survey of energy management in interconnected multi-microgrids". In: *IEEE Access* 7 (2019), pp. 72158–72169.

[5] Alessandra Parisio, Evangelos Rikos, and Luigi Glielmo. "A model predictive control approach to microgrid operation optimization". In: *IEEE Transactions on Control Systems Technology* 22.5 (2014), pp. 1813–1827.

[6] Stefano Raimondi Cominesi, Marcello Farina, Luca Giulioni, Bruno Picasso, and Riccardo Scattolini. "A two-layer stochastic model predictive control scheme for microgrids". In: *IEEE Transactions on Control Systems Technology* 26.1 (2017), pp. 1–13.

[7] Yan Zhang, Lijun Fu, Wanlu Zhu, Xianqiang Bao, and Cang Liu. "Robust model predictive control for optimal energy management of island microgrids with uncertainties". In: *Energy* 164 (2018), pp. 1229–1241.

[8] Ahmed Ouammi, Yasmine Achour, Driss Zejli, and Hanane Dagdougui. "Supervisory model predictive control for optimal energy management of networked smart greenhouses integrated microgrid". In: *IEEE Transactions on Automation Science and Engineering* 17.1 (2019), pp. 117–128.

[9] Ahmed Saad et al. "Data-centric hierarchical distributed model predictive control for smart grid energy management". In: *IEEE Transactions on Industrial Informatics* 15.7 (2018), pp. 4086–4098.

[10] Peng Xie, Youwei Jia, Hongkun Chen, Jun Wu, and Zexiang Cai. "Mixed-stage energy management for decentralized microgrid cluster based on enhanced tube model predictive control". In: *IEEE Transactions on Smart Grid* 12.5 (2021), pp. 3780–3792.

[11] Ying Ji, Jianhui Wang, Jiacan Xu, Xiaoke Fang, and Huaguang Zhang. "Real-time energy management of a microgrid using deep reinforcement learning". In: *Energies* 12.12 (2019), p. 2291.

[12] Glenn Ceusters et al. "Model-predictive control and reinforcement learning in multi-energy system case studies". In: *Applied Energy* 303 (2021), p. 117634.

[13] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[14] Elizaveta Kuznetsova et al. "Reinforcement learning for microgrid energy management". In: *Energy* 59 (2013), pp. 133–146.

[15] Taha Abdelhalim Nakabi and Pekka Toivanen. "Deep reinforcement learning for energy management in a microgrid with flexible demand". In: *Sustainable Energy, Grids and Networks* 25 (2021), p. 100413.

[16] Elham Foruzan, Leen-Kiat Soh, and Sohrab Asgarpoor. "Reinforcement learning approach for optimal distributed energy management in a microgrid". In: *IEEE Transactions on Power Systems* 33.5 (2018), pp. 5749–5758.

[17] P Kofinas, AI Dounis, and GA Vouros. "Fuzzy Q-Learning for multi-agent decentralized energy management in microgrids". In: *Applied Energy* 219 (2018), pp. 53–67.

[18] Fangyuan Li, Jiahu Qin, and Wei Xing Zheng. "Distributed Q-Learning-Based Online Optimization Algorithm for Unit Commitment and Dispatch in Smart Grid". In: *IEEE Transactions on Cybernetics* 50.9 (2019), pp. 4146–4156.

[19] Liyuan Zhao, Ting Yang, Wei Li, and Albert Y Zomaya. "Deep reinforcement learning-based joint load scheduling for household multi-energy system". In: *Applied Energy* 324 (2022), p. 119346.

[20] Mehdi Ahrarinouri, Mohammad Rastegar, Kiana Karami, and Ali Reza Seifi. "Distributed reinforcement learning energy management approach in multiple residential energy hubs". In: *Sustainable Energy, Grids and Networks* 32 (2022), p. 100795.

[21] Linfang Yan, Xia Chen, Yin Chen, and Jinyu Wen. "A Hierarchical Deep Reinforcement Learning-Based Community Energy Trading Scheme for a Neighborhood of Smart Households". In: *IEEE Transactions on Smart Grid* 13.6 (2022), pp. 4747–4758.

[22] Kaile Zhou and Lulu Wen. "Incentive-Based Demand Response with Deep Learning and Reinforcement Learning". In: *Smart Energy Management*. Springer, 2022, pp. 155–182.

[23] Silvio Brandi, Davide Coraci, Davide Borello, and Alfonso Capozzoli. "Energy Management of a Residential Heating System Through Deep Reinforcement Learning". In: *Sustainability in Energy and Buildings 2021*. Springer, 2022, pp. 329–339.

[24] Renzhi Lu et al. "Reward Shaping-Based Actor-Critic Deep Reinforcement Learning for Residential Energy Management". In: *IEEE Transactions on Industrial Informatics* (2022), pp. 1–12.

[25] Lucian Buşoniu, Tim de Bruin, Domagoj Tolić, Jens Kober, and Ivana Palunko. “Reinforcement learning for control: Performance, stability, and deep approximators”. In: *Annual Reviews in Control* 46 (2018), pp. 8–28.

[26] Y Chen, Tien-Ju Yang, Joel Emer, and Vivienne Sze. “Understanding the limitations of existing energy-efficient design approaches for deep neural networks”. In: *Energy* 2.L1 (2018), p. L3.

[27] Sébastien Gros and Mario Zanon. “Data-driven economic nmpc using reinforcement learning”. In: *IEEE Transactions on Automatic Control* 65.2 (2019), pp. 636–648.

[28] Mario Zanon, Sébastien Gros, and Alberto Bemporad. “Practical reinforcement learning of stabilizing economic MPC”. In: *2019 18th European Control Conference (ECC)*. IEEE. 2019, pp. 2258–2263.

[29] Sebastien Gros and Mario Zanon. “Reinforcement Learning for Mixed-Integer Problems Based on MPC”. In: *arXiv preprint arXiv:2004.01430* (2020).

[30] Mario Zanon and Sébastien Gros. “Safe reinforcement learning using robust MPC”. In: *IEEE Transactions on Automatic Control* (2020).

[31] Arash Bahari Kordabad, Wenqi Cai, and Sebastien Gros. “MPC-based reinforcement learning for economic problems with application to battery storage”. In: *2021 European Control Conference (ECC)*. IEEE. 2021, pp. 2573–2578.

[32] Wenqi Cai, Arash B Kordabad, Hossein N Esfahani, Anastasios M Lekkas, and Sébastien Gros. “MPC-based reinforcement learning for a simplified freight mission of autonomous surface vehicles”. In: *2021 60th IEEE Conference on Decision and Control (CDC)*. IEEE. 2021, pp. 2990–2995.

[33] Arash Bahari Kordabad, Wenqi Cai, and Sebastien Gros. “Multi-agent battery storage management using MPC-based reinforcement learning”. In: *2021 IEEE Conference on Control Technology and Applications (CCTA)*. IEEE. 2021, pp. 57–62.

[34] Wenqi Cai, Hossein N Esfahani, Arash B Kordabad, and Sébastien Gros. "Optimal management of the peak power penalty for smart grids using MPC-based reinforcement learning". In: *2021 60th IEEE Conference on Decision and Control (CDC)*. IEEE. 2021, pp. 6365–6370.

[35] Nidhi Hegde, Laurent Massoulié, Theodoros Salonidis, et al. "Optimal control of residential energy storage under price fluctuations". In: *Energy* (2011).

[36] Yang Chen et al. "A comparison study on trading behavior and profit distribution in local energy transaction games". In: *Applied Energy* 280 (2020), p. 115941.

[37] Yuanxiong Guo, Miao Pan, and Yuguang Fang. "Optimal power management of residential customers in the smart grid". In: *IEEE Transactions on Parallel and Distributed Systems* 23.9 (2012), pp. 1593–1606.

[38] Fang Fang, Songyuan Yu, and Mingxi Liu. "An improved Shapley value-based profit allocation method for CHP-VPP". In: *Energy* 213 (2020), p. 118805.

[39] Alvin E Roth. *The Shapley value: essays in honor of Lloyd S. Shapley*. Cambridge University Press, 1988.

[40] Luceny Guzmán Acuña, Diana Ram´ırez R´ıos, Carlos Paternina Arboleda, and Esneyder González Ponzón. "Cooperation model in the electricity energy market using bi-level optimization and Shapley value". In: *Operations Research Perspectives* 5 (2018), pp. 161–168.

[41] Yue Teng, Xiao Li, Peng Wu, and Xiangyu Wang. "Using cooperative game theory to determine profit distribution in IPD projects". In: *International Journal of Construction Management* 19.1 (2019), pp. 32–45.

[42] Javier Castro, Daniel Gómez, and Juan Tejada. "Polynomial calculation of the Shapley value based on sampling". In: *Computers & Operations Research* 36.5 (2009), pp. 1726–1730.

[43] Alf Kimms and Igor Kozeletskyi. “Shapley value-based cost allocation in the cooperative traveling salesman problem under rolling horizon planning”. In: *EURO Journal on Transportation and Logistics* 5.4 (2016), pp. 371–392.

[44] Guillermo Owen. “Multilinear extensions of games”. In: *The Shapley Value. Essays in Honor of Lloyd S. Shapley* (1988), pp. 139–151.

[45] André Casajus and Frank Huettner. “Calculating direct and indirect contributions of players in cooperative games via the multilinear extension”. In: *Economics Letters* 164 (2018), pp. 27–30.

[46] Gilad Zlotkin and Jeffrey S Rosenschein. *Coalition, cryptography, and stability: Mechanisms for coalition formation in task oriented domains*. Alfred P. Sloan School of Management, Massachusetts Institute of Technology, 1994.

[47] David Liben-Nowell, Alexa Sharp, Tom Wexler, and Kevin Woods. “Computing Shapley value in supermodular coalitional games”. In: *International Computing and Combinatorics Conference*. Springer. 2012, pp. 568–579.

[48] Osnat Israeli. “A Shapley-based decomposition of the R-square of a linear regression”. In: *The Journal of Economic Inequality* 5.2 (2007), pp. 199–212.

[49] Ian Covert and Su-In Lee. “Improving KernelSHAP: Practical Shapley Value Estimation Using Linear Regression”. In: *International Conference on Artificial Intelligence and Statistics*. PMLR. 2021, pp. 3457–3465.

[50] Nord Pool Group. *Day-ahead power prices of Trondheim, Norway during November, 2020*. https://www.nordpoolgroup.com/Market-data1/Dayahead/Area-Prices/ALL1/Monthly/?view=table. 2020.

5 | MPC-based RL for Smart Homes

This paper presents a Model Predictive Control (MPC)-based Reinforcement Learning (RL) approach for a Home Energy Management System (HEMS). The house consists of an air-to-water heat pump connected to a hot water tank that supplies thermal energy to a water-based floor heating system. Additionally, it includes a photovoltaic (PV) array and a battery storage system. The HEMS is supposed to exploit the house thermal inertia and battery storage to shift demand from peak hours to off-peak periods and earn benefits by selling excess energy to the utility grid during periods of high electricity prices. However, designing such a HEMS is challenging because the discrepancies due to model mismatch make erroneous predictions of the system dynamics, leading to a non-optimal decision making. Besides, uncertainties in the house thermodynamics, misprediction in the forecasting of PV generation, outdoor temperature, and user load demand make the problem more challenging. We solve this issue by approximating the optimal policy by a parameterized MPC scheme and updating the parameters via a Compatible Delayed Deterministic Actor-Critic (with Gradient Q-learning critic, i.e., CDDAC-GQ) algorithm. Simulation results show that the proposed MPC-based RL HEMS can effectively deliver a policy that satisfies both indoor thermal comfort and economic costs even in the case of inaccurate model and system uncertainties. Furthermore, we conduct a thorough comparison between the CDDAC-GQ algorithm and the conventional Twin Delayed Deep Deterministic policy gradient

(TD3) algorithm, the results of which affirm the efficacy of our proposed method in addressing complex HEMS problems.

Nomenclature

T_{w}	Wall temperature$(^{\circ}C)$
T_{in}	Indoor temperature$(^{\circ}C)$
T_{g}	Ground temperature$(^{\circ}C)$
T_{p}	Water pipeline temperature$(^{\circ}C)$
T_{out}	Outdoor temperature$(^{\circ}C)$
T_{inl}	Inlet water temperature of the pipeline$(^{\circ}C)$
T_{ret}	Return water temperature of the pipeline$(^{\circ}C)$
T_1	Node 1 (of the hot water tank) temperature$(^{\circ}C)$
T_2	Node 2 temperature$(^{\circ}C)$
T_3	Node 3 temperature$(^{\circ}C)$
M_{inl}	Inlet water mass flow rate to the pipeline$(\mathrm{kg/s})$
c_{wat}	Specific heat capacity of water$(\mathrm{J}/(\mathrm{kg}\cdot\mathrm{K}))$
X_{v}	Opening of the valve$(\%)$
COP	Coefficient of performance of the heat pump
E	Battery stored energy (kWh)

B	Electricity buying price$(\mathrm{EUR/MWh})$
S	Electricity selling price$(\mathrm{EUR/MWh})$
P_{hp}	Compressor power of the heat pump(kW)
P_{rad}	Solar irradiance$(\mathrm{kW/m^2})$
P_{pv}	Generated PV power(kW)
P_{ch}	Battery charging power (kW)
P_{dis}	Battery discharging power (kW)
P_{app}	Total loads power of home appliances(kW)
P_{buy}	Buying power from the grid(kW)
P_{sell}	Selling power to the grid(kW)

5.1 Introduction

In recent years, as global warming and the energy crisis have intensified, there has been a growing interest in Home Energy Management Systems (HEMSs), which are considered to have great potential for reducing building energy costs and improving energy efficiency and stability of the grid [1]. HEMSs monitor and manage home energy consumption patterns to achieve specific goals (e.g., cost, comfort, etc.) by reducing or shifting consumption [2]. Initially, the functionality of HEMS is relatively simple, considering only a few of the most critical sources of energy consumption: Heating, Ventilation, and Air Conditioning (HVAC), water heaters, general appliances, and lighting [3]. Therefore, some scheduling strategies based on mathematical optimization are proposed, such as mixed integer nonlinear programming [4], dynamic programming [5], stochastic optimization [6], etc. However, with the development of

sensing, communication, intermittent Renewable Energy Sources (RES), Energy Storage Systems (ESS), and home-to-grid technology, home energy systems have become increasingly complex and highly uncertain. Strategies based on mathematical optimization are apparently no longer able to solve such large and complex optimization problems subject to multiple known and unknown disturbances [7]. Therefore, the literature survey in this paper focuses on more advanced HEMS approaches based on Model Predictive Control (MPC), Data-driven MPC (DMPC), and Reinforcement Learning (RL).

A typical modern HEMS connects household loads (controllable and uncontrollable), RES and ESS (e.g., photovoltaic (PV)-batteries), and the utility grid. It operates on a similar principle: based on the forecasts of customer load consumption, weather, renewable generation, and other factors, it creates an optimal energy consumption policy, while ensuring user comfort, lower energy costs, and constraints satisfaction. The policy is based on electricity price or grid incentive signals to achieve both comfortable and economic purposes by reducing/shifting the power consumption and/or trading power with the utility grid [8]. For example, HEMS can take advantage of battery storage or the house thermal inertia to shift energy consumption to periods of low electricity prices and sell the produced/stored energy to the utility grid during periods of high electricity prices, thereby generating revenue. However, designing an effective HEMS is challenging for two main reasons:

- Thermodynamic models of buildings are extremely nonlinear and complex, and existing models are often simplified forms of real system dynamics. This is because these models usually consider only important factors such as heating/cooling power and weather conditions (heat dissipation), but ignore factors such as air permeability, furniture thermal mass, occupancy changes, etc [9].
- Contemporary HEMSs have many sources of uncertainties, including renewable energy generation, household load demand, electricity price, and weather forecast [10].

As a result, more advanced HEMS strategies based on MPC, DMPC and

RL have emerged in recent years, and we provide a brief summary of these strategies in Table 5.1, Table 5.2, and Table 5.3.

5.1.1 MPC-based HEMS

MPC is one of the mainstream solutions for home energy management, with the advantage of making full use of the house dynamics model and predictive information (e.g. weather, electricity prices) to predict its future behavior. The system constraints are easy to implement and also have strong theoretical support to ensure its feasibility and stability. A critical characteristic of HEMS is its inherent nonlinearity, which can be attributed to factors like nonlinear thermodynamics, appliance behaviors, and fluctuating consumer energy usage patterns. While some studies have favored linearized MPC for its computational traceability [30, 31], this linearization can be limiting. Specifically, for extended prediction horizons or when the system significantly deviates from a certain operating point, a linearized model may fall short of capturing the system's true dynamics. This inadequacy can result in suboptimal or even unviable control strategies. Some studies have sought to address these limitations by adopting Nonlinear Model Predictive Control (NMPC) [32, 33]. While NMPC can handle system nonlinearities more accurately, this improvement comes at the expense of a higher computational load. Regardless of whether they are linear or nonlinear, traditional MPC approaches share a common drawback: they require highly accurate models and are sensitive to uncertainties commonly found in HEMS [1]. To address the uncertainty issues, Robust MPC (RMPC) and Stochastic MPC (SMPC) are the two main variants. RMPC designs consider the worst-case scenario and thus always lead to overly conservative solutions. SMPC is characterized by the use of chance constraints. It is quite effective against probabilistic uncertainties, but the obvious drawback of the SMPC approach is the high computational cost. Compared to classic MPC, SMPC and RMPC can improve the case with disturbances

[1] not comprehensive
[2] not comprehensive
[3] not comprehensive

Table 5.1: A brief summary of different HEMS scheduling strategies: (a) MPC part[1]

	Reference	Advantages	Limitations
Classic MPC	[11–13]	• Exploit the thermodynamic model of the house to predict its behavior. • Sequences of predictive information (e.g. weather, electricity prices) can be incorporated. • System and control constraints are easy to implement.	• Rely on accurate system models. • Cannot handle system uncertainties or inaccurate predictions.
RMPC	[14–16]	• Design the controller based on the estimate of the worst-case scenario. • Effective in cases where uncertainties degrade control performance or even endanger the system stability.	• The control strategy is usually too conservative. (e.g., min-max RMPC.)
SMPC	[17–19]	• Use chance constraints to trade-off control performance against constraint violation. • Effective for cases that suffer from probabilistic uncertainties.	• The computational burden is usually heavy. (e.g., scenario tree SMPC.)

Table 5.2: A brief summary of different HEMS scheduling strategies: (b) DMPC part[2]

	Reference	Advantages	Limitations
DMPC Type I	[20–22]	• Learn prediction models from data with uncertainty quantification. • No prior knowledge of the model is required and uncertainties can be handled to some extent. • Preserves the reliability and "easy-to-implement constraints" from MPC.	• Rely on the quantity and quality of data and is prone to overfitting. • A model-driven approach: learning the MPC model that best fits the real system does not necessarily yield a policy that achieves the best closed-loop performance.
DMPC Type II	[23, 24]	• MPC generates training data for the ML-based controller. • Fast online optimization and uncertainties can be handled to some extent.	• Still rely on the model accuracy as the training data is generated by MPC. • ML controllers lack interpretability and reliability.

Table 5.3: A brief summary of different HEMS scheduling strategies: (c) RL part[3]

	Reference	Advantages	Limitations
Classic RL	[25, 26]	• Classic RL uses Q tables as function approximations. • DRL uses DNN as function approximations.	• Classical RL cannot handle continuous action. • Need for massive data and long training time.
DRL	[27–29]	• No prior knowledge of the model is required and uncertainties can be handled to some extent. • Can handle long-term or even infinite-horizon problems.	• Lack of interpretability and reliability. • Safety is not guaranteed. • Predictive information in the form of time series leads to the curse of dimensionality.

or uncertainties to satisfy constraints (i.e., maintain room temperature or comfort level), but the cost-saving performance is correspondingly weakened [34]. Beyond these classic methods, recent advancements in the field have birthed innovative approaches like Ensemble NMPC (EnNMPC). Ensemble techniques have been explored that can greatly enhance the accuracy of energy forecasting models [35], and one notable study demonstrates how EnNMPC can effectively address uncertainties in battery degradation and fluctuating generation/load patterns while significantly reducing the daily costs for prosumers [36]. However, they may also amplify computational burden and the efficacy of EnNMPC can be undermined if the quality and diversity of the ensemble models are inadequate.

5.1.2 DMPC-based HEMS

DMPC (also known as Learning-based MPC (LBMPC)) combines the advantages of Machine Learning (ML) and MPC, which can effectively mitigate the effects of disturbances and uncertainties. DMPC for HEMS can be divided into two main categories [37]. The first category (noted as Type I) trains predictive models offline from data with uncertainty quantification. The trained models are then used by MPC to predict the output during online optimization. Since the MPC still acts as a controller, this type of HEMS maintains the benefits of MPC in terms of reliability and ease of implementing constraints. However, the accuracy of the trained models depends on sufficient, high-quality data that should fully reflect the state transitions of real systems under the influence of many uncertainties. In addition to the potential problem of overfitting, another easily overlooked issue is that model regression using data falls under the category of "model-driven", yet it has been argued in [38] that learning the MPC model that best fits the real system does not necessarily yield a policy that achieves the best closed-loop performance. The second category (noted as Type II) uses the data generated by MPC rollouts to train an ML controller. Since ML replaces MPC in online optimization, the online computational complexity is greatly reduced. However, the performance of the ML controller still depends on the

MPC model that generates the training data. Another concern is the lack of interpretability and reliability of ML controllers.

5.1.3 RL-based HEMS

RL is gaining increasing interest in HEMS. RL is a model-free approach that seeks the optimal policy by interacting with the environment. RL agents can take different control actions based on different observations (i.e., can handle the uncertainties of HEMS) and can handle long-term problems, e.g., considering the monthly operating cost of a HEMS [3, 39]. The early classic RL could not be applied to continuous action spaces due to the use of Q-table. Deep RL (DRL) using Deep Neural Networks (DNN) as function approximations solves this problem. However, there are still three main obstacles that prevent the wide application of RL in HEMS. The first and most fatal drawback is that training RL agents requires huge amounts of data and extremely long training times due to the complexity, high stochasticity, and long sampling times of HEMS [40]. For example, [41] noted that the RL agent may require at least 47.5 years of training (consider 5 million interactions with a sampling time of 5 minutes) to achieve the same performance as a feedback controller for an HVAC system. Safety is the second drawback. RL requires iterative "trial and error" learning, which can lead the system into potentially dangerous or financially costly states. As a result, the vast majority of RL-based HEMS studies are conducted in simulation environments [40]. Due to the inevitable differences between simulation environments and real systems, there are concerns about whether trained RL controllers can remain safe and effective in real environments. Wasting known information about models and predictions is also a major concern. To avoid the curse of dimensionality, many works have to discard the available sequence of prediction information that could have been used as states. As can be seen from [42], the performance of RL without prediction sequences is only about 38% of the best case (where the model and all prediction information are known). Despite the considerable progress of RL-based HEMS, it has to be recognized that most of the current approaches either ignore certain uncertainties in

the system, ignore the RES or ESS and consider only a few variables, or use quite simplified thermodynamic models as environments, all of which make the training of RL easier. However, in reality, it is still very challenging to design HEMS based only on those highly random and intricate data.

5.1.4 Combining MPC and RL

With this background, we propose to solve the HEMS problem with an MPC-based RL approach, which was first proposed in [43] in 2019. The main idea is that one can parameterize the model, cost function, and constraints of an MPC, and if the parameterization is "rich" enough, then the MPC with optimal parameters can yield the optimal policy even if the model is inaccurate and the system has uncertainties [43, 44]. Some advances have been made in both theory [45–48] and application [49–51] of this approach. In this paper, we first construct a HEMS problem that takes into account an inaccurate system model with uncertainties in house thermodynamics, load demand, renewable energy generation, and weather forecasts. We then approximate the optimal energy consumption policy by a parametric MPC scheme in which the parameters are deployed in the model, cost functions, and constraints of the MPC formulation to overcome model mismatches and uncertainties. The parameters are updated by a proposed RL algorithm according to the principle of simultaneously satisfying thermal comfort and reducing economic costs. Applying the MPC-based RL approach for HEMS design is motivated by the following reasons:

- An approximate model of a house energy system is usually available, and the MPC-based RL approach can leverage this known (though inaccurate) model, thus allowing the use of a smaller dataset.

- Some information for a short period of time in the future, such as electricity prices, outdoor temperature, solar radiation (PV generation), and household appliance loads, is usually either available

or predictable. This information can simply be considered by the MPC-based RL approach without causing the curse of dimensionality.

- Due to the use of the MPC framework, system constraints can be easily implemented. Besides, since "action exploration" is performed on the basis of the secure output of MPC, safety is significantly improved compared to pure RL methods.

- Due to the use of RL, the MPC-based RL approach is designed to adapt to the inherent nonlinear and uncertain properties of the system, making it suitable for dealing with the challenges posed by nonlinearities and uncertainties in HEMS.

- It is a performance-driven approach, which means that the controller is trained to improve the overall performance considering comfort and cost, rather than just to replicate the model more precisely. Moreover, unlike DNNs, parameters in the parameterized MPC are interpretable.

5.1.5 Contributions

To the best of our knowledge, the MPC-based RL approach has never been applied to HEMS, and we hope that this work will provide a new perspective for solving the HEMS problem. The contributions of this paper are summarized below:

- We construct a standard HEMS paradigm that incorporates weather factors, renewable energy sources, energy storage, heat pump floor heating units, other appliance loads, and the utility grid. Based on the above system, an optimization problem is constructed in the framework of RL considering both user comfort and economic cost.

- To overcome the inaccurate model and system uncertainties, the MPC-based RL method is proposed to solve the above optimiza-

tion problem. We describe in detail the procedures and thoughts for constructing the parameterized MPC.

- To optimize the parameter training process, we propose a new learning algorithm, the Compatible Delayed Deterministic Actor-Critic (with Gradient Q-learning critic, i.e., CDDAC-GQ), based on our previously published work [52]. We also explain the improvements of the new algorithm compared to the old one.
- The performance of the proposed MPC-based RL approach is quantified through simulation tests using real data such as electricity prices, weather forecasts, and load demand, and is further examined through a comprehensive comparison with the traditional Twin Delayed Deep Deterministic policy gradient (TD3) algorithm.

The remainder of the paper is constructed as follows. Section 5.2 introduces the whole HEMS, and the problem formulation is given in Section 5.3. Section 5.4 is the main part that presents the proposed MPC-based RL approach and the corresponding algorithm. Finally, the simulation results are discussed in Section 5.5.

5.2 HEMS Model

In this study, we consider a typical smart home model in Nordic countries. The house is heated by an air-to-water Heat Pump (HP) equipped with a Hot Water Tank (HWT) and a water-based floor heating system (see Fig. 5.1). In addition, a residential PV-battery system is included for power generation and storage.

5.2.1 House Thermodynamics

The house model, referring to [53], describes the energy conservation among the house walls, indoor, ground, and water pipeline temperatures.

The thermodynamics are given by

$$\dot{T}_{\mathrm{w}} = \frac{1}{C_{\mathrm{w}}}[k_{\mathrm{w,out}}(T_{\mathrm{out}} - T_{\mathrm{w}}) + k_{\mathrm{w,in}}(T_{\mathrm{in}} - T_{\mathrm{w}})] + \epsilon_1, \tag{5.1a}$$

$$\dot{T}_{\mathrm{in}} = \frac{1}{C_{\mathrm{in}}}[k_{\mathrm{w,in}}(T_{\mathrm{w}} - T_{\mathrm{in}}) + k_{\mathrm{g,in}}(T_{\mathrm{g}} - T_{\mathrm{in}})] + \epsilon_2, \tag{5.1b}$$

$$\dot{T}_{\mathrm{g}} = \frac{1}{C_{\mathrm{g}}}[k_{\mathrm{g,in}}(T_{\mathrm{in}} - T_{\mathrm{g}}) + k_{\mathrm{p,g}}(T_{\mathrm{p}} - T_{\mathrm{g}})] + \epsilon_3, \tag{5.1c}$$

$$\dot{T}_{\mathrm{p}} = \frac{1}{C_{\mathrm{p}}}[k_{\mathrm{p,g}}(T_{\mathrm{g}} - T_{\mathrm{p}}) + M_{\mathrm{inl}}c_{\mathrm{wat}}(T_{\mathrm{inl}} - T_{\mathrm{p}})] + \epsilon_4, \tag{5.1d}$$

where C_{w},C_{in},C_{g}, and C_{p} are the corresponding heat capacities of the wall, indoor-air, ground pavement, and water pipeline; $k_{\mathrm{w,out}}$, $k_{\mathrm{w,in}}$, $k_{\mathrm{g,in}}$, and $k_{\mathrm{p,g}}$ are the heat transfer coefficients between each medium. Variables $\boldsymbol{\epsilon} = \{\epsilon_1, \epsilon_2, \epsilon_3, \epsilon_4\}$ are the internal uncertainties of the house thermodynamics and assumed to be normally distributed. (The physical meaning of the other variables is given in the Nomenclature.) The outdoor temperature forecast T_{out} is one of the external uncertainties of the system and is assumed to be a variable with Gaussian prediction error.

5.2.2 Heat Pump and Hot Water Tank

The convective heat transfer in the HWT is characterized by a three-node model [54], shown as

$$\dot{T}_1 = \frac{1}{m_1 c_{\mathrm{wat}}}[R_1(T_2 - T_1) - R_{\mathrm{w}}(T_1 - T_{\mathrm{out}}) + X_{\mathrm{v}}M_{\mathrm{inl}}c_{\mathrm{wat}}(T_2 - T_1)], \tag{5.2a}$$

$$\dot{T}_2 = \frac{1}{m_2 c_{\mathrm{wat}}}[R_2(T_3 - T_2) - R_2(T_2 - T_1) - R_{\mathrm{w}}(T_2 - T_{\mathrm{out}}) + X_{\mathrm{v}}M_{\mathrm{inl}}c_{\mathrm{wat}}(T_3 - T_2) + \mathrm{COP}P_{\mathrm{hp}}], \tag{5.2b}$$

$$\dot{T}_3 = \frac{1}{m_3 c_{\mathrm{wat}}}[-R_3(T_3 - T_2) - R_{\mathrm{w}}(T_3 - T_{\mathrm{out}}) + X_{\mathrm{v}}M_{\mathrm{inl}}c_{\mathrm{wat}}(T_{\mathrm{ret}} - T_3)], \tag{5.2c}$$

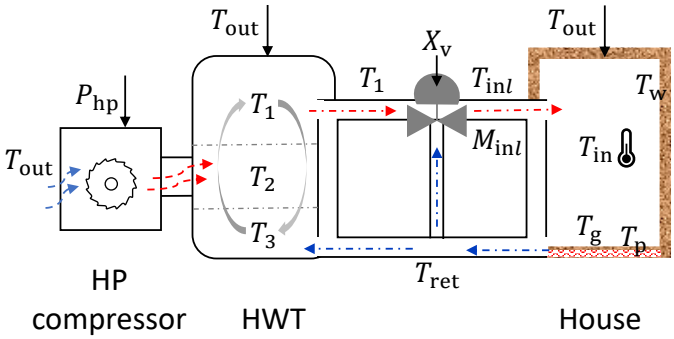

Figure 5.1: Schematic of the floor heating house equipped with a heat pump and a hot water tank. The external air (T_{out}) passes through the heat pump compressor converting electricity into heat. The input power of the heat pump (P_{hp}) directly controls the temperature of the middle layer (T_2) of the water tank. The upper water flow of the tank (T_1) is connected to the floor heating pipes through a valve (X_{v}). After heating the inside air (T_{in}), the water flow returns to the lower layer of the tank (T_3).

with

$$T_{\text{ret}} = (1 - \exp(-\rho))T_{\text{g}} + \exp(-\rho))T_{\text{p}}, \tag{5.3}$$

$$T_{\text{inl}} = X_{\text{v}}(T_1 - T_{\text{ret}}) + T_{\text{ret}}. \tag{5.4}$$

Parameters $R_{1,2,3}$ and R_{w} are the heat transfer coefficients of the HWT layers and tank wall; $m_{1,2,3}$ are the water masses of each layer; ρ is the absorption coefficient of the pipeline. (The physical meaning of the other variables is given in the Nomenclature.) The heat output from the HP compressor is approximated by its coefficient of performance (COP) as $Q_{\text{hp}} = \text{COP}P_{\text{hp}}$, where P_{hp} represents the power supplied to the compressor and COP adopts a time-varying linear function as

$$\text{COP} = a_{\text{cop}}T_{\text{out}} + b_{\text{cop}}\left(\frac{T_2 + T_3}{2}\right) + c_{\text{cop}}. \tag{5.5}$$

Coefficients $a_{\text{cop}}, b_{\text{cop}}, c_{\text{cop}}$ are chosen based on the experimental analysis in [55]. Besides, the temperature of each layer of the HWT, the service

power of the HP, and the valve opening of the HWT should comply with the following constraints

$$15^{\circ}\mathrm{C} \lessapprox T_1, T_2, T_3 \lessapprox 60^{\circ}\mathrm{C}, \tag{5.6}$$
$$0\mathrm{kW} \leq P_{\mathrm{hp}} \leq 3\mathrm{kW}, \tag{5.7}$$
$$20\% \leq X_{\mathrm{v}} \leq 80\%. \tag{5.8}$$

5.2.3 Solar PV

The power generated by PV is assumed as a linear function of the solar irradiance [56], described as

$$P_{\mathrm{pv}}(t) = 10^{-3}\beta A_{\mathrm{pv}} P_{\mathrm{rad}}(t), \tag{5.9}$$

where β and A_{pv} are the conversion efficiency and effective radiant area of the PV panel. The variable P_{rad} with Gaussian error represents an inaccurate prediction of the solar irradiance, which is another external uncertainty of the system. The generated PV power has three destinations: to be sold to the utility grid, to meet the home load demand, and to charge the battery.

5.2.4 Battery

The battery with a capacity of $5\mathrm{kWh}$ is modeled as

$$\dot{E}(t) = \eta P_{\mathrm{ch}}(t) - \frac{1}{\eta} P_{\mathrm{dis}}(t), \tag{5.10}$$

with

$$1\mathrm{kWh} \lessapprox E(t) \lessapprox 4\mathrm{kWh}, \tag{5.11}$$
$$0\mathrm{kW} \leq P_{\mathrm{ch}}(t), P_{\mathrm{dis}}(t) \leq 1\mathrm{kW}, \tag{5.12}$$
$$P_{\mathrm{ch}}(t) P_{\mathrm{dis}}(t) = 0, \tag{5.13}$$

where η is the efficiency coefficient. To prolong the battery life, it is recommended to keep the battery level around $20\% - 80\%$, as described

in (5.11). The charging/discharging power limit is shown in (5.12). Note that the charging and discharging processes should not occur simultaneously, i.e., constraint (5.13) should be satisfied, but due to the presence of η, the solved policy would follow this rule even if (5.13) is not explicitly required. It is worth mentioning that for the sake of clarity and focus, we choose not to delve into the battery degradation issue. However, it should be recognized that its inclusion in the proposed algorithm is straightforward (could be perceived as just another source of system uncertainty) and would not compromise the core of the proposed methodology.

5.2.5 Utility Grid

The power bought from the utility grid $P_{\text{buy}}(t)$ is used to charge the battery and to meet the home load demand, while the sold power $P_{\text{sell}}(t)$ comes from battery discharging and PV generation. It should satisfy

$$P_{\text{buy}}(t)P_{\text{sell}}(t) = 0, \tag{5.14}$$

$$0\text{kW} \leq P_{\text{buy}}(t), P_{\text{sell}}(t) \leq 5\text{kW}, \tag{5.15}$$

where (5.14) is likewise an implicit constraint that the optimal policy would obey due to the gap between the buying and selling prices. And (5.15) takes into account the peak power constraint to ensure a smooth operation of the grid.

5.2.6 Power Balance

Overall, for the power balance in the house, we have

$$P_{\text{app}}(t) + P_{\text{hp}}(t) + P_{\text{ch}}(t) + P_{\text{sell}}(t) = P_{\text{dis}}(t) + P_{\text{buy}}(t) + P_{\text{pv}}(t). \tag{5.16}$$

The left side of the equation is the total consumption power, including the power of household appliances [4], heat pumps, charging batteries, and selling electricity; while the right side of the equation is the total absorption

[4] Those uncontrollable appliances such as lamps, induction cookers, TVs, etc.

power, including the power of discharging batteries, buying electricity, and PV generation. Note that the total load demand of appliances $P_{\text{app}}(t)$, similar to $T_{\text{out}}(t)$ and $P_{\text{rad}}(t)$, is a highly uncertain variable, which is an estimated value with Gaussian error based on the user's consumption pattern. Another interesting point to mention is that, in constructing the model, while for some it may seem intuitive to represent $P_{\text{ch}}(t)/P_{\text{dis}}(t)$ and $P_{\text{buy}}(t)/P_{\text{sell}}(t)$ actions with a singular variable-where positive values indicate one action and negative values indicate the opposite-we opted for two independent variables. This decision was driven primar-

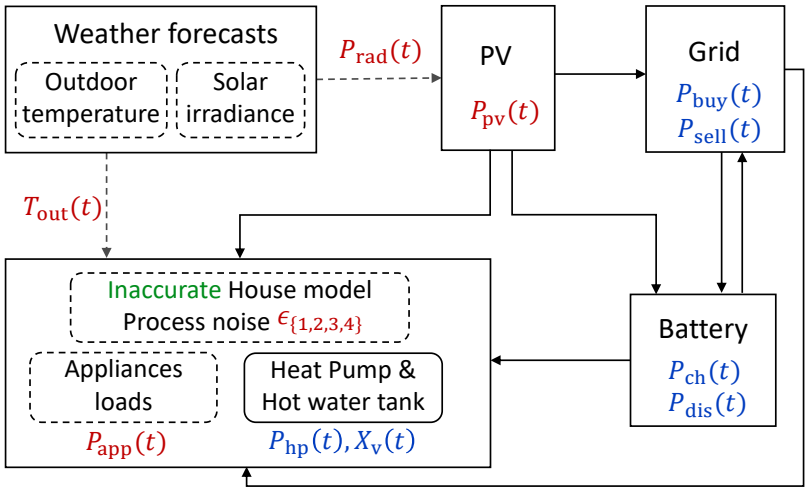

Figure 5.2: Schematic of the home energy management system, where the blue font variables indicate control signals and the red fonts indicate uncertainty sources, including internal uncertainties from the thermodynamics, and external uncertainties from the predictions of the outdoor temperature, solar irradiance/PV generation, and appliances load demand. Besides, as marked in green, we also consider an inaccurate (simplified) house model in the setting, which will be explained in the later section.

ily by two considerations. First, distinct variables ensure the absence of discontinuities and non-differentiable regions in our optimization model, crucial for effective gradient-based optimization. Second, given

that our model inherently differentiates between coefficients for battery charging/discharging and for buying/selling transactions, using a single variable would introduce complications. Specifically, it would necessitate conditional checks within the algorithmic differentiation tool, e.g. CasADi [57], during the MPC formulation, presenting a technical challenge. Consequently, the schematic of the whole HEMS is shown in Fig. 5.2.

5.3 Problem Formulation

5.3.1 Real Model

The HEMS can be formulated as a stochastic optimization problem. The state vector $\mathbf{s} \in \mathbb{R}^8$ reads as

$$\mathbf{s} = \{T_\mathrm{w}, T_\mathrm{in}, T_\mathrm{g}, T_\mathrm{p}, T_1, T_2, T_3, E\}. \tag{5.17}$$

The input vector $\mathbf{a} \in \mathbb{R}^6$ reads as

$$\mathbf{a} = \{P_\mathrm{ch}, P_\mathrm{dis}, P_\mathrm{buy}, P_\mathrm{sell}, P_\mathrm{hp}, X_\mathrm{v}\}. \tag{5.18}$$

The internal uncertainties from the house thermodynamics is $\boldsymbol{\epsilon}$, and the external uncertainties vector could be written as

$$\mathbf{d} = \{P_\mathrm{rad}, P_\mathrm{app}, T_\mathrm{out}\}, \tag{5.19}$$

where P_rad, P_app, and T_out are equal to their baseline estimations plus the corresponding normally distributed estimation errors (Gaussian white noise), i.e., we have

$$P_\mathrm{rad} = \bar{P}_\mathrm{rad} + \xi_\mathrm{rad}, \tag{5.20a}$$
$$P_\mathrm{app} = \bar{P}_\mathrm{app} + \xi_\mathrm{app}, \tag{5.20b}$$
$$T_\mathrm{out} = \bar{T}_\mathrm{out} + \xi_\mathrm{out}. \tag{5.20c}$$

Note that in real-world scenarios, the errors may neither be strictly Gaussian nor exhibit white noise characteristics. Instead, errors in these

predictions can have auto-correlations. However, it is crucial to highlight that the foundation of the following proposed algorithm is not contingent solely upon this Gaussian assumption. Its strength lies in leveraging the inherent characteristics of RL, which excels in learning from interactions with the actual environment or real data. (This means that the proposed algorithm is adept at handling a variety of noise or error distributions and strives to learn the optimal policy regardless of the specific form of the error.)

Applying the Explicit fourth-order Runge-Kutta (ERK4) method with a sampling time of $\Delta t = 15\text{min}$, the system is discretized as

$$\mathbf{s}_{t+1} = f(\mathbf{s}_t, \mathbf{a}_t, \mathbf{d}_t, \boldsymbol{\epsilon}_t). \tag{5.21}$$

5.3.2 RL Objective

The objective of the HEMS is to seek an optimal policy $\boldsymbol{\pi}^\star(\mathbf{s})$ that decreases the customer's net energy costs while satisfying indoor temperature comfort conditions. Therefore, the objective function that we aim to minimize is described as

$$J(\boldsymbol{\pi}) = \mathbb{E}_{\boldsymbol{\pi}}\left[\sum_{t=0}^{N} l(\mathbf{s}_t, \mathbf{a}_t)\middle|\mathbf{a}_t = \boldsymbol{\pi}(\mathbf{s}_t)\right], \tag{5.22}$$

where the stage cost (reward) l is composed of spot-market energy cost l_{spot} and temperature comfort term l_{temp}, written as

$$\begin{aligned} l(\mathbf{s}_t, \mathbf{a}_t) = &\underbrace{[B(t)P_{\text{buy}}(t) - S(t)P_{\text{sell}}(t)]}_{l_{\text{spot}}} + \\ &\underbrace{[c_{\text{low}}(20 - T_{\text{in}}(t))_+ + c_{\text{hig}}(T_{\text{in}}(t) - 24)_+]}_{l_{\text{temp}}}. \end{aligned} \tag{5.23}$$

Variables $B(t), S(t)$ are the spot-market electricity buying and selling prices. With consideration of a 25% tax rate, we approximately assume that $B(t) = 1.25S(t)$. The price for $B(t)$ can be obtained from the day-ahead market prices on Nord Pool [58]. It's imperative to highlight that

this study is grounded in the assumption that electricity prices are known and remain constant. This aligns with the prevalent practices in Norway, where the emphasis is on the spot-market prices (day-ahead market prices). Such prices are determined a day prior, based on anticipated supply and demand, facilitating strategic planning for utilities and market players within a (at least) 12-hour forecast horizon. (Although the more dynamic energy landscape of future power markets may necessitate real-time or intraday markets to promptly respond to unforeseen fluctuations in demand, renewable energy sources, and other grid contingencies, it is beyond the primary scope of this study.) The function $(x)_+$ equals x when $x > 0$ and zero otherwise, which implies that the indoor temperature is supposed to be within $[20°\mathrm{C}, 24°\mathrm{C}]$. Being lower or higher than the bounds would introduce a penalty to the objective function weighted by c_{low} and c_{hig} respectively.

5.3.3 Simplified Control-Oriented Model

Model (5.21) gives a relatively comprehensive but not completely accurate description of the system. This is due to the practical limitation that arises from having a finite number of sensors and gauges, which restricts the amount of information that can be obtained. It can be expected that the dynamics of the real system are much more complex, so (5.21) is only a simplified and inaccurate model of the real system. In reality, precisely modeling the entire home energy system is very challenging because of its intricate dynamics with many variables and uncertainties. Moreover, even if a sufficiently accurate model could be built, solving such a complex model in an optimization problem would be computationally expensive and completely impractical. In this work, in order to verify the effectiveness of our proposed approach, we assume that (5.21) is the real system and an oversimplified model is used for controller design. Therefore, the assumption is made of having poor knowledge about the cumbersome heat transfer process. Specifically, instead of the complete heat transitions between $T_{\mathrm{w}}, T_{\mathrm{in}}, T_{\mathrm{g}}$ and T_{p}, the house thermodynamics is described with only T'_{in} and T'_{g}. Likewise, the heating system (HP and HWT) is represented by only one state T'_2 instead of the sophisticated

three-node model. Although the thermodynamics between the retained states are reconstructed rationally based on the physical sense and the heat transfer coefficients are superimposed accordingly, model errors are inevitable. It is well known that a common method to eliminate model errors is model fitting, i.e., parameterizing the coefficients in the dynamic equations with some unknown parameters, which is also adopted in this paper.

Overall, the simplified model is described as below

$$\dot{T}'_{\mathrm{in}} = \frac{1}{C_{\mathrm{in}}}[\theta_{\mathrm{m}1} k_{\mathrm{w,out}} k_{\mathrm{w,in}} (\bar{T}_{\mathrm{out}} - T'_{\mathrm{in}}) + \theta_{\mathrm{m}2} k_{\mathrm{g,in}} (T'_{\mathrm{g}} - T'_{\mathrm{in}})] + \theta_{\epsilon 2}, \tag{5.24a}$$

$$\dot{T}'_{\mathrm{g}} = \frac{1}{C_{\mathrm{g}}}[\theta_{\mathrm{m}3} k_{\mathrm{g,in}} (T'_{\mathrm{in}} - T'_{\mathrm{g}}) + \theta_{\mathrm{m}4} M_{\mathrm{inl}} c_{\mathrm{wat}} X_{\mathrm{v}} (T'_2 - T'_{\mathrm{g}})] + \theta_{\epsilon 3}, \tag{5.24b}$$

$$\dot{T}'_2 = \frac{1}{(m_1 + m_2 + m_3) c_{\mathrm{wat}}}[-\theta_{\mathrm{m}5} R_{\mathrm{w}} (T'_2 - \bar{T}_{\mathrm{out}}) + \theta_{\mathrm{m}6} X_{\mathrm{v}} M_{\mathrm{inl}} c_{\mathrm{wat}} (T'_{\mathrm{g}} - T'_2) + \mathrm{COP} P_{\mathrm{hp}}], \tag{5.24c}$$

$$\mathrm{COP} = a_{\mathrm{cop}} T_{\mathrm{out}} + b_{\mathrm{cop}} T'_2 + c_{\mathrm{cop}}, \tag{5.24d}$$

$$15^{\circ}\mathrm{C} \leq T'_2 \leq 60^{\circ}\mathrm{C}, \tag{5.24e}$$

$$(5.7), (5.8), (5.9), (5.10), (5.11), (5.12), (5.15), (5.16), \tag{5.24f}$$

which forms the new state vector $\mathbf{s}' \in \mathbb{R}^4$

$$\mathbf{s}' = \{T'_{\mathrm{in}}, T'_{\mathrm{g}}, T'_2, E\}. \tag{5.25}$$

The corresponding discrete system is noted as $f'_{\boldsymbol{\theta}_{\mathrm{m}}}$, with

$$\mathbf{s}'_{t+1} = f'_{\boldsymbol{\theta}_{\mathrm{m}}}(\mathbf{s}'_t, \mathbf{a}_t, \mathbf{d}_t). \tag{5.26}$$

Note that the input vector and external uncertainties vector are the same as (5.18) and (5.19). The model fitting parameters $\boldsymbol{\theta}_{\mathrm{m}} = \{\theta_{\mathrm{m}1}, \theta_{\mathrm{m}2}, \theta_{\mathrm{m}3}, \theta_{\mathrm{m}4}, \theta_{\mathrm{m}5}, \theta_{\mathrm{m}6}, \theta_{\epsilon 2}, \theta_{\epsilon 3}\}$, where the two parameters $\theta_{\{\epsilon 2, \epsilon 3\}}$ are used to replace the dynamics uncertainties $\epsilon_{\{2,3\}}$ in (5.1). Therefore, $f'_{\boldsymbol{\theta}_{\mathrm{m}}}(\cdot)$ is a deterministic model rather than the stochastic one in (5.21), which avoids tackling the computationally expensive stochastic optimization problem.

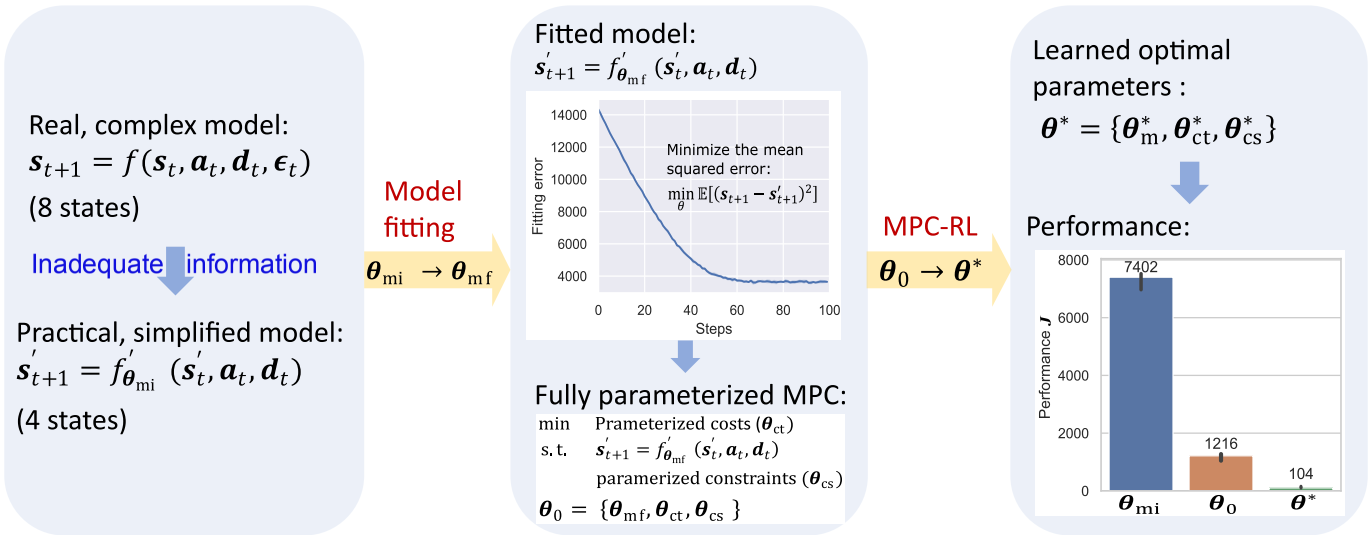

Figure 5.3: The mind map of the problem-solving steps.

5.3.4 Fitting the Simplified Model

The physical parameters of the HEMS are given in Table 5.4. To fit the simplified model, we train the model parameters by minimizing the mean squared error, where the training data are collected by running the real system (5.21) for long enough time. The model parameters are initialized as $\boldsymbol{\theta}_{\mathrm{mi}}$ and are converged to $\boldsymbol{\theta}_{\mathrm{mf}}$ after the fitting process, their values are given in Table 5.5. Note that the fitting error can not decrease to zero due to the intrinsic structure mismatch, therefore, the fitted model $f'_{\boldsymbol{\theta}_{\mathrm{mf}}}(\cdot)$ still can not capture the real system and an MPC scheme that uses the inaccurate fitted model would yield a deviated solution. In summary, data fitting does compensate for some model errors, but it is far from sufficient for the goal of finding the optimal policy.

Fortunately, we show in the next section that, in addition to the model parameterization, proper parameterizations of the MPC cost functions and constraints could further improve the performance (after learning the parameters), and the discrepancies induced by both model errors and system uncertainties can be compensated. The mind map of the process is shown in Fig. 5.3.

Table 5.4: System parameters.

C_{w}	$3.12 \times 10^6[\mathrm{J/K}]$	$k_{\mathrm{w,out}}$	$64.8[\mathrm{W/K}]$
C_{in}	$4.4 \times 10^6[\mathrm{J/K}]$	$k_{\mathrm{w,in}}$	$64.8[\mathrm{W/K}]$
C_{g}	$1.8 \times 10^6[\mathrm{J/K}]$	$k_{\mathrm{g,in}}$	$594.8[\mathrm{W/K}]$
C_{p}	$1.7 \times 10^6[\mathrm{J/K}]$	$k_{\mathrm{p,g}}$	$506.2[\mathrm{W/K}]$
η	90%	R_{w}	$0.99[\mathrm{W/K}]$
M_{inl}	$0.062[\mathrm{kg/s}]$	c_{wat}	$4180[\mathrm{J/(kg \cdot K)}]$
$R_{\{1,2,3\}}$	$18.8[\mathrm{W/K}]$	ρ	7.5
$m_{\{1,2,3\}}$	$66.38[\mathrm{kg}]$	a_{cop}	0.088
β	0.429	b_{cop}	-0.079
A_{pv}	$35[\mathrm{m}^2]$	c_{cop}	7.253
$\epsilon_{\{1,2,3,4\}}$	$\mathcal{N}(0, 2.5 \times 10^{-5})$	ϵ_{rad}	$\mathcal{N}(0, 3.6 \times 10^{-5})$
ϵ_{app}	$\mathcal{N}(0, 1.6 \times 10^{-3})$	ϵ_{out}	$\mathcal{N}(0, 3.6 \times 10^{-1}))$

Table 5.5: Parameters variations in model fitting.

Parameter		Initial		After model fitting		
θ_{m1}	$\boldsymbol{\theta}_{\mathrm{m}}$	1	$\boldsymbol{\theta}_{\mathrm{mi}}$	$9.99454383 \times 10^{-1}$	$\boldsymbol{\theta}_{\mathrm{mf}}$	$\boldsymbol{\theta}_0$
θ_{m2}		1		1.00003276		
θ_{m3}		1		$9.99998532e \times 10^{-1}$		
θ_{m4}		1		$9.99990420 \times 10^{-1}$		
θ_{m5}		1		1.00000006		
θ_{m6}		1		1.00000675		
θ_{e2}		0		$2.54631645 \times 10^{-2}$		
θ_{e3}		0		$-8.29237256 \times 10^{-4}$		
θ_{in1}	$\boldsymbol{\theta}_{\mathrm{ct}}$	N/A		5		
θ_{in2}		N/A		0		
θ_{in3}		N/A		0		
θ_{en1}		N/A		30		
θ_{en2}		N/A		3		
θ_{t2}	$\boldsymbol{\theta}_{\mathrm{cs}}$	N/A		0		
θ_{e}		N/A		0		

5.4 MPC-based Reinforcement Learning

Based on the inaccurate fitted model, this section presents an MPC-based RL approach that could deliver a near-optimal energy consumption policy for the above constructed HEMS problem. This is achieved by parameterizing the MPC cost functions and constraints in addition to the parametric model (5.26). The fully parameterized MPC scheme serves as a policy approximator in RL and the parameters are trained according to minimizing the RL performance (5.22). It has been mathematically proved that, even with an inaccurate model and with system uncertainties, the trained MPC can deliver the optimal policy if the parameterization is sufficiently rich (i.e., the MPC model, cost function, and constraints are adequately parameterized) [43].

5.4.1 MPC-based Policy Approximation

Consider the following MPC-scheme fully parametrized with $\boldsymbol{\theta}$

$$\min_{\hat{\mathbf{s}}',\hat{\mathbf{a}},\boldsymbol{\sigma}} \; T_{\boldsymbol{\theta}}(\hat{\mathbf{s}}'_n) + \boldsymbol{\omega}_n^\top \boldsymbol{\sigma}_n + \sum_{i=1}^{n-1} l_{\boldsymbol{\theta}}(\hat{\mathbf{s}}'_i, \hat{\mathbf{a}}_i) + \boldsymbol{\omega}_l^\top \boldsymbol{\sigma}_i \tag{5.27a}$$

$$\text{s.t.} \quad \forall\, i = 0, \dots, n-1$$

$$\hat{\mathbf{s}}'_{i+1} = f'_{\boldsymbol{\theta}_\mathrm{m}}(\hat{\mathbf{s}}'_i, \hat{\mathbf{a}}_i, \bar{\mathbf{d}}_i), \tag{5.27b}$$

$$15 - \sigma_{\mathrm{t2},i} \leq \hat{T}'_{2,i} + \theta_{\mathrm{t2}},\ \hat{T}'_{2,i} - \theta_{\mathrm{t2}} \leq 60 + \sigma_{\mathrm{t2},i}, \tag{5.27c}$$

$$15 - \sigma_{\mathrm{t2},n} \leq \hat{T}'_{2,n} + \theta_{\mathrm{t2}},\ \hat{T}'_{2,n} - \theta_{\mathrm{t2}} \leq 60 + \sigma_{\mathrm{t2},n}, \tag{5.27d}$$

$$1 - \sigma_{\mathrm{e},i} \leq \hat{E}_i + \theta_{\mathrm{e}},\ \hat{E}_i - \theta_{\mathrm{e}} \leq 4 + \sigma_{\mathrm{e},i}, \tag{5.27e}$$

$$1 - \sigma_{\mathrm{e},n} \leq \hat{E}_n + \theta_{\mathrm{e}},\ \hat{E}_n - \theta_{\mathrm{e}} \leq 4 + \sigma_{\mathrm{e},n}, \tag{5.27f}$$

$$0 \leq \hat{P}_{\mathrm{hp},i} \leq 3, \tag{5.27g}$$

$$0.2 \leq \hat{X}_{\mathrm{v},i} \leq 0.8, \tag{5.27h}$$

$$0 \leq \hat{P}_{\mathrm{ch},i}, \hat{P}_{\mathrm{dis},i} \leq 1, \tag{5.27i}$$

$$0 \leq \hat{P}_{\mathrm{buy},i}, \hat{P}_{\mathrm{sell},i} \leq 5, \tag{5.27j}$$

$$\hat{P}_{\mathrm{app},i} + \hat{P}_{\mathrm{hp},i} + \hat{P}_{\mathrm{ch},i} + \hat{P}_{\mathrm{sell},i} = \hat{P}_{\mathrm{dis},i} + \hat{P}_{\mathrm{buy},i} + \hat{P}_{\mathrm{pv},i}, \tag{5.27k}$$

$$\boldsymbol{\sigma}_i \geq 0, \boldsymbol{\sigma}_n \geq 0, \tag{5.27l}$$

$$\hat{\mathbf{s}}'_0 = \{\hat{T}'_{\mathrm{in},0}, \hat{T}'_{\mathrm{g},0}, \hat{T}'_{2,0}, \hat{E}_0\} = \{T_{\mathrm{in},t}, T_{\mathrm{g},t}, T_{2,t}, E_t\}, \tag{5.27m}$$

where equations (5.27c) and (5.27d) ensure the temperature $\hat{T}'_2$ remains within the desired bounds, and (5.27e) and (5.27f) restrain the battery energy $\hat{E}$. The slack variables vector $\boldsymbol{\sigma} = \{\sigma_{\mathrm{t2},0\dots n}, \sigma_{\mathrm{e},0\dots n}\}$ is used to penalize and minimize deviations from the desired range. By including $\boldsymbol{\sigma}$ weighted by a constant vector $\boldsymbol{\omega}$ in the cost function, any minor violations are penalized, pushing the solution to the desired bounds. On the other hand, these slack variables allow the hard constraints of the MPC to transform into soft constraints, providing not only flexibility in adhering to the constraints but also enhancing the MPC feasibility, especially in scenarios where minor deviations can be tolerated for better

overall performance. (It's noteworthy that, we have already taken into account a certain margin when designing the constraint models for T_2 and E. Therefore, minor violations of these constraints are acceptable.) The subsequent equations (5.27g)-(5.27j) ensure that the input variables remain strictly within their operational limits. The power balance is enforced in (5.27k), and the slack variables are further constrained to be non-negative in (5.27l), indicating that they only serve to represent the magnitude of constraint violations. Lastly, equation (5.27m) describes the MPC initialization conditions. Note that at every time instant, the MPC initial states are extracted from the current full states of the real system.

The details of the parametrization of the model, cost functions, and constraints are explained below:

- As for the parameterization of the MPC model, $f'_{\boldsymbol{\theta}_\mathrm{m}}(\hat{\mathbf{s}}'_i, \hat{\mathbf{a}}_i, \bar{\mathbf{d}}_i)$ uses the same parametrization in (5.27b) as in the simplified model (5.26), and we use the fitted model parameters $\boldsymbol{\theta}_\mathrm{mf}$ as the initial values for learning. One noteworthy point is that we use $\bar{\mathbf{d}} = \{\bar{P}_\mathrm{rad}, \bar{P}_\mathrm{app}, \bar{T}_\mathrm{out}\}$ in the MPC formulation instead of the uncertain values $\mathbf{d}$ to mimic the prediction errors (i.e, system uncertainties) that would be encountered in realistic scenarios.

- As for the parameterization of the cost functions in (5.27a), the parameterized stage cost is designed as

$$l_{\boldsymbol{\theta}}(\hat{\mathbf{s}}_i, \hat{\mathbf{a}}_i) = \underbrace{\theta_{\mathrm{in1}}(\hat{T}_\mathrm{in} - 20 - \theta_{\mathrm{in2}})(\hat{T}_\mathrm{in} - 24 - \theta_{\mathrm{in3}})}_{l^\theta_\mathrm{temp}} + \underbrace{[B\hat{P}_\mathrm{buy} - S\hat{P}_\mathrm{sell}]}_{l^\theta_\mathrm{spot}}, \tag{5.28}$$

  and the parameterized terminal cost is designed as

$$T_{\boldsymbol{\theta}}(\hat{\mathbf{s}}_n) = \theta_{\mathrm{en1}}(\hat{E}_n - \theta_{\mathrm{en2}})^2. \tag{5.29}$$

  Because of the mismatch between the MPC horizon (n) and the RL horizon (N), parameterization of the stage cost function and the extra terminal cost function are essential.

- As for the parameterization of the constraints, we put θ_{t2} and θ_{e} into the state constraints (5.27c)-(5.27f).

Some remarks are listed below regarding the parametric MPC:

- The MPC horizon n can be different from the RL horizon N. To speed up the computation, we usually choose $n << N$. As a result, the terminal cost function of the MPC is critical because it needs to compensate for the "short-sightedness" of the MPC to some extent.
- Due to the differences between the MPC and RL formulations (e.g., mismatches between N and n, $\mathbf{s}$ and $\mathbf{s}'$, ϵ and θ_{ϵ}, $\mathbf{d}$ and $\bar{\mathbf{d}}$), the solution generated by a non-parameterized MPC must be a severely suboptimal solution to (5.22). Therefore, it is sensible to use the parametric MPC (5.27) to compensate for these discrepancies so that the optimal policy can be captured.
- The parameterization could be in principle arbitrary, but its design is usually motivated by some consideration of the physical meaning of the problem and is hence often interpretable.

The approximated policy $\boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathbf{s})$ and action $\mathbf{a}$ are obtained as

$$\boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathbf{s}) = \mathbf{u}_0^{\star}(\mathbf{s}, \boldsymbol{\theta}), \tag{5.30a}$$

$$\mathbf{a} = \boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathbf{s}) + {k_{\varrho}}^{\kappa}\boldsymbol{\varrho}, \tag{5.30b}$$

where $\mathbf{u}_0^{\star}(\mathbf{s}, \boldsymbol{\theta})$ is the first element of the MPC input solution and $\boldsymbol{\varrho}$ is a Gaussian term that adds weighted explorations onto the actions. In the learning, we adopt a decayed exploration by multiplying the exploration by a decay coefficient k_{ϱ} (less than one), which exponentially decreases according to the policy updating times κ. Overall, the parameters vector $\boldsymbol{\theta} \in \mathbb{R}^{15}$ consists of

$$\begin{aligned}\boldsymbol{\theta} = \{&\theta_{\mathrm{m1}}, \theta_{\mathrm{m2}}, \theta_{\mathrm{m3}}, \theta_{\mathrm{m4}}, \theta_{\mathrm{m5}}, \theta_{\mathrm{m6}}, \theta_{\epsilon 2}, \theta_{\epsilon 3}, \theta_{\mathrm{in1}}, \theta_{\mathrm{in2}}, \\ &\theta_{\mathrm{in3}}, \theta_{\mathrm{en1}}, \theta_{\mathrm{en2}}, \theta_{\mathrm{t2}}, \theta_{\mathrm{e}}\}.\end{aligned} \tag{5.31}$$

With an appropriate choice of $\boldsymbol{\theta}$, the short-horizon, deterministic MPC scheme (5.27) can yield the optimal policy $\boldsymbol{\pi}^{\star}(\mathbf{s}, \boldsymbol{\theta}^{\star})$ to the long-horizon, stochastic problem (5.22).

5.4.2 Compatible Delayed Deterministic Actor-Critic

In this section, we present an updating algorithm called CDDAC-GQ to adjust the parameters (5.31) of the MPC scheme (5.27). CDDAC-GQ is derived based on the DPG method elaborated in Section 2.4.2: *"Core Formulas: DPG for MPC-based RL"*. The policy gradient $\nabla_{\boldsymbol{\theta}}$ (a.k.a $\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\pi}_{\boldsymbol{\theta}})$) is computed using the same formulas as in Section 2.4.2, except that the variables are replaced with those defined in this microgrid energy management problem. Specifically, $\boldsymbol{\zeta}$ in (2.17) has the form $\boldsymbol{\zeta} = \{\hat{\mathbf{s}}', \hat{\mathbf{a}}, \boldsymbol{\sigma}\}$, which is the primal decision variable of the MPC (5.27). And in (2.18), $\Omega_{\boldsymbol{\theta}}$ now represents the MPC cost (5.27a), $\mathbf{G}_{\boldsymbol{\theta}}$ gathers the equality constraints and $\mathbf{H}_{\boldsymbol{\theta}}$ collects the inequality constraints of the MPC (5.27).

The critic parameters $\mathbf{w}, \mathbf{v}$ in (2.20), the solution to the Least Squares problem (2.23), obtained by updating using the gradient Q-learning method as described in Equation (2.25) in Section 2.4.2. Besides the gradient Q-learning technique, in CDDAC-GQ, the actor parameters $\boldsymbol{\theta}$ are designed to update less frequently (*delayed*) than the critic, and it employs the ADAptive Moment estimation (ADAM) optimizer. See Algorithm 4 for details. Compared to the Least Squares Temporal Difference based-Deterministic Policy Gradient (LSTD-DPG) used in the previous sections, the CDDAC-GQ is superior in the following aspects:

- Compared to the LSTD-DPG which uses an on-policy manner, the CDDAC-GQ uses off-policy Q-learning to update the critic, which significantly increases the data efficiency.

- Concerning that off-policy Q-learning may diverge with linear function approximators, the gradient Q-learning critic is adopted. With this technique, the critic parameters are updated towards the true gradient descent and convergence is thus ensured [59].

- We adopt the "delayed" policy updates in CDDAC-GQ, i.e., one policy update for every several Q and V-functions updates, which smooths the learning.

- Rather than the Stochastic Gradient Descent (SGD) used in LSTD-DPG, we update the actor parameters via ADAM in CDDAC-GQ. ADAM assigns a learning rate for each parameter and the rate is adapted based on the average of recent magnitudes of gradients [60]. This improves the efficiency and smoothness of the policy learning process.

5.5 Simulation

This section presents the simulation results of our proposed MPC-based RL approach, assessing the performance of the CDDAC-GQ algorithm. Besides, we provide a comprehensive comparative analysis between CDDAC-GQ and the recent RL algorithm TD3 in terms of economic efficiency and computational cost.

5.5.1 Results of the MPC-based RL Approach (CDDAC-GQ)

The parameters used in the MPC scheme and RL updating of CDDAC-GQ are given in Table 5.6. It can be seen that $N = 96$, i.e., the goal of RL is to minimize the combined cost (net energy cost and temperature comfort considerations) of the user for a day. The values of the action exploration $\boldsymbol{\varrho}$, exploration decay coefficient k_{ϱ}, and learning step sizes $\alpha_w, \alpha_v, \alpha_\nu, \alpha_\theta$ are chosen empirically after several trials. The initial values of MPC parameters $\boldsymbol{\theta}_0$ are composed of the fitted model parameters $\boldsymbol{\theta}_{\mathrm{mf}}$, the cost function parameter $\boldsymbol{\theta}_{\mathrm{ct}}$, as well as the constraint parameter $\boldsymbol{\theta}_{\mathrm{cs}}$ (i.e., $\boldsymbol{\theta}_0 = \{\boldsymbol{\theta}_{\mathrm{mf}}, \boldsymbol{\theta}_{\mathrm{ct}}, \boldsymbol{\theta}_{\mathrm{cs}}\}$), see Table 5.5. Although any initial value in the defined domain should be theoretically allowed, the initial values of $\boldsymbol{\theta}_{\mathrm{ct}}$ and $\boldsymbol{\theta}_{\mathrm{cs}}$ are chosen based on certain considerations. For example, the initial value of θ_{in1} is chosen as 5 to be consistent with the coefficient in the RL equation, and θ_{en2} is initialized as 3 to be in the middle of the battery charge.

Algorithm 4: Compatible Delayed DAC with Gradient Q-Learning Critic

Input: initialize policy parameters $\boldsymbol{\theta}$, ADAM parameters $\mathbf{m}, \mathbf{n}$, Q-function parameters $\mathbf{w}$, V-function parameters $\mathbf{v}$, greedy-GQ parameters $\boldsymbol{\nu}$, replay buffer $\mathcal{D}$, number of policy updates κ

repeat
  Randomly initialize starting state $\mathbf{s}$
  for $t = 1, \ldots, T$ in episode **do**
    Observe state $\mathbf{s}$ and select action $\mathbf{a}$ by (5.30b)
    Execute $\mathbf{a}$ in the environment
    Observe next state $\mathbf{s}^\dagger$, reward r, and additional information d
    Store transition $(\mathbf{s}, \mathbf{a}, r, \mathbf{s}^\dagger, d)$ in the replay buffer $\mathcal{D}$
    for $j = 1, \ldots, n_{\text{train}}$ **do**
      Randomly sample a batch of transitions, $B = \{(\mathbf{s}, \mathbf{a}, r, \mathbf{s}^\dagger, d)\}$ from $\mathcal{D}$
      Compute temporal difference errors

$$\delta = r + \gamma Q_{\mathbf{w}}(\mathbf{s}^\dagger, \boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathbf{s}^\dagger))) - Q_{\mathbf{w}}(\mathbf{s}, \mathbf{a}))$$

      Update Q-function and V-function using gradient Q-learning

$$\mathbf{w} \leftarrow \mathbf{w} + \alpha_w \frac{1}{|B|} \sum_B \left[\delta \boldsymbol{\psi}(\mathbf{s}, \mathbf{a}) - \gamma \boldsymbol{\psi}(\mathbf{s}^\dagger, \boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathbf{s}^\dagger))(\boldsymbol{\psi}(\mathbf{s}, \mathbf{a})^\top \boldsymbol{\nu})\right]$$

$$\mathbf{v} \leftarrow \mathbf{v} + \alpha_v \frac{1}{|B|} \sum_B \left[\delta \boldsymbol{\phi}(\mathbf{s}) - \gamma \boldsymbol{\phi}(\mathbf{s}^\dagger)(\boldsymbol{\psi}(\mathbf{s}, \mathbf{a})^\top \boldsymbol{\nu})\right]$$

$$\boldsymbol{\nu} \leftarrow \boldsymbol{\nu} + \alpha_\nu \frac{1}{|B|} \sum_B \left[(\delta - \boldsymbol{\psi}(\mathbf{s}, \mathbf{a})^\top \boldsymbol{\nu})\boldsymbol{\psi}(\mathbf{s}, \mathbf{a})\right]$$

      if $j \mod \text{policy delay} = 0$ **then**
        Compute policy gradient

$$\nabla_{\boldsymbol{\theta}} = \frac{1}{|B|} \sum_B \left[\nabla \boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathbf{s}) \nabla \boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathbf{s})^\top \mathbf{w}\right]$$

        Update policy parameters via ADAM

$$\mathbf{m} \leftarrow \beta_m \mathbf{m} + (1 - \beta_m)\nabla_{\boldsymbol{\theta}}, \quad \hat{\mathbf{m}} = \frac{\mathbf{m}}{1 - \beta_m^\kappa}$$

$$\mathbf{n} \leftarrow \beta_n \mathbf{n} + (1 - \beta_n)(\nabla_{\boldsymbol{\theta}})^2, \quad \hat{\mathbf{n}} = \frac{\mathbf{n}}{1 - \beta_n^\kappa}$$

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \alpha_\theta \frac{\hat{\mathbf{m}}}{\sqrt{\hat{\mathbf{n}}} + \varepsilon}, \quad \kappa = \kappa + 1$$

      end
    end
    One step forward $\mathbf{s} \leftarrow \mathbf{s}^\dagger$
  end
until *convergence*;

By applying the CDDAC-GQ update rule outlined in Algorithm 4, the MPC parameters are effectively learned, achieving convergence at approximately 1000 steps. The variation of its changing from $\boldsymbol{\theta}_0$ to $\boldsymbol{\theta}^\star$ is presented in Fig. 5.4, and the corresponding RL performance $J(\boldsymbol{\pi}_{\boldsymbol{\theta}})$ decreases consequently as presented in Fig. 5.5. (For better comparison with TD3 in the later section, the Y-axis (performance) is on a logarithmic scale.) Note that the results are averaged across five independent experiments for reliability, and the shaded areas are the 95% confidence intervals. It can be observed that even though the uncertainties, initial values, and explorations are random for each trial, the parameters eventually converge to similar values, and the optimal values of the obtained performance $J(\boldsymbol{\pi}_{\boldsymbol{\theta}})$ are almost identical. This consistent convergence signifies the robustness of our proposed MPC-based RL method in varying situations. Thus, it holds potential for real-world HEMS where such variations and uncertainties are commonplace. Another important remark is that identifying an explicit optimal policy is actually a formidable task due to inherent model inaccuracies and system uncertainties. In theory, our MPC-based RL approach can yield the optimal policy even in the face of these complexities [43]. Yet, the practical application may not always reflect this theoretical promise due to limited parameters and other learning-related limitations. Nonetheless, as shown in Fig. 5 (the economic cost before and after training), our approach significantly reduces the cost, and the convergence consistency of the five random experiments shows that our approach always targets the (sub)optimal policy-which is important in cases where the truly optimal policy cannot be determined due to intrinsic uncertainties. For real-world implementations of the MPC-based RL approach, striving for closer alignment with the optimal policy would necessitate richer MPC parameterizations and more effective RL training, which involves a balance between training complexity and economic benefits.

In Fig. 5.6, we show the 24-hour accumulated costs obtained using the parameterized MPC before and after the training. The blue shaded parts represent the spot-market net energy costs $\sum_{24\mathrm{h}} l_{\mathrm{spot}}$, the orange shaded parts represent the temperature penalty costs $\sum_{24\mathrm{h}} l_{\mathrm{temp}}$, and the red dashed lines are the sum of the two, i.e, $\sum_{24\mathrm{h}} l = \sum_{24\mathrm{h}} (l_{\mathrm{spot}} + l_{\mathrm{temp}})$.

It can be seen that before training, although the energy cost is almost zero, the temperature penalty cost is very large, resulting in a large combined cost. While after training, although the energy cost increases, the temperature cost decreases significantly and the total cost is greatly reduced compared to the previous case. The reduced temperature penalty cost after training provides quantifiable evidence that the algorithm is effective in learning from the environment and adapting its policy to achieve a more balanced trade-off between energy savings and user comfort.

Furthermore, we show in Fig. 5.7 and Fig. 5.8 the state solution and input solution of the parameterized MPC before and after training using the proposed CDDAC-GQ algorithm. As can be observed from Fig. 5.7, the indoor temperature T_{in} in the $\boldsymbol{\theta}_0$ case drops almost continuously and is severely out of the comfort range (shaded area in the figure), while the indoor temperature with $\boldsymbol{\theta}^\star$ is within the comfort range most of the time. There are two reasons why the control performance of the untrained MPC is so poor. The first and main reason is that the MPC model is inaccurate, so the input solution derived from this faulty model is invalid; the second reason is that the MPC horizon we used is 6, which is much shorter than the actual length of the problem 96 $(24\mathrm{h})$. Thus, the solution yielded by the short-sighted MPC is inevitably much less effective in dealing with the true problem. On the other hand, the well-trained MPC can compensate for these deficiencies, since the update of parameters is driven by the objective of the true RL problem. It can be seen that the trained MPC buys a large amount of electricity to supply the heat pump to raise the indoor temperature during the lowest electricity price period of the day $(0\mathrm{h} - 5\mathrm{h})$, which exploits the thermal inertia of the air to store energy for the high price period later in the day. This reflects a sophisticated learning capability of our approach, which not only reacts to the environment, but also predicts future states and acts accordingly. This predictive ability demonstrates the predictive power inherent in the MPC framework when fine-tuned through RL. Besides, the observations made in the state and input solutions emphasize the importance of a well-parameterized MPC. The policy of the untrained MPC is greatly improved upon training, emphasizing the role of RL in aligning the MPC

objective more closely with the true system dynamics and constraints.

Table 5.6: CDDAC-GQ: MPC and learning parameters.

$\Delta t, T$	$15\text{min}, 96$
N, n	96, 6
$c_{\text{low}}, c_{\text{hig}}$	$5, 5$
n_{train}	5
batch size	500
γ	0.99
policy delay	5
$\beta_m, \beta_n, \varepsilon$	$0.9, 0.999, 10^{-8}$
$\boldsymbol{\omega}_l, \boldsymbol{\omega}_n$	$[1, 1]$
$\boldsymbol{\varrho}$	$[1, 1, 5, 5, 3, 1] \times \mathcal{N}(0, 6.4 \times 10^{-4})$
k_ϱ	0.9985
$\alpha_w, \alpha_v, \alpha_\nu, \alpha_\theta$	$2 \times 10^{-5}, 2 \times 10^{-5}, 2 \times 10^{-5}, 10^{-4}$

Table 5.7: TD3: learning parameters.

Networks	$2Q, 2Q_{\text{targ}}, \Pi, \Pi_{\text{targ}}$
Hidden sizes	$[256, 256, 256]$
Q-nets activation functions	LeakyReLU, Identity
Π-nets activation functions	ReLU, Tanh
Q, Π-nets learning rate	1e-6
Polyak	0.995
Policy delay	4
Initial steps with random policy	19200
Exploration noise	$[1, 3, 1] \times \mathcal{N}(0, 2.25 \times 10^{-4})$
Smoothing noise added to Π_{targ}	$[1, 3, 1] \times \mathcal{N}(0, 2.25 \times 10^{-4})$
Total learning steps	1.152×10^6

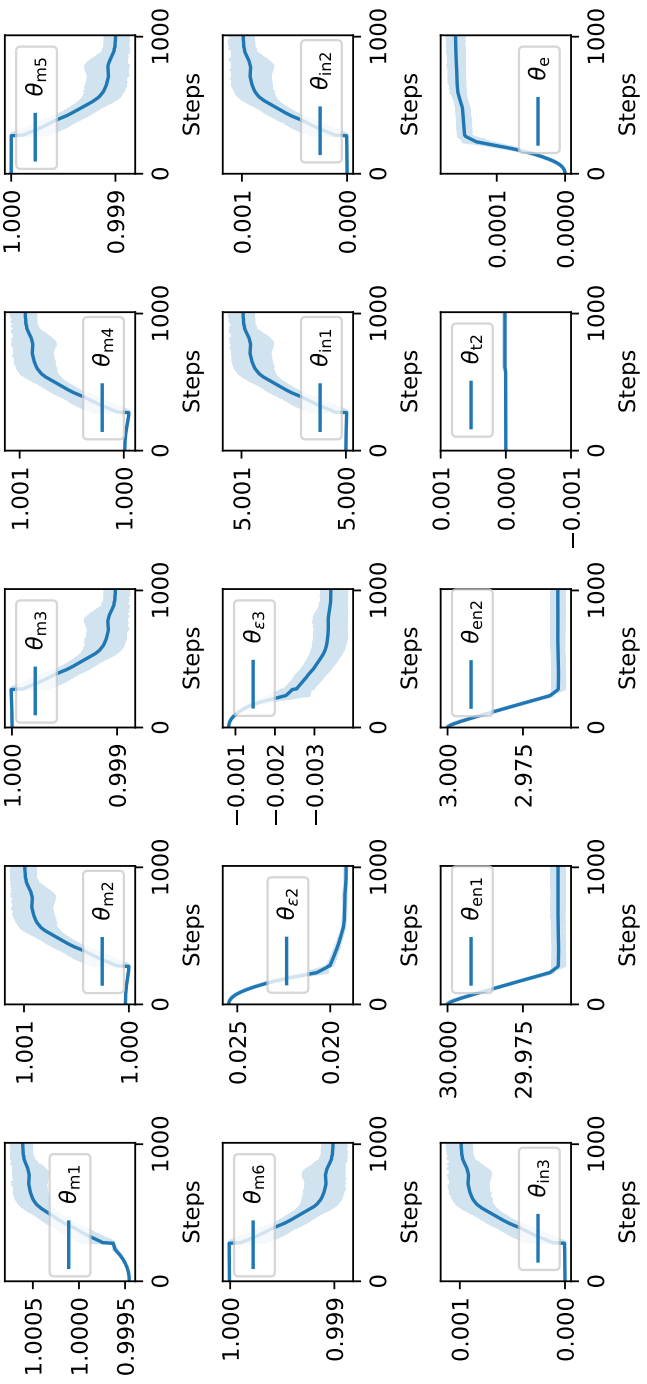

Figure 5.4: CDDAC-GQ: Variations of $\boldsymbol{\theta}$ over learning steps averaged across five experiments (from $\boldsymbol{\theta}_0$ to $\boldsymbol{\theta}^\star$). The solid line is the mean and the shaded area indicates the 95% confidence interval.

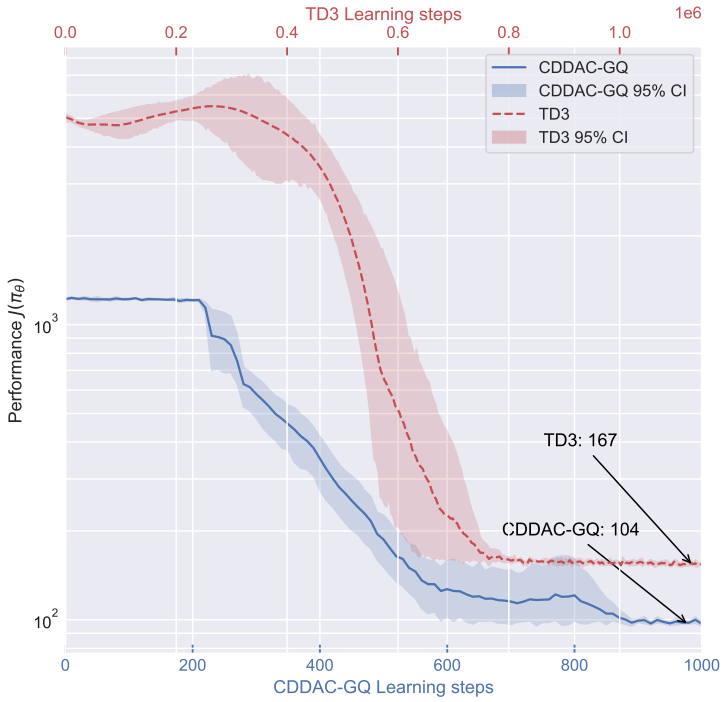

Figure 5.5: Comparative performance evaluation of $J(\boldsymbol{\pi_\theta})$ for CDDAC-GQ and TD3 algorithms over learning steps. The solid line indicates the mean performance for CDDAC-GQ and the dashed line for TD3, with the shaded areas representing their respective 95% confidence intervals, averaged across five experiments.

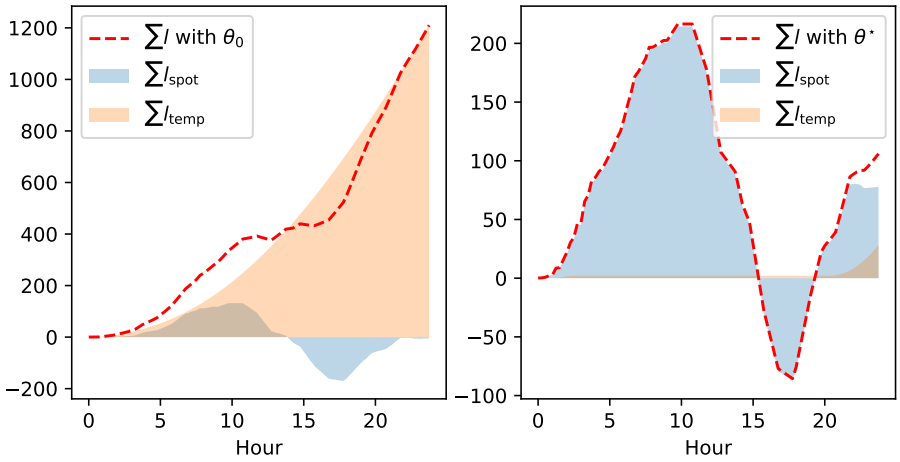

Figure 5.6: CDDAC-GQ: Accumulated costs using the parameterized MPC-scheme before (with $\boldsymbol{\theta}_0$) and after (with $\boldsymbol{\theta}^\star$) the training. The blue shaded parts represent the spot-market net energy costs $\sum_{24\mathrm{h}} l_{\mathrm{spot}}$, the orange shaded parts represent the temperature penalty costs $\sum_{24\mathrm{h}} l_{\mathrm{temp}}$, and the red dash lines are the combined costs $\sum_{24\mathrm{h}} l$.

5.5.2 Comparison with TD3

To fairly assess our proposed MPC-based RL approach, we apply the RL algorithm TD3 to address the same HEMS problem characterized by inaccurate model and system uncertainties. (We initially attempted to use the DDPG algorithm, but DDPG failed to converge effectively due to the significant uncertainties of this problem.)

To circumvent the curse of dimensionality, we follow the approach of [42], which uses time as a state variable instead of the full sequence of prediction information. This would not only avoid the dimensionality issue but also incorporate predictive information to some extent. Unlike the MPC framework which cannot handle conditional statements, we can use fewer action variables in TD3, with single variables representing charging/discharging battery and buying/selling power, denoted as $P_{\mathrm{ch/dis}}$ and $P_{\mathrm{buy/sell}}$, respectively. However, RL methods could not explicitly enforce system constraints like the MPC-based RL method.

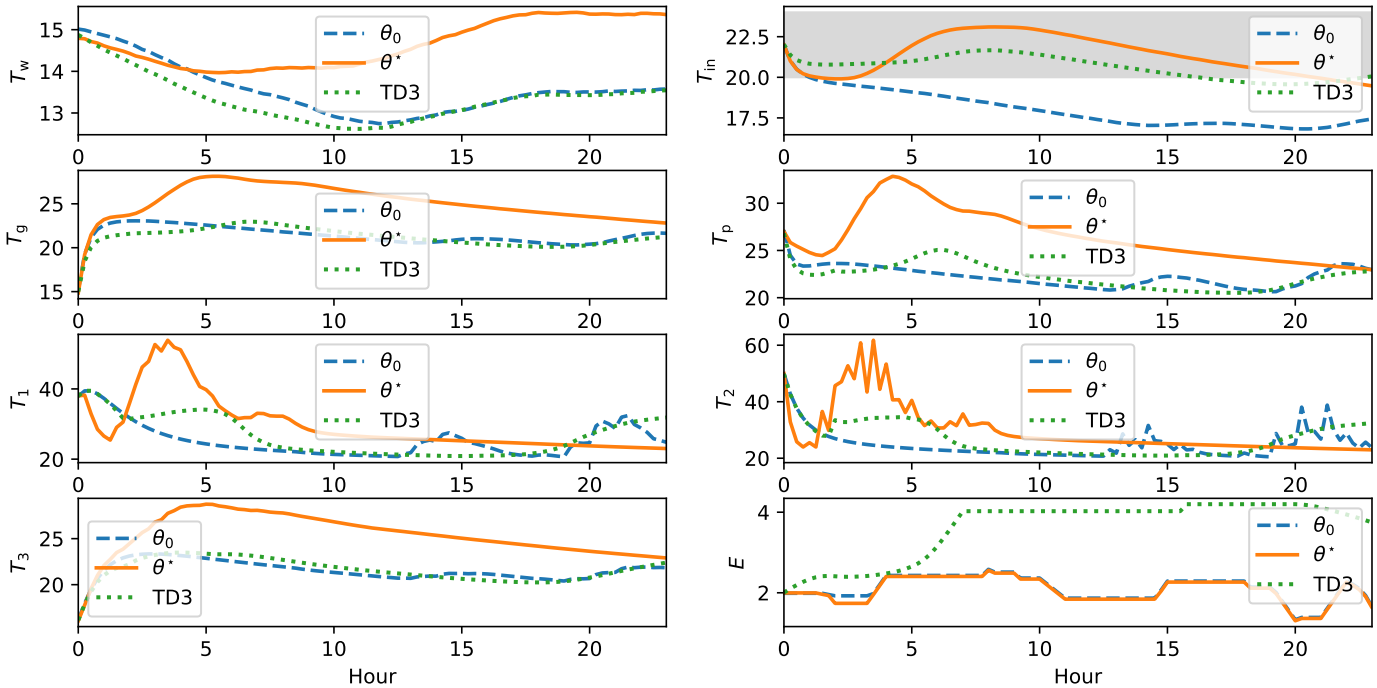

Figure 5.7: Comparative state solutions of CDDAC-GQ and TD3. The figure includes the state solutions of the parameterized MPC scheme generated by CDDAC-GQ before (with $\boldsymbol{\theta}_0$) and after (with $\boldsymbol{\theta}^\star$) training, as well as the solution of the TD3 algorithm. The shaded area in subfigure T_{in} indicates the comfort temperature range $[20°\text{C}, 24°\text{C}]$.

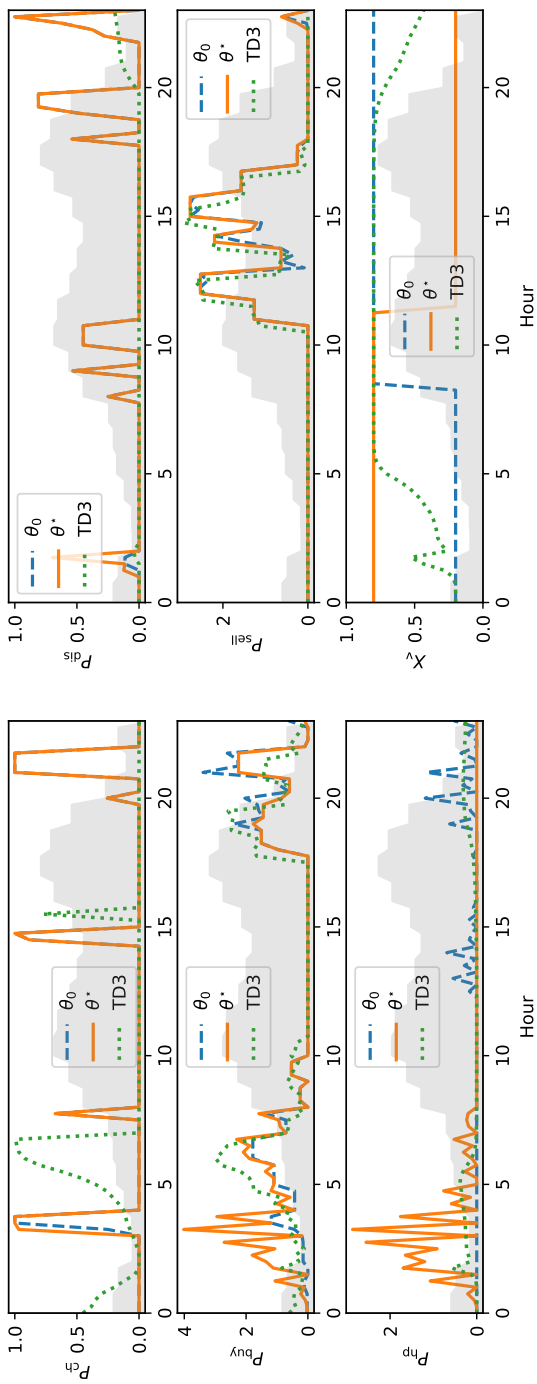

Figure 5.8: Comparative input solutions of CDDAC-GQ and TD3. The figure includes the input solutions of the parameterized MPC scheme generated by CDDAC-GQ before (using $\boldsymbol{\theta}_0$) and after (using $\boldsymbol{\theta}^\star$) training, as well as the solution of the TD3 algorithm. The shaded areas indicate the spot-market electricity buying prices $B(t)$ obtained from the Nord Pool.

They could only implicitly enforce constraints through penalties, making it challenging to strictly adhere to the power balance constraint (5.16)- a mandatory system constraint. To address this, we eliminate the $P_{\text{buy/sell}}$ variable and algebraically express it using the power balance constraint (5.16). This ensures that (5.16) is forcibly met, with penalties only applied for $P_{\text{buy/sell}}$. Consequently, the final action variables are defined as $\mathbf{a} = [P_{\text{ch/dis}}, P_{\text{hp}}, X_{\text{v}}]$. Furthermore, the constraint of battery state E faces a similar issue due to its physical limit, which has to be strictly followed. We therefore implement a "forced action" trick where if the new $P_{\text{ch/dis}}$ would cause the next state E to exceed the range, we would force $P_{\text{ch/dis}} = 0$. There are additional minor technical details, such as considering soft constraints along with hard constraints for a fair comparison with CDDAC-GQ, allowing some slack for soft constraints, and normalizing state and action variables, which are not elaborated here due to space limitations. In essence, from a design perspective, TD3 struggles to incorporate prediction information and handle constraints effectively, particularly when dealing with hard constraints that involve multiple action variables. This requires designing some reasonable tricks based on expert experience, and it is not trivial.

TD3 employs two action-value networks (Q), two target action-value networks (Q_{targ}), a policy network (Π), and a target policy network (Π_{targ}), with other training parameters presented in Table 5.7. Like CDDAC-GQ, we conduct five random experiments, and the result is shown in Fig. 5.5. As can be seen, CDDAC-GQ demonstrates superior performance with a lower cost metric from the beginning of the learning process, as indicated by its consistently lower $J(\boldsymbol{\pi}_{\boldsymbol{\theta}})$ value. The graph illustrates that CDDAC-GQ not only starts stronger but also converges to a better performance value of approximately 104, compared to 167 for TD3, signifying around a 37.7% improvement in cost efficiency. Moreover, the CDDAC-GQ algorithm reaches this performance with 1000 learning steps, which is substantially fewer than the 1152000 steps required by TD3, highlighting the data efficiency of CDDAC-GQ. The 95% confidence intervals represented by the shaded areas further substantiate that CDDAC-GQ has slightly lower variance and slightly higher stability throughout the learning process.

The results yielded by the policy network trained via TD3 are presented in Fig. 5.7 and Fig. 5.8. It is observed that the policy network generated by TD3 has some awareness of future electricity price fluctuations, buying electricity during low-price periods and selling during high-price periods. However, there are two notable deficiencies in the TD3 behaviour compared to CDDAC-GQ: 1. The P_{hp} and T_{in} subplots reveal that TD3 does not fully exploit the thermal inertia of the house for "slightly excessive" heating during low electricity price intervals between $[0\mathrm{h}-7\mathrm{h}]$, resulting in the necessity for heating during the more expensive peak price period later on; 2. As seen from the P_{buy} and P_{sell} subgraphs, unlike CDDAC-GQ, which trades perfectly following price fluctuations, TD3 appears unable to capture the full spectrum of electricity price information, leading to a discrepancy between its trading actions and the price fluctuations. These shortcomings primarily stem from the limited incorporation of predictive information of TD3, only factoring in the time as a state, unlike CDDAC-GQ, which fully integrates predictive information. Consequently, it is within expectations that the performance of TD3 would be approximately 37% lower than that of CDDAC-GQ.

In the following discussion, we compare the computational performance of our proposed CDDAC-GQ algorithm with the TD3 algorithm. The machine used in the simulation is $2.3\mathrm{GHz}$ 8-Core Intel Core i9 @ $16\mathrm{GB}$ $2667\mathrm{MHz}$ DDR4. The proposed CDDAC-GQ algorithm requires approximately $5\mathrm{h}37\mathrm{min}10\mathrm{s}$ for the total training duration, averaging around $20.23\mathrm{s}$ per learning step over 1000 steps. In contrast, the TD3 algorithm completes training in about $2\mathrm{h}25\mathrm{min}5\mathrm{s}$, with an average time of $7.56\mathrm{ms}$ per learning step over 1152000 steps. While TD3 appears faster in terms of per-step computation, the developmental efficiency of the algorithms must be considered. Implementing and tuning TD3 involved substantial time and effort, with considerable trial-and-error in adjusting the reward function, step function, and hyperparameters. As for the real-time implementation of the proposed MPC-based RL approach, the time to solve an MPC instance in CDDAC-GQ is about $0.08\mathrm{s}$, significantly less than the $15\mathrm{min}$ sampling time, indicating its suitability for real-time applications.

In summary, although our proposed CDDAC-GQ algorithm requires a longer training time, this process can be entirely conducted offline.

It can effectively integrate a large sequence of predicted information and excels in adhering to system constraints, thus significantly reducing development and design efforts. Most importantly, the policy derived from CDDAC-GQ shows a performance improvement of approximately 37.7% over TD3, demonstrating its significant advantages in solving complex HEMS problems.

5.6 Conclusion

In this paper, we propose an MPC-based RL approach to solve the HEMS problem. We show that data-driven model fitting may not completely eliminate model errors, and the MPC-based RL approach is precisely a powerful tool for tackling the HEMS problem with model errors and system uncertainties. By parameterizing the model, cost function, and constraints of MPC and training the parameters by RL, an optimal policy that satisfies both economy and comfort is finally obtained. Compared to the conventional TD3 algorithm, the proposed approach has the advantages of higher sampling efficiency, easier incorporation of predictive information, easier realization of constraints, higher robustness, and better cost-effectiveness. However, it should be acknowledged that this work still uses a simulated environment and that the uncertainties are ideally Gaussian distributed. In future work, we will try to further validate the effectiveness of the proposed method in a real system.

References

[1] D Mariano-Hernández, L Hernández-Callejo, A Zorita-Lamadrid, O Duque-Pérez, and F Santos Garc´ıa. “A review of strategies for building energy management system: Model predictive control, demand side management, optimization, and fault detect & diagnosis”. In: *Journal of Building Engineering* 33 (2021), p. 101692.

[2] Bandana Mahapatra and Anand Nayyar. “Home energy management system (HEMS): Concept, architecture, infrastructure, challenges and energy management schemes”. In: *Energy Systems* 13.3 (2022), pp. 643–669.

[3] Karl Mason and Santiago Grijalva. “A review of reinforcement learning for autonomous building energy management”. In: *Computers & Electrical Engineering* 78 (2019), pp. 300–312.

[4] Sereen Althaher, Pierluigi Mancarella, and Joseph Mutale. “Automated demand response from home energy management system under dynamic pricing and power and comfort constraints”. In: *IEEE Transactions on Smart Grid* 6.4 (2015), pp. 1874–1883.

[5] Qinglai Wei, Frank L Lewis, Guang Shi, and Ruizhuo Song. “Error-tolerant iterative adaptive dynamic programming for optimal renewable home energy scheduling and battery management”. In: *IEEE Transactions on Industrial Electronics* 64.12 (2017), pp. 9527–9537.

[6] Zhi Chen and Lei Wu. “Residential appliance DR energy management with electric privacy protection by online stochastic optimization”. In: *IEEE Transactions on Smart Grid* 4.4 (2013), pp. 1861–1869.

[7] Joaquim Leitao, Paulo Gil, Bernardete Ribeiro, and Alberto Cardoso. “A survey on home energy management”. In: *IEEE Access* 8 (2020), pp. 5699–5722.

[8] Bandana Mahapatra and Anand Nayyar. “Home energy management system (HEMS): Concept, architecture, infrastructure, challenges and energy management schemes”. In: *Energy Systems* (2019), pp. 1–27.

[9] Hicham Johra and Per Heiselberg. “Influence of internal thermal mass on the indoor thermal dynamics and integration of phase change materials in furniture for building energy storage: A review”. In: *Renewable and Sustainable Energy Reviews* 69 (2017), pp. 19–32.

[10] Mojtaba Yousefi, Amin Hajizadeh, and Mohsen Nourbakhsh Soltani. “A comparison study on stochastic modeling methods for home energy management systems”. In: *IEEE Transactions on Industrial Informatics* 15.8 (2019), pp. 4799–4808.

[11] Omar Alrumayh and Kankar Bhattacharya. “Model predictive control based home energy management system in smart grid”. In: *2015 IEEE Electrical Power and Energy Conference (EPEC)*. IEEE. 2015, pp. 152–157.

[12] Radu Godina, Eduardo MG Rodrigues, Edris Pouresmaeil, João CO Matias, and João PS Catalão. “Model predictive control home energy management and optimization strategy with demand response”. In: *Applied Sciences* 8.3 (2018), p. 408.

[13] Gianluca Serale, Massimo Fiorentini, Alfonso Capozzoli, Daniele Bernardini, and Alberto Bemporad. “Model predictive control (MPC) for enhancing building and HVAC system energy efficiency: Problem formulation, applications and opportunities”. In: *Energies* 11.3 (2018), p. 631.

[14] Shiyu Yang, Man Pun Wan, Wanyu Chen, Bing Feng Ng, and Deqing Zhai. “An adaptive robust model predictive control for indoor climate optimization and uncertainties handling in buildings”. In: *Building and Environment* 163 (2019), p. 106326.

[15] Yan Zhang, Lijun Fu, Wanlu Zhu, Xianqiang Bao, and Cang Liu. “Robust model predictive control for optimal energy management of island microgrids with uncertainties”. In: *Energy* 164 (2018), pp. 1229–1241.

[16] Seyed Mohsen Hosseini, Raffaele Carli, and Mariagrazia Dotoli. “A residential demand-side management strategy under nonlinear pricing based on robust model predictive control”. In: *2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)*. IEEE. 2019, pp. 3243–3248.

[17] Yushen Long, Shuai Liu, Lihua Xie, and Karl Henrik Johansson. “A scenario-based distributed stochastic MPC for building temperature regulation”. In: *2014 IEEE international conference*

on automation science and engineering (CASE). IEEE. 2014, pp. 1091–1096.

[18] Stefano Raimondi Cominesi, Marcello Farina, Luca Giulioni, Bruno Picasso, and Riccardo Scattolini. "A two-layer stochastic model predictive control scheme for microgrids". In: *IEEE Transactions on Control Systems Technology* 26.1 (2017), pp. 1–13.

[19] Dennis Van der Meer, Guang Chao Wang, and Joakim Munkhammar. "An alternative optimal strategy for stochastic model predictive control of a residential battery energy management system with solar photovoltaic". In: *Applied Energy* 283 (2021), p. 116289.

[20] Louis-Gabriel Maltais and Louis Gosselin. "Energy management of domestic hot water systems with model predictive control and demand forecast based on machine learning". In: *Energy Conversion and Management: X* 15 (2022), p. 100254.

[21] Abdul Afram, Farrokh Janabi-Sharifi, Alan S Fung, and Kaamran Raahemifar. "Artificial neural network (ANN) based model predictive control (MPC) and optimization of HVAC systems: A state of the art review and case study of a residential HVAC system". In: *Energy and Buildings* 141 (2017), pp. 96–113.

[22] Raffaele Soloperto, Matthias A Müller, Sebastian Trimpe, and Frank Allgöwer. "Learning-based robust model predictive control with state-dependent uncertainty". In: *IFAC-PapersOnLine* 51.20 (2018), pp. 442–447.

[23] Ján Drgoňa, Damien Picard, Michal Kvasnica, and Lieve Helsen. "Approximate model predictive building control via machine learning". In: *Applied Energy* 218 (2018), pp. 199–216.

[24] Yue Li and Zheming Tong. "Model predictive control strategy using encoder-decoder recurrent neural networks for smart control of thermal environment". In: *Journal of Building Engineering* 42 (2021), p. 103017.

[25] Frederik Ruelens et al. “Residential demand response of thermostatically controlled loads using batch reinforcement learning”. In: *IEEE Transactions on Smart Grid* 8.5 (2016), pp. 2149–2159.

[26] Frederik Ruelens, Sandro Iacovella, Bert J Claessens, and Ronnie Belmans. “Learning agent for a heat-pump thermostat with a set-back strategy using model-free reinforcement learning”. In: *Energies* 8.8 (2015), pp. 8300–8318.

[27] Renzhi Lu, Seung Ho Hong, and Mengmeng Yu. “Demand response for home energy management using reinforcement learning and artificial neural network”. In: *IEEE Transactions on Smart Grid* 10.6 (2019), pp. 6629–6639.

[28] Aya A Amer, Khaled Shaban, and Ahmed M Massoud. “DRL-HEMS: Deep Reinforcement Learning Agent for Demand Response in Home Energy Management Systems Considering Customers and Operators Perspectives”. In: *IEEE Transactions on Smart Grid* 14.1 (2022), pp. 239–250.

[29] Caomingzhe Si, Yuechuan Tao, Jing Qiu, Shuying Lai, and Junhua Zhao. “Deep reinforcement learning based home energy management system with devices operational dependencies”. In: *International Journal of Machine Learning and Cybernetics* 12.6 (2021), pp. 1687–1703.

[30] Michael Blonsky, Killian McKenna, Jeff Maguire, and Tyrone Vincent. “Home energy management under realistic and uncertain conditions: A comparison of heuristic, deterministic, and stochastic control methods”. In: *Applied Energy* 325 (2022), p. 119770.

[31] Tohid Sattarpour, Daryoush Nazarpour, and Sajjad Golshannavaz. “A multi-objective HEM strategy for smart home energy scheduling: A collaborative approach to support microgrid operation”. In: *Sustainable cities and society* 37 (2018), pp. 26–33.

[32] Chao Sun, Fengchun Sun, and Scott J Moura. “Nonlinear predictive energy management of residential buildings with photovoltaics & batteries”. In: *Journal of Power Sources* 325 (2016), pp. 723–731.

[33] Mohammad Ostadijafari, Anamika Dubey, Yang Liu, Jie Shi, and Nanpeng Yu. “Smart building energy management using nonlinear economic model predictive control”. In: *2019 IEEE Power & Energy Society General Meeting (PESGM)*. IEEE. 2019, pp. 1–5.

[34] Ján Drgoňa et al. “All you need to know about model predictive control for buildings”. In: *Annual Reviews in Control* 50 (2020), pp. 190–232.

[35] Karol Bot, Samira Santos, Inoussa Laouali, Antonio Ruano, and Maria da Graça Ruano. “Design of ensemble forecasting models for home energy management systems”. In: *Energies* 14.22 (2021), p. 7664.

[36] Yang Li, D. Mahinda Vilathgamuwa, Daniel E. Quevedo, Chih Feng Lee, and Changfu Zou. “Ensemble Nonlinear Model Predictive Control for Residential Solar Battery Energy Management”. In: *IEEE Transactions on Control Systems Technology* 31.5 (2023), pp. 2188–2200.

[37] Huiliang Zhang, Sayani Seal, Di Wu, François Bouffard, and Benoit Boulet. “Building Energy Management With Reinforcement Learning and Model Predictive Control: A Survey”. In: *IEEE Access* 10 (2022), pp. 27853–27862.

[38] Mario Zanon and Sébastien Gros. “Safe reinforcement learning using robust MPC”. In: *IEEE Transactions on Automatic Control* (2020).

[39] Paulo Lissa et al. “Deep reinforcement learning for home energy management system control”. In: *Energy and AI* 3 (2021), p. 100043.

[40] Zhe Wang and Tianzhen Hong. “Reinforcement learning for building controls: The opportunities and challenges”. In: *Applied Energy* 269 (2020), p. 115036.

[41] Zhiang Zhang and Khee Poh Lam. “Practical implementation and evaluation of deep reinforcement learning control for a radiant heating system”. In: *Proceedings of the 5th Conference on Systems for Built Environments*. 2018, pp. 148–157.

[42] Liang Yu et al. “Deep reinforcement learning for smart home energy management”. In: *IEEE Internet of Things Journal* 7.4 (2019), pp. 2751–2762.

[43] Sébastien Gros and Mario Zanon. “Data-driven economic nmpc using reinforcement learning”. In: *IEEE Transactions on Automatic Control* 65.2 (2019), pp. 636–648.

[44] Arash Bahari Kordabad, Wenqi Cai, and Sebastien Gros. “MPC-based reinforcement learning for economic problems with application to battery storage”. In: *2021 European Control Conference (ECC)*. IEEE. 2021, pp. 2573–2578.

[45] Sebastien Gros and Mario Zanon. “Learning for mpc with stability & safety guarantees”. In: *Automatica* 146 (2022), p. 110598.

[46] Shambhuraj Sawant and Sebastien Gros. “Bridging the gap between QP-based and MPC-based Reinforcement Learning”. In: *IFAC-PapersOnLine* 55.15 (2022), pp. 7–12.

[47] Sebastien Gros and Mario Zanon. “Economic MPC of Markov Decision Processes: Dissipativity in undiscounted infinite-horizon optimal control”. In: *Automatica* 146 (2022), p. 110602.

[48] Arash Bahari Kordabad, Hossein Nejatbakhsh Esfahani, Wenqi Cai, and Sebastien Gros. “Quasi-Newton Iteration in Deterministic Policy Gradient”. In: *arXiv preprint arXiv:2203.13854* (2022).

[49] Wenqi Cai, Arash B Kordabad, Hossein N Esfahani, Anastasios M Lekkas, and Sébastien Gros. “MPC-based reinforcement learning for a simplified freight mission of autonomous surface vehicles”. In: *2021 60th IEEE Conference on Decision and Control (CDC)*. IEEE. 2021, pp. 2990–2995.

[50] Arash Bahari Kordabad, Wenqi Cai, and Sebastien Gros. “Multi-agent battery storage management using MPC-based reinforcement learning”. In: *2021 IEEE Conference on Control Technology and Applications (CCTA)*. IEEE. 2021, pp. 57–62.

[51] Wenqi Cai, Arash Bahari Kordabad, and Sébastien Gros. "Energy management in residential microgrid using model predictive control-based reinforcement learning and Shapley value". In: *Engineering Applications of Artificial Intelligence* 119 (2023), p. 105793.

[52] Wenqi Cai, Hossein N Esfahani, Arash B Kordabad, and Sébastien Gros. "Optimal management of the peak power penalty for smart grids using MPC-based reinforcement learning". In: *2021 60th IEEE Conference on Decision and Control (CDC)*. IEEE. 2021, pp. 6365–6370.

[53] Soroush Rastegarpour, Luca Ferrarini, and Sebastien Gros. "Economic NMPC for multiple buildings connected to a heat pump and thermal and electrical storages". In: *IFAC-PapersOnLine* 53.2 (2020), pp. 17089–17094.

[54] Soroush Rastegarpour, Sebastien Gros, and Luca Ferrarini. "MPC approaches for modulating air-to-water heat pumps in radiant-floor buildings". In: *Control Engineering Practice* 95 (2020), p. 104209.

[55] Soroush Rastegarpour, Lorenzo Caseri, Luca Ferrarini, and Oliver Gehrke. "Experimental validation of the control-oriented model of heat pumps for MPC applications". In: *2019 IEEE 15th International Conference on Automation Science and Engineering (CASE)*. IEEE. 2019, pp. 1249–1255.

[56] Nurul Akmam Naamandadin, Chew Jian Ming, and Wan Azani Mustafa. "Relationship between Solar Irradiance and Power Generated by Photovoltaic Panel: Case Study at UniCITI Alam Campus, Padang Besar, Malaysia". In: *Journal of Advanced Research in Engineering Knowledge* 5.1 (2018), pp. 16–20.

[57] Joel A E Andersson, Joris Gillis, Greg Horn, James B Rawlings, and Moritz Diehl. "CasADi – A Software Framework for Nonlinear Optimization and Optimal Control". In: *Mathematical Programming Computation* 11.1 (2019), pp. 1–36. DOI: 10.1007/s12532-018-0139-4.

[58] Nord Pool Group. *Day-ahead power prices of Trondheim, Norway during November, 2020*. https://www.nordpoolgroup.com/Market-data1/Dayahead/Area-Prices/ALL1/Monthly/?view=table. 2020.

[59] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[60] Diederik P Kingma and Jimmy Ba. "Adam: A method for stochastic optimization". In: *arXiv preprint arXiv:1412.6980* (2014).

6 | Discussion

In this chapter, we will conclude the thesis by summing up some of its main contributions and a discussion on each contribution. In addition, we will suggest some future research directions for the topics.

6.1 Conclusion

This thesis has successfully showcased the versatility and efficacy of the MPC-based RL approach in addressing the complexities of stochastic systems across various applications. The research specifically targeted the limitations of traditional MPC and machine learning methods in dealing with inaccurate modeling and performance optimization. Through rigorous application in three diverse areas—ASV, residential microgrid energy management, and HEMS—the thesis has not only validated the theoretical principles of MPC-based RL but also demonstrated its practical utility in engineering solutions.

The work on ASVs emphasized improved closed-loop performance, particularly in tasks requiring precision, such as collision-free path tracking and autonomous docking. The research on residential microgrids effectively showcased the ability to manage energy distribution and optimize economic costs under stochastic conditions, introducing the Shapley value method for equitable bill distribution. In the domain of HEMS, the MPC-based RL approach proved robust in handling model inaccuracies and uncertainties, achieving a balance between economic viability and

comfort.

Taken together, these studies underscore the robustness of the MPC-based RL framework, setting a foundation for future advancements in control systems, especially in integrating MPC with RL. This approach opens new possibilities for addressing complex, uncertain, and dynamic systems in various engineering fields.

6.2 Limitations and Future Work

The limitations of this approach and the corresponding reflections are discussed below:

- In those cases where the MPCs to be solved are quite intractable, the proposed MPC-based RL approach may consume longer training time than standard RL approaches (those using NNs as function approximators). Fortunately, the training can be done offline. Besides, the parameterization of the MPC is quite flexible, allowing us to design the MPC cost from a numerical perspective that makes the MPC implementation as easy and effective as possible.

- In those cases with very short sampling times, the MPC-based RL method may struggle a bit in the real-time application. Indeed, the real-time burden comes mostly from solving the MPC (computing the sensitivity is inexpensive). Nonetheless, the progress in the optimization algorithms and in the computational hardware makes the deployment of real-time MPC possible for most of the real applications, as presented in [1].

- For more complex problems, the simple quadratic parameterization of the value function used in this thesis may no longer be sufficient to capture the true function. Alternatively, one could consider more general forms of value functions such as radial basis functions, Lagrangian polynomials, and neural networks, or one could even use MPC itself as a value function. Then updating the function

parameters in the MPC-based RL framework will become a bit more tricky, but it is still feasible.

The discussion of the limitations of this work has accordingly inspired our future works, summarized as follows:

- From the methodology point of view, it is interesting to consider how to reduce the time required for solving the MPC during training or even to avoid this process altogether. For example, our recent work [2] proposes a solution for efficient offline training without explicitly solving the MPC. Another heuristic solution is to perform “cost-matching" between the MPC formulation and the data based on some existing dataset. It is interesting to consider how to make the MPC-based policy approximator more general. For example, our recent work [3] uses convex neural networks as the cost functions for MPC. Furthermore, another promising research direction could delve into developing distributed MPC-based RL algorithms. This advancement aims to enhance scalability and efficiency for large-scale and complex systems, addressing the need for robust control strategies in expansive networked environments.
- From an application perspective, the focus can shift to applying MPC-based RL methods to a variety of industries, such as manufacturing or transportation, which offers exciting opportunities for broader application and impact. Experimental validation and real-world implementation, particularly in smart home applications we are currently investigating, are crucial next steps to validate the efficacy of this approach beyond simulated environments. Addressing the challenges of handling non-Gaussian uncertainties to enhance the robustness and applicability of the approach is also paramount. Lastly, scalability and the application of MPC-based RL to larger, more complex systems are critical areas for future exploration.

References

[1] Milan Vukov et al. “Real-time nonlinear MPC and MHE for a large-scale mechatronic application”. In: *Control Engineering Practice* 45 (2015), pp. 64–78.

[2] Shambhuraj Sawant, Akhil S Anand, Dirk Reinhardt, and Sebastien Gros. “Learning-based MPC from big data using reinforcement learning”. In: *arXiv preprint arXiv:2301.01667* (2023).

[3] Katrine Seel, Arash Bahari Kordabad, Sebastien Gros, and Jan Tommy Gravdahl. “Convex Neural Network-based Cost Modifications for Learning Model Predictive Control”. In: *IEEE Open Journal of Control Systems* (2022), pp. 1–14.

ISBN 978-82-326-8168-6 (printed ver.)
ISBN 978-82-326-8167-9 (electronic ver.)
ISSN 1503-8181 (printed ver.)
ISSN 2703-8084 (online ver.)

NTNU

Norwegian University of
Science and Technology