Doctoral theses at NTNU, 2023:82

Arash Bahari Kordabad

Theoretical Properties of Learning-based Model Predictive Control

Doctoral thesis

NTNU
Norwegian University of Science and Technology
Thesis for the Degree of
Philosophiae Doctor

NTNU
Norwegian University of
Science and Technology

Arash Bahari Kordabad

Theoretical Properties of Learning-based Model Predictive Control

Thesis for the Degree of Philosophiae Doctor

Trondheim, March 2023

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

ISBN 978-82-326-5698-1 (printed ver.)
ISBN 978-82-326-6983-7 (electronic ver.)
ISSN 1503-8181 (printed ver.)
ISSN 2703-8084 (online ver.)

ITK report number 2023-1-W

Doctoral theses at NTNU, 2023:82

Printed by NTNU Grafisk senter

Summary

Recently, the core idea of using Model Predictive Control (MPC) as a function approximator for the Reinforcement Learning (RL) methods has been proposed and justified. More specifically, it has been shown that a parameterized MPC scheme with a possibly inaccurate model can capture the optimal value functions and policy of a given Markov Decision Process (MDP).

The thesis investigates more on this idea and provides theorems supporting and developing this idea and answering some fundamental questions in the intersection of MDP, MPC, Moving Horizon Estimation (MHE), and RL based on the publications during the Ph.D.

We implement MPC-based RL in engineering applications such as Autonomous Surface Vehicle (ASV), including path planning, obstacle avoidance, and docking, and some investigations in the smart grid context, including learning the optimal bang-bang policy and multi-agent batteries with power peak constraint.

In the intersection of MDP and MPC, we provide a theory on the equivalence of optimality criteria for MPC and MDP. We show that an (undiscounted) MPC scheme can capture the optimal value and optimal policy of a (possibly discounted) MDP, even if an inaccurate model is used in the MPC scheme. This equivalence can be established using a proper selection of the stage cost and the terminal cost of an MPC scheme. This observation leads us to parameterize an MPC scheme fully, including the cost function. In practice, Reinforcement Learning algorithms can then be used to tune the parameterized MPC scheme. Using the cost modification idea, we also eliminate the bias of the optimal steady state in the discounted setting.

In the context of MDP and RL, we provide the Quasi-Newton technique with a novel approximated hessian of the performance function that yields a superlinear convergence in the learning using the policy gradient method. In addition, we characterize the stability of MDPs with discounted cost using Economic Model Predictive Control (EMPC) dissipativity theory in the measure space.

In the context of EMPC, we propose the use of Q-learning to capture a valid storage function that satisfies the dissipation inequality and verify the dissipativity for both discounted and undiscounted settings.

Robust Model Predictive Control (RMPC) is used for different purposes and forms. We address the bias issue in the MPC-based policy gradient method when a linear compatible advantage function approximator is used in the actor-critic. When hard constraints restrict the policy, the exploration may not be Centred or Isotropic (non-CI). As a result, the policy gradient estimation can be biased. We solve this issue using the RMPC approach accounting for the exploration based on the first-order Taylor approximation of the constraint-tightening. Moreover, we investigate using RL methods to adjust RMPC with ellipsoidal uncertainty set for stochastic nonlinear systems. Scenario-tree-based RMPC was implemented to handle potential failures of the ship thrusters and Q-learning was used to improve the closed-loop performance.

Moreover, we provide a generic convex function approximator in the stage cost of the MPC scheme and also address the safe RL problem using the Distributionally Robust Model Predictive Control (DRMPC) scheme and chance constraints.

Preface

This thesis is submitted in partial fulfilment of the requirements for the degree of philosophiae doctor (Ph.D.) at the Norwegian University of Science and Technology (NTNU), Trondheim.
The work presented has been conducted at the Department of Engineering Cybernetics at NTNU. My project supervisor is Professor Sebastien Gros, and my co-supervisor is Professor Anastasios Lekkas, both from the Department of Engineering Cybernetics.
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

I am very grateful for the support of my supervisors, who have helped me through the Ph.D. I would, first of all, like to thank my main supervisor Prof. Sebastien Gros. He has always been supportive and helpful almost 24/7 through Skype and sometimes personal meetings and discussions in his office, and our regular group meeting schedule has helped keep me focused on clear goals. Sebastien, it was a great pleasure and an honor to work with you; thank you for everything. Not only in the scientific aspects but also I learned too much about other life areas. This thesis only reflects part of what I have learned from you.

I would also like to thank my co-supervisor, Prof. Anastasios Lekkas, for his great help. Moreover, I got a bunch of great helps from Prof. Mario Zanon, Associate Professor at IMT School for Advanced Studies Lucca, Lucca, Italy. In addition, I spent eight months at Aalborg University, Aalborg, Denmark, as a visiting Ph.D., and got help and input from Prof. Rafael Wisniewski.

Although I have written only my name on the front page, this thesis represents anything

but a single person's work. I have been fortunate enough to collaborate with several people throughout this work. I would like to thank Hossein Nejatbakhsh Esfahani, Wenqi Cai, and Katrine Seel.

Finally, I would like to thank my parents, Ali and Maryam, for their support. I am ending this section with a special and warm thanks to my wife, Haniyeh Malektaj. Without her encouragement and support, I would probably not have even considered doing a Ph.D.

March 2023, Trondheim

Arash Bahari Kordabad

Contents

Abbreviations

AC Actor-Critic 19, 81

ASV Autonomous Surface Vehicle . . . . . iii, 1, 27, 33

CBF Control Barrier Function 191

CI Centred or Isotropic 19, 81

CSTR Continuously Stirred Tank Reactor 167

CVaR Conditional Value at Risk 25, 191

DNN Deep Neural Network 1, 33, 65, 113, 127, 167

DP Dynamic Programming 9, 33, 97, 167

DRMPC Distributionally Robust Model Predictive Control . . . . . iv, 26, 191

DRO Distributionally Robust Optimization 26, 191

EMPC Economic Nonlinear Model Predictive Control 22, 127

EMPC Economic Model Predictive Control . . . . . iii, 28, 113, 231

ESS Energy Storage Systems 65

FSDSD Functional Strong Discounted Strict Dissipativity 22, 113

KKT Karush–Kuhn–Tucker 49, 65

LICQ Linear Independence Constraint Qualification 49

LMI Linear Matrix Inequality 127

LP Linear Programming 191

LQR Linear Quadratic Regulator 97, 127, 167

LS Least Square 10, 65, 113, 127, 191

LSTD Least-Squares Temporal-Difference 16, 49, 65

MDP Partially Observable Markov Decision Process 16, 113

MDP Markov Decision Process . . . . . iii, 1, 7, 25, 33, 49, 65, 81, 97, 113, 167, 191

MHE Moving Horizon Estimation . . . . . iii, 16, 27

ML Machine Learning 2, 15

MPC Model Predictive Control . . . . . iii, iv, 1, 33, 49, 65, 81, 113, 127, 167, 191

NLP Nonlinear Programming 49, 81

NMPC Nonlinear Model Predictive Control 127

NN Neural Network 16, 127

OCP Optimal Control Problem 113, 167, 231

PV Photovoltaic 49, 65

RL Reinforcement Learning . . . . . iii, 1, 7, 33, 49, 65, 81, 97, 113, 127, 167, 191

RMPC Robust Model Predictive Control . . . . . iv, 20, 27, 33, 81

SAA Sample Average Approximation 26, 191

SDP Semi-Definite Program 127

SDSD Strong Discounted Strict Dissipativity 21, 113, 127, 231

SMPC Stochastic Model Predictive Control 191

SOC State of Charge 18, 49, 65

SOS Sum of Squares 127

SOSC Second Order Sufficient Condition 49

TD Temporal-Difference 10, 33, 49, 127

VaR Value at Risk 191

WMR Wheeled Mobile Robot 20, 191

1 Introduction

This chapter contains a brief motivation for the topics covered in this thesis, a summary of the main contributions, and an overview of the publications presented in the thesis. We finally provide an outline of the thesis.

1.1 Motivation

Reinforcement Learning (RL) has drawn increasing attention thanks to its striking accomplishments ranging from computers beating chess and Go masters [1]. Indeed, RL is a powerful tool for tackling Markov Decision Process (MDP) without prior knowledge of the process to be controlled. Most RL methods are based on learning the optimal policy and optimal value functions for the real system, described by an MDP, using a function approximator. The function approximator must be ensured that it is general enough that is able to capture the optimal policy or optimal value function of a given MDP. A common choice in the RL community is to use a Deep Neural Network (DNN). For instance, in [2] the baseline control is employed to ensure stability and tracking performance of an Autonomous Surface Vehicle (ASV), while DNN-based RL is added to handle uncertainties and collision avoidance.

Unfortunately, the closed-loop stability of a system with the optimal policy supported by a DNN or a generic function approximation can be difficult to formally analyze [3]. Moreover, providing meaningful initial weights for the DNN can be very difficult.

Model Predictive Control (MPC) is an optimization-based control approach operating with a receding horizon [4]. MPC employs a (possibly inaccurate) model of the real system dynamics to produce an input-state sequence over a given finite horizon. The resulting trajectory optimizes a given cost function while explicitly enforcing the system constraints. The optimization problem is solved at each time instance based on the current system state, and the first input of the optimal solution is applied to

the system. Due to the finite-horizon scheme and (possibly) model mismatch, MPC usually delivers a reasonable but suboptimal approximation of the optimal policy.

For computational reasons, simple models are usually preferred in the MPC scheme. Hence, the MPC model often does not have the structure required to correctly capture the real system dynamics and stochasticity. As a result, MPC usually delivers a reasonable but suboptimal approximation of the optimal policy.

Recently, the integration of Machine Learning (ML) in MPC has been investigated, with the aim of learning the model of the system, used in the MPC scheme, in a data-driven fashion [5]. While this paradigm has a clear value, it does not eliminate the issues related to model inaccuracies. Indeed, the performance of a policy delivered by an MPC scheme integrating an ML-based model is still only as good as the ML-based model is, and therefore limited by the structure and choices made in the ML tools.

Using MPC as a function approximator of a given MDP has been first proposed and justified in [6]. It was shown that a parameterized MPC is able to capture the optimal policy and optimal value function of a given MDP by modification of the stage cost and terminal cost even if a simple and inaccurate model is used in the MPC scheme. Then RL methods, such as Q-learning and policy gradient method can be used in order to adjust the parameters to achieve the best long-term closed-loop performance.

Considering that the central and primary theories related to this type of learning based on MPC have been recently published, there are still open questions in this field. Therefore, the current thesis has been proposed and carried out in order to cover some of the theoretical challenges. These challenges include some developments in the former work, applying the method for different engineering applications such as ASVs and smart grids, and showing the advantages of the method in solving challenging questions in the context of (E)MPC, RL, and MDP. Summaries of the developed theories are listed in the following of this chapter.

1.2 Publications

Taking into account the collaboration works around the topic, in total 19 papers (12 as the main author), including 13 conf. papers (9 as the main author) and 6 (3 as the main author) journal papers have been prepared during the Ph.D. carrier. Among them, 11 (9 as the main author) of the conf. papers and 4 (2 as the main author) of the journal papers have been published at the time of writing the thesis and the rests are under review. Given below is the list of all 19 publications in the order of submitted time starting from the earliest.

Conference publications

i **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.

ii Hossein Nejatbakhsh Esfahani, **Arash Bahari Kordabad**, and Sebastien Gros. “Reinforcement learning based on MPC/MHE for unmodeled and partially observable dynamics". In: *2021 American Control Conference (ACC)* (2021), pp.2121-2126.

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**, Hossein Nejatbakhsh Esfahani, and Sebastien Gros. “Bias Correction in Deterministic Policy Gradient Using Robust MPC". In: *2021 European Control Conference (ECC)* (2021), pp.1086-1091.

v Hossein Nejatbakhsh Esfahani, **Arash Bahari Kordabad**, and Sebastien Gros. “Approximate Robust NMPC using Reinforcement Learning". In: *2021 European Control Conference (ECC)* (2021), pp.132-137.

vi **Arash Bahari Kordabad** and Sebastien Gros. “Verification of Dissipativity and Evaluation of Storage Function in Economic Nonlinear MPC using Q-Learning". In: *IFAC-PapersOnLine*, vol.54, no.6, (2021), 7th IFAC Conference on Nonlinear Model Predictive Control NMPC, pp.308-313.

vii **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.

$viii$ 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.

ix 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.

[x] **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.

[xi] **Arash Bahari Kordabad**, Hossein Nejatbakhsh Esfahani, wenqi Cai, and Sebastien Gros. “Functional Stability of Discounted Markov Decision Processes Using Economic MPC Dissipativity Theory". In: *2022 European Control Conference (ECC)* (2022), pp.1858-1863.

[xii] **Arash Bahari Kordabad** and Sebastien Gros. “Bias correction of discounted optimal steady state using cost modification". *Submitted to a Conf.* (2023).

[xiii] **Arash Bahari Kordabad**, Dirk Reinhardt, Akhil S Anand, and Sebastien Gros. “Reinforcement Learning for MPC: Fundamentals and Current Challenges". *Submitted to a Conf.* (2023).

Journal publications

[xiv] **Arash Bahari Kordabad** and Sebastien Gros. “Q-learning of the storage function in Economic Nonlinear Model Predictive Control". In: *Engineering Applications of Artificial Intelligence*, vol.116, (2022), pp.105343.

[xv] 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.

[xvi] 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*, vol.1, (2022), pp.366-379.

[xvii] 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: *Submitted to a Journal*, (2022).

[xviii] **Arash Bahari Kordabad**, Mario Zanon, and Sebastien Gros. “Equivalence of Optimality Criteria for Markov Decision Process and Model Predictive Control". In: *arXiv preprint, Submitted to a Journal*, (2022).

[xix] **Arash Bahari Kordabad**, Rafal Wisniewski, and Sebastien Gros. “Safe Reinforcement Learning Using Wasserstein Distributionally Robust MPC and Chance Constraint". In: *IEEE access*, vol.10, (2022), pp.130058-130067.

In Chapter 5, the papers in which the candidate contributed as the first author have been collected. [1]

1.3 Contributions

A summary of contributions covered by the papers of the previous section is listed as follows [2]. These contributions are discussed in detail in Chapter 3.

- Learning MPC for the ASV applications.
  - Papers $\boxed{i}$ and $\boxed{ix}$.
- Learning MPC/MHE for the state estimation and control.
  - Papers $\boxed{ii}$ and $\boxed{xvii}$.
- Learning MPC for bang-bang policies, multi-agent battery storage, and peak power management.
  - Papers $\boxed{iii}$, $\boxed{vii}$, $\boxed{viii}$, and $\boxed{xv}$.
- Learning RMPC for bias correction of policy gradient and adjusting ellipsoidal uncertainty.
  - Papers $\boxed{iv}$ and $\boxed{v}$.
- Q-Learning of the storage function in EMPC schemes.
  - Papers $\boxed{vi}$ and $\boxed{xiv}$.
- Quasi-Newton iteration for deterministic policy gradient.
  - Paper $\boxed{x}$.
- Functional stability of discounted MDPs.
  - Paper $\boxed{xi}$.
- Bias correction of the optimal steady state in discounted OCP.

[1] Journal article $\boxed{xiv}$ has been published as an extension of conference article $\boxed{vi}$. Therefore, we do not include conference paper $\boxed{vi}$ in Chapter 5.

[2] All the papers are included in one of the categories, except paper $\boxed{xiii}$, which provides an overview and challenges on the learning MPC.

  - Paper [xii].

- Generic convex function approximator in the cost of learning MPC.
  - Paper [xvi].
- Optimality equivalency of MDP and MPC.
  - Paper [xviii].
- Probabilistic safe policy using Distributionally Robust MPC.
  - Paper [xix].

1.4 Outline

The rest of the thesis is structured as follows: Chapter 2 contains background on the topics covered in the publications. Chapter 3 gives an in-depth presentation of the contributions of the publications. Chapter 4 provides a summary and conclusion of the thesis and discusses some directions for future works. Finally, Chapter 5 contains the publications that were written as a result of the work on this thesis.

2 | Background

In this chapter, we first present some background on MDPs which is a core concept in the context of RL. MDP is a general description of the real system where its state transitions satisfy the Markov property. Next, we detail RL, as a practical and powerful technique to solve MDPs. We provide Q-learning and policy gradient methods in this section. Then we provide a background on MPC and the detail of the provided policy from an MPC scheme. Finally, we present the combination of MPC and RL and provide a fundamental theorem that recently has developed in order to detail the main scope of the current work.

2.1 Markov Decision Processes

Markov Decision Processes (MDPs) provide a standard framework for the optimal control of discrete-time stochastic processes, where the stage cost and transition probability depend only on the current state and the current input of the system. An MDP operates over given state and action (aka input) spaces S, A, respectively. These spaces can be discrete (i.e. integer sets), continuous, or mixed. We denoted ρ as a conditional probability (measure) defining the dynamics of the system considered, i.e. for a given state-action pair $\mathbf{s}, \mathbf{a} \in S \times A$, the successive state $\mathbf{s}_+$ is distributed according to

$$\mathbf{s}_+ \sim \rho(\cdot|\mathbf{s}, \mathbf{a}) \tag{2.1}$$

Note that (2.1) is a generalization of the classic dynamics, deterministic or not, often considered in the context of control theory, usually cast as

$$\mathbf{s}_+ = \mathbf{F}(\mathbf{s}, \mathbf{a}, \mathbf{w}), \quad \mathbf{w} \sim W \tag{2.2}$$

where $\mathbf{w} \in D$ is a random disturbance from distribution W and $\mathbf{F} : S \times A \times D \to S$ is a Borel-measurable function. In the special case $\mathbf{w} = 0$, (2.2) simply yields

deterministic dynamics. Solving an MDP is then the problem of finding the optimal policy $\boldsymbol{\pi}^\star : S \to A$ solution of:

$$\boldsymbol{\pi}^\star \in \arg\min_{\boldsymbol{\pi}} J(\boldsymbol{\pi}), \tag{2.3}$$

$$\tag{2.4}$$

where $J(\boldsymbol{\pi})$ is the performance function and depends on the optimality criteria describing the MDP.

2.1.1 Discounted setting

In the discounted setting, an MDP is defined by the triplet (L, γ, ρ), where $L : S \times A \to \mathbb{R}$ is a stage cost, $\gamma \in (0, 1]$ a discount factor and the performance function $J(\boldsymbol{\pi})$ is defined as follows:

$$J(\boldsymbol{\pi}) = \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k L\left(\mathbf{s}_k, \mathbf{a}_k\right) \,\middle|\, \mathbf{a}_k = \boldsymbol{\pi}\left(\mathbf{s}_k\right)\right], \tag{2.5}$$

and the expected value operator $\mathbb{E}[.]$ is taken over the (possibly) stochastic closed loop trajectories of the system. Discussing the solution of MDPs is often best done via the Bellman equations defining implicitly the optimal value function $V^\star : S \to \mathbb{R}$ and the optimal action-value function $Q^\star : S \times A \to \mathbb{R}$ as

$$V^\star(\mathbf{s}) = \min_{\mathbf{a}} Q^\star(\mathbf{s}, \mathbf{a}) \tag{2.6a}$$

$$Q^\star(\mathbf{s}, \mathbf{a}) = L(\mathbf{s}, \mathbf{a}) + \gamma \mathbb{E}\left[V^\star\left(\mathbf{s}_+\right) \,|\, \mathbf{s}, \mathbf{a}\right] \tag{2.6b}$$

The optimal policy then reads as:

$$\boldsymbol{\pi}^\star(\mathbf{s}) \in \arg\min_{\mathbf{a}} Q^\star(\mathbf{s}, \mathbf{a}) \tag{2.7}$$

2.1.2 Undiscounted setting

Undiscounted MDPs refer to MDPs with a discount factor $\gamma = 1$. If using $\gamma = 1$ in (2.5), $V^\star$ is in general unbounded and the MDP ill-posed. In order to tackle this issue, alternative optimality criteria are needed. Gain optimality is one of the common criteria in the undiscounted setting. Gain optimality is defined based on the following average-cost problem:

$$\bar{V}^\star(\mathbf{s}) := \min_{\boldsymbol{\pi}} \lim_{N\to\infty} \frac{1}{N} \mathbb{E}\left[\sum_{k=0}^{N-1} L(\mathbf{s}_k, \mathbf{a}_k) \,\middle|\, \mathbf{a}_k = \boldsymbol{\pi}\left(\mathbf{s}_k\right)\right], \tag{2.8}$$

for all initial state $\mathbf{s}_0^{\boldsymbol{\pi}} = \mathbf{s}, \forall \boldsymbol{\pi}$, where $\bar{V}^\star$ is the optimal average cost. We denote the optimal policy solution of (2.8) as $\bar{\boldsymbol{\pi}}^\star$. This optimal policy is called *gain optimal*.

The gain optimal policy $\bar{\boldsymbol{\pi}}^\star$ may not be unique. Moreover, the optimal average cost $\bar{V}^\star$ is commonly assumed to be independent of the initial state $\mathbf{s}$ [7]. This assumption e.g. holds for *unichain* MDPs, in which under any policy any state can be reached in finite time from any other state.

Unfortunately, the gain optimality criterion only considers the optimal steady-state distribution and it overlooks transients. As an alternative, *bias optimality* considers the optimality of the transients. Precisely, bias optimality can be formulated through the following OCP:

$$\tilde{V}^\star(\mathbf{s}) = \min_{\boldsymbol{\pi}} \mathbb{E}\left[\sum_{k=0}^{\infty} \left(L(\mathbf{s}_k, \mathbf{a}_k) - \bar{V}^\star\right) \middle| \, \mathbf{a}_k = \boldsymbol{\pi}\left(\mathbf{s}_k\right)\right], \tag{2.9}$$

where $\tilde{V}^\star$ is the optimal value function associated with bias optimality. Note that (2.9) can be seen as a special case of the discounted setting in (2.5) when $\gamma = 1$ and the optimal average cost $\bar{V}^\star$ is subtracted from the stage cost in (2.5). Therefore, for the rest of the paper, we will consider the discounted setting (2.5). Without loss of generality, we assume that $\bar{V}^\star = 0$ in the case $\gamma = 1$. This choice yields a well-posed optimal value function in the undiscounted setting. Clearly, if this does not hold, one can shift the stage cost to achieve $\bar{V}^\star = 0$.

2.2 Reinforcement Learning

As discussed, solving an MDP refers to finding an optimal policy that minimizes the expected value of a total cumulative cost as a function of the current state. Dynamic Programming (DP) techniques can be used to solve MDPs based on the Bellman equations. However, solving the Bellman equations is typically intractable unless the problem is of very low dimension [8]. This issue is known as the "curse of dimensionality" in the literature [9]. Besides, DP requires the exact transition probability of MDPs, while in most engineering applications, we do not have access to the exact probability transition of the real system.

Reinforcement Learning (RL) is a common technique that tackles these difficulties. The fundamental goal of RL is to use data to deliver an approximation of the optimal policy $\boldsymbol{\pi}^\star$. Indeed, RL offers practical tools for tackling MDPs without having an accurate knowledge of the probability distribution underlying the state transition ρ. RL methods are usually either directly based on an approximation of the optimal policy or indirectly based on an approximation of the action-value function.

The field can be coarsely divided into two large classes of approaches, value-based methods, and policy-based methods. Here we will detail Q-learning and Policy gradient methods as the indirect and direct methods, respectively.

2.2.1 Q-learning

The first class generically labelled Q-learning, approximates the optimal action-value function $Q^\star$ via a parametrized function approximator $Q_{\boldsymbol{\theta}}$. The parameters $\boldsymbol{\theta}$ are then adjusted using data such that $Q_{\boldsymbol{\theta}^\star} \approx Q^\star$ for the optimal parameters $\boldsymbol{\theta}^\star$.

Q-learning solves the following Least Square (LS) problem in order to achieve the best parameters $\boldsymbol{\theta}^\star$, describing the optimal action-value function $Q^\star$:

$$\min_{\boldsymbol{\theta}} \mathbb{E}\left[\left(Q_{\boldsymbol{\theta}}(\mathbf{s}_k, \mathbf{a}_k) - Q^\star(\mathbf{s}_k, \mathbf{a}_k)\right)^2\right]. \tag{2.10}$$

Temporal-Difference (TD) learning is a common way to tackle (2.10). More specifically, a basic TD-based learning step uses the following update rule for the parameters $\boldsymbol{\theta}$ at time instance k in the discounted setting (and the undiscounted setting when $\gamma = 1$):

$$\delta_k = L(\mathbf{s}_k, \mathbf{a}_k) + \gamma V_{\boldsymbol{\theta}}(\mathbf{s}_{k+1}) - Q_{\boldsymbol{\theta}}(\mathbf{s}_k, \mathbf{a}_k) \tag{2.11a}$$

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \zeta \delta_k \nabla_{\boldsymbol{\theta}} Q_{\boldsymbol{\theta}}(\mathbf{s}_k, \mathbf{a}_k) \tag{2.11b}$$

where the scalar $\zeta > 0$ is the learning step-size, δ_k is labelled the TD error and $V_{\boldsymbol{\theta}}$ is the parameterized value function. Note that there are more advanced methods to tackle (2.10) in the literature.

An approximation of the optimal policy $\boldsymbol{\pi}^\star$ can then be obtained using:

$$\hat{\boldsymbol{\pi}}^\star(\mathbf{s}) = \arg\min_{\mathbf{a}} Q_{\boldsymbol{\theta}^\star}(\mathbf{s}, \mathbf{a}) \tag{2.12}$$

2.2.2 Policy Gradient method

The second class approximates $\boldsymbol{\pi}^\star$ directly via a parametrized policy $\boldsymbol{\pi}_{\boldsymbol{\theta}}$, and adjust the parameters $\boldsymbol{\theta}$ from data so as to minimize $J(\boldsymbol{\pi}_{\boldsymbol{\theta}})$. This can, e.g., be done by estimating policy gradients $\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\pi}_{\boldsymbol{\theta}})$, or by building surrogate models of $J(\boldsymbol{\pi}_{\boldsymbol{\theta}})$, used to adjust $\boldsymbol{\theta}$. The former typically uses the gradient descent technique to update the parameters $\boldsymbol{\theta}$ as follows:

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \alpha \nabla_{\boldsymbol{\theta}} J(\boldsymbol{\pi}_{\boldsymbol{\theta}}), \tag{2.13}$$

where $\alpha > 0$ is the step size. Using deterministic policy gradient theory developed in [10], the gradient of J with respect to parameters $\boldsymbol{\theta}$ is obtained as

$$\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\pi}_{\boldsymbol{\theta}}) = \mathbb{E}\left[\nabla_{\boldsymbol{\theta}} \boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathbf{s}) \nabla_{\mathbf{a}} A_{\boldsymbol{\pi}_{\boldsymbol{\theta}}}(\mathbf{s}, \mathbf{a})|_{\mathbf{a}=\boldsymbol{\pi}_{\boldsymbol{\theta}}}\right], \tag{2.14}$$

where $A_{\boldsymbol{\pi}_{\boldsymbol{\theta}}}(\mathbf{s}, \mathbf{a}) = Q_{\boldsymbol{\pi}_{\boldsymbol{\theta}}}(\mathbf{s}, \mathbf{a}) - V_{\boldsymbol{\pi}_{\boldsymbol{\theta}}}(\mathbf{s})$ is the advantage function associated to the policy $\boldsymbol{\pi}_{\boldsymbol{\theta}}$, and where $Q_{\boldsymbol{\pi}_{\boldsymbol{\theta}}}$ and $V_{\boldsymbol{\pi}_{\boldsymbol{\theta}}}$ are the action-value and value functions for the policy $\boldsymbol{\pi}_{\boldsymbol{\theta}}$, respectively. Under some conditions detailed in [10], the action-value function $Q_{\boldsymbol{\pi}_{\boldsymbol{\theta}}}$ in (2.14) can be replaced by an approximation $Q_{\mathbf{w}}$ without affecting the policy gradient. Such an approximation is labelled *compatible* and can, e.g., take the form:

$$Q_{\mathbf{w}}(\mathbf{s}, \mathbf{a}) = \left(\mathbf{a} - \boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathbf{s})\right)^{\top} \nabla_{\boldsymbol{\theta}} \boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathbf{s})^{\top} \mathbf{w} + V_{\mathbf{v}}(\mathbf{s}), \tag{2.15}$$

where $\mathbf{w}$ is a parameters vector estimating the action-value function and $V_{\mathbf{v}} \approx V_{\boldsymbol{\pi}_{\boldsymbol{\theta}}}$ is a baseline function approximating the value function, which can, e.g., take a linear form:

$$V_{\mathbf{v}}(\mathbf{s}) = \Phi(\mathbf{s})^{\top} \mathbf{v}, \tag{2.16}$$

where Φ is a state feature vector and $\mathbf{v}$ is the corresponding parameters vector. The parameters $\mathbf{w}$ and $\mathbf{v}$ of the action-value function approximation (2.15) ought to be the solution of the Least Squares problem:

$$\min_{\mathbf{w},\mathbf{v}} \mathbb{E}\left[\left(Q_{\boldsymbol{\pi}_{\boldsymbol{\theta}}}(\mathbf{s}, \mathbf{a}) - Q_{\mathbf{w}}(\mathbf{s}, \mathbf{a})\right)^2\right]. \tag{2.17}$$

For instance, problem (2.17) can be tackled via Least Squares Temporal Difference (LSTD) [11].

2.3 Model Predictive Control

Model Predictive Control (MPC) is a popular and widely used practical approach to optimal control. MPC is often selected for its capability to handle both input and state constraints [4]. At each time instant, MPC calculates the input and corresponding state sequence minimizing a cost function while satisfying the constraints over a given prediction horizon. For a given system state $\mathbf{s}$, MPC produces control policies based on repeatedly solving an optimal control problem on a finite, receding horizon, often cast as:

$$\min_{\mathbf{x},\mathbf{u}} \quad T(\mathbf{x}_N) + \sum_{k=0}^{N-1} L(\mathbf{x}_k, \mathbf{u}_k) \tag{2.18a}$$

$$\text{s.t.} \quad \mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k), \quad \mathbf{x}_0 = \mathbf{s} \tag{2.18b}$$

$$\mathbf{h}(\mathbf{x}_k, \mathbf{u}_k) \leq 0, \quad \mathbf{u}_k \in A, \tag{2.18c}$$

for a given system state $\mathbf{s}$, where N is the horizon length, T is the terminal cost, $\mathbf{f}$ is a model of the system and $\mathbf{h}$ is the mixed input-state constraint.

Problem (2.18) produces a complete profile of control inputs $\mathbf{u}^\star = \{\mathbf{u}_0^\star, \dots, \mathbf{u}_{N-1}^\star\}$ and corresponding state predictions $\mathbf{x} = \{\mathbf{x}_0^\star, \dots, \mathbf{x}_N^\star\}$. Only the first element $\mathbf{u}_0^\star$ of the input sequence $\mathbf{u}^\star$ is applied to the system. At the next physical sampling time, a new state $\mathbf{s}$ is received, and problem (2.18) is solved again, producing a new $\mathbf{u}^\star$ and a new $\mathbf{u}_0^\star$. MPC hence yields a policy:

$$\boldsymbol{\pi}_{\mathrm{MPC}}(\mathbf{s}) = \mathbf{u}_0^\star, \tag{2.19}$$

with $\mathbf{u}_0^\star$ solution of (2.18) for $\mathbf{s}$ given. For $\gamma \approx 1$, policy (2.19) can provide a good approximation of the optimal policy $\boldsymbol{\pi}^\star$ for an adequate choice of prediction horizon N, terminal cost T and if the MPC model $\mathbf{f}$ approximates the true dynamics (2.1) sufficiently well. In that context, the latter is arguably the major weakness. Indeed, many systems are difficult to model accurately. Furthermore, within a modelling structure, selecting the model $\mathbf{f}$ that yields the best closed-loop performance $J(\boldsymbol{\pi}_{\mathrm{MPC}})$ is very difficult. Indeed, there is in general no guarantee that the model $\mathbf{f}$ that best fits the data collected from the real system is the best model in terms of $J(\boldsymbol{\pi}_{\mathrm{MPC}})$.

2.4 Learning based MPC

The combination of RL and MPC can address the issues raised above. In this section, we provide the central result supporting that statement. To that end, it is useful to construe MPC as a (possibly local) model of the action-value function $Q^\star$. Indeed, consider an MPC-based policy

$$\boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathbf{s}) = \mathbf{u}_0^\star \tag{2.20}$$

where $\mathbf{u}_0^\star$ is part of the solution of:

$$\mathbf{x}^\star, \mathbf{u}^\star = \arg\min_{\mathbf{x},\mathbf{u}} \quad T_{\boldsymbol{\theta}}(\mathbf{x}_N) + \sum_{k=0}^{N-1} L_{\boldsymbol{\theta}}(\mathbf{x}_k, \mathbf{u}_k), \tag{2.21a}$$

$$\text{s.t.} \quad \mathbf{x}_{k+1} = \mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x}_k, \mathbf{u}_k), \quad \mathbf{x}_0 = \mathbf{s}, \tag{2.21b}$$

$$\mathbf{h}_{\boldsymbol{\theta}}(\mathbf{x}_k, \mathbf{u}_k) \leq 0, \quad \mathbf{u}_k \in A. \tag{2.21c}$$

This MPC formulation is identical to (2.18), but the cost, constraints, and dynamics underlying the MPC scheme are now all parametrized in $\boldsymbol{\theta}$, to the exception of the input constraint $\mathbf{u}_k \in U$. This choice is motivated below. An MPC-based model of

$Q^\star$ is then provided by:

$$Q_{\boldsymbol{\theta}}(\mathbf{s},\mathbf{a}) = \min_{\mathbf{x},\mathbf{u}} \quad (2.21a), \tag{2.22a}$$

$$\text{s.t.} \quad (2.21b) - (2.21c), \quad \mathbf{u}_0 = \mathbf{a}, \tag{2.22b}$$

where a constraint $\mathbf{u}_0 = \mathbf{a}$ on the initial input has been added to (2.21). MPC (2.22) is a valid model of $Q^\star$ in the sense that it satisfies the relationships (2.6) and (2.7), i.e.:

$$\boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathbf{s}) = \arg\min_{\mathbf{a}} Q_{\boldsymbol{\theta}}(\mathbf{s},\mathbf{a}), \quad V_{\boldsymbol{\theta}}(\mathbf{s}) = \min_{\mathbf{a}} Q_{\boldsymbol{\theta}}(\mathbf{s},\mathbf{a}), \tag{2.23}$$

where $V_{\boldsymbol{\theta}}(\mathbf{s})$ is the optimal cost resulting from solving MPC (2.21). One can then readily verify that if the MPC parameters $\boldsymbol{\theta}$ are such that $Q_{\boldsymbol{\theta}} = Q^\star$, then MPC scheme (2.21) delivers the optimal policy $\boldsymbol{\pi}^\star$ through (2.20), i.e. $\boldsymbol{\pi}_{\boldsymbol{\theta}} = \boldsymbol{\pi}^\star$. An important question, then, is how effective can an MPC scheme be at approximating $Q^\star$, at least in a neighborhood of $\mathbf{a} = \boldsymbol{\pi}^\star(\mathbf{s})$. In addition, $Q^\star$ is typically built from a discounted sum of the stage costs L, while undiscounted MPC formulations are typically preferred.

The Theorem reported below addresses these concerns and provides the central justification for considering the MPC parametrization (2.21) in learning-based MPC. It establishes that under some mild conditions, (2.22) is able to provide an exact model of $Q^\star$ even if its predictive model (2.21b) is inaccurate. This in turn entails that MPC (2.21) can achieve optimal closed-loop performances even if the MPC model is inaccurate.

Theorem 1. *Suppose that the parameterized stage cost, terminal cost, and constraints in* (2.21) *are universal function approximators with adjustable parameters* $\boldsymbol{\theta}$*. Then there exist parameters* $\boldsymbol{\theta}^\star$ *such that the following identities hold,* $\forall\gamma$*:*

1. $V_{\boldsymbol{\theta}^\star}(\mathbf{s}) = V^\star(\mathbf{s}),\ \forall \mathbf{s} \in \mathcal{S}$
2. $\boldsymbol{\pi}_{\boldsymbol{\theta}^\star}(\mathbf{s}) = \boldsymbol{\pi}^\star(\mathbf{s}),\ \forall \mathbf{s} \in \mathcal{S}$
3. $Q_{\boldsymbol{\theta}^\star}(\mathbf{s},\mathbf{a}) = Q^\star(\mathbf{s},\mathbf{a}),\ \forall \mathbf{s} \in \mathcal{S}$*, for the inputs* $\mathbf{a} \in A$ *such that* $|V^\star(\mathbf{f}_{\boldsymbol{\theta}^\star}(\mathbf{s},\mathbf{a}))| < \infty$

if the set

$$\mathcal{S} =: \left\{ \mathbf{s} \in S \;\middle|\; |[V^\star(\mathbf{x}_k^\star)]| < \infty,\ \forall\, k \leq N \right\} \tag{2.24}$$

is non-empty.

Proof. We select the parameters such that the following holds:

$$T_{\boldsymbol{\theta}^\star}(\mathbf{s}) = V^\star(\mathbf{s}) \tag{2.25a}$$

$$L_{\boldsymbol{\theta}^\star}(\mathbf{s}, \mathbf{a}) = \tag{2.25b}$$

$$\begin{cases} Q^\star(\mathbf{s}, \mathbf{a}) - V^\star(\mathbf{f}_{\boldsymbol{\theta}^\star}(\mathbf{s}, \mathbf{a})) & \text{If } |V^\star(\mathbf{f}_{\boldsymbol{\theta}^\star}(\mathbf{s}, \mathbf{a}))| < \infty \\ \infty & \text{otherwise} \end{cases}$$

The proof then follows from [6, 12]. ■

Theorem 1 states that, for a given MDP, an MPC scheme with a possible inaccurate model can deliver the optimal value functions and the optimal policy of the original MDP. This can be achieved by selecting the proper stage cost, terminal cost, and constraints. Theorem 1 extends to robust MPC, stochastic MPC, and Economic MPC (EMPC), all discounted or not. The assumption in (2.24) can be interpreted as some form of the stability condition on $\mathbf{f}_{\boldsymbol{\theta}^\star}$ under the optimal trajectory $\mathbf{x}^\star$. More specifically, this assumption requires the existence of a non-empty set such that the optimal value function $V^\star$ of the predicted optimal trajectories $\mathbf{x}^\star$ based on the system model is finite with a unitary probability for all initial states starting from this set.

Then We can use RL techniques, detailed in the previous section, such as Q-learning and policy gradient method to tune the parameters $\boldsymbol{\theta}$ of parameterized MPC scheme (2.21) and approach the *optimal* parameter $\boldsymbol{\theta}^\star$.

3 | Contributions

The contributions provided from the published papers can be split into 11 categories as listed in Section 1.3 of Chapter 1. In this chapter, we discuss the detailed contributions of the each category.

3.1 Learning MPC for the ASV applications

Autonomous Surface Vehicles (ASVs) have been extensively investigated recently in industry and research [13–15]. However, designing a control strategy that is able to realize collision-free path planing, docking and handling potential failures of the ship thrusters in a freight mission with time-varying disturbances is still a topic worth exploring. With the development of ML, RL control strategies are getting noticed, as they can exploit real data to reduce the impact of model uncertainties and disturbances.

MPC is a successful control strategy in this field due to its capability for satisfying the state/input constraints while minimizing a finite-horizon cost function. However, model uncertainties can severely impact the performance of the MPC policy. In Robust Model Predictive Control (RMPC), Scenario tree MPC is a useful approach to handle nonlinear systems with finite and discrete uncertainties [16]. Scenario-based MPC approach for ship collision avoidance is presented in [17]. Using the MPC based RL idea presented in [6], RL methods then can be utilized in order to update the MPC scheme and achieve the best closed-loop performance over missions using the collected data form the real system. This approach solves the challenges arised from the wind and ocean current stochasticity, model uncertainty, and potential failures of the ship thrusters.

In paper i, we presented the use of Q-learning based on scenario-tree based RMPC for the obstacle avoidance and path planning of an ASV. The scenario-tree RMPC was used to handle potential failures of the ship thrusters. Besides, the wind and ocean

current were considered as unknown stochastic disturbances in the real system, which were handled via constraints tightening. An economic cost is considered, minimizing the time and energy required to achieve the ship missions. Actually, a trade-off between time and energy was considered to reach a neighborhood of the target as a baseline cost of RL. Moreover, this cost is penalized linearly when approaching the obstacles. The tightening and other cost parameters are adjusted by the RL method in order to achieve the best closed-loop performance with respect to the economic cost. The simulations on a nonlinear 3-DOF model of a scaled version of the Cybership II showed how RL managed to adjust the RMPC parameters in the several missions to improve the performance.

Paper ix provided an MPC based on deterministic policy gradient method for a complete ASV freight mission problem, including obstacle avoidance, path following, and autonomous docking, solved in a stochastic environment. Least-Squares Temporal-Difference (LSTD) method was used in order to update the action-value function.

Paper ix was led by Wenqi Cai [18].

3.2 Learning MPC/MHE for state estimation and control

In many real applications, a state estimator (observer) is needed to provide an estimation of the current system states to the MPC scheme. Moving Horizon Estimation (MHE) is an optimization-based state observer that works on a horizon window covering a limited history of past measurements [19].

Accurate models of dynamical systems are often difficult to obtain due to uncertainties and unknown dynamics. It is also worth noting that even if an accurate model is available, it may be in general too complex to be used in MHE and MPC schemes. However, if the model is imperfect, the inaccuracies can significantly degrade the performance of the MHE-MPC scheme. To cope with this problem, data-driven methods can be used in order to either improve the MPC and MHE models [20–23] or modify the MHE/MPC cost functions [6].

In some real-world control applications, the measurements available from the real system at a given time instant do not constitute a Markov state. In the context of RL, these systems are then formulated as Partially Observable Markov Decision Process (MDP) [24, 25]. To tackle a POMDP, one solution is to formulate a belief MDP where the information about current state is described as a probability distribution over the state space a.k.a belief state. Hence, POMDPs can be regarded as traditional MDPs using the concept of belief states as complete observable states [26]. Most previous works in the context of POMDPs use Neural Network (NN) to summarise the past

observations and learn the optimal policy [27–29]. An NN-based framework (posterior distributions over states) was proposed in [30, 31] in order to estimate a belief state based on historical information.

In paper *ii* (see [32]), a Q-learning method based on MHE-MPC with inaccurate models was proposed for dealing with POMDPs. In this research, we proposed to integrate MHE and MPC to treat the hidden Markovian state evolution and build an optimal policy based on the historic of the available measurements rather than on the full state of the system. More specifically, a structured solution by using a parameterized MHE-MPC scheme as a model based approach was proposed to create a state from the measurement history and a provide a parameterized policy. Then we utilized the Q-learning method to update the parameters of the MHE-MPC scheme in order to improve the performance.

This idea was investigated more in the both stochastic (probability measure space) and deterministic MHE-MPC schemes in *xvii* (see [33]) based on policy gradient method. The effectiveness of the proposed learning-based estimator/controller has been established for two examples including a model mismatch problem and a climate control of smart building where the building model used in the MHE-MPC is simplified and different from the real dynamics.

Papers *ii* and *xvii* were led by Hossein Nejatbakhsh Esfahani [32, 33].

3.3 Learning MPC for bang-bang policies, multi-agent battery storage and peak power management

Making decisions for the energy system in the presence of different forms of uncertainty is the object of recent publications [34, 35]. In smart grids, the uncertainty mainly arises from the imperfect forecasts for the prices, demand, and power generation. Finding a policy minimizing the economic cost of operating the grid in the presence of these uncertainties is highly valuable [36]. Economic costs for smart grids are linear, based on the difference between the profit made by selling electricity to the power grid, and the losses incurred from buying it [37].

MPC is a promising choice for the management of smart grids [35], because it provides a simple way to exploit forecasts on the grid prices, local power demand, and production, while respecting the physical limitations of the system. The stochasticity of the forecasts uncertainty is, however, not straightforward to treat at low computational costs.

In paper *iii*, we investigated a simple, well-known battery storage problem having a

purely economic cost and stochastic dynamics. This example has an optimal policy with a nearly bang-bang structure [38], in the sense that the optimal policy selects inputs that are either in the bounds or zero for a large subset of the state space. We showed that the deterministic policy gradient method is difficult to use for this type of problem because the state trajectories mostly lie in the set where the policy is trivially zero or in the bounds, which impedes the learning. We then proposed a homotopy strategy based on the interior-point method, which smoothness the MPC policy via the barrier parameter associated to the method, allowing for a more homogeneous and faster learning than a classical policy gradient approach.

A multi-agent battery storage system, usually includes several batteries that are connected to a main grid. The main grid exchanges the power with all of the batteries and the batteries attempt to optimize their own cost. Since the total power exchanged by the main grid is limited at each time, finding an optimal policy that satisfies this restriction is challenging.

In paper *vii*, considering the time-varying prediction of the spot market (using real power price data for Trondheim provided by the Nord Pool European Power Exchange [39]) and the production-demand uncertainty, we used a centralized MPC-scheme to minimize the running cost of the multi-agent battery storage system, while penalizing extreme State of Charge (SOC) and respecting the power peak constraint at the connection point of all batteries to the main grid. We supposed that a low level controller monitors the SOC in real time and prevents violating the constraints by buying or selling more power if needed.

In paper *viii*, we first considered a multi-agent residential smart grid system, where each agent has local renewable energy production and energy storage, and all agents are connected to a local transformer. The objective then was to develop an optimal policy that minimizes the economic cost consisting of both the spot-market cost for each consumer and their collective peak-power cost. In the paper, the MPC-based RL method was adopted to seek an optimal smart-grid policy that minimizes the long-term economic costs, including the spot-market cost and the peak-power cost. We used a parametrized MPC-scheme to approximate the optimal policy suffering from varying spot-market prices and inaccurate local agent's power production-consumption forecasts. To improve the closed-loop performance of the MPC-based policy we used deterministic policy gradient method.

Paper *xv* extended the latter paper and presented an Energy Management (EM) strategy for residential micro-grid systems using MPC-based RL. The EM problem was formulated as a Cooperative Coalition Game (CCG). The objective was to 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. Then, at the end of the monthly billing period, we apply the Shapley value approach to equitably distribute the profits (i.e., cost savings) gained through cooperation, that is, to determine the amount of electricity fee each resident should pay.

Papers $viii$ and xv were led by Wenqi Cai [40, 41].

3.4 Learning RMPC for bias correction of policy gradient and adjusting ellipsoidal uncertainty

Actor-Critic (AC) techniques combine the strong points of actor-only (policy search methods) and critic-only (e.g., Q-learning) methods [42]. AC approaches are based on genuine optimality conditions of the closed-loop policy and typically deliver less noisy policy gradients than direct policy search. The deterministic policy gradient is built based on an approximation of the advantage function associated with the policy. To this end, a linear compatible advantage function approximator is a convenient choice, because it provides a correct policy gradient estimation with a given structure and a low number of parameters [10]. For deterministic policies, exploration is required in order to estimate the corresponding policy gradient. For deterministic policies, exploration is required in order to estimate the corresponding policy gradient. In the presence of hard constraints, this exploration can be restricted. As a result, the exploration may not be Centred or Isotropic (non-CI). In [43], it is shown that a linear compatible advantage function approximator can deliver an incorrect policy gradient estimation for a non-CI exploration.

Paper iv proposed to use a RMPC scheme that is robust with respect to a bounded disturbance of its first control input to enable the feasibility of a Centred or Isotropic (CI) exploration. Because RMPC is computationally expensive, we used an inexpensive approximate RMPC instead, feasible to a first-order approximation. The RMPC-based policy ensures that a CI exploration is approximately feasible. A posterior projection technique was used in order to ensure the feasibility of the exploration. We then formally proved that the exploration resulting from RMPC scheme and the projection delivers an unbiased policy gradient estimation.

RMPC has received a great attention recently in the control community. Model-plant mismatch and disturbances can be treated via Robust NMPC (RNMPC) techniques. For linear MPC models and polytopic disturbance models and constraints, tube-based MPC techniques provides computationally effective techniques [44]. Treating nonlinear MPC models or generic disturbances and constraints is more challenging [45]. Researchers in [46] proposed to use a tube-based MPC with a Min-Max differential

inequality. Multi-stage or Scenario-tree NMPC scheme was proposed in [47, 48] as a real-time NMPC that accounts for the uncertain influence and generates decisions to control a nonlinear plant in a robust sense. These approaches remain challenging for problems that are not of small scale.

In paper [v], we modeled the propagation of perturbations in the state dynamics via ellipsoids, based on the linearization of the system dynamics and constraints on the nominal trajectories and using a Gaussian disturbance model. Then we proposed Robust Model Predictive Control (RMPC) based RL frame-work for controlling nonlinear systems in the presence of disturbances and uncertainties. We proposed to adjust the RMPC parameters generating the ellipsoids using the RL method in order to tailor this inaccurate uncertainty model to the real system and achieve a best closed-loop performance. A fast convergence of the adjustable parameters of RNMPC is achieved via a second-order Least Square Temporal Difference Q-learning (LSTDQ). The approach was tested on a simulated Wheeled Mobile Robot (WMR) tracking for a desired trajectory while avoiding static obstacles.

Paper [v] was led by Hossein Nejatbakhsh Esfahani [49].

3.5 Q-Learning of the storage function in EMPC schemes

Tracking Nonlinear Model Predictive Control (NMPC) refers to NMPC schemes that are formulated with a cost function penalizing the deviations of the current state and input from a desired steady-state reference [50]. More formally, the stage cost of a tracking NMPC scheme is lower-bounded by a class-$\mathcal{K}_\infty$ function, usually selected as convex, often quadratic. In contrast, the cost function used in Economic NMPC (ENMPC) does not satisfy such requirement [51–54]. The cost function used in ENMPC is typically an economic cost, often corresponding to the energy, the time or the financial cost of running a system [55]. Thus an ENMPC employs a cost function that is not necessarily lower-bounded by a class-$\mathcal{K}_\infty$ function with respect to any setpoint.

The closed-loop stability of an optimal policy provided by an ENMPC scheme requires the existence of a storage function satisfying dissipativity conditions while the stability of undiscounted tracking MPC schemes is fairly straightforward to establish [4], as the optimal value function can typically be used as a Lyapunov function for the closed-loop system. Finding the storage function for a given problem can be very demanding for nonlinear dynamics and non-quadratic stage costs [56].

In paper [vi], we leveraged on Q-learning technique to compute a data-based storage function for a given ENMPC problem. In order to capture the storage function and

verify dissipativity, we first parameterized the storage function, stage cost, and terminal cost in an undiscounted tracking MPC-scheme. We then used the parameterized tracking MPC, as a function approximator, in order to capture the optimal action-value function resulting from a specified infinite-horizon sum of economic stage costs. The undiscounted tracking MPC then provided a stabilizing policy for the closed-loop system regardless of whether the original ENMPC scheme is dissipative or not, and discounted or not. We showed that, for dissipative problems, if the parameterization is rich enough, then the resulting storage function satisfies the dissipativity conditions for the parameters that capture the optimal action-value function accurately. Finally, Q-learning was used in order to adjust the parameters of the tracking MPC-scheme. For a non-dissipative problem, Q-learning converges to sub-optimal parameters that cannot capture the optimal action-value function of the original ENMPC scheme.

The dissipativity theory for discounted formulations is more involved than for the undiscounted setting. In the former case, the discount factor has a central role to establish the closed-loop stability of the policy. Asymptotic stability requires an additional condition to the discounted strict dissipativity conditions. Recently the dissipativity condition has been extended to the discounted setting [57]. The resulting conditions are called Strong Discounted Strict Dissipativity (SDSD). The SDSD conditions guarantee asymptotic stability of the closed-loop dynamics with the discounted optimal policy.

Paper $\boxed{xiv}$ extends this idea for the discounted EMPC setting and we used an undiscounted tracking MPC-scheme function approximator for both the discounted and undiscounted ENMPC setting and showed that the proposed method works in both cases. Moreover, a detailed explanation of the stage cost parameterization that is lower-bounded by a class-$\mathcal{K}_\infty$ function and practical implementation were added.

3.6 Quasi-Newton iteration for deterministic policy gradient

Deterministic policy gradient algorithms are widely used in RL with continuous action spaces [58]. These methods attempt to learn the optimal parameters of a parameterized policy using only state transitions observed on the real system. These methods commonly use gradient descent methods to optimize a discounted sum of stage costs, called closed-loop performance. Unfortunately, the convergence rate of classical gradient descent is limited, especially when the Hessian of closed-loop performance is far from a scalar multiple of the Identity matrix [59].

Natural policy gradient methods has been attracted many attentions in RL community

recently due to its capability for better convergence [60]. The efficiency of the natural policy gradient in RL was showed in [61]. The natural policy gradient methods use the *Fisher information matrix* as an approximate Hessian [62].

Although the Fisher information matrix, as an approximation for the Hessian, is positive definite, it does not asymptotically converge to the exact Hessian necessarily, when the policy converges to the optimal policy [60]. As a result, the rate of convergence of the natural policy gradient method is linear, i.e., the same as the regular gradient descent [63]. Therefore, providing an approximation of the Hessian (without imposing heavy computation) that converges to the exact Hessian at the optimal policy can improve the convergence rate.

In paper x, we first derived a formulation for exact Hessian of deterministic policy performance with respect to the parameters. Then we provided a model-free approximation for the Hessian of the performance function. We showed that the approximate Hessian converges to the exact Hessian at the optimal policy when the parameterized policy is rich. As a result, it gives a superlinear convergence using a Quasi-Newton optimization.

3.7 Functional stability of discounted MDPs

In the context of MDP, most of the research has been done in order to find the optimal policy or verify the optimality of a given policy. However, in general, optimality may not lead to the stability of the closed-loop Markov Chain. Stability of the Markov Chains has been extensively studied in [64]. However, this framework provides results that are not easily related to MDPs and optimality criteria. To the best of our knowledge, there are limited results characterizing the stability of MDPs as an outcome of the interplay between its objective function and its dynamics.

In order to characterize the closed-loop stability of MDPs, we extend the concept of stability and dissipativity developed in the context of Economic Nonlinear Model Predictive Control (EMPC) [4]. This theory is based on a so-called *storage function* satisfying the dissipativity inequality. Dissipativity is well-known for EMPC schemes having an undiscounted cost and deterministic dynamics. In the discounted setting, finding the Lyapunov function still is challenging even for positive-definite stage costs [65]. In the discounted setting, the discount factor plays a vital role in the closed-loop stability. Recently the dissipativity theory has been extended to the discounted setting with deterministic dynamics [57]. These conditions are called SDSD.

In paper xi, we used the generalization of the classic dissipativity theory by making an argument on the measure space underlying the MDP rather than on the state space

itself. This idea was first discussed in [66], but was limited to undiscounted MDPs, where the dissipativity is fairly straightforward. In this paper, we considered MDPs with a general functional stage cost. We use the concept of D-stability [66] and introduce generalized functional dissipativity conditions for MDPs with a discounted objective function. We labeled these conditions Functional Strong Discounted Strict Dissipativity (FSDSD). These conditions require the transition probability, the stage cost, and the discount factor of the MDP to satisfy certain inequalities. We showed that if a given problem is FSDSD, then the D-stability of MDP follows.

3.8 Bias correction of the optimal steady state in discounted OCPs

One of the central objectives in control engineering, especially in chemical processes, power networks, etc [67, 68], is to steer the closed-loop trajectories of a given system to a steady point that has the minimum stage cost. Mathematically, the optimal steady-state problem can be formulated as a constrained optimization problem, where the cost function is the stage cost, and its constraint is the equilibrium of the point. This concept also appears in Economic Model Predictive Control (EMPC) problems, where the purpose is not tracking but to minimize a generic stage cost, such as time, energy and etc [4].

MPC schemes are generally formulated in an undiscounted setting. However, in some cases, it is reasonable to introduce a discount factor in the objective [69]. Discounted OCP has drawn wide attention in, e.g., economic application [70] and social science [71]. Moreover, a discounted infinite-horizon objective function is often the preferred setting in dynamic programming [8, 72] and reinforcement learning [73, 74].

The optimal steady state, resulting from the discounted OCP, differs from the optimal steady-state obtained from the undiscounted OCP. Although the discounted optimal steady-state is optimal in the sense of discounted OCP, the discounted optimal steady-state point does not result in the minimum one-step stage cost [75]. The bias between the discounted optimal steady-state and the undiscounted optimal steady-state depends on the discount factor, and tends to zero as the discount factor tends to one.

Paper $\boxed{xii}$ provides an inexpensive approximated cost modification using a second-order Taylor expansion of the optimal value function at the optimal steady state. We provide simple tools to compute the gradient and curvature needed for the approximation. Moreover, it was shown that the approximated cost modification preserves the stability of the closed-loop system locally.

3.9 Generic convex function approximator in the cost of learning MPC

One direction of Learning based MPC is the use of learning to build the MPC prediction model. In that context, neural networks (NNs) have typically been used for learning an approximation of the system dynamics from data, which is then used as the prediction model in the MPC scheme, see e.g. [76], [77], [78]. However, it is in general difficult to conclude regarding the closed-loop optimality of the resulting MPC scheme.

Another approach to learning-based MPC, is using cost modifications to handle model imperfection. The idea of compensating model inaccuracy with cost modifications was first established in [6]. However, the theory underlying this result suggests that in principle the cost parametrization should be "rich", i.e. it should be able to capture fairly generic functions. Rich parametrizations of the cost in the context of ENMPC were first considered in [79]. In paper [xvi], we elaborated on this early investigation and propose a more complete framework to provide such a rich parametrization. More specifically, we proposed to use a class of NNs that preserve convexity. This choice has two important benefits. First, ensuring convexity of the MPC cost alleviates the difficulties inherent to solving MPC schemes numerically using sensitivity-based solvers. Second, the stage cost in the MPC scheme must be lower-bounded by a $\mathcal{K}_\infty$-function to ensure stability. A convex function can be designed to satisfy this lower bound, and in turn ensures nominal stability of the resulting MPC scheme.

The main contribution of paper [xvi] is the introduction of convex NNs as cost modifications in MPC schemes with imperfect prediction models. Using the MPC scheme as a function approximator for the value function and the policy, we will use RL to adjust the cost parameters, including the NN weights, in pursuance of the optimal economic policy. The second contribution of the paper is the combination of RL methods for when neither value-based nor policy-based RL methods alone are sufficient.

Paper [xvi] was led by Katrine Seel [80].

3.10 Optimality equivalency of MDP and MPC

Solving an MDP refers to finding an optimal policy that minimizes the expected value of a total cumulative cost as a function of the current state. The cumulative cost can be either discounted or undiscounted with respect to the time instant. Therefore, different

definitions for the cumulative cost yields different optimality criteria for the MDPs.

RL is a model-free method that tackles these difficulties [58]. In most cases, RL requires a function approximator to capture the optimal policy or the optimal value functions underlying the MPD. Deep Neural Networks (DNNs) are common function approximators to capture either the optimal policy underlying the MDP directly or the action-value function from which the optimal policy can be indirectly extracted [81]. However, the formal analysis of closed-loop stability and safety of the policies provided by approximators such as DNNs is challenging. DNNs usually need a large number of tunable parameters. Moreover, a pre-training is often required so that the initial values of the parameters are reasonable.

An MPC scheme can be used as a function approximator without these difficulties [6], where it was shown that the optimal policy of a discounted MDP can be captured by a discounted MPC scheme even if the model is inexact. Stability for discounted MPC schemes is challenging, and for a finite-horizon problem, it is shown in [82] that even if the provided stage cost, terminal cost and terminal set satisfy the stability requirements, the closed-loop might be unstable for some discount factors. Indeed, the discount factor has a critical role in the stability of the closed-loop system under the optimal policy of the discounted cost. Therefore, an undiscounted MPC scheme is more desirable, where the closed-loop stability analysis is straightforward and well-developed.

The equivalency of MDPs criteria (discounted and undiscounted) has been recently discussed in [83] in the case an exact model of MDP is available. However, in practice, the exact probability transition of the MDP might not be available and we usually have a (possibly inaccurate) model of the real system. In paper $\boxed{xviii}$, we first showed that, under some conditions, an undiscounted finite-horizon Optimal Control Problem (OCP) can capture the optimal policy and the optimal value functions of a given MDPs, either discounted or undiscounted, even if an inexact model is used in the undiscounted OCP. We then proposed to use a deterministic (possibly nonlinear) MPC scheme as a particular case of the theorem to formulate the undiscounted OCP as a common MPC scheme. By parameterizing the MPC scheme, and tuning the parameters via RL algorithms one can achieve the best approximation of the optimal policy and the optimal value functions of the original MDP within the adopted MPC structure.

3.11 Probabilistic safe policy using Distributionally Robust MPC

Enforcing safety in the presence of uncertainty and stochasticity of nonlinear dynamical systems is a challenging task [84]. Chance constraints are common way of mathematical modeling of safety that requires a user-specified upper bound for the probability of the constraint violation [85]. However, from the computational point of view, it is challenging to handle a chance constraint due to its nonconvexity. Conditional Value at Risk (CVaR) [86] is a convex risk measure that has received considerable attention in decision-making problems, such as MDP [87, 88].

The theory of stochastic optimal control typically assumes that the probability distribution of the disturbance is fully known. However, this assumption may not hold in many real-world applications and one needs to estimate the probability distribution. In data-driven stochastic optimization, Sample Average Approximation (SAA) is a fundamental way to estimate the probability distribution of the random variables [89]. SAA typically needs quite a large set of data to fulfill risk constraints accurately. Distributionally Robust Optimization (DRO) is an alternative that overcomes this problem. DRO tackles stochastic optimization by considering the worst-case distribution in an ambiguity set. There are several ways to construct ambiguity sets (see e.g., [90–92]). Wasserstein based ambiguity set is well-known in this context, because it provides a probabilistic guarantee based on finite-samples in a tractable formulation [93]. The Wasserstein-based ball is a statistical ball in the space of probability distributions around the empirical distribution such that the radius of this ball is measured using Wasserstein distance.

In paper [xix], we used the DRO in the chance-constrained nonlinear MPC. This approach has been known as Distributionally Robust Model Predictive Control (DRMPC) [94]. We then proposed to use a parameterized nonlinear DRMPC based on the Wasserstein metric as a function approximator for RL in order to generate a family of policies that are safe by construction. DRMPC is subject to the chance constraint that is approximated by the CVaR risk measure and we reformulated Wasserstein DRMPC as a tractable optimization. A safety projection technique was used in order to ensure the safety with a random exploration. Then we used the Q-learning technique to optimize the parameters of the DRMPC scheme to achieve the best closed-loop performance among the safe policies.

4 | 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.

4.1 Conclusion

As discussed in Sections 3.1 and 3.3, MPC-based RL can be successfully applied for engineering applications such as ASV and smart grids, and the advantages of this method were shown in these fields. In ASV applications, the MPC scheme can solve crucial problems, such as path planning, trajectory tracking, autonomous docking, and obstacle avoidance. However, RMPC schemes such as scenario tree MPC is a successful approach to handle the finite and discrete uncertainties, the other uncertainties of the ship model, and stochastic disturbances caused by wind and ocean currents, and the data collected from the real system can improve the performance of the closed-loop system employing RL methods simultaneously. In the smart-grid and energy applications, it is shown that the MPC-based RL can be used in order to provide a smooth and fast learning procedure for bang-bang policies. Moreover, this framework can handle multi-agent battery systems while respecting all the individual state-input constraints and peak power capacity of the system.

MHE is a well-known state estimator that optimizes a moving finite-horizon cost based on a model of the read system in order to find the best estimations of the unobservable states. Similar to the way that RL is used to update MPC parameters, RL methods can also be used to update MHE parameters to compensate for the possible model mismatch. The use of MHE along with MPC can improve the performance of the closed-loop system and refine the state estimation for the systems whose states are not fully observable, known as PAMDPs in the literature of MDP and RL. Learning-based MHE-MPC is discussed in detail in Section 3.2.

RMPC can be used in different forms in the field of learning MPC, which we mentioned two applications in Section 3.4. In the first application, RMPC was used for constrained problems to generate optimal policies that remain feasible with random CI exploration. The CI exploration made it possible to evaluate the policy gradient without bias with respect to the actual policy gradient. RL-based MPC was used in another application to adjust ellipsoids that modeled the propagation of perturbations in the state dynamics for nonlinear systems.

Learning based on MPC is not limited only to the practical aspects but also solves some theoretical problems in the context of data-driven MPC. In Section 3.5, we explained the use of Q-learning to update a tracking MPC along with a parameterized storage function for Economic Model Predictive Control (EMPC) problems. It has been shown that this method leads to learning valid storage functions that satisfy dissipative conditions in both discounted and undiscounted EMPC settings.

Some contributions of this thesis are in the context of MDP and RL and can be used independently of the field of MPC. In Section 3.6, the Quasi-Newton method for deterministic policy gradient was explained using a novel proposed hessian. This method yields a faster learning rate. In Section 3.7, it was explained that one could characterize the closed-loop stability of a given MDP with the discounted optimality criterion by the inspiration of dissipativity theory in EMPC. This concept introduces stability properties on the measure space underlying the MDP rather than the state space itself.

The reason the optimal policy and optimal value functions of a given MDP with different optimality criteria can be captured only by modifying the stage cost and terminal cost function of an undiscounted MPC (even with an inaccurate model) was explained in Section 3.10. In fact, using the idea of parameterizing for the MPC scheme can be reasonable with the explanation of this section. In addition, the cost modification establishes equivalency in the optimality for discounted and undiscounted MDPs. In the field of EMPC, as explained in Section 3.8, the cost modification can correct the bias in the optimal steady state for the discounted setting with respect to the actual optimal steady state point (for the undiscounted setting). As explained in Section 3.9, the parameterized cost function, especially for the stage cost, can have properties such as convexity so that numerical methods do not struggle to solve the optimization problem and other properties such as being lower-bounded by a $\mathcal{K}_\infty$ can be satisfied by construction.

As discussed in Section 3.11, the combination of distributionally robust optimization in the MPC scheme and the use of parameterized DRMPC as an approximator function for RL methods results in the extracted policies being safe by construction for unknown distributions of disturbances.

4.2 Future work

The future directions of the current research can be considered as follows:

Stability of the closed-loop system. The nominal stability of the closed-loop system with the policy provided by an MPC scheme is relatively straightforward, while the stability of the real system in the present uncertainty and stochasticity is generally challenging. Although RMPC offers practical tools to discuss the stability of the system with uncertainty, it imposes a conservatism on the policy. Using the recently developed concept of stability and dissipativity of MDPs, characterizing the properties of the nominal model and MPC scheme to achieve some type of stability of the real system (e.g., functional stability of MDPs) can be valuable research in this field.

Convergence analysis. Specifying the properties of the parameterized MPC scheme, as well as the baseline cost and the algorithm used in RL in order to achieve a sequence of parameters that converge to the optimal parameter can be considered as another future research direction.

Learning the storage function for stochastic systems. Using the learning methods of the storage function provides a platform for applying the technique for stochastic systems and noisy data. Therefore, learning a valid storage function (or possibly functional) and verifying the dissipativity for a stochastic system (possibly MDP in the general case and functional space) using the proposed method can be a direction for future work. Moreover, combining the proposed method with the SOS technique can provide an interesting and fast method for evaluating the storage function.

Distributed MPC-based RL. One can investigate fully distributed MPC-based RL and utilize game theory to solve exciting questions in decentralized and large scales problems.

MPC-based Quasi-Newton RL. MPC can effectively provide a less noisy gradient and curvature of the parameterized action-value function. Therefore, providing actor-critic algorithms based on the MPC scheme using the proposed Hessian can be an interesting research direction.

Model mismatch and joint chance-constrained for safety. Considering inaccurate MPC models and joint-chance constraints in safety-critical systems can cover more practical problems. However, this will have some theoretical challenges, especially in nonlinear problems.

Implementations on real systems. Finally, applying the proposed method to real systems can be practical research.

5 | Publications

Publications

A Reinforcement learning based on scenario-tree MPC for ASVs

Postprint of [95] **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

Reinforcement Learning based on Scenario-tree MPC for ASVs

Arash Bahari Kordabad[1], Hossein Nejatbakhsh Esfahani[1], Anastasios M. Lekkas[1], and Sebastien Gros[1]

[1]Department of Engineering Cybernetics, Norwegian University of Science and Technology (NTNU), Trondheim, Norway.

Abstract: In this paper, we present the use of Reinforcement Learning (RL) based on Robust Model Predictive Control (RMPC) for the control of an Autonomous Surface Vehicle (ASV). The RL-MPC strategy is utilized for obstacle avoidance and target (set-point) tracking. A scenario-tree robust MPC is used to handle potential failures of the ship thrusters. Besides, the wind and ocean currents are considered as unknown stochastic disturbances in the real system, which are handled via constraints tightening. The tightening and other cost parameters are adjusted by RL, using a Q-learning technique. An economic cost is considered, minimizing the time and energy required to achieve the ship missions. The method is illustrated in simulation on a nonlinear 3-DOF model of a scaled version of the Cybership II.

1 Introduction

Autonomous Surface Vehicles (ASVs) have been extensively investigated recently in industry and research [1–3]. However, designing control systems that can tackle obstacle avoidance and tracking control, with severe external time-varying disturbances due to the wind, wave, and ocean currents, is one of the most challenging research topics for ASVs in maritime engineering [4, 5]. In the control literature, the motion control scenarios of such vehicles are divided into target tracking, path following, path tracking, and path maneuvering [6]. This paper focuses on target (set-point) tracking motion control in the presence of static elliptic-shaped obstacles and mission-varying wind and ocean currents. In set-point tracking, only a terminal point is given, which ought to be reached at minimum cost.

Reinforcement Learning (RL) is a powerful tool for tackling Markov Decision Processes (MDP) without prior knowledge of the process to be controlled [7]. Indeed, RL attaches a reward function to each state-action pair and tries to find a policy to optimize the discounted infinite rewards labelled performance [8]. Dynamic Programming (DP) methods can be used to solve MDPs. However, DP requires a knowledge of the MDP dynamics, and its computational complexity is unrealistic in practice for systems having more than a few states and inputs. Instead, most investigations in RL have focused on achieving approximate solutions, while

not requiring a model of the dynamics. Fuzzy Neural Networks and Deep Neural Networks (DNNs) are a common choice to approximate the optimal policy [9]. However, analysing formally the closed-loop behavior of a learned policy based on a DNN, such as stability and constraints satisfaction is challenging. Moreover, providing meaningful initial weights for the DNN can be very difficult. For instance, in [10] the baseline control is employed to ensure stability and tracking performance of ASV, while DNN-based RL is added to handle uncertainties and collision avoidance.

Model Predictive Control (MPC) is a well-known model-based control method that employs a model of the system dynamics to build an input sequence over a given finite horizon such that the resulting predicted state trajectory minimizes a given cost function while respecting the constraints imposed on the system [11]. The first input is applied to the real system, and the problem is solved at each time instant based on the latest state of the system. The advantage of MPC is its ability to explicitly support state and input constraints while producing a nearly optimal policy [12]. However, model uncertainties can severely impact the performance of the MPC policy.

In Robust Model Predictive Control (RMPC), Scenario- tree MPC is a useful approach to handle nonlinear systems with finite and discrete uncertainties. Scenario-based MPC approach for ship collision avoidance is presented in [13]. Tube-based MPC is another technique for RMPC mostly used when the MPC model and constraints are linear and the uncertainties can be contained in a polytope [11].

Data-driven adaptation of the RMPC model, e.g. using system identification, to better fit the real system is a fairly obvious strategy to tackle the issues concerning inaccurate model and unknown disturbance. However, if the model cannot capture the real system dynamics, adapting the model from data does not necessarily improve the performance of the MPC policy. Instead, we propose to use RL to online tune the RMPC formulation using the data obtained from the real system [14]. Unlike DNN, MPC as a function approximator for RL, can explicitly handle constraints satisfaction, stability, and safety [15–18].

In this paper, we use a scenario-tree MPC to manage potential thruster failures. Constraint tightening is used to avoid obstacles in the presence of stochastic wind and ocean currents. We consider a trade-off between time and energy to reach a neighborhood of the target as a baseline cost of RL. This cost is penalized linearly when approaching the obstacles. RL will adjust the tightening parameter and other RMPC parameters to find an optimal policy during some missions.

The paper is structured as follows. Section 2 presents the 3-DOF nonlinear ship's dynamics and its thruster configuration. Section 3 formulates the scenario-tree MPC and RL, and details an RMPC parameterized scheme as a function approximator of Q-learning. Section 4 describes the simulation details and illustrates the results. The target point tracking with back-off constraint in obstacle will be considered, and Q-learning tunes the parameters.

2 Vessel Model

The 3-DOF nonlinear dynamics of the Cybership II can be represented by a pose vector $\boldsymbol{\eta} = [x, y, \psi]^\top \in \mathbb{R}^3$ in the Earth-fixed frame, where x is the North position, y is the East position, ψ is the heading angle. The velocity vector $\boldsymbol{\nu} = [u, v, r]^\top \in \mathbb{R}^3$ includes the surge u and sway v velocities, and yaw rate r decomposed in the body-fixed frame (see Fig.1). The model dynamics can be written as follows [19]:

$$\dot{\boldsymbol{\eta}} = J(\psi)\boldsymbol{\nu} \tag{1a}$$

$$M_{RB}\dot{\boldsymbol{\nu}} + M_A\dot{\boldsymbol{\nu}}_r + C_{RB}(\boldsymbol{\nu})\boldsymbol{\nu} + C_A(\boldsymbol{\nu}_r)\boldsymbol{\nu}_r + D(\boldsymbol{\nu}_r)\boldsymbol{\nu}_r = \boldsymbol{\tau} + \boldsymbol{\tau}_w \tag{1b}$$

where $\boldsymbol{\nu}_r = \boldsymbol{\nu} - \boldsymbol{\nu}_c = [u_r, v_r, r]^\top$ is the ship velocity relative to the ocean current, and

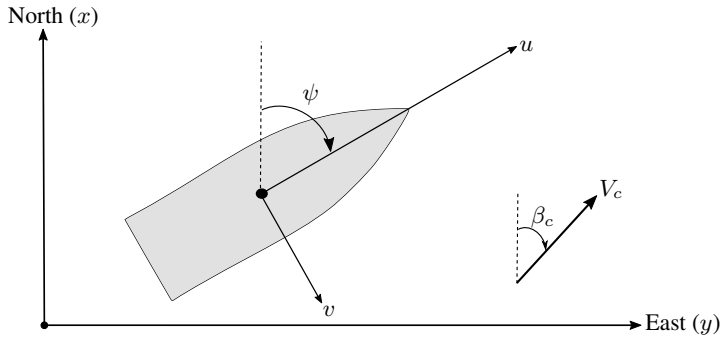


Figure 1: The 3-DOF Ship model in North-East-Down (NED) frame with surge u, sway v and heading angle ψ, and ocean current vector v_c

$\boldsymbol{\nu}_c = J(\psi)^\top \boldsymbol{v}_c$ where $\boldsymbol{v}_c = [V_c \cos\beta_c, V_c \sin\beta_c, 0]^\top$ are the ocean current in the body-fixed and Earth-fixed frames, respectively, and where V_c is the current velocity and β_c is its angle in the Earth-fixed frame. The rotation matrix $J(\psi)$ is given by:

$$J(\psi) = \begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2}$$

The rigid-body inertia matrix M_{RB} and added mass M_A are given by:

$$M_{RB} = \begin{bmatrix} m & 0 & 0 \\ 0 & m & mx_g \\ 0 & mx_g & I_z \end{bmatrix}, \; M_A = \begin{bmatrix} -X_{\dot{u}} & 0 & 0 \\ 0 & -Y_{\dot{v}} & -Y_{\dot{r}} \\ 0 & -N_{\dot{v}} & -N_{\dot{r}} \end{bmatrix} \tag{3}$$

where m is the mass of the ship, I_z is the moment of inertia about the body z_b-axis (yaw axis) and x_g is the distance between the center of gravity and the body x_b-axis. Furthermore, the

rigid-body and hydrodynamic of the Centripetal and Coriolis acceleration matrices read as:

$$\mathbf{C}_{RB}(\boldsymbol{\nu}) = \begin{bmatrix} 0 & 0 & -m(x_g r + v) \\ 0 & 0 & mu \\ m(x_g r + v) & -mu & 0 \end{bmatrix}, \mathbf{C}_A(\boldsymbol{\nu}_r) = \begin{bmatrix} 0 & 0 & c_{13} \\ 0 & 0 & c_{23} \\ -c_{13} & -c_{23} & 0 \end{bmatrix} \tag{4}$$

where $c_{13} = Y_{\dot{v}} v_r + 0.5(N_{\dot{v}} + Y_{\dot{r}})r$, $c_{23} = -X_{\dot{u}} u_r$, and $X_{\dot{u}}$, $Y_{\dot{v}}$, $Y_{\dot{r}}$, $N_{\dot{v}}$ and $N_{\dot{r}}$ are constant model parameters [20]. Moreover, the damping matrix is:

$$\mathbf{D}(\boldsymbol{\nu}_r) = -\begin{bmatrix} d_{11} & 0 & 0 \\ 0 & d_{22} & d_{23} \\ & d_{32} & d_{33} \end{bmatrix} \tag{5}$$

where

$$d_{11} = X_u + X_{|u|u}|u_r| + X_{uuu} u_r^2, \quad d_{22} = Y_v + Y_{|v|v}|v_r| + Y_{|r|v}|r| \tag{5a}$$
$$d_{23} = Y_r + Y_{|v|r}|v_r| + Y_{|r|r}|r|, \quad d_{32} = N_v + N_{|v|v}|v_r| + N_{|r|v}|r| \tag{5b}$$
$$d_{33} = N_r + N_{|v|r}|v_r| + N_{|r|r}|r| \tag{5c}$$

where $X_{(.)}$, $Y_{(.)}$, and $N_{(.)}$ are the hydrodynamic coefficients [20]. The model parameters are taken from [19]. Finally, $\boldsymbol{\tau} = [X, Y, N]^\top$ is the external control forces X, Y and moment N vector and $\boldsymbol{\tau}_w$ is the wind effects disturbance.

2.1 Thruster Allocation

We consider one tunnel thruster (transverse) f_1 and two main propeller thrusters (longitudinal) f_2, f_3 as the thrust configuration (see Fig. 2). Then

$$\boldsymbol{\tau} = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 0 \\ l_x & -l_y & l_y \end{bmatrix} \boldsymbol{a} \tag{6}$$

where $\boldsymbol{a} = [f_1, f_2, f_3]^\top$ is the actuator forces vector subject to the bounds:

$$\boldsymbol{a}_{\min} \leq \boldsymbol{a} \leq \boldsymbol{a}_{\max} \tag{7}$$

2.2 Obstacle Avoidance

For simplicity, we consider obstacles of elliptic shape. Hence, the condition for obstacles avoidance can be seen as the following inequality:

$$\left((x - o_{x,j})/(r_{x,j} + r_o)\right)^2 + \left((y - o_{y,j})/(r_{y,j} + r_o)\right)^2 \geq 1 \tag{8}$$

where $(o_{x,j}, o_{y,j})$ and $(r_{x,j}, r_{y,j})$ are the center and radii of the j^{th} ellipse ($j = 1, \ldots, N_o$), respectively, r_o is radius of the vessel and N_o is number of obstacles.

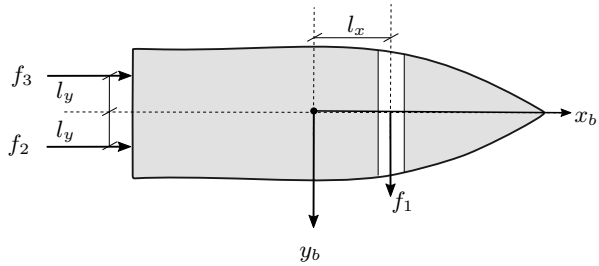


Figure 2: Schematic drawing showing the thrusters configuration in the body-fixed frame $\{b\}$

3 RMPC-based Reinforcement learning

In this section, we formulate the scenario-tree MPC scheme and detail how it can be treated via Q-learning.

3.1 Robust Model Predictive Control

Scenario-tree MPC is a robust MPC technique that can treat finite and discrete uncertainties in the system [21]. Fig. 3 shows the evolution of the system described by a scenario tree, where $\boldsymbol{x}_{k,i}$ and $\boldsymbol{u}_{k,i}$ are the state and input of scenario k at time i, given by:

$$\boldsymbol{x}_{k,i+1} = \boldsymbol{f}_{k,i}\left(\boldsymbol{x}_{k,i}, \boldsymbol{u}_{k,i}\right) \tag{9}$$

where $\boldsymbol{f}_{k,i}$ is the k^{th} (time-varying) model. In this paper, the scenario tree will be used to model the thruster failures in the system, hence each model $\boldsymbol{f}_{k,i}$ corresponds to a specific failure k at a specific time i. Since the number of scenarios grows exponentially with the length of the MPC horizon, it is common to fix the uncertain parameters after a certain period of time called Robust horizon $N_r < N$, where N is the MPC prediction horizon. Then the number of scenarios is $M = m_d^{N_r}$, where m_d is the number of realization (branches) at each time stage. We assumed separate state and control variables for each scenario to enable parallel computations. However, because the uncertainty cannot be anticipated, control action must depend on only the historical realizations of the uncertainty. Then, $\boldsymbol{u}_{k,j} = \boldsymbol{u}_{l,j}, \forall j = 0, ..., i$ if the uncertainty realization for scenario k and l are identical up to and including the time stage i. This restriction is commonly denoted as non-anticipativity constraint. In Fig. 3, $N = 4$, $m_d = 3$, $N_r = 2$ and then, $M = m_d^{N_r} = 9$. Also, $\boldsymbol{u}_{1,0} = \boldsymbol{u}_{2,0} = \boldsymbol{u}_{3,0}, \boldsymbol{u}_{4,0} = \boldsymbol{u}_{5,0} = \boldsymbol{u}_{6,0}, \boldsymbol{u}_{7,0} = \boldsymbol{u}_{8,0} = \boldsymbol{u}_{9,0}$ are the non-anticipativity constraints.

3.2 Reinforcement Learning

Reinforcement Learning considers that the real system is described by a Markov Decision Process (MDP) with state transitions having the underlying conditional probability density

$$x_{i+1} = A_i(\theta)x_i + B_i(\theta)u_i \quad (1a)$$
$$C_i(\theta)x_i + D_i(\theta)u_i \le d_i(\theta) \quad (1b)$$

where $x_i \in \mathbb{R}^n$ and $u_i \in \mathbb{R}^m$ denote the state and control variables respectively. To account for the uncertain parameters, we consider n_d realizations of (1) at each time stage. The evolution of the system can then be described by a scenario tree as depicted in Figure 1.

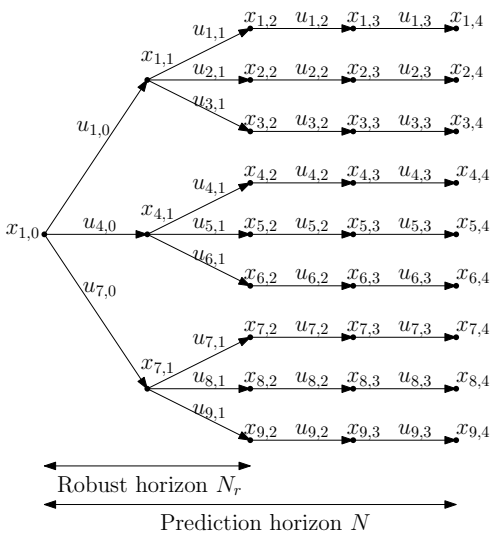


Fig. 1: The evolution of the system represented as a scenario tree. The nodes and branches that are shared between multiple scenarios, the variable corresponding to the scenario with the lowest index is visualized in the tree.

Figure 3: The evolution of the system represented as a scenario tree [21]

$\mathbb{P}[\boldsymbol{s}_+|\boldsymbol{s},\boldsymbol{a}]$, where $\boldsymbol{s},\boldsymbol{a}$ is the current state-input pair and $\boldsymbol{s}_+$ is the subsequent state. The control literature typically uses the notation $\boldsymbol{s}_+ = \boldsymbol{f}^{\text{real}}(\boldsymbol{s},\boldsymbol{a},\boldsymbol{\zeta})$ where $\boldsymbol{\zeta}$ is a random disturbance and $\boldsymbol{f}^{\text{real}}$ is discretized real system dynamics (1) and $\boldsymbol{s} = [\boldsymbol{x}^\top, \boldsymbol{u}^\top]^\top$. We will label $L(\boldsymbol{s},\boldsymbol{a})$ as the baseline stage cost associated with the MDP at each transition. The optimum action-value function $Q_\star$, optimum value function $V_\star$, and optimum policy $\boldsymbol{\pi}_\star$ associated with the MDP are defined by the Bellman equations:

$$V_\star(\boldsymbol{s}) = \min_{\boldsymbol{a}} Q_\star(\boldsymbol{s},\boldsymbol{a}) \quad (10a)$$
$$Q_\star(\boldsymbol{s},\boldsymbol{a}) = L(\boldsymbol{s},\boldsymbol{a}) + \gamma\mathbb{E}[V_\star(\boldsymbol{s}_+)|\boldsymbol{s},\boldsymbol{a}], \quad (10b)$$
$$\boldsymbol{\pi}_\star(\boldsymbol{s}) = \arg\min_{\boldsymbol{a}} Q_\star(s,a) \quad (10c)$$

where $\gamma \in (0,1]$ is the MDP discount factor.

Q-learning is a classical model-free RL algorithm that tries to capture the action value function $Q_{\boldsymbol{\theta}} \approx Q_\star$ via tuning the parameters vector $\boldsymbol{\theta} \in \mathbb{R}^n$. The approximation of the value function $V_{\boldsymbol{\theta}}$ and parametric optimal policy $\boldsymbol{\pi}_{\boldsymbol{\theta}}$ can then be extracted from the Bellman equations. Q-learning uses the following update rule for the parameters $\boldsymbol{\theta}$ at state $\boldsymbol{s}_k$ [22]:

$$\delta_k = L(\boldsymbol{s}_k,\boldsymbol{a}_k) + \gamma V_{\boldsymbol{\theta}}(\boldsymbol{s}_{k+1}) - Q_{\boldsymbol{\theta}}(\boldsymbol{s}_k,\boldsymbol{a}_k) \quad (11a)$$
$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \alpha\delta_k\nabla_{\boldsymbol{\theta}}Q_{\boldsymbol{\theta}}(\boldsymbol{s}_k,\boldsymbol{a}_k) \quad (11b)$$

where the scalar $\alpha > 0$ is the learning step-size, δ_k is labelled the Temporal-Difference (TD) error and the input $\boldsymbol{a}_k$ is selected according to the corresponding parametric policy $\boldsymbol{\pi}_{\boldsymbol{\theta}}(\boldsymbol{s}_k)$ with possible addition of small random exploration.

A. Reinforcement learning based on scenario-tree MPC for ASVs

We define a scenario as a path from the root node to a leaf node, see Figure 1. The number of scenarios resulting from the scenario-tree approach grows exponentially with the length of the MPC horizon, yielding very large optimization problems to be solved online. Hence, approaches proposed to treat the uncertain parameters as constant after a certain period of time. We denote the time period where the parameters can change as

$$\min_{x,u} \sum_{k=1}^{M} \bar{A}_k x_k + \bar{B}_k u_k = b_k,$$
$$\text{s.t.} \quad \bar{A}_k x_k + \bar{B}_k u_k = b_k,$$
$$\bar{C}_k x_k + \bar{D}_k u_k \le d_k,$$
$$\sum_{k=1}^{M} \bar{E}_k u_k = 0$$

where we have introduced the notat
and $u = [u_1^T \dots u_M^T]^T$ for the collect
scenarios, and $V(x_k, u_k)$ denotes a s
function. The dynamics (2b) are def

$$\bar{A}_k = \begin{bmatrix} -I & & \\ A_{k,1} & -I & \\ & \ddots & \\ & & A_{k,} \end{bmatrix}$$

$$\bar{B}_k = \begin{bmatrix} B_{k,0} & & \\ & \ddots & \\ & & B_{k,} \end{bmatrix}$$

and $b_k = [-x_0^T A_{k,0}^T \; 0 \dots 0]^T$, wher
are defined by the matrices:

$$\bar{C}_k = \begin{bmatrix} 0 & \dots & \\ C_{k,1} & & \\ & \ddots & \\ & & C \end{bmatrix}$$

$$\bar{D}_k = \begin{bmatrix} D_{k,0} & & \\ & \ddots & \\ & & D \\ 0 & \dots & \end{bmatrix}$$

and $d_k = [d_{k,0} \dots d_{k,N}]^T$, for $C_{k,i}$ $\in \mathbb{R}^{}$. For notational sim
$\mathcal{Z} = \mathcal{Z}_1 \times \cdots \times \mathcal{Z}_M$.

The formulation of the non-antici
provides some freedom in constructi

Using RMPC as a way of supporting the approximations $V_{\boldsymbol{\theta}}$ and $Q_{\boldsymbol{\theta}}$ has been proposed and justified in [14]. Hereafter, we detail how this can be done for the specific choice of RMPC proposed here.

3.3 RMPC as a function approximator for RL

We propose to use the action-value function approximate $Q_{\boldsymbol{\theta}} \approx Q_\star$ obtained from the following RMPC scheme parameterized by $\boldsymbol{\theta}$ [14]:

$$
\begin{aligned}
Q_{\boldsymbol{\theta}}(\boldsymbol{s},\boldsymbol{a}) = \min_{\boldsymbol{x},\boldsymbol{u},\boldsymbol{\sigma}} \quad & \sum_{k=1}^{M} \Bigg(\gamma^N V_k^f\left(\boldsymbol{x}_{k,N},\boldsymbol{\theta}\right) + \boldsymbol{\omega}_f^\top \boldsymbol{\sigma}_{k,N} + \\
& \sum_{i=0}^{N-1} \left(\gamma^i l_k\left(\boldsymbol{x}_{k,i},\boldsymbol{u}_{k,i},\boldsymbol{\theta}\right) + \boldsymbol{\omega}^\top \boldsymbol{\sigma}_{k,i}\right) \Bigg) && \text{(12a)} \\
\text{s.t.} \quad & \forall i = 0,...,N-1, \forall k = 1,...,M: \\
& \boldsymbol{x}_{k,i+1} = \boldsymbol{f}_{k,i}(\boldsymbol{x}_{k,i},\boldsymbol{u}_{k,i},\boldsymbol{\theta}) && \text{(12b)} \\
& \boldsymbol{h}_{\boldsymbol{\theta}}(\boldsymbol{x}_{k,i},\boldsymbol{u}_{k,i}) + \boldsymbol{B}_{k,i}(\boldsymbol{\theta}) \leq \sigma_{k,i} && \text{(12c)} \\
& \boldsymbol{h}^f_{\boldsymbol{\theta}}(\boldsymbol{x}_{k,N}) + \boldsymbol{B}^f_{k,N}(\boldsymbol{\theta}) \leq \sigma_{k,N} && \text{(12d)} \\
& \boldsymbol{g}(\boldsymbol{u}_{k,i}) \leq 0 && \text{(12e)} \\
& \boldsymbol{u}_{k,i} = \boldsymbol{u}_{l,i} \text{ if } \boldsymbol{x}_{k,j} = \boldsymbol{x}_{l,j}, \\
& \forall k,l \in \{1,..,M\}, \forall j \in \{1,...,i\} && \text{(12f)} \\
& \boldsymbol{x}_{k,0} = \boldsymbol{s} && \text{(12g)} \\
& \boldsymbol{u}_{k,0} = \boldsymbol{a} && \text{(12h)} \\
& \boldsymbol{\sigma}_{k,i} \geq 0, \quad \boldsymbol{\sigma}_{k,N} \geq 0 && \text{(12i)}
\end{aligned}
$$

where $\boldsymbol{x} = \{\boldsymbol{x}_{1,0},\ldots,\boldsymbol{x}_{M,N}\}$, $\boldsymbol{u} = \{\boldsymbol{u}_{1,0},\ldots,\boldsymbol{u}_{M,N-1}\}$ and $\boldsymbol{\sigma} = \{\boldsymbol{\sigma}_{1,0},\ldots,\boldsymbol{\sigma}_{M,N}\}$ are the primal decision variables, M is the number of scenarios, N is the prediction horizon, $\boldsymbol{f}_{\{1,\ldots,M\},i}$ are M different (possibly) time-varying models supporting the discrete uncertainties, $l_{1,\ldots,M}$ and $V^f_{1,\ldots,M}$ are the stage and terminal costs for the different scenarios, respectively. The constraint tightening is performed in (12c) and (12d), where $\boldsymbol{B}_{k,i}(\boldsymbol{\theta}) \geq 0$ and $\boldsymbol{B}^f_{k,N}(\boldsymbol{\theta}) \geq 0$ are the (possibly) time-varying tightening parameters. Variables $\boldsymbol{\sigma}_{k,i}$ and $\boldsymbol{\sigma}_{k,N}$ are slacks for the relaxation of the mixed state-input constraints, using the positive weights vectors $\boldsymbol{\omega}$ and $\boldsymbol{\omega}_f$, respectively. The relaxation prevents the infeasibility of the tightened constraints of the RMPC in the presence of disturbances and mismatching models $\boldsymbol{f}_{k,i}$ to the real system $\boldsymbol{f}^{\text{real}}$. Constraint (12e) represents the input inequality constraints which is defined in (7) for the ASV. Constraint (12f) is the non-anticipativity constraint [23]

In (12), $\boldsymbol{\theta}$ is the parameters vector that can be modified by RL to shape the action-value function. Under some mild assumptions (see [14] for the technical details), if the parametrization is rich enough, the MPC scheme is able to capture the true optimal action-value function $Q_\star$,

value function $V_\star$ and policy $\boldsymbol{\pi}_\star$ jointly, even if the RMPC models $\boldsymbol{f}_{k,i}$ do not capture the real system dynamics (1).

One can verify that the parameterized value function $V_{\boldsymbol{\theta}}$ that satisfies the Bellman equations can be obtained by solving (12) without constraint (12h). Moreover, the parameterized deterministic policy $\boldsymbol{\pi}_{\boldsymbol{\theta}}$ reads as follows:

$$\boldsymbol{\pi}_{\boldsymbol{\theta}}(\boldsymbol{s}) = \boldsymbol{u}^\star_{k,0}(\boldsymbol{s}, \boldsymbol{\theta}) \tag{13}$$

where $\boldsymbol{u}^\star_{k,0}(\boldsymbol{s}, \boldsymbol{\theta})$ is the first element of $\boldsymbol{u}^\star$, solution of the RMPC scheme (12) when constraint (12h) is removed. Therefore, the value function $V_{\boldsymbol{\theta}}(\boldsymbol{s})$ can be acquired together with the policy $\boldsymbol{\pi}_{\boldsymbol{\theta}}(\boldsymbol{s})$ by solving a classic MPC scheme, while the action value function results from solving the same MPC scheme with its first input constrained to a specific value $\boldsymbol{a}$.

The sensitivity $\nabla_{\boldsymbol{\theta}} Q_{\boldsymbol{\theta}}(\boldsymbol{s}, \boldsymbol{a})$ required in (11b) is given by [14]:

$$\nabla_{\boldsymbol{\theta}} Q_{\boldsymbol{\theta}}(\boldsymbol{s}, \boldsymbol{a}) = \nabla_{\boldsymbol{\theta}} \mathcal{L}_{\boldsymbol{\theta}}(\boldsymbol{s}, \boldsymbol{a}, \boldsymbol{y}^\star) \tag{14}$$

where $\mathcal{L}$ is the Lagrange function associated to the scenario-tree RMPC (12), i.e.:

$$\mathcal{L}_{\boldsymbol{\theta}}(\boldsymbol{s}, \boldsymbol{a}, \boldsymbol{y}) = \Phi_{\boldsymbol{\theta}} + \boldsymbol{\lambda}^\top \boldsymbol{G}_{\boldsymbol{\theta}} + \boldsymbol{\mu}^\top \boldsymbol{H}_{\boldsymbol{\theta}} \tag{15}$$

where $\Phi_{\boldsymbol{\theta}}$ is the cost (12a), $\boldsymbol{G}_{\boldsymbol{\theta}}$ gathers the equality constraints (12b), (12f), (12g), (12h), $\boldsymbol{H}_{\boldsymbol{\theta}}$ collects the inequalities (12c), (12d), (12e), (12i), and $\boldsymbol{\lambda}, \boldsymbol{\mu}$ are the associated dual variables. Argument $\boldsymbol{y}$ reads as $\boldsymbol{y} = \{\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{\sigma}, \boldsymbol{\lambda}, \boldsymbol{\mu}\}$ and $\boldsymbol{y}^\star$ is the solution to (12).

4 Simulation

In this section, we consider a target tracking problem in the presence of static obstacles modelled as ellipsoids, random wind and ocean currents, and discrete uncertainties in the dynamics. The objective is to reach the terminal (target) point while achieving an optimal trade-off between time and energy.

We consider the nominal system as the first scenario ($k = 1$) and the failure of thrusters f_2 or f_3 ($k = 2, 3$) as the discrete uncertainties in the system. As a result, by considering $N_r = 1$, this formulation has $M = m_d = 3$ scenarios and realization at each time instance.

We consider a stage cost that minimizes both energy and time. Also, the stage cost is in the form:

$$L(\boldsymbol{s}, \boldsymbol{a}) = \underbrace{|X.u| + |Y.v| + |N.r|}_{\text{power}} + \underbrace{T}_{\text{time}} + \underbrace{\boldsymbol{c}^\top \max(0, \boldsymbol{h}_{\boldsymbol{\theta}} + \boldsymbol{d})}_{\text{obstacles penalty}} \tag{16}$$

where T is a constant introducing a penalty on the time to reach the target. The term $\boldsymbol{c}^\top \max(0, \boldsymbol{h}_{\boldsymbol{\theta}} + \boldsymbol{d})$ penalizes violations of the relaxed inequality constraints $\boldsymbol{h}_{\boldsymbol{\theta}} + \boldsymbol{d} \leq 0$ with a weight vector $\boldsymbol{c}$. The parameter $\boldsymbol{d}$ can be interpreted as the dangerous distance from the

obstacles. Indeed, when $0 < \boldsymbol{h}_{\boldsymbol{\theta}} + \boldsymbol{d}$, RL tries to increase the distance by adjusting the MPC tightening parameters. Since the task is episodic here, we can use an undiscounted cost in RL i.e. $\gamma = 1$.

The obstacles constraints tightening is parametrized as follows:

$$\boldsymbol{B}_{k,i}(\boldsymbol{\theta}) = \boldsymbol{B}^{f}_{k,N}(\boldsymbol{\theta}) = \boldsymbol{\theta}^{h}_{k} \tag{17}$$

where $\boldsymbol{\theta}^{h}_{k} = \boldsymbol{\theta}^{h}_{k,\{1,\dots,N_o\}}$ is horizon-invariant parameter and we use $N_o = 2$ obstacles.

The stochastic ocean current is represented as $\boldsymbol{\zeta} = \{V_c, \beta_c\}$. We generate a random current map for each mission independently, using the gradient of the Gaussian Radial Basis Functions set, as follows:

$$\boldsymbol{\upsilon}_c = \frac{\partial}{\partial \boldsymbol{p}} \sum_{l=1}^{N_c} q_l \exp\left(-\frac{\|\boldsymbol{p} - \boldsymbol{b}_l\|^2}{2\rho_l^2}\right) \tag{18}$$

where $\boldsymbol{p} = [x, y]^\top$ is the position vector, $\{q_l, \boldsymbol{b}_l, \rho_l\}$ are random values and N_c is the number of Gaussian functions which we take $N_c = 2$ here. Then V_c and β_c are obtained as magnitude and angle of the vector $\boldsymbol{\upsilon}_c$.

We consider $N = 20$ the prediction horizon. A sampling time of dt $= 0.5$s was chosen for the discretization of the system dynamics (1), and the actuators bounds as $\boldsymbol{a}_{\max} = [2, 8, 8]^\top$N and $\boldsymbol{a}_{\min} = [-2, 0, 0]^\top$N in (7). In addition, the stage and terminal costs of the RMPC scheme can be represented as the following weighted vector norm:

$$l_k(\boldsymbol{x}_{k,i}, \boldsymbol{u}_{k,i}, \boldsymbol{\theta}) = \left\| \left[\left(\boldsymbol{x}_{k,i} - \boldsymbol{X}_{ref}\right)^\top, \boldsymbol{u}_{k,i}^\top \right]^\top \right\|_{\Theta^l_k} \tag{19a}$$

$$V^f_k(\boldsymbol{x}_{k,i}, \boldsymbol{\theta}) = \|\boldsymbol{x}_{k,i} - \boldsymbol{X}_{ref}\|_{\Theta^V_k} \tag{19b}$$

where $\boldsymbol{X}_{ref}$ is the reference state in the target-tracking and parameters Θ^l_k and Θ^V_k are the weights of the vector norm. They can be tune by RL as well. The RL parameters read as:

$$\boldsymbol{\theta} = \left\{ \boldsymbol{\theta}^h, \Theta^l_1, \dots, \Theta^l_M, \Theta^V_1, \dots, \Theta^V_M \right\} \tag{20}$$

Fig. 4 shows the path for the first simulated mission. The corresponding random wind and ocean current map is shown as well. The failure scenario prediction and nominal scenario are specified by red and green, respectively. The learning process continues until RMPC predicts the target point as the terminal state for the first time. Once the target point is within the RMPC horizon, a different control scheme ought to be used.

Fig. 5 illustrates the paths over missions for the nominal system. We simulated seven missions and for the sake of brevity, four missions were selected for illustration. It can be seen that the paths are nearing obstacles during the missions until the RL penalty is activated and find the optimal distance to handle disturbance.

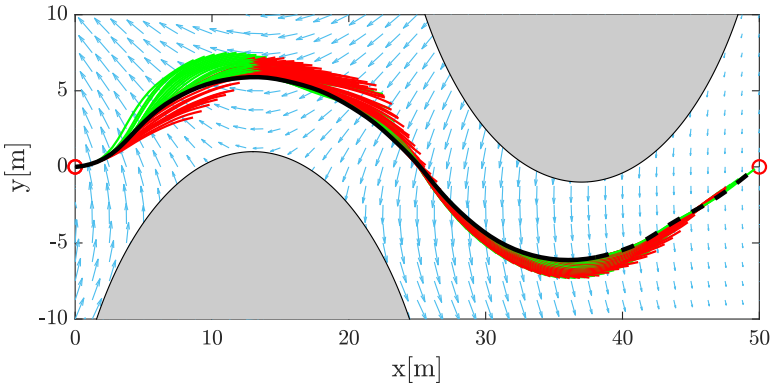


Figure 4: The path of the ship (black) in the first mission and random current, fail prediction ($k = 2, 3$): red and nominal prediction ($k = 1$): green. RL updating is stopped in the dashed line (the MPC prediction at the end of missions).

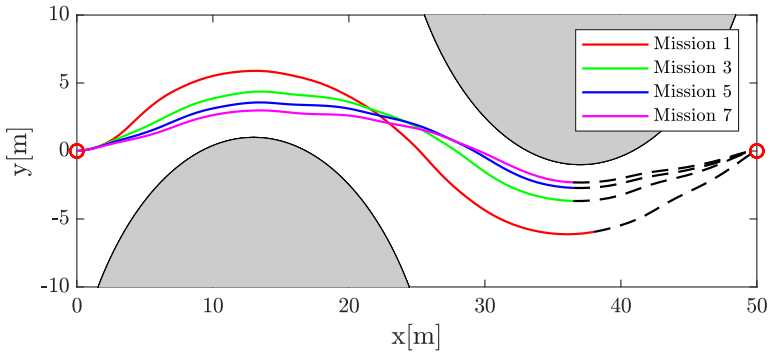


Figure 5: The path of the ship over missions.

Fig. 6 shows the surge u, sway v, and yaw r velocities over the missions. $\beta(t) = \arctan(\frac{v(t)}{u(t)})$ is the sideslip angle. The wind and ocean current disturbance and parametric uncertainties in the ship's model are effective factors in increasing the absolute value of this angle.

The control inputs (thruster forces) are provided in Fig. 7 for the nominal system. As it is observed, the propeller thrusters f_2 and f_3 work in their upper bounds as expected to reduce the cost of the route to the target point.

The back-off RL parameters $\boldsymbol{\theta}^h$ changes during the learning is demonstrated in Fig. 8. As can be seen, in the first mission, which has a large distance from the obstacles, the parameters are reduced in order to approach the obstacles until certain values.

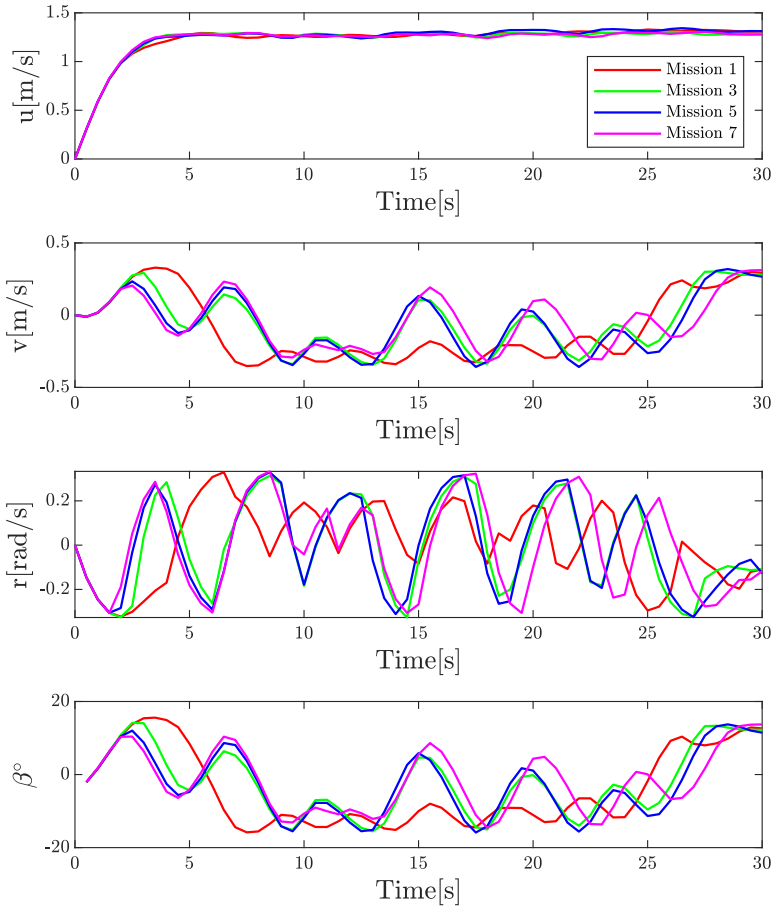


Figure 6: The surge u, sway v and yaw r velocities and sideslip angle β.

Finally, Fig. 9 illustrates the closed-loop performance of each mission. This performance is obtained by summing of baseline stage cost $L(\boldsymbol{s}, \boldsymbol{a})$ during each episode. As can be seen, the closed-loop performance is reduced by about 12% over seven missions.

5 Conclusion

This paper proposed an RL-based RMPC technique for controling an ASV in a target-tracking scenario in the presence of obstacles and stochastic wind and ocean currents. A parameterized scenario-tree-based MPC was used to approximate the action-value function, modelling a potential propeller thrusters failure. Additionally, constraint tightening was used in the MPC scheme to handle uncertain wind and current disturbances. The MPC tightening was adjustable

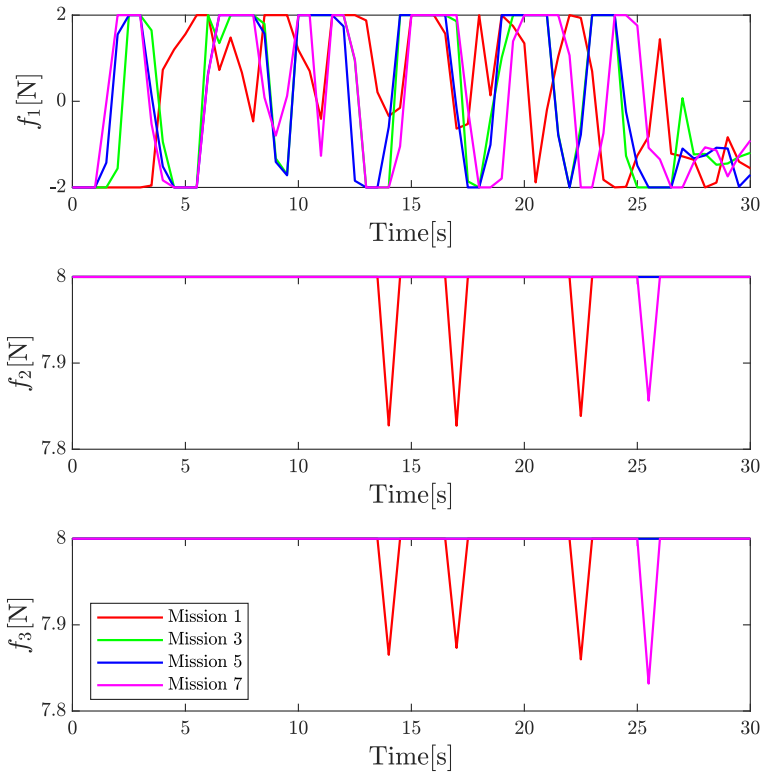


Figure 7: Actuator forces: f_2 and f_3 are the propeller, and f_1 is the tunnel thrusters.

by RL. A mixed energy and time cost was used as the RL's baseline cost, with the addition of a penalty when the ship's trajectory was too close to the obstacles. We started the mission with a conservative tightening, yielding a fairly large distance from the obstacles, and let RL adjust the tightening. The simulations show how RL manages to adjust the tightening to better values. The adaptation of more parameters in the MPC scheme will be considered in the future.

References

[1] Hossein Nejatbakhsh Esfahani, Rafal Szlapczynski, and Hossein Ghaemi. "High performance super-twisting sliding mode control for a maritime autonomous surface ship (MASS) using ADP-Based adaptive gains and time delay estimation". In: *Ocean Engineering* 191 (2019), p. 106526.

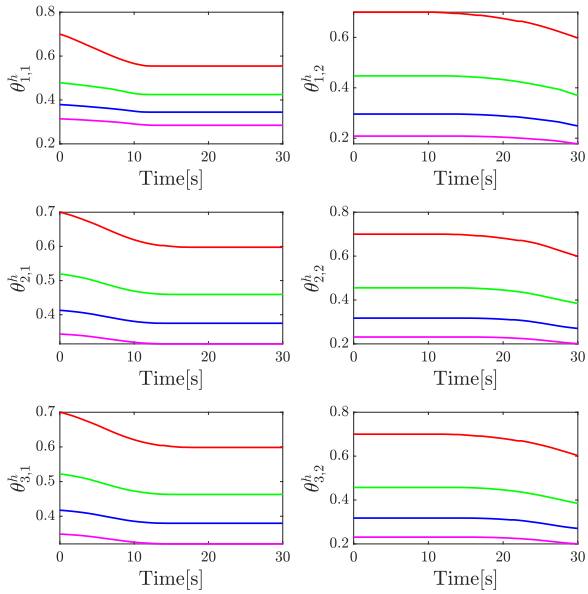


Figure 8: RL-parameters in constraint back-off $\boldsymbol{\theta}^h$. Red, Green, Blue, and Magenta for missions 1, 3, 5, and 7, respectively.

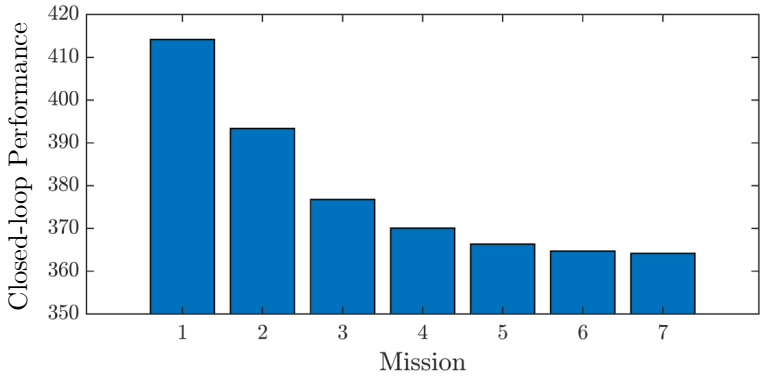


Figure 9: Histogram of Closed-loop performance over mission

[2] Andreas B Martinsen et al. "Optimization-Based Automatic Docking and Berthing of ASVs Using Exteroceptive Sensors: Theory and Experiments". In: *IEEE Access* 8 (2020), pp. 204974–204986.

[3] Glenn Bitar et al. "Two-Stage Optimized Trajectory Planning for ASVs Under Polygonal Obstacle Constraints: Theory and Experiments". In: *IEEE Access* 8 (2020), pp. 199953–199969.

[4] Joohyun Woo, Chanwoo Yu, and Nakwan Kim. “Deep reinforcement learning-based controller for path following of an unmanned surface vehicle”. In: *Ocean Engineering* 183 (2019), pp. 155–166.

[5] Andreas B. Martinsen et al. “Reinforcement Learning-Based Tracking Control of USVs in Varying Operational Conditions”. In: *Frontiers in Robotics and AI* 7 (2020), p. 32.

[6] Morten Breivik. “Topics in guided motion control of marine vehicles”. PhD thesis. Norwegian University of Science and Technology, 2010.

[7] Dimitri P Bertsekas. *Reinforcement learning and optimal control*. Athena Scientific Belmont, MA, 2019.

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

[9] Arash Bahari Kordabad and Mehrdad Boroushaki. “Emotional Learning Based Intelligent Controller for MIMO Peripheral Milling Process”. In: *Journal of Applied and Computational Mechanics* 6.3 (2020), pp. 480–492.

[10] Qingrui Zhang, Wei Pan, and Vasso Reppa. *Model-reference reinforcement learning for collision-free tracking control of autonomous surface vehicles*. 2020.

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

[12] Kai Wang et al. “Parallel Explicit Tube Model Predictive Control”. In: *IEEE 58th Conference on Decision and Control (CDC)*. IEEE. 2019, pp. 7696–7701.

[13] Tor A Johansen, Andrea Cristofaro, and Tristan Perez. “Ship collision avoidance using scenario-based model predictive control”. In: *IFAC-PapersOnLine* 49.23 (2016), pp. 14–21.

[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] M. Zanon and S. Gros. “Safe Reinforcement Learning Using Robust MPC”. In: *Transaction on Automatic Control, (accepted))*. https://arxiv.org/abs/1906.04005. 2021.

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

[17] Arash Bahari Kordabad, Wenqi Cai, and Sebastian Gros. “MPC-based Reinforcement Learning for Economic Problems with Application to Battery Storage”. In: *20th European Control Conference (ECC) (Accepted)*. IEEE. 2021.

[18] Hossein Nejatbakhsh Esfahani, Arash Bahari Kordabad, and Sébastien Gros. “Reinforcement Learning based on MPC/MHE for Unmodeled and Partially Observable Dynamics (accepted)”. In: *2021 American Control Conference (ACC)*. IEEE. 2021.

[19] 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.

[20] Thor I Fossen. *Handbook of marine craft hydrodynamics and motion control*. John Wiley & Sons, 2011.

[21] E. Klintberg et al. “An improved dual Newton strategy for scenario-tree MPC”. In: *IEEE 55th Conference on Decision and Control (CDC)*. 2016, pp. 3675–3681.

[22] Csaba Szepesvári. “Algorithms for reinforcement learning”. In: *Synthesis lectures on artificial intelligence and machine learning* 4.1 (2010).

[23] Sergio Lucia et al. “Handling uncertainty in economic nonlinear model predictive control: A comparative case study”. In: *Journal of Process Control* 24.8 (2014), pp. 1247–1259.

B MPC-based reinforcement learning for economic problems with application to battery storage

Postprint of [96] **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. DOI: 10.23919/ECC54610.2021.9654852

MPC-based Reinforcement Learning for Economic Problems with Application to Battery Storage

Arash Bahari Kordabad[1], Wenqi Cai[1], and Sebastien Gros[1]

[1]Department of Engineering Cybernetics, Norwegian University of Science and Technology (NTNU), Trondheim, Norway.

Abstract: In this paper, we are interested in optimal control problems with purely economic costs, which often yield optimal policies having a (nearly) bang-bang structure. We focus on policy approximations based on Model Predictive Control (MPC) and the use of the deterministic policy gradient method to optimize the MPC closed-loop performance in the presence of unmodelled stochasticity or model error. When the policy has a (nearly) bang-bang structure, we observe that the policy gradient method can struggle to produce meaningful steps in the policy parameters. To tackle this issue, we propose a homotopy strategy based on the interior-point method, providing a relaxation of the policy during the learning. We investigate a specific well-known battery storage problem and show that the proposed method delivers homogeneous and faster learning than a classical policy gradient approach.

1 Introduction

Making decisions for the energy system in the presence of different forms of uncertainty is the object of recent publications [1, 2]. In smart grids, uncertainty mainly arises from imperfect forecasts for the prices, demand, and power generation. Finding a policy minimizing the economic cost of operating the grid in the presence of these uncertainties is highly valuable [3]. Economic costs for smart grids are linear, based on the difference between the profit made by selling electricity to the power grid, and the losses incurred from buying it [4].

Reinforcement Learning (RL) offers tools for tackling Markov Decision Processes (MDP) without having an accurate knowledge of the probability distribution underlying the state transition [5, 6]. RL seeks to optimize the parameters underlying a given policy in view of minimizing the expected discounted sum of a given baseline stage cost $L(\boldsymbol{s}, \boldsymbol{a}) \in \mathbb{R}$, where $\boldsymbol{s}, \boldsymbol{a}$ are the system states and inputs.

RL methods are usually either directly based on an approximation of the optimal policy or indirectly based on an approximation of the action-value function. Policy gradient methods directly seek to find the optimal policy parameters [7, 8]. Different variants of Temporal Difference (TD) methods are at the core of many RL techniques for estimating the different value

functions associated with the MDP. Least-Squares Temporal-Difference (LSTD) techniques are widely used because of their reliability and efficient use of data [9].

Model Predictive Control (MPC) is a control strategy that employs a (possibly inaccurate) model of the real system dynamics to produce an input-state sequence over a given finite horizon such that the resulting predicted state trajectory minimizes a given cost function while explicitly enforcing the input-state constraints imposed on the system trajectories [10]. The problem is solved at each time instant, and only the first input of the input sequence is applied to the real system. By solving the entire problem at each time instant based on the current state of the system in a receding-horizon fashion, MPC delivers a policy for the real system.

For computational reasons, simple models are usually preferred in the MPC scheme. Hence, the MPC model often does not have the structure required to correctly capture the real system dynamics and stochasticity. As a result, MPC usually delivers a reasonable but suboptimal approximation of the optimal policy. Choosing the MPC parameters that maximize the closed-loop performance for the selected MPC formulation is a difficult problem. Indeed, e.g. selecting the MPC model parameters that best fit the model to the real system is not guaranteed to yield the best closed-loop performance that the MPC scheme can achieve [11]. In [11, 12], it is shown that adjusting the MPC model, cost and constraints can be beneficial to achieve the best closed-loop performances, and RL is proposed as a possible approach to perform that adjustment in practice. Further recent research have focused on MPC-based policy approximation for RL [12–15].

MPC is a promising choice for the management of smart grids [2], because it provides a simple way to exploit forecasts on the grid prices, local power demand, and production while respecting the physical limitations of the system. The stochasticity of the forecast uncertainty is, however, not straightforward to treat at low computational costs. In this paper, we investigate a simple, well-known battery storage problem having a purely economic cost and stochastic dynamics. This example has an optimal policy with a nearly bang-bang structure [16], in the sense that the optimal policy selects inputs that are either in the bounds or zero for a large subset of the state space. We show that the deterministic policy gradient method is difficult to use for this type of problem because the state trajectories mostly lie in the set where the policy is trivially zero or in bounds, which impedes the learning.

In this paper we propose a homotopy strategy based on the interior-point method [17], which smoothens the MPC policy via the barrier parameter associated with the method, allowing for more homogeneous and faster learning. The policy smoothing is gradually removed over the learning to recover the optimal policy. The paper is structured as follows. Section 2 presents the battery storage dynamics and provides its optimal policy of an economic cost. Section 3 formulates the LSTD-based deterministic policy gradient method. Section 4 details the use of MPC-scheme as a function approximator in RL. The difficulties of applying the policy gradient method for (nearly) bang-bang policies is analyzed. And the main contribution of this paper is presented. Section 5 provides the simulation results for the proposed approach and compares it with the classical implementation of the policy gradient methods. Finally, section 6 delivers the conclusions.

2 A simple motivational example

Photovoltaic (PV) battery systems allow households to participate in a more sustainable energy system [2]. The local electric demand is covered by the PV battery system or the connection to the public distribution grid. A simple model for the battery storage reads as [1]:

$$\boldsymbol{s}_{k+1} = \boldsymbol{s}_k + \alpha \left(\Delta_k + \boldsymbol{a}_k \right), \tag{1}$$

where $\boldsymbol{s}_k \in [0, 1]$ is the State-of-Charge (SOC) of the battery and the interval $[0, 1]$ represents the SOC levels considered as non-damaging for the battery (typically 20%-80% range of the physical SOC). Constant α is a positive value that reflects the battery size. Variable $\Delta_k \sim \mathcal{N}\left(\bar{\delta}^X, \sigma^X\right)$ is the difference between the local power production and demand, which– for the sake of simplicity–is considered as a Normal centered random variable here, where $\bar{\delta}^X$ and σ^X are the mean and variance of the Gaussian distribution. Input $\boldsymbol{a}_k \in [-\bar{U}, \bar{U}]$ is the power bought from (for $\boldsymbol{a}_k > 0$) and sold to (for $\boldsymbol{a}_k < 0$) the power grid. The economic stage cost can be written as follows:

$$L(\boldsymbol{s}_k, \boldsymbol{a}_k) = \begin{cases} \phi_b \boldsymbol{a}_k & \text{if} \quad \boldsymbol{a}_k \geq 0 \\ \phi_s \boldsymbol{a}_k & \text{if} \quad \boldsymbol{a}_k < 0 \end{cases}, \tag{2}$$

where $\phi_b \geq 0$ is the buying price and $\phi_s \geq 0$ is the selling price, and we assume that $\phi_b \geq \phi_s$. For the sake of simplicity, we consider the prices ϕ_b and ϕ_s as constants. Appendix 6 provides the model parameters we use in this paper. More complex models will be considered in the future.

In the deterministic policy gradient context, the optimal policy can be defined as follows:

$$\boldsymbol{\pi}^\star = \arg\min_{\boldsymbol{\pi}} \mathbb{E}_{\boldsymbol{\pi}} \left[\sum_{k=0}^{\infty} \gamma^k \tilde{L}\left(\boldsymbol{s}_k, \boldsymbol{a}_k\right) \middle| \boldsymbol{a}_k = \boldsymbol{\pi}(\boldsymbol{s}_k) \right], \tag{3}$$

where $\gamma \in (0, 1]$ is the discount factor, and for the battery storage dynamics (1) with stage cost (2), the modified stage cost $\tilde{L}(\boldsymbol{s}_k, \boldsymbol{a}_k)$ is defined as follows [8]:

$$\tilde{L}(\boldsymbol{s}_k, \boldsymbol{a}_k) = L(\boldsymbol{s}_k, \boldsymbol{a}_k) + p \max(\boldsymbol{s}_k - 1, 0) + p \max(-\boldsymbol{s}_k, 0), \tag{4}$$

where p is a large constant. The expected value $\mathbb{E}_{\boldsymbol{\pi}}$ is taken over the Markov Chain distribution resulting from the real system in closed-loop with policy $\boldsymbol{\pi}$. Since the state $\boldsymbol{s}_k$ ought to stay in the interval $[0, 1]$, a large penalty is introduced in the RL stage cost for $\boldsymbol{s}_k \notin [0, 1]$.

The example is selected such that its optimal policy $\boldsymbol{\pi}^\star$ can be solved via Dynamic Programming (DP), see fig. 1, and used as a baseline to assess the policies learned via RL. As can be seen in fig. 1, the optimal policy has a bang-bang-like structure. When the battery is at $\boldsymbol{s} \approx 0$, maximum buying is the optimal policy. Then for a fairly large subset of the states ($\boldsymbol{s} \approx [0.05, 0.5]$), no exchange with the grid is the optimal policy. For a high SOC ($\boldsymbol{s} \approx [0.55, 1]$), maximum selling is optimum.

Note that the computational complexity makes DP unrealistic for systems more complex than this example. Instead, most investigations in RL (e.g., policy gradient methods) focus on

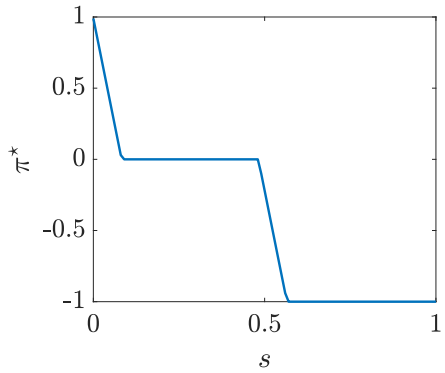


Figure 1: Optimal policy $\boldsymbol{\pi}^\star$ resulted from DP.

achieving approximate solutions, which do not require a model of the dynamics. The next section details the RL algorithm that obtains an optimal policy based on the observed data from the (stochastic) real system.

3 Deterministic policy gradient method

In the context of the deterministic policy gradient method [8], the policy $\boldsymbol{\pi_\theta}$ is parameterized by parameters $\boldsymbol{\theta}$, which are optimized directly according to the closed-loop performance using the gradient of the performance J defined as:

$$J(\boldsymbol{\pi_\theta}) = \mathbb{E}_{\boldsymbol{\pi_\theta}}\left[\sum_{k=0}^{\infty} \gamma^k \tilde{L}\left(\boldsymbol{s}_k, \boldsymbol{a}_k\right) \middle| \boldsymbol{a}_k = \boldsymbol{\pi_\theta}(\boldsymbol{s}_k)\right]. \tag{5}$$

The gradient of J with respect to parameters $\boldsymbol{\theta}$ is obtained as follows:

$$\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\pi_\theta}) = \mathbb{E}_{\boldsymbol{\pi_\theta}}\left[\nabla_{\boldsymbol{\theta}} \boldsymbol{\pi_\theta}(\boldsymbol{s}) \nabla_{\boldsymbol{a}} A_{\boldsymbol{\pi_\theta}}(\boldsymbol{s}, \boldsymbol{a})|_{\boldsymbol{a}=\boldsymbol{\pi_\theta}}\right], \tag{6}$$

where $A_{\boldsymbol{\pi_\theta}}(\boldsymbol{s}, \boldsymbol{a}) = Q_{\boldsymbol{\pi_\theta}}(\boldsymbol{s}, \boldsymbol{a}) - V_{\boldsymbol{\pi_\theta}}(\boldsymbol{s})$ is the advantage function associated to $\boldsymbol{\pi_\theta}$, and where $Q_{\boldsymbol{\pi_\theta}}$ and $V_{\boldsymbol{\pi_\theta}}$ are the action-value and value functions for the policy $\boldsymbol{\pi_\theta}$, respectively. Under some conditions detailed in [8], the action-value function $Q_{\boldsymbol{\pi_\theta}}$ in (6) can be replaced by an approximation $Q_{\boldsymbol{w}}$ without affecting the policy gradient. Such an approximation is labelled *compatible* and can, e.g., take the form:

$$Q_{\boldsymbol{w}}\left(\boldsymbol{s}, \boldsymbol{a}\right) = \left(\boldsymbol{a} - \boldsymbol{\pi_\theta}\left(\boldsymbol{s}\right)\right)^\top \nabla_{\boldsymbol{\theta}} \boldsymbol{\pi_\theta}(\boldsymbol{s})^\top \boldsymbol{w} + V_{\boldsymbol{v}}\left(\boldsymbol{s}\right), \tag{7}$$

where $\boldsymbol{w}$ is a parameters vector estimating the action-value function and $V_{\boldsymbol{v}} \approx V_{\boldsymbol{\pi_\theta}}$ is a baseline function approximating the value function, which can, e.g., take a linear form:

$$V_{\boldsymbol{v}}\left(\boldsymbol{s}\right) = \Phi\left(\boldsymbol{s}\right)^\top \boldsymbol{v}, \tag{8}$$

where Φ is a state feature vector and $\boldsymbol{v}$ is the corresponding parameters vector. The parameters $\boldsymbol{w}$ and $\boldsymbol{v}$ of the action-value function approximation (7) ought to be the solution of the Least Squares problem:

$$\min_{\boldsymbol{w},\boldsymbol{v}} \mathbb{E}\left[\left(Q_{\boldsymbol{\pi_\theta}}(\boldsymbol{s},\boldsymbol{a}) - Q_{\boldsymbol{w}}(\boldsymbol{s},\boldsymbol{a})\right)^2\right]. \tag{9}$$

In this paper, problem (9) is tackled via Least Squares Temporal Difference (LSTD) [9].

Next section details using an MPC scheme to approximate the optimal policy and proposes a smoothing approach based on the interior-point method for the (nearly) bang-bang policies.

4 MPC-based RL

Using MPC as a way of supporting the approximations of the value function, action-value function, and policy $\boldsymbol{\pi_\theta}$ has been proposed and justified in [11]. In this paper, we focus on the approximation of the optimal policy. Consider the following MPC scheme parameterized with $\boldsymbol{\theta}$:

$$\begin{aligned}
\min_{\boldsymbol{x},\boldsymbol{u},\boldsymbol{\sigma}} \quad & T_{\boldsymbol{\theta}}\left(\boldsymbol{x}_N\right) + \boldsymbol{\omega}_{\mathrm{f}}^{\top}\boldsymbol{\sigma}_N && \text{(10a)}\\
& + \sum_{i=0}^{N-1} \gamma^i(\ell_{\boldsymbol{\theta}}\left(\boldsymbol{x}_i,\boldsymbol{u}_i\right) + \boldsymbol{\omega}^{\top}\boldsymbol{\sigma}_i) \\
\text{s.t.} \quad & \boldsymbol{x}_{i+1} = \boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x}_i,\boldsymbol{u}_i), \quad \boldsymbol{x}_0 = \boldsymbol{s} && \text{(10b)}\\
& \boldsymbol{h}_{\boldsymbol{\theta}}(\boldsymbol{x}_i,\boldsymbol{u}_i) \leq \boldsymbol{\sigma}_i, \quad \boldsymbol{h}^{\mathrm{f}}_{\boldsymbol{\theta}}(\boldsymbol{x}_N) \leq \sigma_N && \text{(10c)}\\
& \boldsymbol{g}(\boldsymbol{u}_i) \leq 0, \quad \boldsymbol{\sigma}_i \geq 0, \quad \boldsymbol{\sigma}_N \geq 0, && \text{(10d)}
\end{aligned}$$

where $T_{\boldsymbol{\theta}}$ and $\ell_{\boldsymbol{\theta}}$ are the MPC terminal and stage costs, respectively. Function $\boldsymbol{f}_{\boldsymbol{\theta}}$ is the model dynamics, $\boldsymbol{g}$ is the pure input constraint and $\boldsymbol{h}_{\boldsymbol{\theta}}$ and $\boldsymbol{h}^{\mathrm{f}}_{\boldsymbol{\theta}}$ are the stage and terminal inequality constraints, respectively. Vectors $\boldsymbol{x} = \{\boldsymbol{x}_0,\ldots,\boldsymbol{x}_N\}$, $\boldsymbol{u} = \{\boldsymbol{u}_0,\ldots,\boldsymbol{u}_{N-1}\}$ and $\boldsymbol{\sigma} = \{\boldsymbol{\sigma}_0,\ldots,\boldsymbol{\sigma}_N\}$ are the primal decision variables, N is the prediction horizon and $\boldsymbol{s}$ is the current state of the system. Variables $\boldsymbol{\sigma}_i$ and $\boldsymbol{\sigma}_N$ are slacks for the relaxation of the state constraints, weighted by the positive vectors $\boldsymbol{\omega}$ and $\boldsymbol{\omega}_{\mathrm{f}}$. The relaxation prevents the infeasibility of the constraints of MPC in the presence of disturbances. The parameterized deterministic policy can be obtained as:

$$\boldsymbol{\pi_\theta}(\boldsymbol{s}) = \boldsymbol{u}_0^{\star}(\boldsymbol{s},\boldsymbol{\theta}), \tag{11}$$

where $\boldsymbol{u}_0^{\star}(\boldsymbol{s},\boldsymbol{\theta})$ is the first element of $\boldsymbol{u}^{\star}$, which is the solution of the MPC scheme (10).

Theoretically, under some assumptions detailed in [11], if the parametrization is rich enough, the MPC scheme is capable of capturing the optimal policy $\boldsymbol{\pi}^{\star}$ in the presence of disturbances and model error [11].

4.1 Primal-dual interior-point method

In the following, we will use the primal-dual interior-point method to solve the MPC scheme (10). Let us cast (10) as the generic Nonlinear Program (NLP):

$$\min_{\boldsymbol{z}} \quad \Psi_{\boldsymbol{\theta}}\left(\boldsymbol{z}\right) \tag{12a}$$

$$\text{s.t.} \quad \boldsymbol{G}_{\boldsymbol{\theta}}\left(\boldsymbol{z},\boldsymbol{s}\right)=0, \quad \boldsymbol{H}_{\boldsymbol{\theta}}\left(\boldsymbol{z}\right)\leq 0, \tag{12b}$$

where $\boldsymbol{z}=\{\boldsymbol{x},\boldsymbol{u},\boldsymbol{\sigma}\}$, function $\Psi_{\boldsymbol{\theta}}$ gathers the cost of (10), and $\boldsymbol{G}_{\boldsymbol{\theta}}$, $\boldsymbol{H}_{\boldsymbol{\theta}}$ are its equality and inequality constraints, respectively. We denote $\mathcal{L}_{\boldsymbol{\theta}}(\boldsymbol{y})=\Psi_{\boldsymbol{\theta}}+\boldsymbol{\lambda}^{\top}\boldsymbol{G}_{\boldsymbol{\theta}}+\boldsymbol{\mu}^{\top}\boldsymbol{H}_{\boldsymbol{\theta}}$ as the Lagrange function associated to (12), where $\boldsymbol{y}=\{\boldsymbol{z},\boldsymbol{\lambda},\boldsymbol{\mu}\}$ is the primal-dual variables vector, and where $\boldsymbol{\lambda}$ and $\boldsymbol{\mu}$ are the dual variables corresponding to the equality and inequality constraints, respectively. The primal-dual interior-point method is then based on the relaxed Karush–Kuhn–Tucker (KKT) conditions associated to (12) as follows:

$$\boldsymbol{r}\left(\boldsymbol{y},\boldsymbol{s},\boldsymbol{\theta}\right)=\begin{bmatrix}\nabla_{\boldsymbol{z}}\mathcal{L}_{\boldsymbol{\theta}}\left(\boldsymbol{y}\right)\\ \boldsymbol{G}_{\boldsymbol{\theta}}\left(\boldsymbol{z},\boldsymbol{s}\right)\\ \operatorname{diag}(\boldsymbol{\mu}_{\tau})\boldsymbol{H}_{\boldsymbol{\theta}}\left(\boldsymbol{z}\right)+\tau\mathbf{1}\end{bmatrix}, \tag{13}$$

and we denote its primal-dual solution by $\boldsymbol{y}_{\tau}=\{\boldsymbol{z}_{\tau},\boldsymbol{\lambda}_{\tau},\boldsymbol{\mu}_{\tau}\}$ for each $(\boldsymbol{s},\boldsymbol{\theta})$ pair, i.e:

$$\boldsymbol{r}\left(\boldsymbol{y}_{\tau},\boldsymbol{s},\boldsymbol{\theta}\right)=0, \tag{14}$$

where τ is the barrier parameter associated with the primal-dual interior-point method. Operator "diag" gathers the vector elements on the diagonal elements of a square matrix and $\mathbf{1}$ is a vector with unit elements and suitable size. If satisfying the Linear Independence Constraint Qualification (LICQ) and the Second Order Sufficient Condition (SOSC), $\boldsymbol{y}_{\tau}$ approximates a local solution of (12) at the order of $\mathcal{O}(\tau)$ [17].

4.2 Policy sensitivity

The policy gradient method requires one to compute $\nabla_{\boldsymbol{\theta}}\boldsymbol{\pi}_{\boldsymbol{\theta}}(\boldsymbol{s})$ for every state $\boldsymbol{s}$ encountered by the policy (see Eq. (6)). It is therefore crucial to be able to compute $\nabla_{\boldsymbol{\theta}}\boldsymbol{\pi}_{\boldsymbol{\theta}}$ from data efficiently. We ought to recall here that $\boldsymbol{\pi}_{\boldsymbol{\theta}}$ is given by the first element of the input profile included in $\boldsymbol{z}$, delivered by NLP (12). In this paper, we will replace that solution with its interior-point approximation $\boldsymbol{z}_{\tau}$. The problem of computing $\boldsymbol{\pi}_{\boldsymbol{\theta}}$ then becomes the problem of differentiating the parametric solution $\boldsymbol{z}_{\tau}\left(\boldsymbol{s},\boldsymbol{\theta}\right)$ of (14) with respect to $\boldsymbol{\theta}$. If the original NLP (12) satisfies LICQ and SOSC, then the sensitivity of $\boldsymbol{z}_{\tau}$ is readily given by the Implicit Function Theorem, i.e.:

$$\left(\frac{\partial\boldsymbol{r}}{\partial\boldsymbol{y}}\frac{\partial\boldsymbol{y}}{\partial\boldsymbol{\theta}}+\frac{\partial\boldsymbol{r}}{\partial\boldsymbol{\theta}}\right)\Bigg|_{\boldsymbol{y}=\boldsymbol{y}_{\tau}}=0 \tag{15}$$

holds. The policy sensitivity $\nabla_{\boldsymbol{\theta}}\boldsymbol{\pi}_{\boldsymbol{\theta}}$ can then be extracted from (15) as follows [11]:

$$\nabla_{\boldsymbol{\theta}}\boldsymbol{\pi}_{\boldsymbol{\theta}}\left(\boldsymbol{s}\right)=-\nabla_{\boldsymbol{\theta}}\boldsymbol{r}\left(\boldsymbol{y}_{\tau},\boldsymbol{s},\boldsymbol{\theta}\right)\nabla_{\boldsymbol{y}}\boldsymbol{r}(\boldsymbol{y}_{\tau},\boldsymbol{s},\boldsymbol{\theta})^{-1}\frac{\partial\boldsymbol{y}}{\partial\boldsymbol{u}_{0}} \tag{16}$$

4.3 Smoothing strategy for (nearly) bang-bang policies

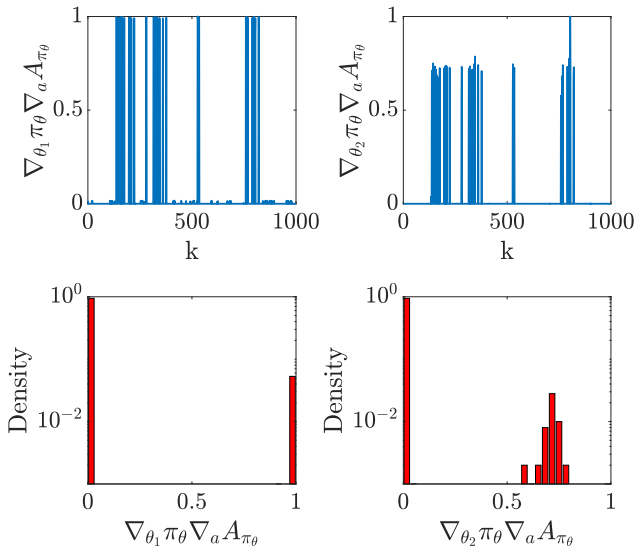
The solution of NLP (12) can be seen as a function of the NLP parameters $\boldsymbol{s}, \boldsymbol{\theta}$, and can be a non-differentiable or even discontinuous function of $\boldsymbol{s}, \boldsymbol{\theta}$ when changes of active set occur. In that context, parameter τ acts as a "smoothing" factor in the NLP solution, in the sense that for $\tau > 0$, the parametric solution $\boldsymbol{z}_\tau(\boldsymbol{s}, \boldsymbol{\theta})$ obtained from solving (14) becomes a smooth function of $\boldsymbol{s}, \boldsymbol{\theta}$. For $\tau \to 0$, $\boldsymbol{z}_\tau$ tends asymptotically to the non-smooth solution of NLP (12), and the derivatives of $\boldsymbol{z}_\tau$ can become unbounded for some $\boldsymbol{s}, \boldsymbol{\theta}$. In contrast, for τ larger, all derivatives of $\boldsymbol{z}_\tau$ remain bounded, and of lower magnitudes.

When the optimal policy has a (nearly) bang-bang structure–such as in the storage example investigated here–it is beneficial to adopt a policy approximation $\boldsymbol{\pi_\theta}$ that approximates that structure well while remaining smooth, such that the policy gradient (6) is guaranteed to be valid. If such a policy approximation can be made arbitrarily close to the bang-bang structure, then (6) remains asymptotically well-defined, and the approximation can approach the optimal policy.

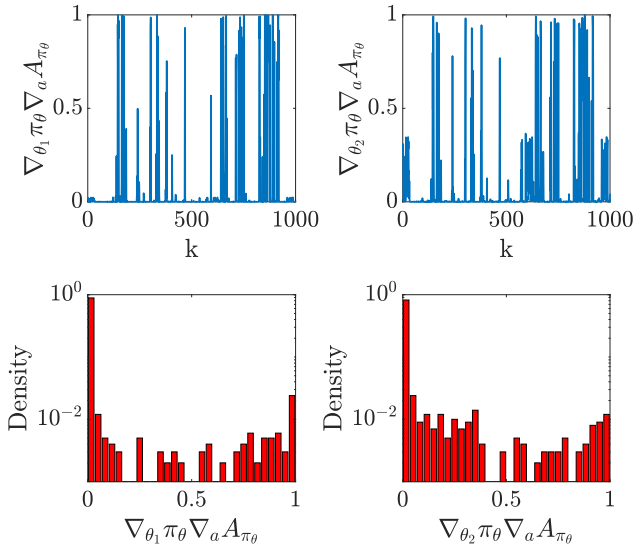
For non-episodic problems, such as the battery storage example considered here, the expected value operator $\mathbb{E}_{\boldsymbol{\pi_\theta}}[.]$ in the policy gradient (6) is meant to be taken over the steady-state distribution of the Markov Chain resulting from applying the policy $\boldsymbol{\pi_\theta}$ on the real system. If the MPC policy $\boldsymbol{\pi_\theta}$ has a purely bang-bang structure meant to approximate $\boldsymbol{\pi}^\star$, for $\tau \to 0$, the interior-point policy approximation is asymptotically bang-bang. Then, the gradient of the policy $\nabla_{\boldsymbol{\theta}} \boldsymbol{\pi_\theta}$, while remaining well-defined everywhere, tends to be nearly zero on large parts of the state space, and takes very large (asymptotically infinite) values when the policy switches between the different input levels. Hence, while the policy gradient (6) remains formally correct, evaluating it via sampling the distribution of the Markov Chain becomes very difficult, because the set of states where $\nabla_{\boldsymbol{\theta}} \boldsymbol{\pi_\theta} \approx 0$ has a measure close to unity, while $\nabla_{\boldsymbol{\theta}} \boldsymbol{\pi_\theta}$ is very large on a set of very small measure. As a result, sample-based estimations of (6) have a very large variance, which impedes the learning.

For nearly bang-bang policy structures, the difficulties can be less severe than for purely bang-bang structures but they remain an issue. That issue can be observed in the battery storage problem considered in this paper. Figure 2 shows the normalized $\nabla_{\boldsymbol{\theta}} \boldsymbol{\pi_\theta} \nabla_{\boldsymbol{a}} A_{\boldsymbol{\pi_\theta}}(\boldsymbol{s}, \boldsymbol{a})|_{\boldsymbol{a}=\boldsymbol{\pi_\theta}}$ for a given $\boldsymbol{\pi_\theta}$ during a fairly long closed-loop trajectory for different values of τ. Parameters $\boldsymbol{\theta} = [\theta_1, \theta_2]$ are the MPC parameters that will be introduced in detail in the simulation section. When using $\tau = 10^{-4}$, it can be seen from Fig. 2(a) that the gradients are very close to zero for almost every time instance, while they are fairly large at some states $\boldsymbol{s}_k$ that are very close to the switching conditions in the bang-bang policy. This observation is clear in the density plot, where it can be seen that the value of the gradient is either zero or takes large values, without intermediate values. This indicates that during the learning, most of the time the policy gradient evaluation $\nabla_{\boldsymbol{\theta}} J$ is close to zero and takes large values when state trajectories yield large contributions $\nabla_{\boldsymbol{\theta}} \boldsymbol{\pi_\theta} \nabla_{\boldsymbol{a}} A_{\boldsymbol{\pi_\theta}}(\boldsymbol{s}, \boldsymbol{a})|_{\boldsymbol{a}=\boldsymbol{\pi_\theta}}$. In contrast, for results of $\tau = 10^{-2}$ as displayed in Fig. 2(b), the distribution of the gradients is more uniform, avoiding the issues detailed above.

Figure 3 shows the MPC policy and the state distribution of the closed-loop system for

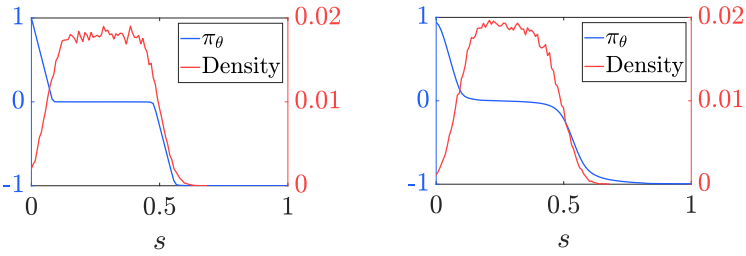


(a) $\tau = 10^{-4}$



(b) $\tau = 10^{-2}$

Figure 2: Normalized $\nabla_{\boldsymbol{\theta}}\boldsymbol{\pi}_{\boldsymbol{\theta}}\nabla_{\boldsymbol{a}}A_{\boldsymbol{\pi}_{\boldsymbol{\theta}}}(\boldsymbol{s},\boldsymbol{a})|_{\boldsymbol{a}=\boldsymbol{\pi}_{\boldsymbol{\theta}}}$ with respect to θ_1 and θ_2 of a closed-loop trajectory for different values of τ and their densities. (The densities are in logarithmic scale.)



(a) Non-smooth policy with state distribution for $\tau = 10^{-4}$

(b) Smooth policy with state distribution for $\tau = 10^{-2}$

Figure 3: MPC policy and state distribution of the closed-loop system

two different values of τ. It can be seen that for both τ, the state density is mainly in the interval where the policy is trivially zero, hence the state trajectories rarely visit the set where $\nabla_{\boldsymbol{\theta}} \boldsymbol{\pi}_{\boldsymbol{\theta}} \neq 0$. Besides, for $\tau = 10^{-4}$, the non-zero gradient occurs in a small subset of states. The policy $\boldsymbol{\pi}_{\boldsymbol{\theta}}$ tends to be the non-smooth solution of NLP (12) and the values of $\nabla_{\boldsymbol{\theta}} \boldsymbol{\pi}_{\boldsymbol{\theta}} \nabla_{\boldsymbol{a}} A_{\boldsymbol{\pi}_{\boldsymbol{\theta}}}$ are relatively large for those data collected around the switching conditions. In contrast, for larger τ ($\tau = 10^{-2}$), the policy $\boldsymbol{\pi}_{\boldsymbol{\theta}}$ is smoother. As a result, the values of sensitivity $\nabla_{\boldsymbol{\theta}} \boldsymbol{\pi}_{\boldsymbol{\theta}} \nabla_{\boldsymbol{a}} A_{\boldsymbol{\pi}_{\boldsymbol{\theta}}}$ remain bounded and with lower magnitude for large τ and they provide a meaningful gradient in a wider range of states compared with small τ.

In this paper, we exploit the smoothing effect of the barrier parameter τ to facilitate the use of the policy gradient method on MDPs that are difficult to treat because of the bang-bang-like structure of their optimal policy. More specifically, we propose to set the barrier parameter τ at large values at the beginning of the learning to smoothen the policy and facilitate the learning and decrease it–in a homotopy fashion– to small values as the learning progresses towards the optimal policy. We adopt a linearly varying τ here that decreases from a large value to the targeted $\bar{\tau}$ of the interior-point method, i.e.

$$\tau \leftarrow \max(\tau - \beta, \bar{\tau}) \tag{17}$$

where $\beta > 0$ is the progression step for τ, and $\bar{\tau}$ is the final barrier parameter targeted for the interior point method. The starting τ and target $\bar{\tau}$ are problem dependent. Alternative progression rules to (17) can clearly be considered, including more advanced adaptive strategies.

5 Simulation

In this section, we illustrate the difficulties encountered when using the LSTD-based policy gradient method to learn the nearly bang-bang optimal policy for the battery storage problem. We then demonstrate the proposed smoothing strategy as explained in section 4.3. We ought to stress here that, this example has a policy that is not fully bang-bang, which allows the

classical policy gradient method to work even without using the proposed technique. However, it requires significantly more RL steps and struggles with a high variance in the gradient estimation. A more extreme example with a pure bang-bang policy is likely to make the classic policy gradient method fail unless the proposed technique is used. Appendix 6 gives the parameters of the model and RL used in the simulations.

The explicit form of MPC scheme (10) used in the simulation is as follows:

$$\begin{aligned}
\min_{\boldsymbol{x},\boldsymbol{u},\boldsymbol{\sigma}} \quad & \theta_2^2(\boldsymbol{x}_{10}-0.5)^2+10\boldsymbol{\sigma}_{10} && \text{(18a)}\\
& +\sum_{i=0}^{9}(0.99)^i\left(L\left(\boldsymbol{x}_i,\boldsymbol{u}_i\right)+0.1\theta_1^2(\boldsymbol{x}_i-0.5)^2+10\boldsymbol{\sigma}_i\right)\\
\text{s.t.} \quad & \boldsymbol{x}_{i+1}=\boldsymbol{x}_i+1/12\boldsymbol{u}_i && \text{(18b)}\\
& \begin{bmatrix}\boldsymbol{x}_i-1\\ -\boldsymbol{x}_i\end{bmatrix}\leq\boldsymbol{\sigma}_i, \quad \boldsymbol{u}_i\in[-1,1] && \text{(18c)}\\
& \boldsymbol{x}_0=\boldsymbol{s}, \quad \boldsymbol{\sigma}_i\geq 0, \quad \boldsymbol{\sigma}_{10}\geq 0. && \text{(18d)}
\end{aligned}$$

We use quadratic stage and terminal costs with 0.5 as their reference points. Parameters θ_1 and θ_2 tune the curvature of the costs, and are squared to ensure their positive definiteness, i.e. $\boldsymbol{\theta}:=[\theta_1,\theta_2]^\top$. Based on our simulation results, this parameterization is sufficient to capture the optimal policy.

Figure 4 displays the policy improvement process for a fixed $\tau=10^{-4}$ using the LSTD-based policy gradient algorithm. Figure 6 (blue curves) displays the policy parameters over the learning. One can observe that the learning progresses very slowly for long periods of time, when the state evolves in regions where $\nabla_{\boldsymbol{\theta}}\boldsymbol{\pi}_{\boldsymbol{\theta}_k}\approx 0$, and undergoes some infrequent, sudden changes otherwise. One can see in Fig. 4 that the policy gradient manages to approximate the optimal policy $\boldsymbol{\pi}^\star$ well, but the convergence is uneven.

Figure 5 shows the policy improvement process resulting from τ starting at a relative large value and being progressively reduced to $\bar{\tau}$ using (17). The method starts with a large $\tau=10^{-2}$, and the target $\bar{\tau}$ is 10^{-4}. The step for decreasing τ is selected as $\beta=5\cdot 10^{-5}$. With this choice of τ, the policy is fairly smooth. The resulting learning can be seen in Fig. 6 (light red curves). One can observe a significantly faster progression of the parameters, with convergence in about 200 steps, as well as a significantly faster progression of the performance throughout the learning process, see Fig. 6 lower graph. Starting with a larger τ when the optimal policy approximation is still inaccurate and reducing τ during the learning allows for a better learning progression and a better performance, while still delivering a policy having the correct structure because τ is reduced to a small value eventually.

Hence, the proposed smoothing strategy not only accelerates learning but also solves the dilemma between the smoothness of the policy improvement process and the accuracy of the obtained policy.

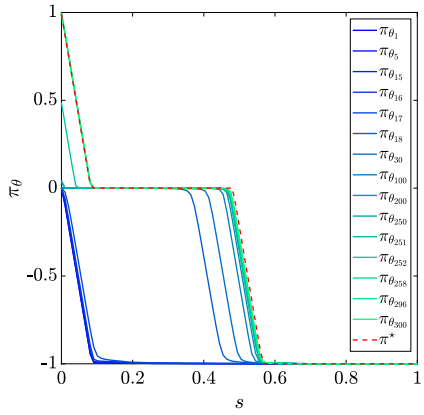


Figure 4: The policy improvement process of the policy gradient method during 300 steps with $\tau = 10^{-4}$.

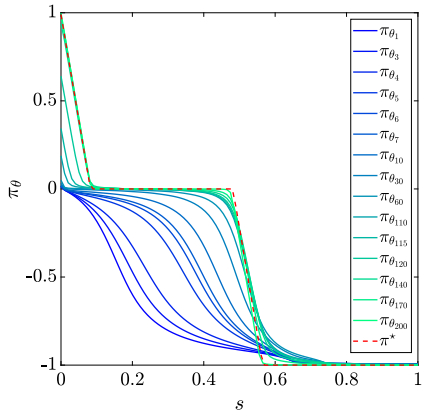


Figure 5: The policy improvement process of the policy gradient method during 200 steps with τ linearly decreases from 10^{-2} to 10^{-4}.

6 Conclusion

In this paper, we discuss the use of the policy gradient method on policies having (nearly) bang-bang structures supported via MPC schemes. We detail why this kind of policy structure is difficult to treat in the deterministic policy gradient context and propose a simple approach to alleviating the problem. A homotopy strategy is used to adapt the barrier parameter in the interior-point method that is used to solve the MPC scheme online. The proposed smoothing approach is illustrated on a classic battery storage problem with an economic stage cost. We show that a classical implementation of the policy gradient method results in a slow convergence, occurring through sudden progressions, while the proposed method offers a more

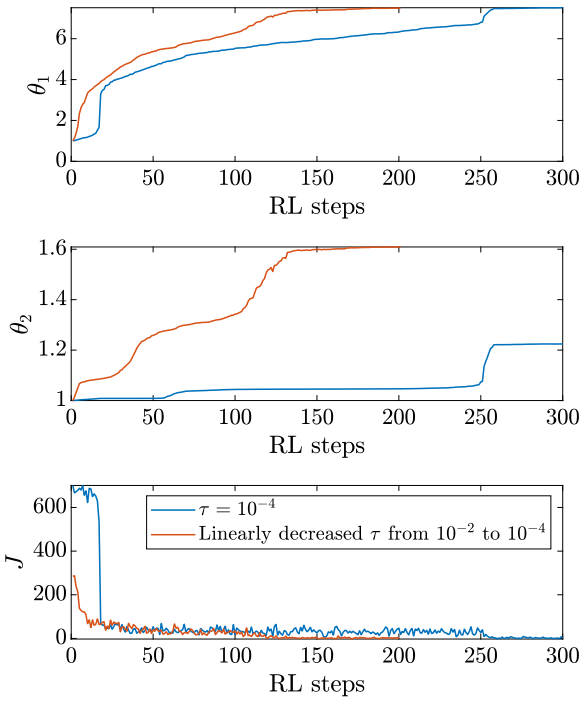


Figure 6: The evolutions of the policy parameters θ_1, θ_2, and the closed-loop performance J of the policy gradient method for the small $\tau = 10^{-4}$ and linearly decreased τ from 10^{-2} to 10^{-4}.

homogeneous and faster convergence, resulting in a better closed-loop performance throughout the learning process. In future work, we will consider more sophisticated techniques to adapt the barrier parameter, analyze the convergence more formally, and tackle challenging economic problems with complex models.

Appendices

Parameters of the dynamics and RL

Dynamics	ϕ_b	5
	ϕ_s	2.5
	α	1/12
	$\bar{U}$	1
	Δ	$\mathcal{N}(0, 0.05)$
RL	Φ	$\left[(s-0.5)^2, s, 1\right]^\top$
	p	1000

References

[1] Arne Groß et al. “Using Probabilistic Forecasts in Stochastic Optimization”. In: *2020 International Conference on Probabilistic Methods Applied to Power Systems (PMAPS)*. IEEE. 2020, pp. 1–6.

[2] Arne Groß, Christof Wittwer, and Moritz Diehl. “Stochastic Model Predictive Control of Photovoltaic Battery Systems using a Probabilistic Forecast Model”. In: *European Journal of Control* (2020).

[3] Warren B Powell and Stephan Meisel. “Tutorial on stochastic optimization in energy—Part I: Modeling and policies”. In: *IEEE Transactions on Power Systems* 31.2 (2015), pp. 1459–1467.

[4] Pavithra Harsha and Munther Dahleh. “Optimal management and sizing of energy storage under dynamic pricing for the efficient integration of renewable energy”. In: *IEEE Transactions on Power Systems* 30.3 (2014), pp. 1164–1181.

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

[6] Dimitri P Bertsekas. *Reinforcement learning and optimal control*. Athena Scientific Belmont, MA, 2019.

[7] Richard S Sutton et al. “Policy gradient methods for reinforcement learning with function approximation”. In: *Advances in neural information processing systems*. 2000, pp. 1057–1063.

[8] 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.

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

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

[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] Sebastien Gros and Mario Zanon. “Reinforcement learning for mixed-integer problems based on MPC”. In: *IFAC-PapersOnLine* 53.2 (2020), pp. 5219–5224.

[13] Torsten Koller et al. “Learning-based model predictive control for safe exploration”. In: *2018 IEEE Conference on Decision and Control (CDC)*. IEEE. 2018, pp. 6059–6066.

[14] Arash Bahari Kordabad et al. “Reinforcement Learning based on Scenario-tree MPC for ASVs”. In: *2021 American Control Conference (ACC)*. 2021, pp. 1985–1990.

[15] Hossein Nejatbakhsh Esfahani, Arash Bahari Kordabad, and Sébastien Gros. “Reinforcement learning based on MPC/MHE for unmodeled and partially observable dynamics”. In: *2021 American Control Conference (ACC)*. IEEE. 2021, pp. 2121–2126.

[16] Doron Lifshitz and George Weiss. “Optimal energy management for grid-connected storage systems”. In: *Optimal Control Applications and Methods* 36.4 (2015), pp. 447–462.

[17] Lorenz T Biegler. *Nonlinear programming: concepts, algorithms, and applications to chemical processes*. SIAM, 2010.

C Multi-agent Battery Storage Management using MPC-based Reinforcement Learning

Postprint of [97] **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. DOI: 10.1109/CCTA48906.2021.9659202

Multi-agent Battery Storage Management using MPC-based Reinforcement Learning

Arash Bahari Kordabad[1], Wenqi Cai[1], and Sebastien Gros[1]

[1]Department of Engineering Cybernetics, Norwegian University of Science and Technology (NTNU), Trondheim, Norway.

Abstract: In this paper, we present the use of Model Predictive Control (MPC) based on Reinforcement Learning (RL) to find the optimal policy for a multi-agent battery storage system. A time-varying prediction of the power price and production-demand uncertainty are considered. We focus on optimizing an economic objective cost while avoiding very low or very high state of charge, which can damage the battery. We consider the bounded power provided by the main grid and the constraints on the power input and state of each agent. A parametrized MPC scheme is used as a function approximator for the deterministic policy gradient method and RL optimizes the closed-loop performance by updating the parameters. Simulation results demonstrate that the proposed method is able to tackle the constraints and deliver the optimal policy.

1 Introduction

Increasingly many electricity consumers actively participate in the power system through bidirectional power trades [1]. In order to improve the efficiency of power transmission and the power quality, one of the key technologies is based on the Energy Storage Systems (ESS) [2]. A multi-agent battery storage system usually includes several batteries that are connected to a main grid. The main grid exchanges the power with all of the batteries and the batteries attempt to optimize their own cost. Since the total power exchanged by the main grid is limited at each time, finding an optimal policy that satisfies this restriction is challenging.

Making decisions for the power system to optimize an economic cost in the presence of different forms of uncertainties is the object of recent publications [3, 4]. In smart grids, uncertainties mainly arise from the imperfect forecasts of long-term prices and the power production demand. Reinforcement Learning (RL) offers tools for tackling Markov Decision Processes (MDP) without having an accurate knowledge of the probability distribution underlying the state transition [5, 6]. RL seeks to optimize the parameters underlying a given policy in view of minimizing the expected sum of a given stage cost. RL methods are usually either directly based on an approximation of the optimal policy or indirectly based on an approximation of the action-value function. Policy gradient methods directly attempt to find the optimal

policy parameters by optimizing the closed-loop performance. Q-learning and Least Squares Temporal Different (LSTD) are among the algorithms that capture the action-value function [7]. Regarding the approximation of the generic optimal policy and optimal action-value function, Fuzzy Neural Networks and Deep Neural Networks (DNNs) are common choices [8].

In the smart grid context, usually, there are reasonable forecasts of the statistics of the uncertainties and knowledge of the dynamics of the system. Therefore, using a structured function approximation such as Model Predictive Control (MPC) scheme can be beneficial. Indeed, MPC uses the predicted information and model to provide a reasonable but usually suboptimal policy [9]. Moreover, MPC is able to handle the high dimensionality of the forecasts. In [10], it is shown that adjusting the model, cost, and constraints of the MPC could achieve the best closed-loop performance, and RL is proposed as a possible approach to perform that adjustment in practice. Recent research have developed further the combination of RL and MPC (see e.g. [11–15]).

In this paper, considering the time-varying prediction of the spot market and the production-demand uncertainty, we use an MPC scheme to minimize the running cost of the system, while penalizing extreme State-of-Charge (SOC). A low-level controller monitors the SOC in real-time and prevents violating the constraints by buying or selling more power if needed [16]. We suppose that all the agents are connected to a main grid, and each battery stores or releases a limited amount of power at every time instant. The deterministic policy gradient method and the LSTD method are adopted to update the policy parameters and action-value parameters, respectively. The simulation results show that our proposed MPC-based RL method is capable of finding the optimal MPC parameters for the multi-agent battery storage system.

The rest of the paper is structured as follows. Section 2 provides the multi-agent battery storage dynamics and details the economic objective and constraints of the problem. Section 3 formulates the centralized MPC-scheme method via the MPC-based policy and it presents the policy gradient method that is used to find the optimal policy. Section 4 presents the simulations and section 5 delivers a conclusion.

2 Problem Formulation

In this section, we formulate the battery storage dynamics, the economic objective function with state constraints for a multi-agent system, and peak power constraints over time.

2.1 Dynamics

Photovoltaic (PV) battery systems allow households to participate in a more sustainable energy system ([3]). The battery storage dynamics can be written as the following linear system:

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

battery storage dynamics can be written as the following linear system:

$$\mathrm{soc}^i_{k+1} = \mathrm{soc}^i_k + \alpha^i \left(\Delta^i_k + b^i_k - s^i_k\right), \qquad (1)$$

where $i \in [1, \ldots, n]$ is the i^{th} battery, n is the number of batteries, subscript $k = 0, 1, \ldots$ denotes the physical time, $\mathrm{soc}^i_k \in [0, 1]$ is the State-of-Charge (SOC) of the battery and the interval $[0, 1]$ represents the SOC levels considered as non-damaging for the battery (typically 20%-80% range of the physical SOC). Constant α^i is a positive value that reflects the battery size. Process noise $\Delta^i_k \sim \mathcal{N}(\delta^i, \sigma^i)$ is the difference between the local power production-demand over the sampling time interval $[k, k+1]$, which for the sake of simplicity is considered as a Normal centred random variable, where δ^i and σ^i are the mean and variance of the Gaussian distribution. Input $b^i_k(s^i_k) \in [0, \bar{U}^i]$ is the power grid over time interval $[k, k+1]$, where $\bar{U}^i$ is the bound for the buying (selling) energy for the i^{th} battery. Fig. 1 illustrates the multi-agent battery system, where the batteries are connected to the main grid at point T.

where $i \in [1, \ldots, n]$ is the i^{th} battery, n is the number of batteries, subscript $k = 0, 1, \ldots$ denotes the physical time, $\mathrm{soc}^i_k \in [0, 1]$ is the State-of-Charge (SOC) of the battery and the interval $[0, 1]$ represents the SOC levels considered as non-damaging for the battery (typically 20%-80% range of the physical SOC). Constant α^i is a positive value that reflects the battery size. Process noise $\Delta^i_k \sim \mathcal{N}(\delta^i, \sigma^i)$ is the difference between the local power production-demand over the sampling time interval $[k, k+1]$, which for the sake of simplicity is considered as a Normal centred random variable, where δ^i and σ^i are the mean and variance of the Gaussian distribution. Input $b^i_k(s^i_k) \in [0, \bar{U}^i]$ is the power grid over time interval $[k, k+1]$, where $\bar{U}^i$ is the bound for the buying (selling) energy for the i^{th} battery. Fig. 1 illustrates the multi-agent battery system, where the sub-load batteries are connected to the main grid at point T.

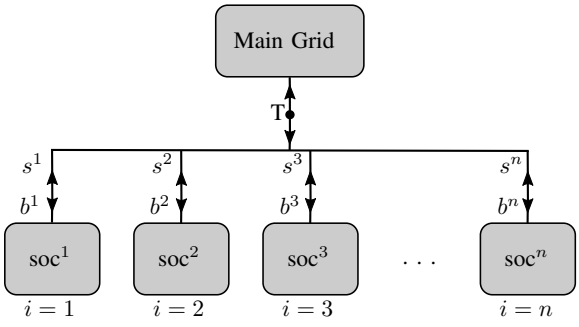


Figure 1: Multi-agent battery storage system
Fig. 1. Multi-agent battery storage system

B. Objective Function

Economic costs for smart grids are usually linear, based on the difference between the profit made by selling electricity to the power grid, and the losses incurred from buying it (see e.g., [18]). Hence, each battery has the following economic stage cost:

2.2 Objective Function

Economic costs for smart grids are usually linear, based on the difference between the profit made by selling electricity to the power grid, and the losses incurred from buying it (see e.g., [17]). Hence, each battery has the following economic stage cost:

$$L(b^i_k, s^i_k) = \phi^i_b b^i_k - \phi^i_s s^i_k, \qquad (2)$$

$$L(b^i_k, s^i_k) = \phi^i_b b^i_k - \phi^i_s s^i_k, \qquad (2)$$

where $\phi^i_b \geq 0$ and $\phi^i_s \geq 0$ are the (time-varying) buying and selling prices, respectively.

In the context of RL, we seek a control policy $\boldsymbol{\pi}$ that maps the state space to the input space and minimizes a closed-loop performance, which can be defined as an infinite-horizon expected sum stage costs. For the battery storage dynamics (1) with stage cost (2) and constraint $\mathrm{soc}^i \in [0, 1]$ for all $1 \leq i \leq n$, the modified stage cost $\tilde{L}$ for the centralized system can be defined as follows:

$$\tilde{L}(\boldsymbol{s}_k, \boldsymbol{a}_k) = \sum_{i=1}^{n} \Big(L(b^i_k, s^i_k) + p^i \max(\mathrm{soc}^i_k - 0.9, 0) + p^i \max(0.1 - \mathrm{soc}^i_k, 0) \Big), \qquad (3)$$

where p^i is a large constant that penalizes the state constraints within 10% of the bound $\mathrm{soc}_k^i \in [0,1]$. Vectors $\boldsymbol{s}_k = \mathrm{soc}_k^{1,\dots,n}$ and $\boldsymbol{a}_k = \{b_k^1 - s_k^1, \dots, b_k^n - s_k^n\}$ describe the entire system states and inputs vectors, respectively. A very low or very high state of charge decreases the battery lifetime [18]. Note that under optimality condition, the buying and selling variables can not be non-zero at the same time, then the difference of buying and selling $b_k^i - s_k^i$ can be considered as the input of the system ([11]). The closed-loop performance J reads as:

$$J(\boldsymbol{\pi}) = \mathbb{E}_{\boldsymbol{\pi}}\left[\sum_{k=0}^{\infty} \gamma^k \tilde{L}(\boldsymbol{s}_k, \boldsymbol{a}_k) \middle| \boldsymbol{a}_k = \boldsymbol{\pi}(\boldsymbol{s}_k)\right], \tag{4}$$

where $\gamma \in (0,1]$ is the discount factor and expectation $\mathbb{E}_{\boldsymbol{\pi}}$ is taken over the distribution of the Markov chain in closed-loop with policy $\boldsymbol{\pi}$. The performance for agent i^{th} is then defined as:

$$\begin{aligned} J^i(\boldsymbol{\pi}) = \mathbb{E}_{\boldsymbol{\pi}}\Bigg[\sum_{k=0}^{\infty} \gamma^k \Big(& L(b_k^i, s_k^i) + p^i \max(\mathrm{soc}_k^i - 0.9, 0) \\ & + p^i \max(0.1 - \mathrm{soc}_k^i, 0)\Big) \Bigg| \boldsymbol{a}_k = \boldsymbol{\pi}(\boldsymbol{s}_k)\Bigg]. \end{aligned} \tag{5}$$

2.3 Peak Power Constraint at point T

Electricity customers usually have different power demands during the day. In a multi-agent battery problem with a common main grid, optimizing the power peaks is critical. The methods for flattening the load curve are often called *peak shaving*. In order to formulate peak power constraints at point T (see Fig.1), we first define P_k as the maximum power amount exchanged with the main grid, i.e.:

$$P_k = \max\left(\sum_{i=1}^{n} b_k^i, \sum_{i=1}^{n} s_k^i\right). \tag{6}$$

Assume that P_k is restricted by the following upper bound over time:

$$P_k \leq \bar{P}\,, \quad \forall k \geq 0, \tag{7}$$

where $\bar{P} > 0$ is the maximum allowed power amount that can be exchanged with the main grid. The maximum grid power $\bar{P}$ is assumed to be less than the sum of the maximum exchanged power for each battery. i.e.:

$$\bar{P} < \sum_{i=1}^{n} \bar{U}^i\,. \tag{8}$$

Otherwise, (7) holds by construction.

Next section details the parametrization of the MPC scheme that is used as an approximator for the RL method and provides the policy gradient formulation to update the parameters.

3 MPC-based Deterministic Policy Gradient

Using MPC to support the approximations of the value function, the action-value function, and the policy has been proposed and justified in [10]. In this section, we detail this approach. We utilize the deterministic policy gradient method to adjust the MPC parameters and improve the closed-loop performance.

3.1 Centralized MPC-scheme

We focus on an MPC-based approximation of the optimal policy. RL is used to adjust the parameters $\boldsymbol{\theta}$ in the MPC scheme to handle model uncertainties and the process noise Δ^i. Furthermore, RL will tune the parameters so as to push the SOC to a safe region ($10\% - 90\%$ of the state of the charge). Note that the outside the interval $[0.1, 0.9]$ for soc, even if it is feasible, may damage the battery and reduce its lifetime. In order to provide an MPC-based policy approximator for RL, consider the following MPC scheme parameterized by $\boldsymbol{\theta}$:

$$\min_{\hat{\mathrm{soc}},\hat{b},\hat{s},\boldsymbol{\sigma}} \quad \sum_{i=1}^{n} \Bigg({\boldsymbol{\omega}_f^i}^\top \boldsymbol{\sigma}_N^i + T_{\boldsymbol{\theta}}(\hat{\mathrm{soc}}_N^i) + \sum_{j=0}^{N-1} \gamma^j \left(L_{\boldsymbol{\theta}}(\hat{b}_j^i, \hat{s}_j^i) + \phi_{\boldsymbol{\theta}}(\hat{\mathrm{soc}}_j^i) + {\boldsymbol{\omega}^i}^\top \boldsymbol{\sigma}_j^i \right) \Bigg) \tag{9a}$$

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

$$\hat{\mathrm{soc}}_{j+1}^i = \hat{\mathrm{soc}}_j^i + \theta_\alpha^i (\hat{b}_j^i - \hat{s}_j^i) + \theta_\delta^i, \tag{9b}$$

$$[\hat{\mathrm{soc}}_j^i - 0.9, 0.1 - \hat{\mathrm{soc}}_j^i]^\top \leq \boldsymbol{\sigma}_j^i, 0 \leq \boldsymbol{\sigma}_j^i \tag{9c}$$

$$[\hat{\mathrm{soc}}_N^i - 0.9, 0.1 - \hat{\mathrm{soc}}_N^i]^\top \leq \boldsymbol{\sigma}_N^i, 0 \leq \boldsymbol{\sigma}_N^i \tag{9d}$$

$$0 \leq \hat{b}_j^i \leq \bar{U}^i, \qquad 0 \leq \hat{s}_j^i \leq \bar{U}^i, \tag{9e}$$

$$\sum_{i=1}^{n} \hat{b}_j^i \leq \bar{P}, \qquad \sum_{i=1}^{n} \hat{s}_j^i \leq \bar{P}, \tag{9f}$$

$$\hat{\mathrm{soc}}_0^i = \mathrm{soc}_k^i, \tag{9g}$$

where $\hat{\mathrm{soc}} = \hat{\mathrm{soc}}_{0,\ldots,N}^{1,\ldots,n}$, $\hat{b} = \hat{b}_{0,\ldots,N-1}^{1,\ldots,n}$, $\hat{s} = \hat{s}_{0,\ldots,N-1}^{1,\ldots,n}$, $\boldsymbol{\sigma} = \boldsymbol{\sigma}_{0\ldots,N}^{1,\ldots,n}$ are the primal decision variables for the predicted state, buying, selling, and slacks, respectively. Subscript j is the MPC prediction step and N is the horizon length. We relax the stage and terminal state inequalities by the positive slack variables $\boldsymbol{\sigma}_j^i$ and $\boldsymbol{\sigma}_N^i$, and penalize them by positive constant weights $\boldsymbol{\omega}^i$ and $\boldsymbol{\omega}_f^i$, respectively. This prevents the infeasibility of the MPC in the presence of the process noise in the real system (1) out of the interval $[0.1, 0.9]$ for the states. Stage cost $\phi_{\boldsymbol{\theta}}$ and terminal cost $T_{\boldsymbol{\theta}}$ are the additional parametric costs, depending on the states that allow the MPC scheme (9) to provide a more generic function approximator. Moreover, because of the stochasticity of the real system and the existence of different uncertainties in the system, we select the parameterized economic cost $L_{\boldsymbol{\theta}}$ as a generic function different from the true L

in (2) and let RL adjust its parameters. Parameters θ_α^i and θ_δ^i, among the adjustable parameters $\boldsymbol{\theta}$, are dedicated to capture the model correction. We summarize (9) as follows:

- Cost (9a) includes the discounted economic cost $L_{\boldsymbol{\theta}}$, additional stage cost $\phi_{\boldsymbol{\theta}}$ and terminal cost $T_{\boldsymbol{\theta}}$ and penalty for the slack variables $\boldsymbol{\sigma}_j^i$ and $\boldsymbol{\sigma}_N^i$.
- Equality constraint (9b) represents the parameterization for the deterministic model of the real system (1).
- Inequality constraints (9c) and (9d) are the relaxed state constraints with positive slacks for each battery.
- Inequality constraint (9e) are the input constraints for each battery.
- Inequality constraints (9f) represent the power peak constraint for the grid.
- Equality constraint (9g) initializes the MPC-scheme at current state soc_k^i.

The parameterized deterministic policy for agent i at time k can be obtained as:

$$\pi_{\boldsymbol{\theta}}^i(\boldsymbol{s}_k) = \hat{b}_0^{i\star}(\boldsymbol{s}_k, \boldsymbol{\theta}) - \hat{s}_0^{i\star}(\boldsymbol{s}_k, \boldsymbol{\theta}), \tag{10}$$

where $\hat{b}_0^{i\star}$ and $\hat{s}_0^{i\star}$ are the first elements of $\hat{b}^{i\star}$ and $\hat{s}^{i\star}$, which are the solutions of the MPC scheme (9) associated to the decision variables $\hat{b}^i$ and $\hat{s}^i$. Then the parametric centralized policy extracted from the MPC-scheme (9) is written as follows:

$$\boldsymbol{\pi}_{\boldsymbol{\theta}}(\boldsymbol{s}_k) = [\pi_{\boldsymbol{\theta}}^1, \ldots, \pi_{\boldsymbol{\theta}}^n]^\top \tag{11}$$

The input $\boldsymbol{a}_k$ is selected according to the corresponding parametric policy $\boldsymbol{\pi}_{\boldsymbol{\theta}}$ in (11) with the possible addition of small random exploration.

3.2 Low-level Control

In the smart grid context, there is usually a low-level control that monitors the current state of charge (which has a 1h sampling time) and power demand/production. If the states tend to violate the constraints $\mathrm{soc}_k^i \in [0, 1]$, the low-level control (which works at a lower sampling time, e.g., every second) would decide to buy or sell more power to keep the states in the feasible interval [16, 19].

3.3 Deterministic Policy Gradient Method

The deterministic policy gradient method optimizes the policy parameters directly via gradient descent steps on the performance function J, defined in (4). The update rule is as follows:

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \alpha \nabla_{\boldsymbol{\theta}} J(\boldsymbol{\pi}_{\boldsymbol{\theta}}), \tag{12}$$

where $\alpha > 0$ is the step size. Applying the deterministic policy gradient method, developed by [20], the gradient of J with respect to parameters $\boldsymbol{\theta}$ is obtained as:

$$\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\pi_\theta}) = \mathbb{E}\left[\nabla_{\boldsymbol{\theta}} \boldsymbol{\pi_\theta}(\boldsymbol{s}) \nabla_{\boldsymbol{a}} A_{\boldsymbol{\pi_\theta}}(\boldsymbol{s}, \boldsymbol{a})|_{\boldsymbol{a}=\boldsymbol{\pi_\theta}}\right], \tag{13}$$

where $A_{\boldsymbol{\pi_\theta}}(\boldsymbol{s}, \boldsymbol{a}) = Q_{\boldsymbol{\pi_\theta}}(\boldsymbol{s}, \boldsymbol{a}) - V_{\boldsymbol{\pi_\theta}}(\boldsymbol{s})$ is the *advantage function* associated to $\boldsymbol{\pi_\theta}$, and where $Q_{\boldsymbol{\pi_\theta}}$ and $V_{\boldsymbol{\pi_\theta}}$ are the action-value function and value function of the policy $\boldsymbol{\pi_\theta}$, respectively, defined as follows:

$$Q_{\boldsymbol{\pi_\theta}}(\boldsymbol{s}, \boldsymbol{a}) = \tilde{L}(\boldsymbol{s}, \boldsymbol{a}) + \gamma \mathbb{E}\left[V_{\boldsymbol{\pi_\theta}}\left(\boldsymbol{s}^+ | \boldsymbol{s}, \boldsymbol{a}\right)\right] \tag{14a}$$

$$V_{\boldsymbol{\pi_\theta}}(\boldsymbol{s}) = Q_{\boldsymbol{\pi_\theta}}(\boldsymbol{s}, \boldsymbol{\pi_\theta}(\boldsymbol{s})), \tag{14b}$$

where $\boldsymbol{s}^+$ is the subsequent state of the state-input pair $(\boldsymbol{s}, \boldsymbol{a})$. Under some conditions [20], the action-value function $Q_{\boldsymbol{\pi_\theta}}$ in (13) can be replaced by an approximator $Q_{\boldsymbol{w}}$ without affecting the policy gradient. Such an approximation is labelled *compatible* and can, e.g., take the form:

$$Q_{\boldsymbol{w}}(\boldsymbol{s}, \boldsymbol{a}) = (\boldsymbol{a} - \boldsymbol{\pi_\theta}(\boldsymbol{s}))^\top \nabla_{\boldsymbol{\theta}} \boldsymbol{\pi_\theta}(\boldsymbol{s})^\top \boldsymbol{w} + V^{\boldsymbol{v}}(\boldsymbol{s}), \tag{15}$$

where $\boldsymbol{w}$ is a parameter vector estimating the action-value function $Q_{\boldsymbol{\pi_\theta}}$ and $V^{\boldsymbol{v}} \approx V_{\boldsymbol{\pi_\theta}}$ is a baseline function approximating the value function. The parameterized value function $V^{\boldsymbol{v}}$ can, e.g., take the linear form:

$$V^{\boldsymbol{v}}(\boldsymbol{s}) = \Phi(\boldsymbol{s})^\top \boldsymbol{v}, \tag{16}$$

where $\Phi(\boldsymbol{s})$ is a state feature vector and $\boldsymbol{v}$ is the corresponding parameter vector. The parameters $\boldsymbol{w}$ and $\boldsymbol{v}$ of the action-value function approximation (15) ought to be the solution of the Least Squares (LS) problem:

$$\min_{\boldsymbol{w}, \boldsymbol{v}} \mathbb{E}\left[\left(Q_{\boldsymbol{\pi_\theta}}(\boldsymbol{s}, \boldsymbol{a}) - Q_{\boldsymbol{w}}(\boldsymbol{s}, \boldsymbol{a})\right)^2\right]. \tag{17}$$

In this paper, the LS problem in (17) is tackled via the Least Squares Temporal Difference (LSTD) method (see e.g., [7]) based on the stage cost $\tilde{L}$. LSTD belongs to *batch method*, seeking to find the best fitting value function and action-value function, and it is more sample efficient than other methods.

The primal-dual Karush–Kuhn–Tucker (KKT) conditions underlying the MPC scheme (9) is written as:

$$\boldsymbol{R} = \left[\begin{array}{cccc} \nabla_{\boldsymbol{\xi}} \mathcal{L}_{\boldsymbol{\theta}} & \boldsymbol{G_\theta} & \operatorname{diag}(\boldsymbol{\mu}) \boldsymbol{H_\theta} \end{array}\right]^\top, \tag{18}$$

where $\boldsymbol{\xi} = \{\hat{\mathrm{soc}}, \hat{b}, \hat{s}, \boldsymbol{\sigma}\}$ is the primal decision variable. Operator "diag" assigns the vector elements onto the diagonal position of a square matrix. $\mathcal{L}_{\boldsymbol{\theta}}$ is the associated Lagrange function of the MPC (9), written as:

$$\mathcal{L}_{\boldsymbol{\theta}}(\boldsymbol{y}) = \Psi_{\boldsymbol{\theta}} + \boldsymbol{\lambda}^\top \boldsymbol{G}_\theta + \boldsymbol{\mu}^\top \boldsymbol{H}_{\boldsymbol{\theta}}, \tag{19}$$

where Ψ_θ is the MPC cost (9a), $\boldsymbol{G}_\theta$ gathers the equality constraints and $\boldsymbol{H}_\theta$ collects the inequality constraints of the MPC (9). Vectors $\boldsymbol{\lambda}, \boldsymbol{\mu}$ are the associated dual variables. Argument

$\boldsymbol{y}$ reads as $\boldsymbol{y} = \{\boldsymbol{\xi}, \boldsymbol{\lambda}, \boldsymbol{\mu}\}$ and $\boldsymbol{y}^\star$ refers to the solution of the MPC (9). The policy sensitivity $\nabla_{\boldsymbol{\theta}} \boldsymbol{\pi}_{\boldsymbol{\theta}}$ required in Eq. (13) can then be obtained as follows ([10]):

$$\nabla_{\boldsymbol{\theta}} \boldsymbol{\pi}_{\boldsymbol{\theta}}(\boldsymbol{s}) = -\nabla_{\boldsymbol{\theta}} \boldsymbol{R}(\boldsymbol{y}^\star, \boldsymbol{s}, \boldsymbol{\theta}) \nabla_{\boldsymbol{y}} \boldsymbol{R}(\boldsymbol{y}^\star, \boldsymbol{s}, \boldsymbol{\theta})^{-1} \frac{\partial \boldsymbol{y}}{\partial \boldsymbol{u}_0}, \tag{20}$$

where $\boldsymbol{u}_0$ is the first input variable, defined as follows:

$$\boldsymbol{u}_0 = [\hat{b}_0^1 - \hat{s}_0^1, \ldots, \hat{b}_0^n - \hat{s}_0^n]^\top. \tag{21}$$

The next section provides the simulation results of the proposed method for a simple configuration of the multi-agent battery storage system.

4 Simulation

In this section, we illustrate the simulation results of the MPC-based deterministic policy gradient method for a 3-agent battery storage problem.

The state feature $\Phi(\boldsymbol{s})$ for the value function approximator $V^{\boldsymbol{v}}(\boldsymbol{s})$ in (16) is selected as a vector of quadratic monomials as follows:

$$\Phi(\boldsymbol{s}) = \left[(\hat{\mathrm{soc}}^1)^2, (\hat{\mathrm{soc}}^2)^2, (\hat{\mathrm{soc}}^3)^2, \hat{\mathrm{soc}}^1, \hat{\mathrm{soc}}^2, \hat{\mathrm{soc}}^3, 1\right]^\top. \tag{22}$$

For the sake of simplicity, we don't consider the joint state effects in the value function.

The parameterized economic cost $L_{\boldsymbol{\theta}}$, additional stage cost $\phi_{\boldsymbol{\theta}}$, and terminal cost $T_{\boldsymbol{\theta}}$ in the MPC-scheme (9) are selected as follows:

$$L_{\boldsymbol{\theta}}(\hat{b}_j^i, \hat{s}_j^i) = (\phi_b^i + \theta_b^i)\hat{b}_j^i - (\phi_s^i + \theta_s^i)\hat{s}_j^i \tag{23a}$$

$$\phi_{\boldsymbol{\theta}}(\hat{\mathrm{soc}}_j^i) = \phi_1^i(\hat{\mathrm{soc}}_j^i)^2 + \phi_2^i \hat{\mathrm{soc}}_j^i + \phi_3^i \tag{23b}$$

$$T_{\boldsymbol{\theta}}(\hat{\mathrm{soc}}_N^i) = T_1^i(\hat{\mathrm{soc}}_N^i)^2 + T_2^i \hat{\mathrm{soc}}_N^i + T_3^i, \tag{23c}$$

where θ_b^i, θ_s^i, $\phi_{1,2,3}^i$, and $T_{1,2,3}^i$ are among the adjustable parameters $\boldsymbol{\theta}$, i.e:

$$\boldsymbol{\theta} = \{\theta_\alpha^{1,\ldots,n}, \theta_\delta^{1,\ldots,n}, \theta_b^{1,\ldots,n}, \theta_s^{1,\ldots,n}, \phi_{1,2,3}^{1,\ldots,n}, T_{1,2,3}^{1,\ldots,n}\}. \tag{24}$$

One can use more generic function approximators in (23), however, in [11], it shows that, for this kind of battery storage problem, quadratic parameterizations for the stage and terminal costs in the MPC-based policy approximator are rich enough to capture the optimal policy.

The rest of the parameter values used in the simulation are given in Table 1.

We use the time-varying power prices of Trondheim in the simulation, which is collected from the website provided by the Nord Pool European Power Exchange [21]. Fig. 2 illustrates the 24-hour buying price ϕ_b for five sampled days of Nov.2020. For the selling price ϕ_s, we use $\phi_s = 0.5\phi_s$ at every time step. Note that the prediction horizon is selected as $N = 12$, because the power prices are usually accessible for 12-hours ahead [21].

Table 1: Parameter values.

Symbol	Value	Symbol	Value
γ	0.99	n	3
Sampling time	1h	N	12
α^i	$1/12$	Δ^i	$\mathcal{N}(0, 0.5)$
$\bar{U}^i$	1	$\bar{P}$	1.5
$\boldsymbol{\omega}^i, \boldsymbol{\omega}_f^i$	$[20, 20]^\top$	p^i	1000
α	5e$-$8	soc_0^i	0.5

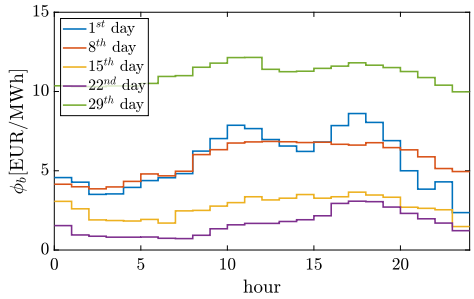


Figure 2: The 24-hour buying price ϕ_b of Trondheim for five sampled days in Nov.2020.

We run the simulation for 100 months. Each month we use a repetitive 30-days, where the states soc^i start from 0.6 at the beginning of the day, and we apply the time-varying prices and consider different stochasticity for each agent. We average along 30 days to approximate the expectations ($\mathbb{E}$) in the policy gradient (13) and LS (17), and update the parameters of the value function, action-value function, and policy at the end of each month.

Figure 3 shows the state and policy trajectories over time for each agent during the first and last month of the learning. The red trajectories show the states and policies for the first month. As can be seen, at the beginning month of the learning, the MPC scheme has not been learned yet and the states are sometimes in the position of lower than 10% of the SOC capacity, i.e, the soc^i of the three agents are sometimes below 10%. The blue trajectories correspond to the last month of the learning. It can be seen that with learning, RL pushes the states up so as to prevent being close to the bounds of the state constraints.

Figure 4 illustrates the norm of policy gradient $\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\pi}_{\boldsymbol{\theta}})$ over RL-steps. Since the existence of the process noise and random exploration, the gradient is noisy, but the overall behaviour is decreasing as the parameters approach their optimal points.

The variation of the closed-loop performance J is shown in Fig.5. It can be seen that 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 twenty months, the performance J drops faster in this range.

Figure 6 presents the maximum power amount P exchanged with the main grid for five

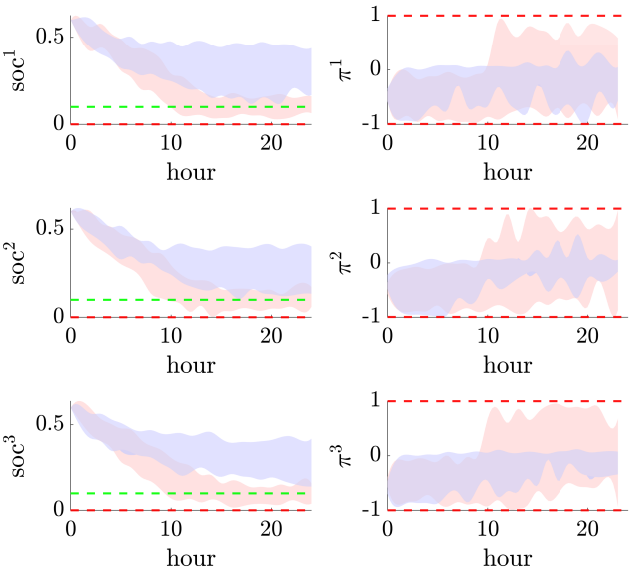


Figure 3: The state and policy trajectories over time for each agent. Red: the first month of learning, Blue: the last month of learning.

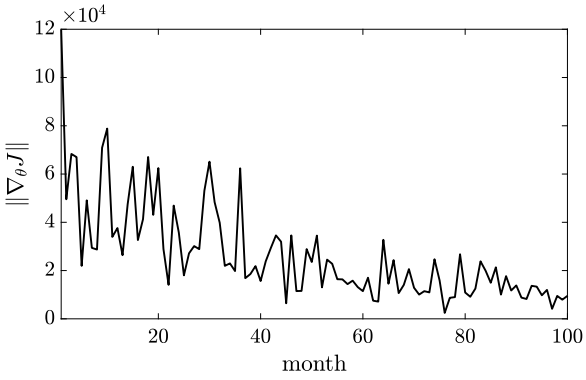


Figure 4: Norm of the policy gradient $\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\pi}_{\boldsymbol{\theta}})$ over RL-steps.

sampled days in the last learning month. As can be seen, the values of P comply with the upper bound constraints $\bar{P}$, which means the optimal policy we find can not only render the minimum economic cost for the whole system but also meet the power peak constraints on the main grid.

Figure 7 (Left) shows the learned policy for each agent after the last RL step. From the previous work ([11]), we know that the linear economic stage cost often yields a (nearly) bang-bang structure optimal policy when the battery dynamics are stochastic and linear. This figure

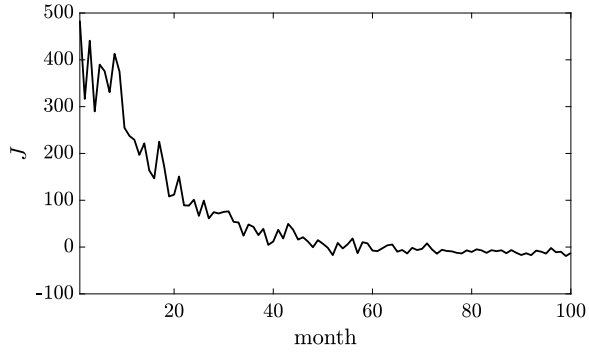


Figure 5: Closed-loop performance J over RL-steps.

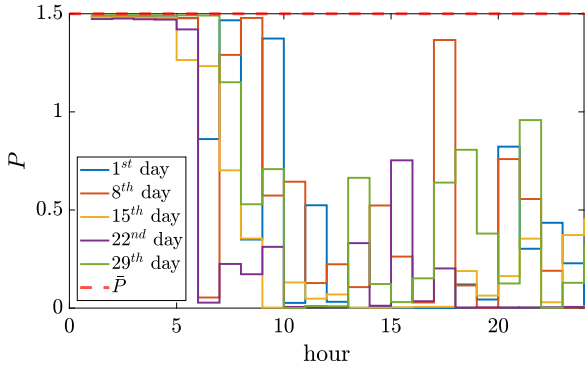


Figure 6: The maximum power amount P exchanged with the main grid for five sampled days.

demonstrates the similar optimal policy consequence as expected. Fig.7 (Right) illustrates the improvement of the closed-loop performance J^i for each agent during the learning.

Fig.8 illustrates the convergence of the parameter θ_δ^i. Note that there are 24 parameters in this simulation, and we select 3 representative parameters for the sake of brevity.

5 Conclusion

In this paper, we propose an MPC-based RL approach to seek for an optimal policy for the multi-agent battery storage system. The objective is to minimize an economic cost considering the battery health using penalties for the very low and high state of charge. We consider the production-demand uncertainty as well as the constraints for the peak power exchanged with the main grid. We parameterize an MPC scheme and use the deterministic policy gradient method to learn the optimal policy subject to the power peak constraints of the main grid. The

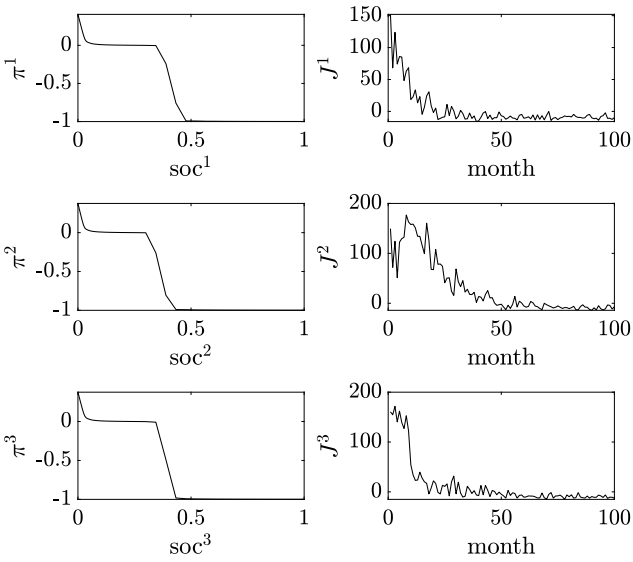


Figure 7: (Left) The learned policy of each agent. (Right) The closed-loop performance of each agent.

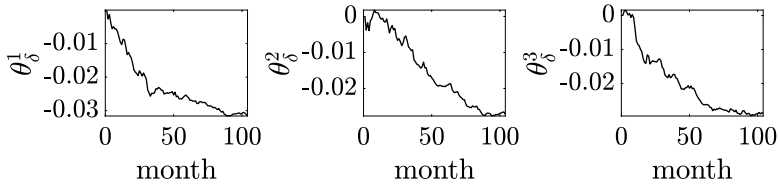


Figure 8: Convergence of one of the policy parameters θ^i_δ.

simulation results prove the feasibility of the proposed method. For future works, we will use decentralized learning on more comprehensive power systems, where the dynamics are more sophisticated and contain other uncertainties in the systems.

References

[1] Soon-Jeong Lee et al. “Coordinated control algorithm for distributed battery energy storage systems for mitigating voltage and frequency deviations”. In: *IEEE Transactions on Smart Grid* 7.3 (2015), pp. 1713–1722.

[2] DM Rastler. *Electricity energy storage technology options: a white paper primer on applications, costs and benefits*. Electric Power Research Institute, 2010.

[3] Arne Groß, Christof Wittwer, and Moritz Diehl. “Stochastic Model Predictive Control of Photovoltaic Battery Systems using a Probabilistic Forecast Model”. In: *European Journal of Control* (2020).

[4] Arne Groß et al. “Using Probabilistic Forecasts in Stochastic Optimization”. In: *2020 International Conference on Probabilistic Methods Applied to Power Systems (PMAPS)*. IEEE. 2020, pp. 1–6.

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

[6] Dimitri P Bertsekas. *Reinforcement learning and optimal control*. Athena Scientific Belmont, MA, 2019.

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

[8] Arash Bahari Kordabad and Mehrdad Boroushaki. “Emotional Learning Based Intelligent Controller for MIMO Peripheral Milling Process”. In: *Journal of Applied and Computational Mechanics* 6.3 (2020), pp. 480–492.

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

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

[11] 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.

[12] Hossein Nejatbakhsh Esfahani, Arash Bahari Kordabad, and Sébastien Gros. “Reinforcement learning based on MPC/MHE for unmodeled and partially observable dynamics”. In: *2021 American Control Conference (ACC)*. IEEE. 2021, pp. 2121–2126.

[13] Arash Bahari Kordabad et al. “Reinforcement learning based on scenario-tree MPC for ASVs”. In: *2021 American Control Conference (ACC)*. IEEE. 2021, pp. 1985–1990.

[14] Torsten Koller et al. “Learning-based model predictive control for safe exploration”. In: *2018 IEEE Conference on Decision and Control (CDC)*. 2018, pp. 6059–6066.

[15] Arash Bahari Kordabad and Sebastien Gros. “Verification of dissipativity and evaluation of storage function in Economic Nonlinear MPC using Q-learning”. In: *IFAC-PapersOnLine* 54.6 (2021), pp. 308–313.

[16] JF Araujo Leao et al. “Lead-acid battery modeling and state of charge monitoring”. In: *2010 Twenty-Fifth Annual IEEE Applied Power Electronics Conference and Exposition (APEC)*. IEEE. 2010, pp. 239–243.

[17] Pavithra Harsha and Munther Dahleh. “Optimal management and sizing of energy storage under dynamic pricing for the efficient integration of renewable energy”. In: *IEEE Transactions on Power Systems* 30.3 (2014), pp. 1164–1181.

[18] Evelina Wikner and Torbjörn Thiringer. “Extending battery lifetime by avoiding high SOC”. In: *Applied Sciences* 8.10 (2018), p. 1825.

[19] Luis Omar Avila et al. “State of charge monitoring of Li-ion batteries for electric vehicles using GP filtering”. In: *Journal of Energy Storage* 25 (2019), p. 100837.

[20] 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.

[21] 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.

D Bias Correction in Deterministic Policy Gradient Using Robust MPC

Postprint of [98] **Arash Bahari Kordabad**, Hossein Nejatbakhsh Esfahani, and Sebastien Gros. "Bias Correction in Deterministic Policy Gradient Using Robust MPC". in: *2021 European Control Conference (ECC)* (2021), pp. 1086–1091. DOI: 10.23919/ECC54610.2021.9654962

Bias Correction in Deterministic Policy Gradient Using Robust MPC

Arash Bahari Kordabad[1], Hossein Nejatbakhsh Esfahani[1], and Sebastien Gros[1]

[1]Department of Engineering Cybernetics, Norwegian University of Science and Technology (NTNU), Trondheim, Norway.

Abstract: In this paper, we discuss the deterministic policy gradient using the Actor-Critic methods based on the linear compatible advantage function approximator, where the input spaces are continuous. When the policy is restricted by hard constraints, the exploration may not be Centred or Isotropic (non-CI). As a result, the policy gradient estimation can be biased. We focus on constrained policies based on Model Predictive Control (MPC) schemes and to address the bias issue, we propose an approximate Robust MPC approach accounting for the exploration. The RMPC-based policy ensures that a Centered and Isotropic (CI) exploration is approximately feasible. A posterior projection is used to ensure its exact feasibility, we formally prove that this approach does not bias the gradient estimation.

1 Introduction

Reinforcement learning (RL) provides powerful tools for tackling Markov Decision Processes (MPDs) without depending on the probability distribution underlying the state transition [1, 2]. RL methods attempt to enhance the closed-loop performance of a control policy deployed on the MDP, using the observed realization of the state transitions and of the corresponding stage cost. RL methods are usually either direct, based on an approximation of the optimal policy (e.g., deterministic and stochastic policy gradient methods [3]), or indirect, based on an approximation of the action-value function (e.g., Q-learning). Unstructured function approximation techniques (e.g., Deep Neural Networks) are often used to carry these approximations. Unfortunately, the closed-loop behavior of such approximators can be challenging to analyze formally. In contrast, structured function approximations such as Model Predictive Control (MPC) schemes provide a formal framework to analyze the stability and feasibility of the closed-loop system [4]. Recent research have focused on MPC-based policy approximation for RL [5–10].

For computational reasons, simple models are usually preferred in the MPC scheme. Hence, the MPC model often does not have the structure required to correctly capture the real system dynamics and stochasticity. As a result, while MPC can deliver a reasonable approximation of the optimal policy, it is usually suboptimal [11]. Choosing the MPC model parameters that

maximize the closed-loop performance of the MPC scheme is a difficult problem, and the parameters that best fit the MPC model to the real system are not guaranteed to yield the best MPC policy [6]. In [6, 9], it is shown that adjusting not only the MPC model, but also the cost and constraints can be beneficial to achieve the best closed-loop performances, and RL is proposed as a possible approach to perform that adjustment in practice. In the presence of uncertainties and stochasticity, if constraints satisfaction is critical, Robust Model Predictive Control (RMPC) provides tools to ensure that the constraints are satisfied, and can be used in the RL context [12].

Actor-Critic (AC) techniques combine the strong points of actor-only (policy search methods) and critic-only (e.g., Q-learning) methods [13]. AC approaches are based on genuine optimality conditions of the closed-loop policy and typically deliver less noisy policy gradients than direct policy search. The deterministic policy gradient is built based on an approximation of the advantage function associated with the policy. To this end, a linear compatible advantage function approximator is a convenient choice, because it provides a correct policy gradient estimation with a given structure and a low number of parameters [3]. For deterministic policies, exploration is required in order to estimate the corresponding policy gradient. In the presence of hard constraints, this exploration can be restricted. As a result, the exploration may become non-CI. In [14] it is shown that a linear compatible advantage function approximator can deliver an incorrect policy gradient estimation for a non-CI exploration.

In this paper, we propose to use an RMPC scheme that is robust with respect to a bounded disturbance of its first control input to enable the feasibility of a CI exploration. Because RMPC is computationally expensive, we use an inexpensive approximate RMPC instead, feasible to a first-order approximation. To ensure the feasibility of the exploration, a posterior projection technique is used. As a main result of this paper, we formally prove that the exploration resulting from the RMPC scheme delivers an unbiased policy gradient estimation.

The paper is structured as follows. Section 2 provides background material on RL and details the bias problem. Section 3 presents the RMPC-based approach that tackles the problem. For the sake of simplicity, we will consider a formulation robust with respect to the exploration only, while in practice the formulation can also be robust against model uncertainties and the stochasticity of the real system, as in [12]. Section 4 presents the projection approach required for nonlinear problems. Section 5 describes the main theorem in the gradient bias correction using RMPC-based policy and proves that the resulting approach asymptotically yields a correct policy gradient. Section 6 provides numerical examples of the method. Section 7 delivers a conclusion.

2 Background

For a given MDP with continuous state-input space, a deterministic policy parametrized by $\boldsymbol{\theta}$ delivers an input $\boldsymbol{a} \in \mathbb{R}^m$ as a function of state $\boldsymbol{s} \in \mathbb{R}^n$ as, $\boldsymbol{\pi_\theta}(\boldsymbol{s}) : \mathbb{R}^n \to \mathbb{R}^m$. If delivered by an MPC scheme, this policy is obtained as:

$$\boldsymbol{\pi_\theta}\left(\boldsymbol{s}\right) = \boldsymbol{u}_0^\star\left(\boldsymbol{s}, \boldsymbol{\theta}\right), \tag{1}$$

where $\boldsymbol{u}_0^\star$ is the first element of the solution $\boldsymbol{u}^\star$ given by:

$$\min_{\boldsymbol{u},\boldsymbol{x}} \quad V_{\boldsymbol{\theta}}(\boldsymbol{x}_N) + \sum_{k=0}^{N-1} \gamma^k \ell_{\boldsymbol{\theta}}(\boldsymbol{x}_k, \boldsymbol{u}_k), \tag{2a}$$

$$\text{s.t.} \quad \boldsymbol{x}_{k+1} = \boldsymbol{f}_{\boldsymbol{\theta}}\left(\boldsymbol{x}_k, \boldsymbol{u}_k\right), \;\; \boldsymbol{x}_0 = \boldsymbol{s}, \tag{2b}$$

$$\boldsymbol{h}_{\boldsymbol{\theta}}\left(\boldsymbol{x}_k, \boldsymbol{u}_k\right) \leq 0, \quad \boldsymbol{h}_{\boldsymbol{\theta}}^{\mathrm{f}}\left(\boldsymbol{x}_N\right) \leq 0, \tag{2c}$$

where $V_{\boldsymbol{\theta}}$ and $\ell_{\boldsymbol{\theta}}$ are the MPC terminal and stage costs, respectively. Function $\boldsymbol{f}_{\boldsymbol{\theta}}$ is the model dynamics and $\boldsymbol{h}_{\boldsymbol{\theta}}$ and $\boldsymbol{h}_{\boldsymbol{\theta}}^{\mathrm{f}}$ are the stage and terminal inequality constraints, respectively. Vector $\boldsymbol{x} = \{\boldsymbol{x}_{0,\ldots,N}\}$ is the predicted state trajectory and $\boldsymbol{u} = \{\boldsymbol{u}_{0,\ldots,N-1}\}$ is the input profile. State $\boldsymbol{s}$ is the current state of the system, N is the horizon length, and $\gamma \in [0, 1]$ is the discount factor. For the following theoretical developments, it will be useful to consider a single-shooting formulation of MPC (2) resulting in a parametric Nonlinear Program (NLP):

$$\min_{\boldsymbol{u}} \quad \Phi_{\boldsymbol{\theta}}(\boldsymbol{s}, \boldsymbol{u}), \tag{3a}$$

$$\text{s.t.} \quad \boldsymbol{H}_{\boldsymbol{\theta}}\left(\boldsymbol{s}, \boldsymbol{u}\right) \leq 0, \tag{3b}$$

delivering the input profile of (2) for all $\boldsymbol{\theta}, \boldsymbol{s}$ for some cost $\Phi_{\boldsymbol{\theta}}$ and inequality constraints $\boldsymbol{H}_{\boldsymbol{\theta}}$. We seek the policy parameters $\boldsymbol{\theta}$ that minimize the overall closed-loop cost J of the policy $\boldsymbol{\pi}_{\boldsymbol{\theta}}$ defined as follows:

$$J(\boldsymbol{\pi}_{\boldsymbol{\theta}}) = \mathbb{E}_{\boldsymbol{\pi}_{\boldsymbol{\theta}}}\left[\sum_{k=0}^{\infty} \gamma^k L(\boldsymbol{s}_k, \boldsymbol{a}_k) \;\middle|\; \boldsymbol{a}_k = \boldsymbol{\pi}_{\boldsymbol{\theta}}\left(\boldsymbol{s}_k\right)\right], \tag{4}$$

where $L(\boldsymbol{s}, \boldsymbol{a}) \in \mathbb{R}$ is the baseline stage cost evaluating the policy performance. It is shown in [6] that using an MPC stage cost $\ell_{\boldsymbol{\theta}}$ different from the baseline stage cost L can be beneficial when the MPC model is not exact. The expectation $\mathbb{E}_{\boldsymbol{\pi}_{\boldsymbol{\theta}}}$ is taken over the distribution of the Markov chain in closed-loop with the policy $\boldsymbol{\pi}_{\boldsymbol{\theta}}$. The policy gradient for the deterministic policy $\boldsymbol{\pi}_{\boldsymbol{\theta}}$ is obtained as follows [3]:

$$\nabla_{\boldsymbol{\theta}}\, J(\boldsymbol{\pi}_{\boldsymbol{\theta}}) = \mathbb{E}_{\boldsymbol{s}}\left[\nabla_{\boldsymbol{\theta}} \boldsymbol{\pi}_{\boldsymbol{\theta}}(\boldsymbol{s})\, \nabla_{\boldsymbol{a}} A_{\boldsymbol{\pi}_{\boldsymbol{\theta}}}(\boldsymbol{s}, \boldsymbol{\pi}_{\boldsymbol{\theta}}(\boldsymbol{s}))\right], \tag{5}$$

where $A_{\boldsymbol{\pi}_{\boldsymbol{\theta}}}(\boldsymbol{s}, \boldsymbol{a}) = Q_{\boldsymbol{\pi}_{\boldsymbol{\theta}}}(\boldsymbol{s}, \boldsymbol{a}) - V_{\boldsymbol{\pi}_{\boldsymbol{\theta}}}(\boldsymbol{s})$ is the advantage function associated to $\boldsymbol{\pi}_{\boldsymbol{\theta}}$, and where $Q_{\boldsymbol{\pi}_{\boldsymbol{\theta}}}$ and $V_{\boldsymbol{\pi}_{\boldsymbol{\theta}}}$ are the action-value and value functions for the policy $\boldsymbol{\pi}_{\boldsymbol{\theta}}$, respectively. In a non-episodic context, the expectation $\mathbb{E}_{\boldsymbol{s}}$ is taken over the steady-state distribution of the Markov chain. In an RL context, the advantage function $A_{\boldsymbol{\pi}_{\boldsymbol{\theta}}}$ must be approximated and evaluated from data. In the following, we label the advantage function approximation as $A_{\boldsymbol{\pi}_{\boldsymbol{\theta}}}^{\boldsymbol{w}}$ with parameter vector $\boldsymbol{w}$. The corresponding estimation of the policy gradient in (5) reads as:

$$\widehat{\nabla_{\boldsymbol{\theta}}\, J(\boldsymbol{\pi}_{\boldsymbol{\theta}})} = \mathbb{E}_{\boldsymbol{\pi}_{\boldsymbol{\theta}}}\left[\nabla_{\boldsymbol{\theta}} \boldsymbol{\pi}_{\boldsymbol{\theta}}(\boldsymbol{s})\, \nabla_{\boldsymbol{a}} A_{\boldsymbol{\pi}_{\boldsymbol{\theta}}}^{\boldsymbol{w}}(\boldsymbol{s}, \boldsymbol{\pi}_{\boldsymbol{\theta}}(\boldsymbol{s}))\right]. \tag{6}$$

The following theorem provides the condition allowing one to replace the exact advantage $A_{\boldsymbol{\pi}_{\boldsymbol{\theta}}}$ in (5) by an approximation $A_{\boldsymbol{\pi}_{\boldsymbol{\theta}}}^{\boldsymbol{w}}$, without affecting the policy gradient.

Theorem 1. *[3] If $A_{\boldsymbol{\pi}_{\boldsymbol{\theta}}}^{\boldsymbol{w}}$ satisfies*

i. $\nabla_{\boldsymbol{a}} A^{\boldsymbol{w}}_{\boldsymbol{\pi_\theta}} = \nabla_{\boldsymbol{\theta}} \boldsymbol{\pi}_{\boldsymbol{\theta}}^{\top} \boldsymbol{w}$,

ii. $\boldsymbol{w}$ *minimizes the following mean-squared error:*

$$\boldsymbol{w} = \arg\min_{\boldsymbol{w}} \frac{1}{2} \mathbb{E}_{\boldsymbol{s}} \left[\left\| \nabla_{\boldsymbol{a}} A_{\boldsymbol{\pi_\theta}} - \nabla_{\boldsymbol{a}} A^{\boldsymbol{w}}_{\boldsymbol{\pi_\theta}} \right\|^2 \right], \tag{7}$$

where the gradients are evaluated at $\boldsymbol{a} = \boldsymbol{\pi_\theta}$*, then we have:*

$$\widehat{\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\pi_\theta})} = \nabla_{\boldsymbol{\theta}} J(\boldsymbol{\pi_\theta}). \tag{8}$$

Proof. See [3]. ■

An advantage function approximator that achieves (8) is labelled *compatible*. A linear compatible advantage function approximator $A^{\boldsymbol{w}}_{\boldsymbol{\pi_\theta}}$, parametrized by $\boldsymbol{w}$ can read as [3]:

$$A^{\boldsymbol{w}}_{\boldsymbol{\pi_\theta}}(\boldsymbol{s}, \boldsymbol{a}) = \boldsymbol{w}^{\top} \nabla_{\boldsymbol{\theta}} \boldsymbol{\pi}_{\boldsymbol{\theta}} (\boldsymbol{a} - \boldsymbol{\pi_\theta}). \tag{9}$$

It is well known that estimating $\nabla_{\boldsymbol{a}} A_{\boldsymbol{\pi_\theta}}$ directly is very difficult [3]. As a surrogate to (7), the least-squares problem:

$$\boldsymbol{w} = \arg\min_{\boldsymbol{w}} \frac{1}{2} \mathbb{E}_{\boldsymbol{\pi_\theta}} \left[\left(Q_{\boldsymbol{\pi_\theta}} - \hat{V}_{\boldsymbol{\pi_\theta}} - A^{\boldsymbol{w}}_{\boldsymbol{\pi_\theta}} \right)^2 \right], \tag{10}$$

is used, where the value function estimation $\hat{V}_{\boldsymbol{\pi_\theta}} \approx V_{\boldsymbol{\pi_\theta}}$ is a baseline supporting the evaluation of $\boldsymbol{w}$. In order to obtain $\boldsymbol{w}$ from (10), the input $\boldsymbol{a}$ applied to the real system must be different from the actual policy $\boldsymbol{\pi_\theta}$, i.e. the input $\boldsymbol{a}$ applied to the real system should include some exploration in order to depart from the given policy $\boldsymbol{\pi_\theta}$. One common choice of exploration is to add a random disturbance $\boldsymbol{e}$ to the policy as follows:

$$\boldsymbol{a} = \boldsymbol{\pi_\theta}(\boldsymbol{s}) + \boldsymbol{e}. \tag{11}$$

For the sake of clarity, we define hereafter a CI exploration.

Definition 1. *An exploration* $\boldsymbol{e}$ *is Centred and Isotropic (CI) if* $\mathbb{E}_{\boldsymbol{e}}[\boldsymbol{e}] = 0$*, and there exists a scalar* p *such that,* $\mathbb{E}_{\boldsymbol{e}}[\boldsymbol{e}\boldsymbol{e}^{\top}] = pI$*. Otherwise, it is non-CI.*

Since the policy $\boldsymbol{\pi_\theta}$ is subject to the hard constraints (3b), an arbitrary input $\boldsymbol{a}$ resulting from a random exploration $\boldsymbol{e}$ may not be feasible. Hence the exploration ought to be restricted such that it respects the constraints. A possible solution for this problem is, e.g., to use a projection of $\boldsymbol{a}$ on the feasible set of NLP (3). In the following, we provide a definition for the projection operator.

Definition 2. *For an arbitrary input* $\boldsymbol{a}$*, the projection operator* $P(\boldsymbol{s}, \boldsymbol{a})$ *is defined as follows:*

$$P(\boldsymbol{s}, \boldsymbol{a}) = \boldsymbol{u}_0^{\perp}, \tag{12a}$$

$$\boldsymbol{u}^{\perp} = \arg\min_{\boldsymbol{u}} \quad \frac{1}{2} \|\boldsymbol{u}_0 - \boldsymbol{a}\|^2, \tag{12b}$$

$$\text{s.t.} \quad \boldsymbol{H_\theta}(\boldsymbol{s}, \boldsymbol{u}) \leq 0, \tag{12c}$$

where $\boldsymbol{u}_0^{\perp}$ *is the first element of the input profile* $\boldsymbol{u}^{\perp}_{0,\ldots,N-1}$ *solution of* (12b)-(12c).

In particular, at a given state $\boldsymbol{s}$, the input $\boldsymbol{a}_\perp$ resulting from projecting the exploration is given by:

$$\boldsymbol{a}_\perp = P\left(\boldsymbol{s}, \boldsymbol{\pi}_{\boldsymbol{\theta}}\left(\boldsymbol{s}\right) + \boldsymbol{e}\right). \tag{13}$$

Then the projected exploration $\boldsymbol{e}_\perp$ is given by:

$$\boldsymbol{e}_\perp = \boldsymbol{a}_\perp - \boldsymbol{\pi}_{\boldsymbol{\theta}}. \tag{14}$$

Unfortunately, even if the selected exploration $\boldsymbol{e}$ is CI, the projected exploration $\boldsymbol{e}_\perp$ may not be [14]. It is shown in [14] that the linear compatible function approximator (9) using the fitting problem (10) delivers a correct estimated policy gradient (6) only for a CI exploration.

In this paper, we modify (3) to find a policy $\hat{\boldsymbol{\pi}}_\theta\left(\boldsymbol{s}\right)$ for which a CI exploration is feasible. This policy $\hat{\boldsymbol{\pi}}_\theta\left(\boldsymbol{s}\right)$ is based on creating a small distance from the boundaries of the constraints so that a small CI exploration is feasible. To perform this modification, in the next section, we will introduce an approximate RMPC scheme having a computational complexity similar to a standard MPC scheme. This RMPC scheme delivers a policy that can be disturbed with an additive perturbation in a given ball while keeping feasibility to a first-order approximation.

3 RMPC-based deterministic policy

In this section, we propose a modified policy $\hat{\boldsymbol{\pi}}_\theta$ based on an RMPC-scheme such that any input $\hat{\boldsymbol{a}}$ resulting from:

$$\hat{\boldsymbol{a}} = \hat{\boldsymbol{\pi}}_\theta\left(\boldsymbol{s}\right) + \hat{\boldsymbol{e}}, \quad \forall \hat{\boldsymbol{e}} \in B(0, \eta), \tag{15}$$

is feasible for the MPC (2), where $B(0,\eta)$ is a ball of radius η. For the sake of brevity, we consider in the following that the exploration $\hat{\boldsymbol{e}}$ is uniformly distributed in the ball $B(0,\eta)$. In that specific case, the exploration $\hat{\boldsymbol{e}}$ is CI with $p = \frac{1}{3}\eta^2$. To generate $\hat{\boldsymbol{\pi}}_\theta$ we tighten the inequality constraint (3b) of NLP (3) as follows:

$$\min_{\boldsymbol{u}} \quad \Phi_{\boldsymbol{\theta}}(\boldsymbol{s}, \boldsymbol{u}), \tag{16a}$$

$$\text{s.t.} \quad \boldsymbol{H}_{\boldsymbol{\theta}}\left(\boldsymbol{s}, \boldsymbol{u}\right) + \boldsymbol{\Delta}_{\boldsymbol{\theta}}\left(\boldsymbol{s}, \boldsymbol{u}\right) \leq 0, \tag{16b}$$

where $\boldsymbol{\Delta}_{\boldsymbol{\theta}}\left(\boldsymbol{s}, \boldsymbol{u}\right) \geq 0$ is a back-off term added to ensure that the NLP (3) is feasible for any additive perturbation $\hat{\boldsymbol{e}} \in B\left(0, \eta\right)$ of the input $\boldsymbol{u}_0$ obtained from (16). In general, evaluating $\boldsymbol{\Delta}_{\boldsymbol{\theta}}\left(\boldsymbol{s}, \boldsymbol{u}\right)$ is difficult. To address this issue, we propose to compute $\boldsymbol{\Delta}_{\boldsymbol{\theta}}\left(\boldsymbol{s}, \boldsymbol{u}\right)$ using a first-order approximation of the constraint (3b). More specifically, we will impose the approximated constraint:

$$\boldsymbol{H}_{\boldsymbol{\theta}}\left(\boldsymbol{s}, \hat{\boldsymbol{u}}\right) \approx \boldsymbol{H}_{\boldsymbol{\theta}}\left(\boldsymbol{s}, \boldsymbol{u}\right) + \left.\frac{\partial \boldsymbol{H}_{\boldsymbol{\theta}}}{\partial \boldsymbol{u}_0}\right|_{\boldsymbol{u}} \hat{\boldsymbol{e}} \leq 0, \tag{17}$$

where $\hat{\boldsymbol{u}}$ is the input profile resulting from perturbing $\boldsymbol{u}$ with the exploration $\hat{\boldsymbol{e}} \in B\left(0, \eta\right)$ in the first input $\boldsymbol{u}_0$. The following Lemma provides an explicit form for (17).

Lemma 1. *Inequality* (17) *holds tightly for all* $\hat{\boldsymbol{e}} \in B(0,\eta)$ *if*

$$\boldsymbol{H}_{\boldsymbol{\theta}}^{i}(\boldsymbol{s},\boldsymbol{u}) + \left\| \frac{\partial \boldsymbol{H}_{\boldsymbol{\theta}}^{i}}{\partial \boldsymbol{u}_0}\bigg|_{\boldsymbol{u}} \right\| \eta \leq 0 \tag{18}$$

holds, where $\boldsymbol{H}_{\boldsymbol{\theta}}^{i}$ *is the* i^{th} *element of the vector* $\boldsymbol{H}_{\boldsymbol{\theta}}$.

Proof. The following inequality

$$\frac{\partial \boldsymbol{H}_{\boldsymbol{\theta}}^{i}}{\partial \boldsymbol{u}_0}\bigg|_{\boldsymbol{u}} \hat{\boldsymbol{e}} \leq \left\| \frac{\partial \boldsymbol{H}_{\boldsymbol{\theta}}^{i}}{\partial \boldsymbol{u}_0}\bigg|_{\boldsymbol{u}} \right\| \eta\,, \quad \forall \hat{\boldsymbol{e}} \in B(0,\eta), \tag{19}$$

holds and is tight, where $\|.\|$ indicates an Euclidean norm. ■

The principles detailed above readily apply to MPC scheme (2). More specifically, an input disturbance $\hat{\boldsymbol{e}}$ yields:

$$\boldsymbol{h}_{\boldsymbol{\theta}} \approx \boldsymbol{h}_{\boldsymbol{\theta}}(\boldsymbol{x}_k,\boldsymbol{u}_k) + \left(\frac{\partial \boldsymbol{h}_{\boldsymbol{\theta}}}{\partial \boldsymbol{x}_k}\frac{\partial \boldsymbol{x}_k}{\partial \boldsymbol{u}_0} + \frac{\partial \boldsymbol{h}_{\boldsymbol{\theta}}}{\partial \boldsymbol{u}_0} \right) \hat{\boldsymbol{e}}, \tag{20}$$

where the left-hand side is evaluated of the perturbed trajectory and $\frac{\partial \boldsymbol{x}_k}{\partial \boldsymbol{u}_0}$ is obtained from the following linear dynamics:

$$\frac{\partial \boldsymbol{x}_k}{\partial \boldsymbol{u}_0} = \left(\frac{\partial \boldsymbol{f}_{\boldsymbol{\theta}}}{\partial \boldsymbol{x}_{k-1}}\frac{\partial \boldsymbol{x}_{k-1}}{\partial \boldsymbol{u}_0} + \frac{\partial \boldsymbol{f}_{\boldsymbol{\theta}}}{\partial \boldsymbol{u}_0} \right)\Big|_{\boldsymbol{x}_{k-1},\boldsymbol{u}_{k-1}}, \tag{21}$$

with the initial condition $\frac{\partial \boldsymbol{x}_0}{\partial \boldsymbol{u}_0} = 0$.

Imposing an arbitrary exploration radius η may be infeasible for some state $\boldsymbol{s}$. To avoid this issue, we consider the radius as a decision variable $\nu \in [0,\bar{\eta}]$ whose optimal solution is η. We label $\bar{\eta}$ the maximum desired radius for the exploration. The RMPC-based policy $\hat{\boldsymbol{\pi}}_{\boldsymbol{\theta}}$ is then obtained as the first element of the input sequence given by:

$$\min_{\boldsymbol{u},\boldsymbol{x},\nu} \quad -w\nu + V_{\boldsymbol{\theta}}(\boldsymbol{x}_N) + \sum_{k=0}^{N-1} \gamma^k \ell_{\boldsymbol{\theta}}(\boldsymbol{x}_k,\boldsymbol{u}_k), \tag{22a}$$

$$\text{s.t.} \quad \boldsymbol{x}_{k+1} = \boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x}_k,\boldsymbol{u}_k)\,, \;\; \boldsymbol{x}_0 = \boldsymbol{s}, \tag{22b}$$

$$\boldsymbol{h}_{\boldsymbol{\theta}}(\boldsymbol{x}_0,\boldsymbol{u}_0)_i + \left\| \left(\frac{\partial \boldsymbol{h}_{\boldsymbol{\theta}}}{\partial \boldsymbol{u}_0} \right)_i \right\| \nu \leq 0, \tag{22c}$$

$$\boldsymbol{h}_{\boldsymbol{\theta}}(\boldsymbol{x}_k,\boldsymbol{u}_k)_i + \left\| \left(\frac{\partial \boldsymbol{h}_{\boldsymbol{\theta}}}{\partial \boldsymbol{x}_k}\frac{\partial \boldsymbol{x}_k}{\partial \boldsymbol{u}_0} \right)_i \right\| \nu \leq 0, \;\; k>0,$$

$$\boldsymbol{h}_{\boldsymbol{\theta}}^{\text{f}}(\boldsymbol{x}_N)_i + \left\| \left(\frac{\partial \boldsymbol{h}_{\boldsymbol{\theta}}^{\text{f}}}{\partial \boldsymbol{x}_N}\frac{\partial \boldsymbol{x}_N}{\partial \boldsymbol{u}_0} \right)_i \right\| \nu \leq 0 \tag{22d}$$

$$\frac{\partial \boldsymbol{x}_{k+1}}{\partial \boldsymbol{u}_0} = \frac{\partial \boldsymbol{f}_{\boldsymbol{\theta}}}{\partial \boldsymbol{x}_k}\frac{\partial \boldsymbol{x}_k}{\partial \boldsymbol{u}_0}, \quad k = 2,\ldots,N-1, \tag{22e}$$

$$\frac{\partial \boldsymbol{x}_1}{\partial \boldsymbol{u}_0} = \frac{\partial \boldsymbol{f}_{\boldsymbol{\theta}}}{\partial \boldsymbol{u}_0}, \quad 0 \leq \nu \leq \bar{\eta}, \tag{22f}$$

where w is a positive constant weight, chosen large enough such that $\eta = \bar{\eta}$ when feasible. Index i indicates the i^{th} element of the vectors.

One can observe that this RMPC scheme is feasible if the original MPC scheme (2) is feasible. Indeed, the choice $\nu = 0$ makes the RMPC and MPC schemes equivalent. It follows that the RMPC scheme (22) inherits the recursive feasibility of (2). We ought to stress again here that the recursive feasibility of (2) may require the robust formulation to be extended to take the stochastic disturbances and model errors into account, as e.g. in [12]. We have omitted this aspect here for the sake of brevity and simplicity. The theory presented hereafter is applicable to that extension. Additionally, one ought to note that RMPC (22) is accounting for a disturbance on the initial input only. However, exploration is meant to be applied at all times. This could be reflected in the RMPC by accounting for a disturbance of the entire input profile, with minor modifications of the formulation. These modifications would, however, unnecessarily reduce the feasible domain of (22). The proposed formulation arguably avoids that issue and ensures feasibility by introducing the exploration radius as a decision variable in the NLP. Finally, stabilizing feedback ought to be considered when forming the sensitivities (21), especially when the dynamics (22e) are unstable. This additional feedback is a classic tool to reduce the conservatism of the RMPC schemes. It is not presented here for the sake of brevity.

Since a first-order approximation of the constraints is used when forming (22), its solution may not ensure the feasibility of all exploration $\hat{\boldsymbol{e}} \in B(0, \eta)$. In the next section, we will address this problem with a posterior projection technique. We will show that this projection does not bias the policy gradient estimation.

4 Ensuring feasibility

Because we considered a first-order approximation of the constraints when forming the RMPC (22), a posterior projection ought to be used to ensure the feasibility of the exploration. Using (12), we apply the projection of $\hat{\boldsymbol{a}}$ on the feasible set as:

$$\hat{\boldsymbol{a}}_{\perp} = P\left(\boldsymbol{s}, \hat{\boldsymbol{\pi}}_{\boldsymbol{\theta}} + \hat{\boldsymbol{e}}\right). \tag{23}$$

Using (15), let us define the projection correction $\boldsymbol{\epsilon}$ as:

$$\boldsymbol{\epsilon} := \hat{\boldsymbol{a}}_{\perp} - \hat{\boldsymbol{a}}, \tag{24}$$

and using (14), the feasible projected exploration $\hat{\boldsymbol{e}}_{\perp}$ can be written as follows:

$$\hat{\boldsymbol{e}}_{\perp} := \hat{\boldsymbol{a}}_{\perp} - \hat{\boldsymbol{\pi}}_{\boldsymbol{\theta}} = \hat{\boldsymbol{e}} + \boldsymbol{\epsilon}. \tag{25}$$

In the following, we will show that the norm of $\boldsymbol{\epsilon}$ is in the order of η^2 for small enough $\bar{\eta}$. To this end, we make the following mild assumption for the constraints.

Assumption 1. *$\boldsymbol{H}_{\boldsymbol{\theta}}$ is a second order differentiable function and we have:*

$$\forall i \ , \ \left\| \frac{\partial \boldsymbol{H}^i_{\boldsymbol{\theta}}}{\partial \boldsymbol{u}_0}\Big|_{\hat{\boldsymbol{\pi}}_{\boldsymbol{\theta}}} \right\| \neq 0. \tag{26}$$

Note that if the constraints satisfy Linear Independence Constraint Qualification (LICQ), then (26) is satisfied.

Lemma 2. *For the projection error $\boldsymbol{\epsilon}$ defined in* (24) *and small enough $\bar{\eta}$, there exists a positive α such that:*

$$\|\boldsymbol{\epsilon}\| \leq \alpha \eta^2. \tag{27}$$

Proof. Let us define $\mathcal{H}_{\boldsymbol{\theta}}$ as,

$$\mathcal{H}_{\boldsymbol{\theta}}(\boldsymbol{s}, \boldsymbol{u}_0) := \boldsymbol{H}_{\boldsymbol{\theta}}(\boldsymbol{s}, \tilde{\boldsymbol{u}}), \tag{28}$$

where $\tilde{\boldsymbol{u}} := \left\{\boldsymbol{u}_0, \boldsymbol{u}_1^{\perp}, \ldots, \boldsymbol{u}_{N-1}^{\perp}\right\}$. We define $\mathcal{H}_{\boldsymbol{\theta}}^i$ as the i^{th} element of vector $\mathcal{H}_{\boldsymbol{\theta}}$. Consider the exploration described by its unitary direction $\boldsymbol{v}$, i.e. $\|\boldsymbol{v}\| = 1$, and magnitude $\zeta \leq \eta$, i.e. $\boldsymbol{e} = \zeta \boldsymbol{v}$. We observe that:

$$\mathcal{H}_{\boldsymbol{\theta}}^i\left(\boldsymbol{s}, \hat{\boldsymbol{\pi}}_{\boldsymbol{\theta}} + \hat{\boldsymbol{e}}\right) \leq \mathcal{H}_{\boldsymbol{\theta}}^i\left(\boldsymbol{s}, \hat{\boldsymbol{\pi}}_{\boldsymbol{\theta}}\right) + (\mathcal{H}_{\boldsymbol{\theta}}^i)' \hat{\boldsymbol{e}} + R\left(\hat{\boldsymbol{e}}\right), \tag{29}$$

where $(\mathcal{H}_{\boldsymbol{\theta}}^i)' := \frac{\partial \mathcal{H}_{\boldsymbol{\theta}}^i}{\partial \boldsymbol{u}_0}|_{\hat{\boldsymbol{\pi}}_{\boldsymbol{\theta}}}$. The inequality (29) holds for all $\hat{\boldsymbol{e}} \in B\left(0, \eta\right)$ for some continuous function $R\left(\hat{\boldsymbol{e}}\right)$, and there is a constant c such that:

$$\left|R\left(\hat{\boldsymbol{e}}\right)\right| \leq c\|\hat{\boldsymbol{e}}\|^2. \tag{30}$$

Additionally:

$$\mathcal{H}_{\boldsymbol{\theta}}^i\left(\boldsymbol{s}, \hat{\boldsymbol{\pi}}_{\boldsymbol{\theta}}\right) \leq -\eta \left\|(\mathcal{H}_{\boldsymbol{\theta}}^i)'\right\|, \tag{31}$$

because $\left\{\hat{\boldsymbol{\pi}}_{\boldsymbol{\theta}}, \boldsymbol{u}_1^{\perp}, \ldots, \boldsymbol{u}_{N-1}^{\perp}\right\}$ is feasible for the RMPC scheme (22). Consider any sequence $\eta_k > 0$ converging uniformly to 0, and a corresponding sequence $t_k = \max(1 - \alpha\eta_k, 0)$ for some positive constant α. One can readily observe that $t_k \hat{\boldsymbol{e}} \in B\left(0, \eta\right)$. Additionally, by construction, there exists an index k_0 such that for all $k \geq k_0$, $t_k = 1 - \alpha\eta_k$ holds. Using $t_k \boldsymbol{e}$ as the exploration in the right side of (29), for $k \geq k_0$ we have:

$$\begin{aligned}
&\mathcal{H}_{\boldsymbol{\theta}}^i\left(\boldsymbol{s}, \hat{\boldsymbol{\pi}}_{\boldsymbol{\theta}}\right) + (\mathcal{H}_{\boldsymbol{\theta}}^i)' \hat{\boldsymbol{e}}\left(1 - \alpha\eta_k\right) + R\left(\left(1 - \alpha\eta_k\right)\hat{\boldsymbol{e}}\right) \leq \\
&- \eta_k \left\|(\mathcal{H}_{\boldsymbol{\theta}}^i)'\right\| + (\mathcal{H}_{\boldsymbol{\theta}}^i)' \zeta \boldsymbol{v}\left(1 - \alpha\eta_k\right) + c\left(1 - \alpha\eta_k\right)^2 \zeta^2 \leq \\
&- \eta_k \left\|(\mathcal{H}_{\boldsymbol{\theta}}^i)'\right\| + \left\|(\mathcal{H}_{\boldsymbol{\theta}}^i)'\right\| \eta_k \left(1 - \alpha\eta_k\right) + c\left(1 - \alpha\eta_k\right)^2 \zeta^2 \\
&= - \left\|(\mathcal{H}_{\boldsymbol{\theta}}^i)'\right\| \alpha\eta_k^2 + c\left(1 - \alpha\eta_k\right)^2 \eta_k^2 \leq 0,
\end{aligned} \tag{32}$$

where the first inequality uses (30) and (31). The second inequality is obtained by selecting $\zeta = \eta_k$ and using the Cauchy–Schwarz inequality. Using Assumption 1, the last inequality holds for $c\left\|(\mathcal{H}_{\boldsymbol{\theta}}^i)'\right\|^{-1} \leq \alpha$. Therefore, $t_k \hat{\boldsymbol{e}}$ is a feasible exploration for (3) and has a larger (or equal) error than the projection error. Then we have:

$$\begin{aligned}
&\|\boldsymbol{\epsilon}\| = \|\hat{\boldsymbol{a}}_{\perp} - \hat{\boldsymbol{a}}\| = \|\hat{\boldsymbol{e}}_{\perp} - \hat{\boldsymbol{e}}\| \leq \|t_k \hat{\boldsymbol{e}} - \hat{\boldsymbol{e}}\| = \\
&\|(1 - t_k)\hat{\boldsymbol{e}}\| = \|\alpha\eta_k \hat{\boldsymbol{e}}\| \leq \alpha\eta^2.
\end{aligned}$$

■

The following theorem provides some useful properties on the statistics of $\hat{\boldsymbol{e}}_\perp$.

Theorem 2. *The projected exploration $\hat{\boldsymbol{e}}_\perp$ defined in* (23-25)*, for the policy resulting from RMPC* (22)*, has the following properties:*

$$\lim_{\bar{\eta}\to 0} \mathbb{E}_{\hat{\boldsymbol{e}}_\perp}[\hat{\boldsymbol{e}}_\perp] = 0, \tag{33a}$$

$$\lim_{\bar{\eta}\to 0} \mathbb{E}_{\hat{\boldsymbol{e}}_\perp}\left[\frac{1}{\eta^2}\,\hat{\boldsymbol{e}}_\perp \hat{\boldsymbol{e}}_\perp{}^\top\right] = \frac{1}{3} I, \tag{33b}$$

$$\lim_{\bar{\eta}\to 0} \mathbb{E}_{\hat{\boldsymbol{e}}_\perp}\left[\frac{1}{\eta^2}\,\hat{\boldsymbol{e}}_\perp\, \xi(\hat{\boldsymbol{e}}_\perp)\right] = 0, \tag{33c}$$

where η is the solution of ν in the RMPC (22) *and ξ is any scalar function satisfying $|\xi(.)| \leq r\|.\|^2$ for some positive r.*

Proof. We have $\lim_{\bar{\eta}\to 0} \eta = 0$, because $\eta \in [0, \bar{\eta}]$. Using Lemma 2, we have:

$$\begin{aligned} \lim_{\bar{\eta}\to 0} \|\mathbb{E}[\boldsymbol{\epsilon}]\| \leq \lim_{\bar{\eta}\to 0} \mathbb{E}\left[\|\boldsymbol{\epsilon}\|\right] \leq \lim_{\bar{\eta}\to 0} \alpha\eta^2 = 0 \\ \Rightarrow \lim_{\bar{\eta}\to 0} \mathbb{E}[\boldsymbol{\epsilon}] = 0. \end{aligned} \tag{34}$$

Taking the expectation from (25) and using that the exploration $\hat{\boldsymbol{e}}$ is CI, we have:

$$\lim_{\bar{\eta}\to 0} \mathbb{E}[\hat{\boldsymbol{e}}_\perp] = \lim_{\bar{\eta}\to 0} \Big(\mathbb{E}[\hat{\boldsymbol{e}}] + \mathbb{E}[\boldsymbol{\epsilon}]\Big) = 0. \tag{35}$$

Using (25), the second moment can be written as follows:

$$\begin{aligned} \lim_{\bar{\eta}\to 0} \mathbb{E}\left[\frac{1}{\eta^2}\,\hat{\boldsymbol{e}}_\perp \hat{\boldsymbol{e}}_\perp{}^\top\right] = \lim_{\bar{\eta}\to 0} \Bigg(\mathbb{E}\left[\frac{1}{\eta^2}\hat{\boldsymbol{e}}\hat{\boldsymbol{e}}^\top\right] + \\ \mathbb{E}\left[\frac{1}{\eta^2}\boldsymbol{\epsilon}\hat{\boldsymbol{e}}^\top\right] + \mathbb{E}\left[\frac{1}{\eta^2}\hat{\boldsymbol{e}}\boldsymbol{\epsilon}^\top\right] + \mathbb{E}\left[\frac{1}{\eta^2}\boldsymbol{\epsilon}\boldsymbol{\epsilon}^\top\right]\Bigg). \end{aligned} \tag{36}$$

For the first term we use that the exploration $\hat{\boldsymbol{e}}$ is CI with $p = \frac{1}{3}\eta^2 I$, i.e:

$$\mathbb{E}[\hat{\boldsymbol{e}}\hat{\boldsymbol{e}}^\top] = \frac{1}{3}\eta^2 I \Rightarrow \mathbb{E}\left[\frac{1}{\eta^2}\hat{\boldsymbol{e}}\hat{\boldsymbol{e}}^\top\right] = \frac{1}{3} I. \tag{37}$$

Using (27), for the second term we have:

$$\begin{aligned} \lim_{\bar{\eta}\to 0} \left\|\mathbb{E}\left[\frac{1}{\eta^2}\boldsymbol{\epsilon}\hat{\boldsymbol{e}}^\top\right]\right\| \leq \lim_{\bar{\eta}\to 0} \mathbb{E}\left[\frac{1}{\eta^2}\|\boldsymbol{\epsilon}\|\|\hat{\boldsymbol{e}}\|\right] \leq \\ \lim_{\bar{\eta}\to 0} \alpha\mathbb{E}[\|\hat{\boldsymbol{e}}\|] = 0 \Rightarrow \lim_{\bar{\eta}\to 0} \mathbb{E}\left[\frac{1}{\eta^2}\boldsymbol{\epsilon}\hat{\boldsymbol{e}}^\top\right] = 0. \end{aligned} \tag{38}$$

The third term will vanish in the similar way and for the forth term we can write:

$$\lim_{\bar{\eta}\to 0}\left\|\mathbb{E}\left[\frac{1}{\eta^2}\boldsymbol{\epsilon}\boldsymbol{\epsilon}^\top\right]\right\| \leq \lim_{\bar{\eta}\to 0}\mathbb{E}\left[\frac{1}{\eta^2}\|\boldsymbol{\epsilon}\|\|\boldsymbol{\epsilon}\|\right] \leq \tag{39}$$
$$\lim_{\bar{\eta}\to 0}\alpha\eta^2 \leq \lim_{\bar{\eta}\to 0}\alpha\bar{\eta}^2 = 0 \Rightarrow \lim_{\bar{\eta}\to 0}\mathbb{E}\left[\frac{1}{\eta^2}\boldsymbol{\epsilon}\boldsymbol{\epsilon}^\top\right] = 0.$$

Then, they deliver (33b). Finally for (33c), we have:

$$\|\hat{\boldsymbol{e}}_\perp\| = \|\hat{\boldsymbol{e}}+\boldsymbol{\epsilon}\| \leq \|\hat{\boldsymbol{e}}\|+\|\boldsymbol{\epsilon}\| \leq \eta+\alpha\eta^2. \tag{40}$$

Then:

$$\lim_{\bar{\eta}\to 0}\left\|\mathbb{E}\left[\frac{1}{\eta^2}\,\hat{\boldsymbol{e}}_\perp\,\xi(\hat{\boldsymbol{e}}_\perp)\right]\right\| \leq \lim_{\bar{\eta}\to 0}\mathbb{E}\left[\frac{1}{\eta^2}\|\hat{\boldsymbol{e}}_\perp\|\,|\xi(\hat{\boldsymbol{e}}_\perp)|\right]$$
$$\leq \lim_{\bar{\eta}\to 0}\mathbb{E}\left[\frac{r}{\eta^2}\|\hat{\boldsymbol{e}}_\perp\|^3\right] \leq \lim_{\bar{\eta}\to 0} r\eta(1+\alpha\eta)^3 = 0, \tag{41}$$

which delivers (33c). ■

5 Corrected Policy Gradient

In this section, we will show that the robust policy $\hat{\boldsymbol{\pi}}_{\boldsymbol{\theta}}$ delivers the true gradient as $\bar{\eta}\to 0$. Indeed, the deterministic policy gradient method uses "small" exploration and all results are valid in the sense of $\bar{\eta}\to 0$. We propose the compatible advantage function:

$$A^{\boldsymbol{w}}_{\hat{\boldsymbol{\pi}}_{\boldsymbol{\theta}}}(\boldsymbol{s},\hat{\boldsymbol{a}}_\perp) = \frac{\bar{\eta}^2}{\eta^2}\boldsymbol{w}^\top\nabla_{\boldsymbol{\theta}}\hat{\boldsymbol{\pi}}_{\boldsymbol{\theta}}\left(\hat{\boldsymbol{a}}_\perp-\hat{\boldsymbol{\pi}}_{\boldsymbol{\theta}}\right), \tag{42}$$

where the factor $\frac{\bar{\eta}^2}{\eta^2}$ is required to account for the varying exploration radius η and $\boldsymbol{w}$ is obtained as follows:

$$\boldsymbol{w} = \arg\min_{\boldsymbol{w}}\ \frac{1}{2}\mathbb{E}_{\hat{\boldsymbol{\pi}}_{\boldsymbol{\theta}},\hat{\boldsymbol{e}}_\perp}\left[\frac{1}{\bar{\eta}^2}\left(Q_{\hat{\boldsymbol{\pi}}_{\boldsymbol{\theta}}}-\hat{V}_{\hat{\boldsymbol{\pi}}_{\boldsymbol{\theta}}}-A^{\boldsymbol{w}}_{\hat{\boldsymbol{\pi}}_{\boldsymbol{\theta}}}\right)^2\right] \tag{43}$$

where $\mathbb{E}_{\hat{\boldsymbol{\pi}}_{\boldsymbol{\theta}},\hat{\boldsymbol{e}}_\perp} = \mathbb{E}_{\hat{\boldsymbol{\pi}}_{\boldsymbol{\theta}}}[\mathbb{E}_{\hat{\boldsymbol{e}}_\perp}[.|\boldsymbol{s}]]$ and $\bar{\eta}^{-2}$ is introduced such that (43) remains well-posed for $\bar{\eta}\to 0$.

Assumption 2. *$Q_{\hat{\boldsymbol{\pi}}_{\boldsymbol{\theta}}}$ is analytic and at least twice differentiable for almost every feasible $\boldsymbol{s}$ and $\nabla^2_{\boldsymbol{a}}Q_{\hat{\boldsymbol{\pi}}_{\boldsymbol{\theta}}}$ is bounded.*

Assumption 2 is usually satisfied in practice, as $Q_{\hat{\boldsymbol{\pi}}_{\boldsymbol{\theta}}}$ tends to be at least piecewise smooth for the many problems based on continuous state-input spaces. This assumption can be relaxed, but it requires more technical developments.

Theorem 3. *The RMPC-based policy gradient estimation using the compatible advantage function in* (42) *with* $\boldsymbol{w}$ *given by* (43) *asymptotically converges to exact gradient, i.e.:*

$$\lim_{\bar{\eta}\to 0} \widehat{\nabla_{\boldsymbol{\theta}} J(\hat{\boldsymbol{\pi}}_{\boldsymbol{\theta}})} = \nabla_{\boldsymbol{\theta}} J(\hat{\boldsymbol{\pi}}_{\boldsymbol{\theta}}). \tag{44}$$

Proof. The solution of (43) is given by:

$$\mathbb{E}_{\boldsymbol{\pi}_{\boldsymbol{\theta}},\hat{\boldsymbol{e}}_\perp}\left[\frac{1}{\eta^2}\nabla_{\boldsymbol{\theta}}\hat{\boldsymbol{\pi}}_{\boldsymbol{\theta}}\ \hat{\boldsymbol{e}}_\perp\ \left(Q_{\hat{\boldsymbol{\pi}}_{\boldsymbol{\theta}}} - \hat{V}_{\hat{\boldsymbol{\pi}}_{\boldsymbol{\theta}}} - A^{\boldsymbol{w}}_{\hat{\boldsymbol{\pi}}_{\boldsymbol{\theta}}}\right)\right] = 0. \tag{45}$$

Using Assumption 2, the Taylor expansions of $Q_{\hat{\boldsymbol{\pi}}_{\boldsymbol{\theta}}}$ and $A^{\boldsymbol{w}}_{\hat{\boldsymbol{\pi}}_{\boldsymbol{\theta}}}$ are valid almost everywhere. They read as:

$$\begin{aligned} Q_{\hat{\boldsymbol{\pi}}_{\boldsymbol{\theta}}}(\boldsymbol{s},\hat{\boldsymbol{a}}_\perp) &= V_{\hat{\boldsymbol{\pi}}_{\boldsymbol{\theta}}}(\boldsymbol{s}) + \nabla_{\boldsymbol{a}} A_{\hat{\boldsymbol{\pi}}_{\boldsymbol{\theta}}}(\boldsymbol{s},\hat{\boldsymbol{\pi}}_{\boldsymbol{\theta}}(\boldsymbol{s}))^\top\ \hat{\boldsymbol{e}}_\perp + \boldsymbol{\xi}, \\ A^{\boldsymbol{w}}_{\hat{\boldsymbol{\pi}}_{\boldsymbol{\theta}}}(\boldsymbol{s},\hat{\boldsymbol{a}}_\perp) &= \nabla_{\boldsymbol{a}} A^{\boldsymbol{w}}_{\hat{\boldsymbol{\pi}}_{\boldsymbol{\theta}}}(\boldsymbol{s},\hat{\boldsymbol{\pi}}_{\boldsymbol{\theta}}(\boldsymbol{s}))^\top\ \hat{\boldsymbol{e}}_\perp, \end{aligned} \tag{46}$$

where $\boldsymbol{\xi}$ is the second-order remainder of the Taylor expansion of $Q_{\hat{\boldsymbol{\pi}}_{\boldsymbol{\theta}}}$ at $\hat{\boldsymbol{e}}_\perp = 0$ and the identity $\nabla_{\boldsymbol{a}} Q_{\hat{\boldsymbol{\pi}}_{\boldsymbol{\theta}}} = \nabla_{\boldsymbol{a}} A_{\hat{\boldsymbol{\pi}}_{\boldsymbol{\theta}}}$ was used. Using Assumption 2, $\boldsymbol{\xi}$ is of order $\|\hat{\boldsymbol{e}}_\perp\|^2$ for almost every feasible $\boldsymbol{s}$. By substitution of (46) in (45), we have:

$$\begin{aligned} &\mathbb{E}_{\boldsymbol{s},\hat{\boldsymbol{e}}_\perp}\left[\frac{1}{\eta^2}\nabla_{\boldsymbol{\theta}}\hat{\boldsymbol{\pi}}_{\boldsymbol{\theta}}\hat{\boldsymbol{e}}_\perp{\hat{\boldsymbol{e}}_\perp}^\top\left(\nabla_{\boldsymbol{a}} A_{\hat{\boldsymbol{\pi}}_{\boldsymbol{\theta}}} - \nabla_{\boldsymbol{a}} A^{\boldsymbol{w}}_{\hat{\boldsymbol{\pi}}_{\boldsymbol{\theta}}}\right)\right] + \\ &\mathbb{E}_{\boldsymbol{s},\hat{\boldsymbol{e}}_\perp}\left[\frac{1}{\eta^2}\nabla_{\boldsymbol{\theta}}\hat{\boldsymbol{\pi}}_{\boldsymbol{\theta}}\hat{\boldsymbol{e}}_\perp\boldsymbol{\xi}\right] + \\ &\mathbb{E}_{\boldsymbol{s},\hat{\boldsymbol{e}}_\perp}\left[\frac{1}{\eta^2}\nabla_{\boldsymbol{\theta}}\hat{\boldsymbol{\pi}}_{\boldsymbol{\theta}}\ \hat{\boldsymbol{e}}_\perp\ \left(V_{\hat{\boldsymbol{\pi}}_{\boldsymbol{\theta}}} - \hat{V}_{\hat{\boldsymbol{\pi}}_{\boldsymbol{\theta}}}\right)\right] = 0. \end{aligned} \tag{47}$$

Using Theorem 2, the second and third terms will vanish in the sense of $\bar{\eta}\to 0$ and the first term will be:

$$\lim_{\bar{\eta}\to 0} \mathbb{E}_{\boldsymbol{s}}\left[\nabla_{\boldsymbol{\theta}}\hat{\boldsymbol{\pi}}_{\boldsymbol{\theta}}\left(\nabla_{\boldsymbol{a}} A_{\hat{\boldsymbol{\pi}}_{\boldsymbol{\theta}}} - \nabla_{\boldsymbol{a}} A^{\boldsymbol{w}}_{\hat{\boldsymbol{\pi}}_{\boldsymbol{\theta}}}\right)\right] = 0, \tag{48}$$

which delivers (44). ■

In addition, under mild conditions, the RMPC-based policy resulting from (22) converges to the main MPC-based policy resulting from (2) as $\bar{\eta}\to 0$. For the sake of brevity, we do not formalize this statement here.

6 Numerical Simulation

In this section, we propose two numerical examples in order to illustrate the theoretical developments. The first example directly compares the MPC-based policy and the RMPC-based policy and the optimal policy with a nonlinear constraint. We consider the deterministic

scalar MDP $s^+ = s + a$ with stage cost $L(s, a) = s^2 + a^2$, constraint $s^2 + 5a^2 \leq 1$ and discount factor $\gamma = 0.9$. Then we use the following MPC scheme to extract the approximated policy:

$$\min_{\boldsymbol{x},\boldsymbol{u}} \quad x_N^2 + \sum_{k=0}^{N-1} \gamma^k (\theta x_k^2 + u_k^2), \tag{49a}$$

$$s.t. \quad x_{k+1} = x_k + u_k \, , \, x_k^2 + 5u_k^2 \leq 1 \, , \, x_0 = s, \tag{49b}$$

then $\pi_\theta(s) = u_0^\star(s)$ obtained from the first element of the input solution. We can build the RMPC scheme according to (22) and extract the modified policy $\hat{\pi}_\theta(s)$. Fig.1 (top) illustrates the posterior projected error $\|\epsilon\|$ and the approximated feasible radius η for $\bar{\eta} = 0.05$. As it can be seen, e.g., a fixed radius exploration with $\eta = 0.05$ may be infeasible at $s = \pm 1$. Fig.1 (bottom) compares these policies with the MDP optimal policy. This simple example shows that the RMPC-based policy makes a distance with the feasible bound to guarantee the feasibility of the exploration in both directions. While classic MPC is on the feasible set bound, a feasible exploration should only be in one direction.

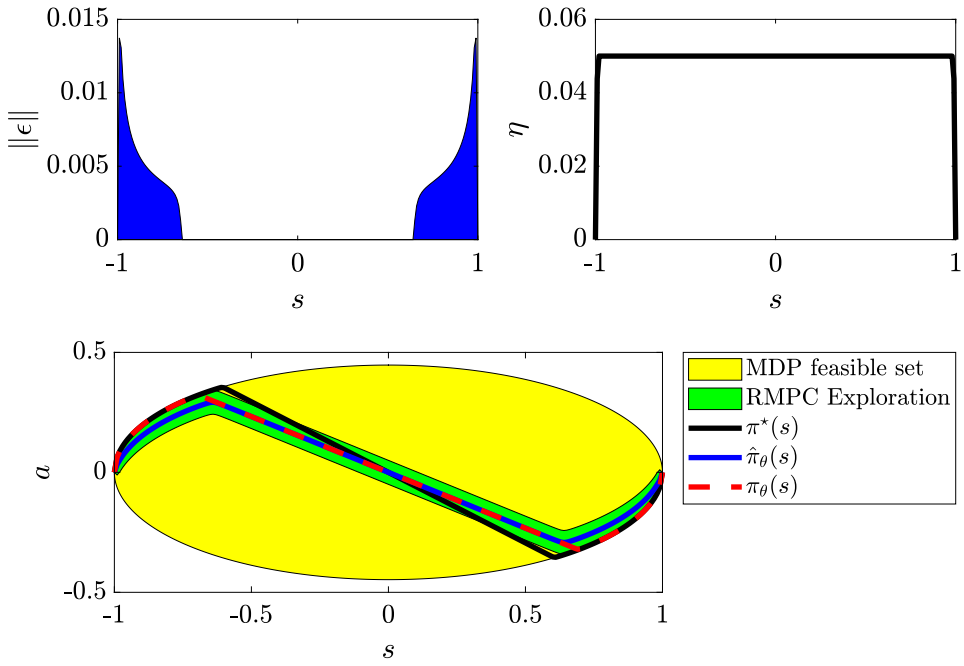


Figure 1: Top-left: Blue region shows norm of the posterior projected error $\|\epsilon\|$. Top-right: The approximated feasible radius η. Bottom: The optimal policy and parametrized policies from MPC and RMPC scheme for $\theta = 0.5$.

The second example compares the gradient of the RMPC-based and MPC-based policies with the true policy gradient. We consider linear scalar dynamics $s^+ = 0.97s + 0.1a + d$ where $d \sim$

$\mathcal{U}(-10^{-3}, 10^3)$ is a scalar uniform noise. RL stage cost is $L(s,a) = 20(s-0.5)^2 + (a-2)^2$ with $\gamma = 0.9$. The policy is extracted from the following MPC scheme:

$$\min_{u} \quad \sum_{k=0}^{50} \gamma^k (10(x_k - 1/3)^2 + (u_k - u_{\text{ref}}(\theta))^2), \tag{50a}$$

$$\text{s.t.} \quad x_{k+1} = 0.97x_k + 0.1u_k, \quad x_0 = s, \tag{50b}$$

$$u_k \leq \theta, \tag{50c}$$

where $u_{\text{ref}}(\theta) = 0.2 - \theta$. The initial RL parameter $\theta = 0.1$ is selected. The MPC policy can be adjusted by increasing the input bound in (50c) and raising the input reference u_{ref}. However, raising the input bound in (50c) by increasing θ results in decreasing the input reference u_{ref}, such that these terms are in contradiction to find the optimal policy. Fig. 2 shows the policy gradient over the RL iterations. The red (dashed) curve is the outcome of learning from the classic MPC, while the blue (solid) curve is the one from the RMPC. As it can be seen, the RMPC gradient $\nabla_{\boldsymbol{\theta}} \widehat{J(\hat{\boldsymbol{\pi}}_{\boldsymbol{\theta}})}$ delivers a very close gradient to the true gradient $\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\pi}_{\boldsymbol{\theta}})$. However, the MPC policy gradient $\nabla_{\boldsymbol{\theta}} \widehat{J(\boldsymbol{\pi}_{\boldsymbol{\theta}})}$ has an obvious bias in both cases. Note that the closed-loop performance loss from this bias issue is not necessarily large for this example.

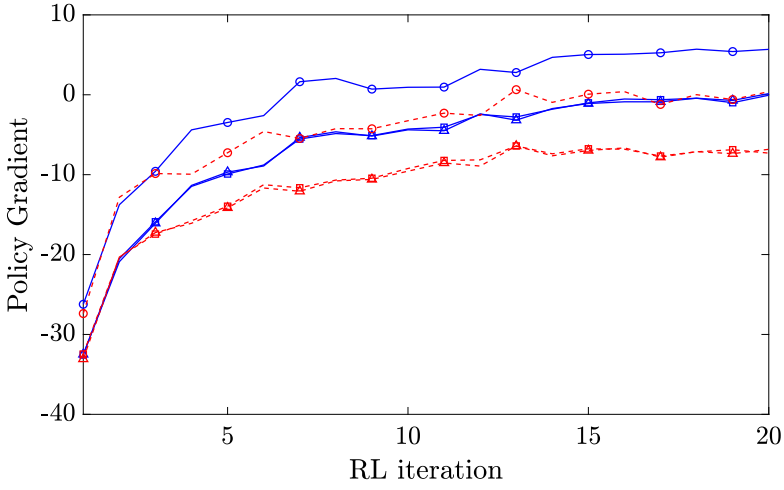


Figure 2: The policy gradients over the RL iterations. The outcome of learning using policy gradients from MPC (red-dashed) and RMPC scheme (blue-solid). ($\circ$: $\nabla_{\boldsymbol{\theta}} \widehat{J(\boldsymbol{\pi}_{\boldsymbol{\theta}})}$. $\square$: $\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\pi}_{\boldsymbol{\theta}})$. $\triangle$: $\nabla_{\boldsymbol{\theta}} \widehat{J(\hat{\boldsymbol{\pi}}_{\boldsymbol{\theta}})}$.

A more complex example demonstrating the theory on a nonlinear example would be useful. For the sake of brevity, such an example will be considered in the future.

7 Conclusion

This paper presented the AC approach using a linear compatible advantage function approximation for the MPC-based deterministic policies. When the policy is restricted by hard constraints, the exploration may be non-CI and delivers a bias in the policy gradient. We proposed RMPC using constraint tightening to provide an approximated feasible CI exploration. A posterior projection is used to ensure feasibility and formally we showed that the RMPC-based policy gradient converges to the true policy gradient for a small enough radius of exploration.

References

[1] Dimitri P Bertsekas. *Reinforcement learning and optimal control*. Athena Scientific Belmont, MA, 2019.

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

[3] David Silver et al. "Deterministic Policy Gradient Algorithms". In: *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32*. ICML'14. Beijing, China: JMLR.org, 2014, I–387–I–395.

[4] Kim P Wabersich and Melanie N Zeilinger. "Safe exploration of nonlinear dynamical systems: A predictive safety filter for reinforcement learning". In: *arXiv preprint arXiv:1812.05506* (2018).

[5] Torsten Koller et al. "Learning-based model predictive control for safe exploration". In: *2018 IEEE Conference on Decision and Control (CDC)*. 2018, pp. 6059–6066.

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

[7] Mario Zanon, Vyacheslav Kungurtsev, and Sébastien Gros. "Reinforcement learning based on real-time iteration NMPC". In: *IFAC-PapersOnLine* 53.2 (2020), pp. 5213–5218.

[8] Arash Bahari Kordabad et al. "Reinforcement learning based on scenario-tree MPC for ASVs". In: *2021 American Control Conference (ACC)* (2021), pp. 1985–1990.

[9] Sebastien Gros and Mario Zanon. "Reinforcement learning for mixed-integer problems based on MPC". In: *IFAC-PapersOnLine* 53.2 (2020), pp. 5219–5224.

[10] Hossein Nejatbakhsh Esfahani, Arash Bahari Kordabad, and Sébastien Gros. "Reinforcement learning based on MPC/MHE for unmodeled and partially observable dynamics". In: *2021 American Control Conference (ACC)* (2021), pp. 2121–2126.

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

[12] Mario Zanon and Sébastien Gros. "Safe reinforcement learning using robust MPC". In: *IEEE Transactions on Automatic Control* 66.8 (2020), pp. 3638–3652.

[13] Vijay R Konda and John N Tsitsiklis. "Actor-critic algorithms". In: *Advances in neural information processing systems*. 2000, pp. 1008–1014.

[14] 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.

E Quasi-Newton Iteration in Deterministic Policy Gradient

Postprint of [99] **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. DOI: 10.23919/ACC53348.2022.9867217

Quasi-Newton Iteration in Deterministic Policy Gradient

Arash Bahari Kordabad[1], Hossein Nejatbakhsh Esfahani[1], Wenqi Cai[1], and Sebastien Gros[1]

[1]Department of Engineering Cybernetics, Norwegian University of Science and Technology (NTNU), Trondheim, Norway.

Abstract: This paper presents a model-free approximation for the Hessian of the performance of deterministic policies to use in the context of Reinforcement Learning based on Quasi-Newton steps in the policy parameters. We show that the approximate Hessian converges to the exact Hessian at the optimal policy, and allows for a superlinear convergence in the learning, provided that the policy parametrization is rich. The natural policy gradient method can be interpreted as a particular case of the proposed method. We analytically verify the formulation in a simple linear case and compare the convergence of the proposed method with the natural policy gradient in a nonlinear example.

1 Introduction

Markov Decision Processes (MDPs) provide the standard framework for (stochastic) control problem. The Bellman equations provide the exact solution for a given MDP, and can be solved via Dynamic Programming (DP) [1]. Unfortunately, this is impractical because of the *curse of dimensionality of DP*. In practice, Reinforcement learning (RL) provides model-free tools to obtain an approximate solutions for the MDPs.

Deterministic policy gradient algorithms are widely used in RL with continuous action spaces [2]. These methods attempt to learn the optimal parameters of a parameterized policy $\boldsymbol{\pi_\theta}$ using only state transitions observed on the real system. These methods commonly use gradient descent methods to optimize a discounted sum of stage costs, called closed-loop performance $J(\boldsymbol{\theta})$. Depending on the policy type, these approaches are divided into the deterministic and the stochastic policy gradient methods. In the stochastic policy gradient methods, a parametrized distribution of action $\boldsymbol{a}$ conditioned on each state $\boldsymbol{s}$ taking the form of $\boldsymbol{\pi_\theta}(\boldsymbol{a}|\boldsymbol{s})$ is considered, while deterministic policy methods use $\boldsymbol{a} = \boldsymbol{\pi_\theta}(\boldsymbol{s})$ to specify a deterministic action for each state $\boldsymbol{s}$. Both methods adjust the parameter vector $\boldsymbol{\theta}$ in order to optimize J. In practice, stochastic policy gradient may need more data when the action space has many dimensions [3]. Hence, in this paper we focus on the deterministic policies.

Unfortunately, the convergence rate of classical gradient descent is limited, especially when the

Hessian of closed-loop performance J is far from a scalar multiple of the Identity matrix [4]. In [5], the global convergence of policy gradient methods has been investigated for the Linear Quadratic Regulator (LQR) problems. Various studies propose to use the Hessian of the policy performance in a Newton-type methods in order to deliver a faster learning [6].

Natural policy gradient methods has been attracted many attentions in RL community recently due to its capability for better convergence [7]. The efficiency of the natural policy gradient in RL was showed in [8]. The natural policy gradient methods use the *Fisher information matrix* as an approximate Hessian [9]. In [10], a natural policy gradient method is developed for Constrained MDPs. A Quasi-Newton method is developed in [11] for Temporal Difference (TD) learning in order to get faster convergence. Natural Actor-critic has been investigated in [12]. Although the Fisher information matrix, as an approximation for the Hessian, is positive definite, it does not asymptotically converge to the exact Hessian necessarily, when the policy converges to the optimal policy [7]. As a result, the rate of convergence of the natural policy gradient method is linear, i.e., the same as the regular gradient descent [6]. Therefore, providing an approximation of the Hessian (without imposing heavy computation) that converges to the exact Hessian at the optimal policy can improve the convergence rate.

In this paper, we first derive a formulation for exact Hessian of deterministic policy performance with respect to the parameters. Then we provide a model-free approximation for the Hessian of the performance function J. We show that the approximate Hessian converges to the exact Hessian at the optimal policy when the parameterized policy is rich. As a result, it gives a superlinear convergence using a Quasi-Newton optimization.

2 Hessian of the Policy Performance

In the RL context, the problem is assumed to be an MDP with an initial state distribution $p_1(\boldsymbol{s}_0)$ and transition probability density $p(\boldsymbol{s}^+|\boldsymbol{s},\boldsymbol{a})$ where $\boldsymbol{s} \in \mathcal{S} \subseteq \mathbb{R}^{n_s}$, $\boldsymbol{a} \in \mathcal{A} \subseteq \mathbb{R}^{n_a}$, and $\boldsymbol{s}^+$ are the current state, input, and subsequent state, respectively, and $\boldsymbol{s}_0$ is the initial state. Every transition imposes a real scalar stage cost $\ell(\boldsymbol{s},\boldsymbol{a})$. A deterministic policy denoted by $\boldsymbol{\pi} : \mathcal{S} \to \mathcal{A}$ specifies how the input $\boldsymbol{a}$ is chosen for each state $\boldsymbol{s}$. We consider a parametrized policy $\boldsymbol{\pi_\theta}$ with parameter vector $\boldsymbol{\theta} \in \mathbb{R}^{n_\theta}$ and seek an optimal policy by adjusting parameter $\boldsymbol{\theta}$. The value function $V^{\boldsymbol{\pi_\theta}}$ and action-value function $Q^{\boldsymbol{\pi_\theta}}(\boldsymbol{s},\boldsymbol{a})$ are defined as follows:

$$Q^{\boldsymbol{\pi_\theta}}(\boldsymbol{s},\boldsymbol{a}) = \ell(\boldsymbol{s},\boldsymbol{a}) + \gamma \mathbb{E}_{p(\cdot|\boldsymbol{s},\boldsymbol{a})}\left[V^{\boldsymbol{\pi_\theta}}(\boldsymbol{s}^+)|\boldsymbol{s},\boldsymbol{a}\right], \tag{1a}$$

$$V^{\boldsymbol{\pi_\theta}}(\boldsymbol{s}) = Q^{\boldsymbol{\pi_\theta}}(\boldsymbol{s},\boldsymbol{\pi_\theta}(\boldsymbol{s})), \tag{1b}$$

where $\gamma \in (0,1]$ is a discount factor. The performance objective $J(\boldsymbol{\theta})$ is given as follows:

$$J(\boldsymbol{\theta}) = \mathbb{E}_{\boldsymbol{s}_0}\left[V^{\boldsymbol{\pi_\theta}}(\boldsymbol{s}_0)\right] = \mathbb{E}_{\boldsymbol{s}}\left[\ell(\boldsymbol{s},\boldsymbol{\pi_\theta}(\boldsymbol{s}))\right]. \tag{2}$$

Note that we simplified the expectation notation $\mathbb{E}_{\boldsymbol{s}_0\sim p_1(\boldsymbol{s}_0)}[\cdot] = \mathbb{E}_{\boldsymbol{s}_0}[\cdot]$ and $\mathbb{E}_{\boldsymbol{s}}[\cdot]$ is taken over the expected sum of the discounted state distribution of the Markov chain in closed-loop with policy $\boldsymbol{\pi_\theta}$. The purpose is solving the following optimization problem:

$$\boldsymbol{\theta}^\star \in \arg\min_{\boldsymbol{\theta}} J(\boldsymbol{\theta}). \tag{3}$$

In the following we make an assumption in order to guarantee the existence of the policy gradient and we recall the deterministic policy gradient theorem.

Assumption 1. *$p(\boldsymbol{s}'|\boldsymbol{s},\boldsymbol{a})$, $\nabla_{\boldsymbol{a}} p(\boldsymbol{s}'|\boldsymbol{s},\boldsymbol{a})$, $\boldsymbol{\pi}_{\boldsymbol{\theta}}(\boldsymbol{s})$, $\nabla_{\boldsymbol{\theta}}\boldsymbol{\pi}_{\boldsymbol{\theta}}(\boldsymbol{s})$, $\ell(\boldsymbol{s},\boldsymbol{a})$, $\nabla_{\boldsymbol{a}}\ell(\boldsymbol{s},\boldsymbol{a})$, $p_1(\boldsymbol{s})$ are continuous in all parameters and variables $\boldsymbol{s}$, $\boldsymbol{a}$, $\boldsymbol{s}'$, $\boldsymbol{\theta}$. Also there exist b and L such that:*

$$\begin{aligned} &\sup_{\boldsymbol{s}} p_1(\boldsymbol{s}) < b, && \sup_{\{\boldsymbol{a},\boldsymbol{s},\boldsymbol{s}'\}} p(\boldsymbol{s}'|\boldsymbol{s},\boldsymbol{a}) < b, \\ &\sup_{\{\boldsymbol{a},\boldsymbol{s}\}} \|\nabla_{\boldsymbol{a}}\ell(\boldsymbol{s},\boldsymbol{a})\| < L, && \sup_{\{\boldsymbol{a},\boldsymbol{s},\boldsymbol{s}'\}} \|\nabla_{\boldsymbol{a}} p(\boldsymbol{s}'|\boldsymbol{s},\boldsymbol{a})\| < L. \end{aligned} \tag{4}$$

Moreover, there exists a policy $\boldsymbol{\pi}_{\boldsymbol{\theta}}$ such that $J(\boldsymbol{\theta})$ is finite.

Assumption 1 is a standard assumption which is made in [3] in order to derive policy gradients. All derivatives are also bounded for a smooth enough p, such as the Gaussian distribution. Moreover, one can select the initial state distribution from a bounded probability function. The existence of a policy that makes the performance $J(\boldsymbol{\theta})$ finite can be interpreted as a controllability assumption in the control literature. Policy gradient methods usually solve (3) using gradient descent method, i.e., at each iteration k, we update $\boldsymbol{\theta}$ as follows:

$$\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k - \alpha \nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta})|_{\boldsymbol{\theta}=\boldsymbol{\theta}_k}, \tag{5}$$

where α is a positive step-size.

Theorem 1. *(Deterministic Policy Gradient) Suppose that the MDP satisfies Assumption 1; then $\nabla_{\boldsymbol{a}} Q^{\boldsymbol{\pi}_{\boldsymbol{\theta}}}$ exists and the deterministic policy gradient reads as:*

$$\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) = \mathbb{E}_{\boldsymbol{s}}\Big[\nabla_{\boldsymbol{\theta}}\boldsymbol{\pi}_{\boldsymbol{\theta}}(\boldsymbol{s}) \nabla_{\boldsymbol{a}} Q^{\boldsymbol{\pi}_{\boldsymbol{\theta}}}(\boldsymbol{s},\boldsymbol{a})|_{\boldsymbol{a}=\boldsymbol{\pi}_{\boldsymbol{\theta}}(\boldsymbol{s})}\Big]. \tag{6}$$

Proof. See in [3]. ■

The next standard assumption will be made to ensure the existence of the Hessian of the policy with respect to the policy parameters $\boldsymbol{\theta}$ and the Hessian of action-value function with respect to the input $\boldsymbol{a}$.

Assumption 2. *$\nabla^2_{\boldsymbol{a}} p(\boldsymbol{s}'|\boldsymbol{s},\boldsymbol{a})$, $\nabla^2_{\boldsymbol{\theta}}\boldsymbol{\pi}_{\boldsymbol{\theta}}(\boldsymbol{s})$, $\nabla^2_{\boldsymbol{a}}\ell(\boldsymbol{s},\boldsymbol{a})$, are continuous in all parameters and variables $\boldsymbol{s}$, $\boldsymbol{a}$, $\boldsymbol{s}'$, $\boldsymbol{\theta}$. Moreover, there exists M such that:*

$$\sup_{\boldsymbol{a},\boldsymbol{s},\boldsymbol{s}'} \|\nabla^2_{\boldsymbol{a}} p(\boldsymbol{s}'|\boldsymbol{s},\boldsymbol{a})\| < M, \quad \sup_{\boldsymbol{a},\boldsymbol{s}} \|\nabla^2_{\boldsymbol{a}}\ell(\boldsymbol{s},\boldsymbol{a})\| < M. \tag{7}$$

Similar to the assumption 1, assumption 2 is made to derive the Hessian of the performance. In practice, the assumption is satisfied for a smooth enough transition p, policy $\boldsymbol{\pi}$ and stage cost ℓ. In the following we provide the exact Hessian of the deterministic policy performance with respect to the policy parameters.

Definition 1. *In this paper, we use the operation* $\otimes : \mathbb{R}^{n_1 \times n_2 \times n_3} \times \mathbb{R}^{n_3} \to \mathbb{R}^{n_1 \times n_2}$ *for the product of a tensor* T *and a vector* $\boldsymbol{v}$*, such that:*

$$T \otimes \boldsymbol{v} \triangleq \sum_{i=1}^{n_3} v_i T_{(:,:,i)}, \tag{8}$$

where scalar v_i *is the* i^{th} *element of vector* $\boldsymbol{v}$ *and matrix* $[T_{(:,:,i)}]_{n_1 \times n_2}$ *is the* i^{th} *frontal slice of tensor* T *[13].*

Theorem 2. *(Deterministic Policy Hessian) Under Assumptions 1 and 2,* $\nabla^2_{\boldsymbol{a}} Q^{\boldsymbol{\pi_\theta}}$ *and the deterministic policy Hessian exist. The latter is given by:*

$$\nabla^2_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) = H(\boldsymbol{\theta}) + \gamma \Lambda(\boldsymbol{\theta}), \tag{9}$$

where $H(\boldsymbol{\theta})$ *and* $\Lambda(\boldsymbol{\theta})$ *are defined as follows:*

$$H(\boldsymbol{\theta}) \triangleq \mathbb{E}_{\boldsymbol{s}} \Big[\nabla^2_{\boldsymbol{\theta}} \boldsymbol{\pi}_{\boldsymbol{\theta}}(\boldsymbol{s}) \otimes \nabla_{\boldsymbol{a}} Q^{\boldsymbol{\pi_\theta}}(\boldsymbol{s}, \boldsymbol{a}) \Big|_{\boldsymbol{a}=\boldsymbol{\pi}_{\boldsymbol{\theta}}} + \nabla_{\boldsymbol{\theta}} \boldsymbol{\pi}_{\boldsymbol{\theta}}(\boldsymbol{s}) \nabla^2_{\boldsymbol{a}} Q^{\boldsymbol{\pi_\theta}}(\boldsymbol{s}, \boldsymbol{a}) \Big|_{\boldsymbol{a}=\boldsymbol{\pi}_{\boldsymbol{\theta}}} \nabla_{\boldsymbol{\theta}} \boldsymbol{\pi}_{\boldsymbol{\theta}}(\boldsymbol{s})^\top \Big], \tag{10a}$$

$$\Lambda(\boldsymbol{\theta}) \triangleq \mathbb{E}_{\boldsymbol{s}} \Big[\int \nabla_{\boldsymbol{\theta}} p(\boldsymbol{s}'|\boldsymbol{s}, \boldsymbol{\pi}_{\boldsymbol{\theta}}(\boldsymbol{s})) \nabla_{\boldsymbol{\theta}} V^{\boldsymbol{\pi_\theta}}(\boldsymbol{s}')^\top \mathrm{d}\boldsymbol{s}' + \int \nabla_{\boldsymbol{\theta}} V^{\boldsymbol{\pi_\theta}}(\boldsymbol{s}') \nabla_{\boldsymbol{\theta}} p(\boldsymbol{s}'|\boldsymbol{s}, \boldsymbol{\pi}_{\boldsymbol{\theta}}(\boldsymbol{s}))^\top \mathrm{d}\boldsymbol{s}' \Big] \tag{10b}$$

Proof. See Appendix. ■

The terms in (10a) only depend on the policy and the action-value function, but the terms in (10b) depend on the gradient of the transition probability $p(\boldsymbol{s}'|\boldsymbol{s}, \boldsymbol{a})$, which is difficult to calculate directly from data. Hence, we use $H(\boldsymbol{\theta})$ as a model-free approximator of the exact Hessian $\nabla^2_{\boldsymbol{\theta}} J$. Next section, we will show that the approximate Hessian $H(\boldsymbol{\theta})$ converges to the exact Hessian $\nabla^2_{\boldsymbol{\theta}} J$ at the optimal policy.

Remark 1. *Note that one can approximate* $p(\boldsymbol{s}'|\boldsymbol{s}, \boldsymbol{a})$ *from observed data in order to obtain a more accurate Hessian, e.g., using system identification techniques [14]. Such an estimation can require a heavy computation if the state-action space of the problem is not small. Hence, in order to provide a model-free approximator and for sake of brevity we ignore such evaluation in this paper.*

3 Quasi-Newton Policy Improvement

Quasi-Newton methods are alternative to Newton's approach where the Hessian of the cost function is unavailable or too expensive to compute at every iteration. A Quasi-Newton update rule for the optimization problem (3) can be written as follows:

$$\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k - \alpha H^{-1}(\boldsymbol{\theta}_k) \nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta})|_{\boldsymbol{\theta}=\boldsymbol{\theta}_k}, \tag{11}$$

where H is an approximation of Hessian of the performance function J. Note that using a Hessian in the policy optimization is advantageous when the different parameters would require very different step sizes in a first-order method, i.e., when $\nabla^2 J$ is far from being a multiple of the identity matrix. This is often the case in practice, unless a pre-scaling is performed on the policy formulation. From the computational viewpoint, the Hessian of a policy is usually dense, and it can be troublesome to use in (11) for a policy parametrization using a very large number of parameters. Hence the proposed second-order method is arguably best for policies using a few dozens, up to a few hundreds of parameters. E.g., policy parametrizations based on model predictive control techniques fall in that range of parameters [15]. Next mild assumptions are made to allow one to use the Newton-type optimization in the policy gradient methods.

Assumption 3. *1. The parameterized policy $\boldsymbol{\pi}_{\boldsymbol{\theta}}$ is rich enough. I.e., there exists $\boldsymbol{\theta}^\star$ such that $\boldsymbol{\pi}_{\boldsymbol{\theta}^\star}(\boldsymbol{s}) = \boldsymbol{\pi}^\star(\boldsymbol{s})$.*

2. $J(\boldsymbol{\theta})$ has a Lipschitz continuous Hessian and $\nabla^2_{\boldsymbol{\theta}} J(\boldsymbol{\theta})^{-1}$ exists in a neighbourhood of $\boldsymbol{\theta}^\star$.

The first statement of Assumption 3 is a standard assumption in the theoretical developments associated to the policy gradient method. For instance, for a Linear dynamic with Quadratic cost, a policy in the form of $\boldsymbol{\pi}_{\boldsymbol{\theta}}(\boldsymbol{s}) = \Theta_1 \boldsymbol{s} + \Theta_2$ with proper matrix dimension Θ_1 and Θ_2 satisfies Assumption 3.1, where $\boldsymbol{\theta} = \{\Theta_1, \Theta_2\}$. In practice, for a general problem such assumption is satisfied approximately by choosing a generic function approximator for the deterministic policy, e.g., Deep Neural Networks [16] and Fuzzy Neural Networks [17]. Then a richer policy satisfies the assumption asymptotically. A key consequence of this assumption is that the optimal policy $\boldsymbol{\pi}^\star$ is independent of the distribution of the initial state $p_1(\boldsymbol{s}_0)$. The second statement guarantees the continuity of the Hessian and allows one to use a Quasi-Newton approach.

Lemma 1. *Assume that $\boldsymbol{f} : \mathbb{R}^n \to \mathbb{R}^m$ is a bounded, continuous function of $\boldsymbol{x} \in \mathbb{R}^n$ and for any probability density $g(\boldsymbol{x})$, we have $\mathbb{E}_{\boldsymbol{x}\sim g}[\boldsymbol{f}(\boldsymbol{x})] = \mathbf{0}$. Then $\boldsymbol{f}(\boldsymbol{x}) = \mathbf{0}$ holds almost everywhere in Lebesgue measure.*

Proof. If $\boldsymbol{f}(\boldsymbol{x}) \neq \mathbf{0}$ holds on a measurable set, then there exists a probability density $\tilde{g}$ on that set such that $\mathbb{E}_{\boldsymbol{x}\sim\tilde{g}}[\boldsymbol{f}(\boldsymbol{x})] \neq 0$ ■

Theorem 3. *Under Assumptions 1-3, the approximate Hessian $H(\boldsymbol{\theta})$ converges to the exact Hessian $\nabla^2_{\boldsymbol{\theta}} J(\boldsymbol{\pi}_{\boldsymbol{\theta}})$ at the optimal policy, i.e.,*

$$\lim_{\boldsymbol{\theta}\to\boldsymbol{\theta}^\star} \Lambda(\boldsymbol{\theta}) = 0. \tag{12}$$

Proof. The initial distribution $p_1(\boldsymbol{s}_0)$ is independent of the policy parameters $\boldsymbol{\theta}$. From the optimality condition of (2), we have:

$$\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) = \nabla_{\boldsymbol{\theta}} \mathbb{E}_{\boldsymbol{s}_0}[V^{\boldsymbol{\pi}_{\boldsymbol{\theta}}}(\boldsymbol{s}_0)] = \mathbb{E}_{\boldsymbol{s}_0}[\nabla_{\boldsymbol{\theta}} V^{\boldsymbol{\pi}_{\boldsymbol{\theta}}}(\boldsymbol{s}_0)] = 0$$

at $\boldsymbol{\theta} = \boldsymbol{\theta}^\star$ for any initial distribution $p_1(\boldsymbol{s}_0)$ (Assumption 3.1). Using Lemma 1, it implies $\nabla_{\boldsymbol{\theta}} V^{\boldsymbol{\pi}_{\boldsymbol{\theta}}}(\boldsymbol{s}) \equiv 0$ at $\boldsymbol{\theta} = \boldsymbol{\theta}^\star$. Under Assumptions 3 and for any bounded $\nabla_{\boldsymbol{\theta}} p$, it reads:

$$\int \nabla_{\boldsymbol{\theta}} p(\boldsymbol{s}'|\boldsymbol{s}, \boldsymbol{\pi}_{\boldsymbol{\theta}}(\boldsymbol{s})) \nabla_{\boldsymbol{\theta}} V^{\boldsymbol{\pi}_{\boldsymbol{\theta}}}(\boldsymbol{s}')^\top \mathrm{d}\boldsymbol{s}' = \int \nabla_{\boldsymbol{\theta}} V^{\boldsymbol{\pi}_{\boldsymbol{\theta}}}(\boldsymbol{s}') \nabla_{\boldsymbol{\theta}} p(\boldsymbol{s}'|\boldsymbol{s}, \boldsymbol{\pi}_{\boldsymbol{\theta}}(\boldsymbol{s}))^\top \mathrm{d}\boldsymbol{s}' = 0 \quad (13)$$

at $\boldsymbol{\theta} = \boldsymbol{\theta}^\star$. Then, from the continuity of the Hessian (Assumption 3.2) and (10b), it implies (12). Note that Assumption 1 guarantees the boundedness of $\nabla_{\boldsymbol{\theta}} p$. ■

Next theorem provides necessary and sufficient conditions for the superlinear[1] convergence of the Quasi-Newton method.

Theorem 4. *(superlinear convergence of Quasi-Newton methods) Suppose that $f : \mathbb{R}^n \to \mathbb{R}$ is twice continuously differetiable. Consider the iteration $x_{k+1} = x_k - B_k^{-1} \nabla f_k$. Let us assume that $\{x_k\}$ converges to a point such that $\nabla f(x^\star) = 0$ and $\nabla^2 f(x^\star)$ is positive definite. Then $\{x_k\}$ converges superlinearly to $x^\star$ if and only if:*

$$\lim_{k \to \infty} \frac{\|(B_k - \nabla^2 f(x^\star)) B_k^{-1} \nabla f_k\|}{\|B_k^{-1} \nabla f_k\|} = 0. \quad (14)$$

Proof. See Theorem 3.7 in [4]. ■

Next corollary concludes that the proposed Hessian implies a superlinear converges.

Corollary 1. ***(From theorem 3 and 4):*** *Under assumption 3 and the assumptions in the theorem 4, the policy parameters $\boldsymbol{\theta}_k$ converge to the optimal policy parameters $\boldsymbol{\theta}^\star$ superlinearly, when $H(\boldsymbol{\theta})$ defined in* (10a) *is an approximator of the exact Hessian* (9) *with $J(\boldsymbol{\theta})$ defined in* (2) *and the Quasi-Newton update rule* (11) *is used.*

Natural policy gradient utilizes Fisher information matrix as its approximate Hessian in the policy gradient method. The Fisher matrix for deterministic policies can be written as follows [18]:

$$F(\boldsymbol{\theta}) = \mathbb{E}_{\boldsymbol{s}} \left[\nabla_{\boldsymbol{\theta}} \boldsymbol{\pi}_{\boldsymbol{\theta}}(\boldsymbol{s}) \nabla_{\boldsymbol{\theta}} \boldsymbol{\pi}_{\boldsymbol{\theta}}(\boldsymbol{s})^\top \right]. \quad (15)$$

The following corollary connects our proposed Hessian with the Fisher Information matrix.

Corollary 2. *Fisher Information matrix, defined in* (15)*, is positive definite and by comparison with* (10a) *and this matrix can be written equal to* (10a) *under the following conditions:*

1. $\nabla^2_{\boldsymbol{a}} Q^{\boldsymbol{\pi}_{\boldsymbol{\theta}}}(\boldsymbol{s}, \boldsymbol{a})|_{\boldsymbol{a}=\boldsymbol{\pi}_{\boldsymbol{\theta}}} = I$,
2. $\nabla^2_{\boldsymbol{\theta}} \boldsymbol{\pi}_{\boldsymbol{\theta}}(\boldsymbol{s}) \otimes \nabla_{\boldsymbol{a}} Q^{\boldsymbol{\pi}_{\boldsymbol{\theta}}}(\boldsymbol{s}, \boldsymbol{a})|_{\boldsymbol{a}=\boldsymbol{\pi}_{\boldsymbol{\theta}}} = 0$.

[1] The sequence x_k is said to converge superlinearly to L if $\lim_{k \to \infty} \frac{|x_{k+1} - L|}{|x_k - L|} = 0$.

Then clearly $F(\boldsymbol{\theta})$ does not converge to the exact Hessian at the optimal policy necessarily. I.e., the parameters will not converge superlinearly to the optimal parameters if the Fisher information matrix is used as a Hessian approximation (see Theorem 4).

Remark 2. *Under assumptions 1-3, $H(\boldsymbol{\theta})$ is positive definite in a neighborhood of $\boldsymbol{\theta}^\star$. Nevertheless $H(\boldsymbol{\theta})$ is not necessarily positive definite for a parameter $\boldsymbol{\theta}$ that is far from the optimal parameter $\boldsymbol{\theta}^\star$ because of the term $\nabla^2_{\boldsymbol{\theta}} \boldsymbol{\pi}_{\boldsymbol{\theta}}(\boldsymbol{s}) \otimes \nabla_{\boldsymbol{a}} Q^{\boldsymbol{\pi}_\theta}(\boldsymbol{s}, \boldsymbol{a})|_{\boldsymbol{a}=\boldsymbol{\pi}_{\boldsymbol{\theta}}}$, while the Fisher information matrix $F(\boldsymbol{\theta})$ is (semi) positive definite by construction. A regularization of H may be needed in practice, and one can use the Fisher information matrix F to regularize the approximate Hessian H, when H is not positive definite. This regularization can be applied using a Hessian in the form of $H + \beta F$ at every step, where $\beta \geq 0$ is a constant that must be ideally selected at every step. However, other methods e.g., trust-region methods can effectively take advantage of indefinite Hessian approximations.*

Remark 3. *Many RL methods deliver a sequence of parameters $\boldsymbol{\theta}_k$ that is stochastic by nature, because they are based on measurements taken from a stochastic system. From the theoretical viewpoint, all of the results in this paper are valid for large data sets, where sample averages converge to the true expectations. However, in practice, one can use the method to improve the stochastic convergence rate and derive an extension of the current theorems.*

4 Analytical Example

In this section, we consider a simple Linear Quadratic Regulator (LQR) problem in order to verify the method analytically. Consider the following scalar linear dynamics:

$$s^+ = s + a + w, \tag{16}$$

where $w \sim \mathcal{N}(0, \sigma^2)$, i.i.d., $\mathbb{E}_w[wa] = 0$ and $\mathbb{E}_w[ws] = 0$. Transition probability of the MDP (16) reads as follows:

$$p(s'|s,a) = \frac{1}{\sqrt{2\pi}\sigma} \exp\Big(- \frac{(s'-s-a)^2}{2\sigma^2} \Big). \tag{17}$$

Initial state distribution is $p_1(s_0) \sim \mathcal{N}(0, \sigma_0^2)$, deterministic policy reads as $\pi_\theta = -\theta s$ and stage cost is $\ell(s,a) = 0.5(s^2 + a^2)$. We assume value function in the from of $V^{\pi_\theta}(s) = p_\theta s^2 + q_\theta$ and we show it will satisfy the fundamental Bellman equations (1), then we have:

$$\begin{aligned} V^{\pi_\theta}(s) &= \ell(s, \pi_\theta(s)) + \gamma \mathbb{E}_w[V^{\pi_\theta}(s - \theta s + w)] \\ &= 0.5 s^2 (1 + \theta^2) + \gamma (1-\theta)^2 p_\theta s^2 + \gamma p_\theta \sigma^2 + \gamma q_\theta. \end{aligned} \tag{18}$$

It implies:

$$p_\theta = \frac{0.5(1+\theta^2)}{1 - \gamma(1-\theta)^2}, \qquad q_\theta = \frac{\gamma \sigma^2}{1-\gamma} p_\theta. \tag{19}$$

Using the Bellman equations (1), the action-value function $Q^{\pi_\theta}(s,a)$ can be evaluated as follows:

$$\begin{aligned} Q^{\pi_\theta}(s,a) &= \ell(s,a) + \gamma \mathbb{E}\left[V^{\pi_\theta}(s^+|s,a)\right] = 0.5(s^2+a^2) + \gamma \mathbb{E}[p_\theta(s+a+w)^2 + q_\theta] \\ &= (0.5+\gamma p_\theta)s^2 + 2\gamma p_\theta sa + (0.5+\gamma p_\theta)a^2 + q_\theta. \end{aligned} \tag{20}$$

One can check the identity $V^{\pi_\theta}(s) = Q^{\pi_\theta}(s,\pi(s))$. Then:

$$\nabla_\theta \pi_\theta \nabla_a Q^{\pi_\theta}(s,a)|_{a=\pi_\theta} = \frac{\gamma\theta^2+\theta-\gamma}{1-\gamma(1-\theta)^2} s^2 \tag{21a}$$

$$\nabla_\theta \pi_\theta \nabla_a^2 Q^{\pi_\theta}(s,a)|_{a=\pi_\theta} \nabla_\theta \pi_\theta = s^2(1+2\gamma p_\theta). \tag{21b}$$

Note that $\nabla_\theta^2 \pi_\theta = 0$. The closed-loop performance J reads:

$$J(\theta) = \mathbb{E}_{s_0}[V^{\pi_\theta}(s_0)] = \frac{0.5(1+\theta^2)}{1-\gamma(1-\theta)^2}(\sigma_0^2 + \frac{\gamma\sigma^2}{1-\gamma}). \tag{22}$$

Then, by taking derivation of J with respect to the parameters $\boldsymbol{\theta}$:

$$J'(\theta) = \frac{\gamma\theta^2+\theta-\gamma}{(1-\gamma(1-\theta)^2)^2}(\sigma_0^2 + \frac{\gamma\sigma^2}{1-\gamma}). \tag{23}$$

From policy gradient (6) and (21a), we can write:

$$J'(\theta) = \mathbb{E}_s[\nabla_\theta \pi_\theta \nabla_a Q^{\pi_\theta}(s,a)|_{a=\pi_\theta}] = \mathbb{E}_s[\frac{\gamma\theta^2+\theta-\gamma}{1-\gamma(1-\theta)^2} s^2]. \tag{24}$$

Then (23) and (24) imply:

$$\mathbb{E}_s[s^2] = \frac{(\sigma_0^2 + \frac{\gamma\sigma^2}{1-\gamma})}{1-\gamma(1-\theta)^2}. \tag{25}$$

From (22), the exact Hessian of the performance J reads:

$$J''(\theta) = p''_\theta(\sigma_0^2 + \frac{\gamma\sigma^2}{1-\gamma}) = \frac{-2\gamma^2\theta^3 - 3\gamma\theta^2 + 6\gamma^2\theta - 4\gamma^2 + \gamma - 1}{(1-\gamma(1-\theta)^2)^3}(\sigma_0^2 + \frac{\gamma\sigma^2}{1-\gamma}). \tag{26}$$

From (10a) and (21b), the approximate Hessian $H(\theta)$ reads:

$$H(\theta) = \mathbb{E}_s[s^2(1+2\gamma p_\theta)] = \frac{1+2\gamma\theta}{(1-\gamma(1-\theta)^2)^2}(\sigma_0^2 + \frac{\gamma\sigma^2}{1-\gamma}). \tag{27}$$

From (10b) and (17), we can write:

$$\begin{aligned} \Lambda(\theta) =& 2\int_{\mathcal{S}} \nabla_\theta V^{\pi_\theta}(s') \nabla_\theta p(s'|s,\pi_\theta) \mathrm{d}s' \\ =& 2\int_{-\infty}^{\infty} -p'_\theta((s')^2 + \frac{\gamma\sigma^2}{1-\gamma}) \frac{s(s'-s+\theta s)}{\sqrt{2\pi}\sigma^3} \exp\Big(-\frac{(s'-s+\theta s)^2}{2\sigma^2}\Big) \mathrm{d}s' \\ =& -4p'_\theta s^2(1-\theta) = \frac{-4(\gamma\theta^2+\theta-\gamma)(1-\theta)}{(1-\gamma(1-\theta)^2)^3}(\sigma_0^2 + \frac{\gamma\sigma^2}{1-\gamma}). \end{aligned} \tag{28}$$

Therefore, one can easily verify (9) by substitution (26), (27) and (28) in (9). Note that we used the following integration in (28):

$$\int_{-\infty}^{\infty}(x^2+a)(x-b)\exp(-c(x-b)^2)\mathrm{d}x=\frac{\sqrt{\pi}b}{c^{\frac{3}{2}}}, \tag{29}$$

where a, b and $c>0$ are constraints. Fig. 1 (right) compares the exact Hessian $\nabla_\theta^2 J(\theta)$, the proposed approximate Hessian $H(\theta)$ and the Fisher matrix $F(\theta)$ for this example with $\gamma=0.9$ and $\sigma_0^2=\sigma^2=0.1$. As can be seen, $\nabla_\theta^2 J$ meets $H(\theta)$ at the optimal parameter. Fig. 1 (left) shows the superlinear convergence of the policy parameters during the learning using Quasi-Newton policy gradient method, while the (first order) policy gradient method and natural policy gradient method result a linear convergence during the learning.

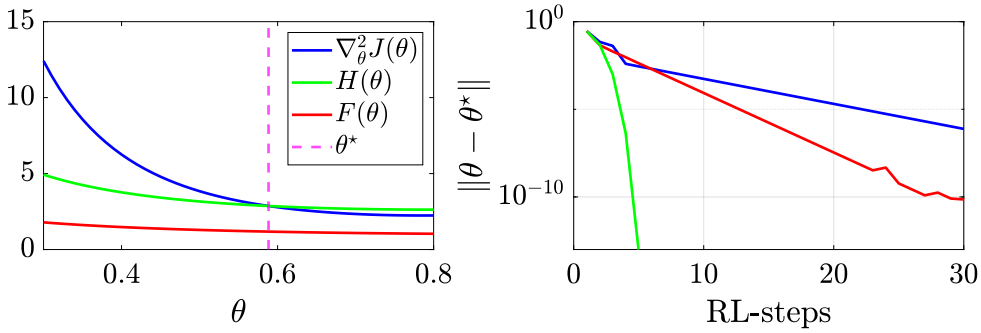


Figure 1: Right: Superlinear convergence of the proposed method. blue: policy gradient method, red: natural policy gradient method, green: proposed method. Left: Comparison of the exact Hessian $\nabla_\theta^2 J(\theta)$, the proposed approximate Hessian $H(\theta)$ and the Fisher matrix $F(\theta)$.

5 Numerical Simulation

Cart-Pendulum balancing is a well-known benchmark in the RL community. The dynamics of a cart-pendulum system, shown in fig. 2, reads as:

$$(M+m)\ddot{x}+\frac{1}{2}ml\ddot{\phi}\cos\phi=\frac{1}{2}ml\dot{\phi}^2\sin\phi+u, \tag{30a}$$

$$\frac{1}{3}ml^2\ddot{\phi}+\frac{1}{2}ml\ddot{x}\cos\phi=-\frac{1}{2}mgl\sin\phi, \tag{30b}$$

where M and m are the cart mass and pendulum mass, respectively, l is the pendulum length and ϕ is its angle from the vertical axis. Force u is the control input, x is the cart displacement and g is gravity. We used the Runge-Kutta 4^{th}-order method to discretize (30) with a sampling time $\mathrm{d}t=0.1\text{s}$ and cast it in the form of $\boldsymbol{s}^+=\boldsymbol{f}(\boldsymbol{s},\boldsymbol{a})+\boldsymbol{\xi}$, where $\boldsymbol{s}=[\dot{x},x,\dot{\phi},\phi]^\top$ is the

state, $\boldsymbol{a} = u$ is the input, $\boldsymbol{\xi}$ is a Gaussian noise and $\boldsymbol{f}$ is a nonlinear function representing (30) in discrete time. A stabilizing quadratic stage cost is considered as $\ell(\boldsymbol{s}, \boldsymbol{a}) = \boldsymbol{s}^\top \boldsymbol{s} + 0.01 \boldsymbol{a}^\top \boldsymbol{a}$, and the deterministic policy is considered in the form of $\boldsymbol{\pi_\theta} = -\boldsymbol{\theta s}$. Fig. 3 (right) shows

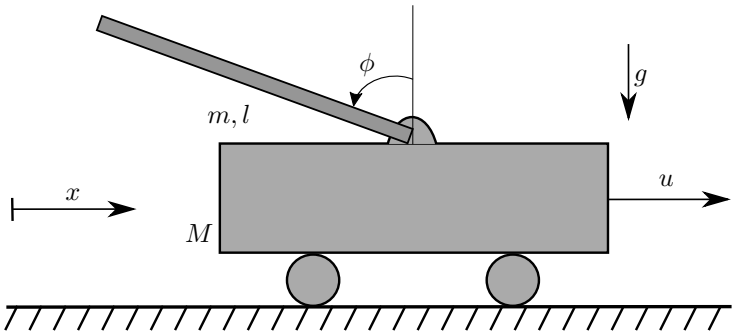


Figure 2: The cart-pendulum system. We use $M = 0.5\text{kg}$, $m = 0.2\text{kg}$, $l = 0.3\text{m}$ and $g = 9.8\text{m/s}^2$ for the simulation.

the closed-loop performance J using the proposed Hessian $H(\boldsymbol{\theta})$ (green) and natural policy gradient method (red). Moreover, the deterministic policy parameters $\boldsymbol{\theta}$ is shown in fig. 3 (left).

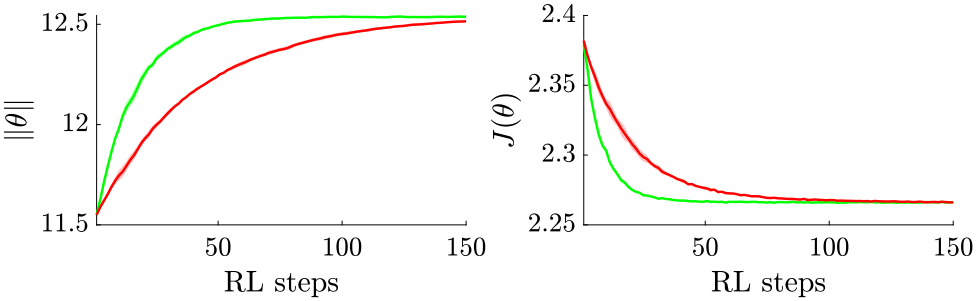


Figure 3: Right: Closed-loop performance $J(\boldsymbol{\theta})$; Left: Convergence of the policy parameters $\boldsymbol{\theta}$ using the proposed Hessian (green) and natural policy gradient method (red).

6 Conclusion

In this work, we provided a Hessian approximation for the performance of deterministic policies. We use the model-independent terms of the exact Hessian as an approximate Hessian, and we showed that the resulting approximate Hessian converges to the exact Hessian at the optimal policy. Therefore, the approximate Hessian can be used in the Quasi-Newton optimization to provide a superlinear convergence. We analytically verified our formulation

in a simple example, and we compare our method with the natural policy gradient in a cart-pendulum system. In the future, we will investigate actor-critic algorithms for the proposed Hessian.

References

[1] Dimitri P Bertsekas. *Dynamic programming and optimal control*. Vol. 1. 2. Athena scientific Belmont, MA, 1995.

[2] Dimitri P Bertsekas. *Reinforcement learning and optimal control*. Athena Scientific Belmont, MA, 2019.

[3] David Silver et al. "Deterministic Policy Gradient Algorithms". In: *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32*. ICML'14. Beijing, China: JMLR.org, 2014, I–387–I–395.

[4] Jorge Nocedal and Stephen Wright. *Numerical optimization*. Springer Science & Business Media, 2006.

[5] Maryam Fazel et al. "Global convergence of policy gradient methods for the linear quadratic regulator". In: *International Conference on Machine Learning*. PMLR. 2018, pp. 1467–1476.

[6] Thomas Furmston, Guy Lever, and David Barber. "Approximate newton methods for policy search in markov decision processes". In: *The Journal of Machine Learning Research* 17.1 (2016), pp. 8055–8105.

[7] Kay Hansel, Janosch Moos, and Cedric Derstroff. "Benchmarking the Natural Gradient in Policy Gradient Methods and Evolution Strategies". In: *Reinforcement Learning Algorithms: Analysis and Applications* (2021), pp. 69–84.

[8] Shun-ichi Amari. "Natural Gradient Works Efficiently in Learning". In: *Neural Computation* 10.2 (1998), pp. 251–276.

[9] Sham M Kakade. "A natural policy gradient". In: *Advances in neural information processing systems*. 2002, pp. 1531–1538.

[10] Dongsheng Ding et al. "Natural Policy Gradient Primal-Dual Method for Constrained Markov Decision Processes". In: *Advances in Neural Information Processing Systems* 33 (2020).

[11] Arash Givchi and Maziar Palhang. "Quasi Newton temporal difference learning". In: *Asian Conference on Machine Learning*. PMLR. 2015, pp. 159–172.

[12] Jan Peters, Sethu Vijayakumar, and Stefan Schaal. "Natural actor-critic". In: *European Conference on Machine Learning*. Springer. 2005, pp. 280–291.

[13] Karen Braman. "Third-order tensors as linear operators on a space of matrices". In: *Linear Algebra and its Applications* 433.7 (2010), pp. 1241–1253.

[14] Andreas B Martinsen, Anastasios M Lekkas, and Sébastien Gros. "Combining system identification with reinforcement learning-based MPC". In: *IFAC-PapersOnLine* 53.2 (2020), pp. 8130–8135.

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

[16] Vincent François-Lavet et al. "An introduction to deep reinforcement learning". In: *Foundations and Trends® in Machine Learning* 11.3-4 (2018), pp. 219–354.

[17] Arash Bahari Kordabad and Mehrdad Boroushaki. "Emotional learning based intelligent controller for mimo peripheral milling process". In: *Journal of Applied and Computational Mechanics* 6.3 (2020), pp. 480–492.

[18] J. Andrew (Drew) Bagnell and Jeff Schneider. "Covariant Policy Search". In: *Proceedings of the International Joint Conference on Artifical Intelligence*. 2003, pp. 1019–1024.

Proof of Theorem 2

Proof. We first calculate the Hessian of $V^{\boldsymbol{\pi}_{\boldsymbol{\theta}}}(\boldsymbol{s})$ as follows:

$$\begin{aligned}\nabla^2_{\boldsymbol{\theta}} V^{\boldsymbol{\pi}_{\boldsymbol{\theta}}}(\boldsymbol{s}) =& \nabla^2_{\boldsymbol{\theta}} Q^{\boldsymbol{\pi}_{\boldsymbol{\theta}}}(\boldsymbol{s},\boldsymbol{a})|_{\boldsymbol{a}=\boldsymbol{\pi}_{\boldsymbol{\theta}}(\boldsymbol{s})} = \nabla^2_{\boldsymbol{\theta}} \left(\ell(\boldsymbol{s},\boldsymbol{\pi}_{\boldsymbol{\theta}}(\boldsymbol{s})) + \int_{\mathcal{S}} \gamma p(\boldsymbol{s}'|\boldsymbol{s},\boldsymbol{\pi}_{\boldsymbol{\theta}}(\boldsymbol{s})) V^{\boldsymbol{\pi}_{\boldsymbol{\theta}}}(\boldsymbol{s}') \mathrm{d}\boldsymbol{s}' \right) \\ =& \nabla^2_{\boldsymbol{\theta}} \boldsymbol{\pi}_{\boldsymbol{\theta}}(\boldsymbol{s}) \otimes \nabla_{\boldsymbol{a}} \ell(\boldsymbol{s},\boldsymbol{a})|_{\boldsymbol{a}=\boldsymbol{\pi}_{\boldsymbol{\theta}}(\boldsymbol{s})} + \nabla_{\boldsymbol{\theta}} \boldsymbol{\pi}_{\boldsymbol{\theta}}(\boldsymbol{s}) \nabla^2_{\boldsymbol{a}} \ell(\boldsymbol{s},\boldsymbol{a})|_{\boldsymbol{a}=\boldsymbol{\pi}_{\boldsymbol{\theta}}} \nabla_{\boldsymbol{\theta}} \boldsymbol{\pi}_{\boldsymbol{\theta}}(\boldsymbol{s})^\top \\ &+ \nabla^2_{\boldsymbol{\theta}} \int_{\mathcal{S}} \gamma p(\boldsymbol{s}'|\boldsymbol{s},\boldsymbol{a}) V^{\boldsymbol{\pi}_{\boldsymbol{\theta}}}(\boldsymbol{s}') \mathrm{d}\boldsymbol{s}' \end{aligned} \tag{A.1}$$

The third term can be calculated as follows:

$$\begin{aligned}\nabla^2_{\boldsymbol{\theta}} \int_{\mathcal{S}} \gamma p(\boldsymbol{s}'|\boldsymbol{s},\boldsymbol{a}) V^{\boldsymbol{\pi}_{\boldsymbol{\theta}}}(\boldsymbol{s}') \mathrm{d}\boldsymbol{s}' =& \int_{\mathcal{S}} \gamma V^{\boldsymbol{\pi}_{\boldsymbol{\theta}}}(\boldsymbol{s}') \nabla^2_{\boldsymbol{\theta}} p(\boldsymbol{s}'|\boldsymbol{s},\boldsymbol{\pi}_{\boldsymbol{\theta}}(\boldsymbol{s})) \mathrm{d}\boldsymbol{s}' \\ &+ \int_{\mathcal{S}} \gamma \nabla_{\boldsymbol{\theta}} p(\boldsymbol{s}'|\boldsymbol{s},\boldsymbol{\pi}_{\boldsymbol{\theta}}(\boldsymbol{s})) \nabla_{\boldsymbol{\theta}} V^{\boldsymbol{\pi}_{\boldsymbol{\theta}}}(\boldsymbol{s}')^\top \mathrm{d}\boldsymbol{s}' + \int_{\mathcal{S}} \gamma \nabla_{\boldsymbol{\theta}} V^{\boldsymbol{\pi}_{\boldsymbol{\theta}}}(\boldsymbol{s}') \nabla_{\boldsymbol{\theta}} p(\boldsymbol{s}'|\boldsymbol{s},\boldsymbol{\pi}_{\boldsymbol{\theta}}(\boldsymbol{s}))^\top \mathrm{d}\boldsymbol{s}' \\ &+ \int_{\mathcal{S}} \gamma p(\boldsymbol{s}'|\boldsymbol{s},\boldsymbol{\pi}_{\boldsymbol{\theta}}(\boldsymbol{s})) \nabla^2_{\boldsymbol{\theta}} V^{\boldsymbol{\pi}_{\boldsymbol{\theta}}}(\boldsymbol{s}') \mathrm{d}\boldsymbol{s}' \end{aligned} \tag{A.2}$$

The first term can be extended as follows:

$$\begin{aligned}\int_{\mathcal{S}} \gamma V^{\boldsymbol{\pi}_{\boldsymbol{\theta}}}(\boldsymbol{s}') \nabla^2_{\boldsymbol{\theta}} p(\boldsymbol{s}'|\boldsymbol{s},\boldsymbol{\pi}_{\boldsymbol{\theta}}(\boldsymbol{s})) \mathrm{d}\boldsymbol{s}' =& \int_{\mathcal{S}} \gamma V^{\boldsymbol{\pi}_{\boldsymbol{\theta}}}(\boldsymbol{s}') \nabla^2_{\boldsymbol{\theta}} \boldsymbol{\pi}_{\boldsymbol{\theta}}(\boldsymbol{s}) \otimes \nabla_{\boldsymbol{a}} p(\boldsymbol{s}'|\boldsymbol{s},\boldsymbol{a})|_{\boldsymbol{a}=\boldsymbol{\pi}_{\boldsymbol{\theta}}(\boldsymbol{s})} \mathrm{d}\boldsymbol{s}' + \\ &+ \int_{\mathcal{S}} \gamma V^{\boldsymbol{\pi}_{\boldsymbol{\theta}}}(\boldsymbol{s}') \nabla_{\boldsymbol{\theta}} \boldsymbol{\pi}_{\boldsymbol{\theta}}(\boldsymbol{s}) \nabla^2_{\boldsymbol{a}} p(\boldsymbol{s}'|\boldsymbol{s},\boldsymbol{a})|_{\boldsymbol{a}=\boldsymbol{\pi}_{\boldsymbol{\theta}}(\boldsymbol{s})} \nabla_{\boldsymbol{\theta}} \boldsymbol{\pi}_{\boldsymbol{\theta}}(\boldsymbol{s})^\top \mathrm{d}\boldsymbol{s}' \end{aligned} \tag{A.3}$$

By rearranging (A.1), we can write:

$$\begin{aligned}\nabla^2_{\boldsymbol{\theta}} V^{\boldsymbol{\pi}_{\boldsymbol{\theta}}}(\boldsymbol{s}) = \nabla^2_{\boldsymbol{\theta}} \boldsymbol{\pi}_{\boldsymbol{\theta}}(\boldsymbol{s}) \otimes \nabla_{\boldsymbol{a}} (\ell(\boldsymbol{s},\boldsymbol{a})|_{\boldsymbol{a}=\boldsymbol{\pi}_{\boldsymbol{\theta}}(\boldsymbol{s})} + \int_{\mathcal{S}} \gamma p(\boldsymbol{s}'|\boldsymbol{s},\boldsymbol{a})|_{\boldsymbol{a}=\boldsymbol{\pi}_{\boldsymbol{\theta}}(\boldsymbol{s})} V^{\boldsymbol{\pi}_{\boldsymbol{\theta}}}(\boldsymbol{s}') \mathrm{d}\boldsymbol{s}') \\ + \nabla_{\boldsymbol{\theta}} \boldsymbol{\pi}_{\boldsymbol{\theta}}(\boldsymbol{s}) \nabla^2_{\boldsymbol{a}} (\ell(\boldsymbol{s},\boldsymbol{a})|_{\boldsymbol{a}=\boldsymbol{\pi}_{\boldsymbol{\theta}}(\boldsymbol{s})} + \int_{\mathcal{S}} \gamma p(\boldsymbol{s}'|\boldsymbol{s},\boldsymbol{a})|_{\boldsymbol{a}=\boldsymbol{\pi}_{\boldsymbol{\theta}}(\boldsymbol{s})} V^{\boldsymbol{\pi}_{\boldsymbol{\theta}}}(\boldsymbol{s}') \mathrm{d}\boldsymbol{s}') \nabla_{\boldsymbol{\theta}} \boldsymbol{\pi}_{\boldsymbol{\theta}}(\boldsymbol{s})^\top \\ + \int_{\mathcal{S}} \gamma \nabla_{\boldsymbol{\theta}} p(\boldsymbol{s}'|\boldsymbol{s},\boldsymbol{\pi}_{\boldsymbol{\theta}}(\boldsymbol{s})) \nabla_{\boldsymbol{\theta}} V^{\boldsymbol{\pi}_{\boldsymbol{\theta}}}(\boldsymbol{s}')^\top \mathrm{d}\boldsymbol{s}' + \int_{\mathcal{S}} \gamma \nabla_{\boldsymbol{\theta}} V^{\boldsymbol{\pi}_{\boldsymbol{\theta}}}(\boldsymbol{s}') \nabla_{\boldsymbol{\theta}} p(\boldsymbol{s}'|\boldsymbol{s},\boldsymbol{\pi}_{\boldsymbol{\theta}}(\boldsymbol{s}))^\top \mathrm{d}\boldsymbol{s}' \\ + \int_{\mathcal{S}} \gamma p(\boldsymbol{s}'|\boldsymbol{s},\boldsymbol{\pi}_{\boldsymbol{\theta}}(\boldsymbol{s})) \nabla^2_{\boldsymbol{\theta}} V^{\boldsymbol{\pi}_{\boldsymbol{\theta}}}(\boldsymbol{s}') \mathrm{d}\boldsymbol{s}' = \mathcal{F}_{\boldsymbol{\theta}}(\boldsymbol{s}) + \int_{\mathcal{S}} \gamma p(\boldsymbol{s}'|\boldsymbol{s},\boldsymbol{\pi}_{\boldsymbol{\theta}}(\boldsymbol{s})) \nabla^2_{\boldsymbol{\theta}} V^{\boldsymbol{\pi}_{\boldsymbol{\theta}}}(\boldsymbol{s}') \mathrm{d}\boldsymbol{s}'\end{aligned} \tag{A.4}$$

where $\mathcal{F}_{\boldsymbol{\theta}}(\boldsymbol{s})$ is defined as follows:

$$\begin{aligned}\mathcal{F}_{\boldsymbol{\theta}}(\boldsymbol{s}) \triangleq \nabla^2_{\boldsymbol{\theta}} \boldsymbol{\pi}_{\boldsymbol{\theta}}(\boldsymbol{s}) \otimes \nabla_{\boldsymbol{a}} Q^{\boldsymbol{\pi}_{\boldsymbol{\theta}}}(\boldsymbol{s},\boldsymbol{a})|_{\boldsymbol{a}=\boldsymbol{\pi}_{\boldsymbol{\theta}}(\boldsymbol{s})} + \nabla_{\boldsymbol{\theta}} \boldsymbol{\pi}_{\boldsymbol{\theta}}(\boldsymbol{s}) \nabla^2_{\boldsymbol{a}} Q^{\boldsymbol{\pi}_{\boldsymbol{\theta}}}(\boldsymbol{s},\boldsymbol{a})|_{\boldsymbol{a}=\boldsymbol{\pi}_{\boldsymbol{\theta}}(\boldsymbol{s})} \nabla_{\boldsymbol{\theta}} \boldsymbol{\pi}_{\boldsymbol{\theta}}(\boldsymbol{s})^\top + \\ \int_{\mathcal{S}} \gamma \nabla_{\boldsymbol{\theta}} p(\boldsymbol{s}'|\boldsymbol{s},\boldsymbol{\pi}_{\boldsymbol{\theta}}(\boldsymbol{s})) \nabla_{\boldsymbol{\theta}} V^{\boldsymbol{\pi}_{\boldsymbol{\theta}}}(\boldsymbol{s}')^\top \mathrm{d}\boldsymbol{s}' + \int_{\mathcal{S}} \gamma \nabla_{\boldsymbol{\theta}} V^{\boldsymbol{\pi}_{\boldsymbol{\theta}}}(\boldsymbol{s}') \nabla_{\boldsymbol{\theta}} p(\boldsymbol{s}'|\boldsymbol{s},\boldsymbol{\pi}_{\boldsymbol{\theta}}(\boldsymbol{s}))^\top \mathrm{d}\boldsymbol{s}'\end{aligned} \tag{A.5}$$

where we used:

$$Q^{\boldsymbol{\pi}_{\boldsymbol{\theta}}}(\boldsymbol{s},\boldsymbol{a}) = \ell(\boldsymbol{s},\boldsymbol{a})|_{\boldsymbol{a}=\boldsymbol{\pi}_{\boldsymbol{\theta}}(\boldsymbol{s})} + \int_{\mathcal{S}} \gamma p(\boldsymbol{s}'|\boldsymbol{s},\boldsymbol{a})|_{\boldsymbol{a}=\boldsymbol{\pi}_{\boldsymbol{\theta}}(\boldsymbol{s})} V^{\boldsymbol{\pi}_{\boldsymbol{\theta}}}(\boldsymbol{s}') \mathrm{d}\boldsymbol{s}' \tag{A.6}$$

Now, we can go one step further for the last term of (A.4):

$$\begin{aligned}\nabla^2_{\boldsymbol{\theta}} V^{\boldsymbol{\pi}_{\boldsymbol{\theta}}}(\boldsymbol{s}) = \mathcal{F}_{\boldsymbol{\theta}}(\boldsymbol{s}) + \int_{\mathcal{S}} \gamma p(\boldsymbol{s}'|\boldsymbol{s},\boldsymbol{\pi}_{\boldsymbol{\theta}}(\boldsymbol{s})) \mathcal{F}_{\boldsymbol{\theta}}(\boldsymbol{s}') \mathrm{d}\boldsymbol{s}' + \\ \int_{\mathcal{S}} \int_{\mathcal{S}} \gamma^2 p(\boldsymbol{s}'|\boldsymbol{s},\boldsymbol{\pi}_{\boldsymbol{\theta}}(\boldsymbol{s})) p(\boldsymbol{s}''|\boldsymbol{s}',\boldsymbol{\pi}_{\boldsymbol{\theta}}(\boldsymbol{s}')) \nabla^2_{\boldsymbol{\theta}} V^{\boldsymbol{\pi}_{\boldsymbol{\theta}}}(\boldsymbol{s}'') \mathrm{d}\boldsymbol{s}' \mathrm{d}\boldsymbol{s}''\end{aligned} \tag{A.7}$$

where we have used the following equality:

$$\nabla^2_{\boldsymbol{\theta}} V^{\boldsymbol{\pi}_{\boldsymbol{\theta}}}(\boldsymbol{s}') = \mathcal{F}_{\boldsymbol{\theta}}(\boldsymbol{s}') + \int_{\mathcal{S}} \gamma p(\boldsymbol{s}''|\boldsymbol{s}',\boldsymbol{\pi}_{\boldsymbol{\theta}}(\boldsymbol{s}')) \nabla^2_{\boldsymbol{\theta}} V^{\boldsymbol{\pi}_{\boldsymbol{\theta}}}(\boldsymbol{s}'') \mathrm{d}\boldsymbol{s}'' \tag{A.8}$$

We can define:

$$p(\boldsymbol{s} \to \boldsymbol{s}'', 2, \boldsymbol{\pi}_{\boldsymbol{\theta}}) = \int_{\mathcal{S}} p(\boldsymbol{s}'|\boldsymbol{s},\boldsymbol{\pi}_{\boldsymbol{\theta}}(\boldsymbol{s})) p(\boldsymbol{s}''|\boldsymbol{s}',\boldsymbol{\pi}_{\boldsymbol{\theta}}(\boldsymbol{s}')) \mathrm{d}\boldsymbol{s}'$$

and interpret it probability of transition from $\boldsymbol{s}$ to $\boldsymbol{s}''$ in 2 steps by policy $\boldsymbol{\pi}_{\boldsymbol{\theta}}$. Then in last term we can alter integral notation $\boldsymbol{s}'' \to \boldsymbol{s}'$ and rewrite (A.7) as follows:

$$\begin{aligned}\nabla^2_{\boldsymbol{\theta}} V^{\boldsymbol{\pi}_{\boldsymbol{\theta}}}(\boldsymbol{s}) = \mathcal{F}_{\boldsymbol{\theta}}(\boldsymbol{s}) + \int_{\mathcal{S}} \gamma p(\boldsymbol{s}'|\boldsymbol{s},\boldsymbol{\pi}_{\boldsymbol{\theta}}(\boldsymbol{s})) \mathcal{F}_{\boldsymbol{\theta}}(\boldsymbol{s}') \mathrm{d}\boldsymbol{s}' + \\ \int_{\mathcal{S}} \gamma^2 p(\boldsymbol{s} \to \boldsymbol{s}', 2, \boldsymbol{\pi}_{\boldsymbol{\theta}}) \nabla^2_{\boldsymbol{\theta}} V^{\boldsymbol{\pi}_{\boldsymbol{\theta}}}(\boldsymbol{s}') \mathrm{d}\boldsymbol{s}'\end{aligned} \tag{A.9}$$

By continuing this procedure, we have:

$$\nabla^2_{\boldsymbol{\theta}} V^{\boldsymbol{\pi}_{\boldsymbol{\theta}}}(\boldsymbol{s}) = \int_{\mathcal{S}} \sum_{t=0}^{\infty} \gamma^t p(\boldsymbol{s} \to \boldsymbol{s}', t, \boldsymbol{\pi}_{\boldsymbol{\theta}}) \mathcal{F}_{\boldsymbol{\theta}}(\boldsymbol{s}') \mathrm{d}\boldsymbol{s}' \tag{A.10}$$

where

$$p(\boldsymbol{s} \to \boldsymbol{s}', t, \boldsymbol{\pi}_{\boldsymbol{\theta}}) = \int_{\mathcal{S}} p(\boldsymbol{s} \to \hat{\boldsymbol{s}}, t-1, \boldsymbol{\pi}_{\boldsymbol{\theta}}) p(\boldsymbol{s}' | \hat{\boldsymbol{s}}, \boldsymbol{\pi}_{\boldsymbol{\theta}}(\hat{\boldsymbol{s}})) \mathrm{d}\hat{\boldsymbol{s}}$$

starting from $p(\boldsymbol{s} \to \boldsymbol{s}', 1, \boldsymbol{\pi}_{\boldsymbol{\theta}}) = p(\boldsymbol{s}' | \boldsymbol{s}, \boldsymbol{\pi}_{\boldsymbol{\theta}}(\boldsymbol{s}))$. Then, tacking the expectation over p_1 for Hessian of policy we have:

$$\begin{aligned}
\nabla^2_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) = \nabla^2_{\boldsymbol{\theta}} \int_{\mathcal{S}} p_1(\boldsymbol{s}) V^{\boldsymbol{\pi}_{\boldsymbol{\theta}}}(\boldsymbol{s}) \mathrm{d}\boldsymbol{s} = \int_{\mathcal{S}} p_1(\boldsymbol{s}) \nabla^2_{\boldsymbol{\theta}} V^{\boldsymbol{\pi}_{\boldsymbol{\theta}}}(\boldsymbol{s}) \mathrm{d}\boldsymbol{s} = \int_{\mathcal{S}} \int_{\mathcal{S}} \sum_{t=0}^{\infty} \gamma^t p_1(\boldsymbol{s}) \\
p(\boldsymbol{s} \to \boldsymbol{s}', t, \boldsymbol{\pi}_{\boldsymbol{\theta}}) \Big[\nabla^2_{\boldsymbol{\theta}} \boldsymbol{\pi}_{\boldsymbol{\theta}}(\boldsymbol{s}') \otimes \nabla_{\boldsymbol{a}} Q_{\boldsymbol{\pi}_{\boldsymbol{\theta}}}(\boldsymbol{s}', \boldsymbol{a})|_{\boldsymbol{a}=\boldsymbol{\pi}_{\boldsymbol{\theta}}(\boldsymbol{s}')} + \\
\nabla_{\boldsymbol{\theta}} \boldsymbol{\pi}_{\boldsymbol{\theta}}(\boldsymbol{s}') \nabla^2_{\boldsymbol{a}} Q^{\boldsymbol{\pi}_{\boldsymbol{\theta}}}(\boldsymbol{s}', \boldsymbol{a})|_{\boldsymbol{a}=\boldsymbol{\pi}_{\boldsymbol{\theta}}(\boldsymbol{s}')} \nabla_{\boldsymbol{\theta}} \boldsymbol{\pi}_{\boldsymbol{\theta}}(\boldsymbol{s}')^\top + \int_{\mathcal{S}} \gamma \nabla_{\boldsymbol{\theta}} p(\boldsymbol{s}'' | \boldsymbol{s}', \boldsymbol{\pi}_{\boldsymbol{\theta}}(\boldsymbol{s}')) \\
\nabla_{\boldsymbol{\theta}} V^{\boldsymbol{\pi}_{\boldsymbol{\theta}}}(\boldsymbol{s}'')^\top \mathrm{d}\boldsymbol{s}'' + \int_{\mathcal{S}} \gamma \nabla_{\boldsymbol{\theta}} V^{\boldsymbol{\pi}_{\boldsymbol{\theta}}}(\boldsymbol{s}'') \nabla_{\boldsymbol{\theta}} p(\boldsymbol{s}'' | \boldsymbol{s}', \boldsymbol{\pi}_{\boldsymbol{\theta}}(\boldsymbol{s}'))^\top \mathrm{d}\boldsymbol{s}'' \Big] \mathrm{d}\boldsymbol{s}' \mathrm{d}\boldsymbol{s}
\end{aligned}$$

Or equivalently:

$$\begin{aligned}
\nabla^2_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) = \mathbb{E}_{\boldsymbol{s}} \Big[\nabla^2_{\boldsymbol{\theta}} \boldsymbol{\pi}_{\boldsymbol{\theta}}(\boldsymbol{s}) \otimes \nabla_{\boldsymbol{a}} Q^{\boldsymbol{\pi}_{\boldsymbol{\theta}}}(\boldsymbol{s}, \boldsymbol{a})|_{\boldsymbol{a}=\boldsymbol{\pi}_{\boldsymbol{\theta}}} + \nabla_{\boldsymbol{\theta}} \boldsymbol{\pi}_{\boldsymbol{\theta}}(\boldsymbol{s}) \nabla^2_{\boldsymbol{a}} Q^{\boldsymbol{\pi}_{\boldsymbol{\theta}}}(\boldsymbol{s}, \boldsymbol{a})|_{\boldsymbol{a}=\boldsymbol{\pi}_{\boldsymbol{\theta}}} \nabla_{\boldsymbol{\theta}} \boldsymbol{\pi}_{\boldsymbol{\theta}}(\boldsymbol{s})^\top \\
+ \int \gamma \nabla_{\boldsymbol{\theta}} V^{\boldsymbol{\pi}_{\boldsymbol{\theta}}}(\boldsymbol{s}') \nabla_{\boldsymbol{\theta}} p(\boldsymbol{s}' | \boldsymbol{s}, \boldsymbol{\pi}_{\boldsymbol{\theta}}(\boldsymbol{s}))^\top \mathrm{d}\boldsymbol{s}' + \int \gamma \nabla_{\boldsymbol{\theta}} p(\boldsymbol{s}' | \boldsymbol{s}, \boldsymbol{\pi}_{\boldsymbol{\theta}}(\boldsymbol{s})) \nabla_{\boldsymbol{\theta}} V^{\boldsymbol{\pi}_{\boldsymbol{\theta}}}(\boldsymbol{s}')^\top \mathrm{d}\boldsymbol{s}' \Big]
\end{aligned} \tag{A.11}$$

where $\mathbb{E}_{\boldsymbol{s}}[\cdot]$ is taken over discounted state distribution of the Markov chain in closed-loop with policy $\boldsymbol{\pi}_{\boldsymbol{\theta}}$. ■

F Functional Stability of Discounted Markov Decision Processes Using Economic MPC Dissipativity Theory

Postprint of [100] **Arash Bahari Kordabad** and Sebastien Gros. "Functional Stability of Discounted Markov Decision Processes Using Economic MPC Dissipativity Theory". In: *2022 European Control Conference (ECC)* (2022), pp. 1858–1863. DOI: 10.23919/ECC55457.2022.9838064

Functional Stability of Discounted Markov Decision Processes Using Economic MPC Dissipativity Theory

Arash Bahari Kordabad[1] and Sebastien Gros[1]

[1]Department of Engineering Cybernetics, Norwegian University of Science and Technology (NTNU), Trondheim, Norway.

Abstract: This paper discusses the functional stability of closed-loop Markov Chains under optimal policies resulting from a discounted optimality criterion, forming Markov Decision Processes (MDPs). We investigate the stability of MDPs in the sense of probability measures (densities) underlying the state distributions and extend the dissipativity theory of Economic Model Predictive Control in order to characterize the MDP stability. This theory requires a so-called storage function satisfying a dissipativity inequality. In the probability measures space and for the discounted setting, we introduce new dissipativity conditions ensuring the MDP stability. We then use finite-horizon optimal control problems in order to generate valid storage functionals. In practice, we propose to use Q-learning to compute the storage functionals.

1 Introduction

Markov Decision Processes (MDPs) provide a generic and standard framework for optimal stochastic control of discrete-time dynamical systems, where the stage cost and transition probability depend only on the current state and the current input of the system [1]. For an MDP, a policy is a mapping from the state space into the input space and determines how to select the input based on the observation of the current state. Solving an MDP refers to finding an optimal policy that typically minimizes the expected value of the discounted infinite-horizon sum of stage costs. Reinforcement Learning (RL) and Dynamic programming are two common techniques to solve MDPs [2].

Most of the research has been done in order to find the optimal policy or verify the optimality of a given policy. However, in general, optimality may not lead to the stability of the closed-loop Markov Chain. The stability of the Markov Chains has been extensively studied in [3]. However, this framework provides results that are not easily related to MDPs and optimality criteria. To the best of our knowledge, there are limited results characterizing the stability of MDPs as an outcome of the interplay between its objective function and its dynamics.

In order to characterize the closed-loop stability of MDPs, we extend the concept of stability

and dissipativity developed in the context of Economic Model Predictive Control (EMPC) [4]. EMPC optimizes a sum of stage costs that is not necessarily positive definite [5]. Dissipativity is a key concept in EMPC to argue about the asymptotic stability of the closed-loop system under the optimal policy [6]. This theory is based on a so-called *storage function* satisfying the dissipativity inequality. The storage function can be used to convert an EMPC to a *tracking* MPC having a stage cost that is lower bounded by a $\mathcal{K}_\infty$ function. Under the dissipativity condition, one can show that the tracking MPC has the same optimal policy as the EMPC. Moreover, the value function resulting from the tracking MPC can be used as a Lyapunov function to show the closed-loop stability of the system under the optimal policy.

Dissipativity is well-known for EMPC schemes having an undiscounted cost and deterministic dynamics. In the discounted setting, finding the Lyapunov function still is challenging even for positive-definite stage costs [7]. In the discounted setting, the discount factor plays a vital role in closed-loop stability. Recently the dissipativity theory has been extended to the discounted setting with deterministic dynamics [8]. These conditions are called *Strong Discounted Strict Dissipativity (SDSD)*.

We use the generalization of the classic dissipativity theory by making an argument on the measure space underlying the MDP rather than on the state space itself. This idea was first discussed in [9], but was limited to undiscounted MDPs, where the dissipativity is fairly straightforward. In this paper, we consider MDPs with a general functional stage cost. We use the concept of D-stability [9] and introduce generalized functional dissipativity conditions for MDPs with a discounted objective function. We label these conditions Functional Strong Discounted Strict Dissipativity (FSDSD). These conditions require the transition probability, the stage cost, and the discount factor of the MDP to satisfy certain inequalities. We show that if a given problem is FSDSD, then the D-stability of MDP follows.

Moreover, [9] covers only the stability analysis, while we discuss it in the learning context and provide practical aspects of the method. Indeed, first, we show that an undiscounted finite-horizon Optimal Control Problem (OCP) is able to capture the optimal value functionals and policy resulting from a discounted infinite-horizon OCP. Then we use a parameterized undiscounted finite-horizon OCP to approximate the action-value functional and show that this framework yields a valid storage function that satisfies FSDSD conditions. Q-learning will be proposed as a practical way of learning the OCP parameters.

2 Problem Setting

In this section, we detail Markov Decision Processes (MDPs) and formulate their representation in the state density space. We consider an MDP with the following transition probability density:

$$\xi(\boldsymbol{s}_{k+1}|\boldsymbol{s}_k, \boldsymbol{a}_k) \ , \tag{1}$$

where $\boldsymbol{s}_k \in \mathcal{X} \subset \mathbb{R}^n$, $\boldsymbol{a}_k \in \mathcal{U} \subset \mathbb{R}^m$, and $\boldsymbol{s}_{k+1}$ are the current state, input, and subsequent state, respectively, and $k \in \mathbb{I}_{\geq 0}$ is the discrete-time index. The input $\boldsymbol{a}_k$ applied to the system

for a given state $\boldsymbol{s}_k$ is selected by a deterministic policy $\boldsymbol{\pi} : \mathcal{X} \to \mathcal{U}$. We label $\mathcal{P}$ the set of policies such that the conditional measure (1) is σ-finite, i.e., $\boldsymbol{\pi} \in \mathcal{P}$. We denote $\rho_0 \in \Xi$ as the initial state $\boldsymbol{s}_0$ distribution, i.e $\boldsymbol{s}_0 \sim \rho_0$, where Ξ is the set of measures supported on $\mathcal{X}$. We define probability measure sequences $\rho_k^{\boldsymbol{\pi}} \in \Xi$ generated by the closed-loop Markov Chain $\xi(\boldsymbol{s}^+|\boldsymbol{s}, \boldsymbol{\pi}(\boldsymbol{s}))$ with policy $\boldsymbol{\pi}$, as:

$$\rho_{k+1}^{\boldsymbol{\pi}}(\cdot) = \mathcal{T}_{\boldsymbol{\pi}} \rho_k^{\boldsymbol{\pi}}(\cdot) = \int_{\mathcal{X}} \xi(\cdot|\boldsymbol{s}, \boldsymbol{\pi}(\boldsymbol{s})) \rho_k^{\boldsymbol{\pi}}(\mathrm{d}\boldsymbol{s}) \ , \tag{2}$$

where $\mathcal{T}_{\boldsymbol{\pi}} : \Xi \to \Xi$ is defined as the transition operator on measures and $\rho_0^{\boldsymbol{\pi}} = \rho_0, \forall \boldsymbol{\pi}$. In general, characterization of convergence of the state sequences $\{\boldsymbol{s}_k\}_{k=0}^{\infty}$ resulting from the closed-loop Markov Chain $\xi(\boldsymbol{s}^+|\boldsymbol{s}, \boldsymbol{\pi}(\boldsymbol{s}))$ is very difficult. To tackle this issue, in this paper, instead of working with state sequences $\boldsymbol{s}_k$, we propose to work with probability measure sequences $\{\rho_k^{\boldsymbol{\pi}}\}_{k=0}^{\infty}$, describing the probability distribution of the states $\boldsymbol{s}_k$ over time. The selected (possibly nonlinear) stage cost functional, denoted by $\mathcal{L} : \Xi \times \mathcal{U} \to \mathbb{R}$, does not have a specific structure and it will be an important point in the rest of the paper. One can select it as follows:

$$\mathcal{L}[\rho_k^{\boldsymbol{\pi}}, \boldsymbol{\pi}] = \mathbb{E}_{\boldsymbol{s} \sim \rho_k^{\boldsymbol{\pi}}} \left[\ell(\boldsymbol{s}, \boldsymbol{\pi}(\boldsymbol{s})) \right] \ , \tag{3}$$

where $\ell : \mathcal{X} \times \mathcal{U} \to \mathbb{R}$ is a stage cost function. In fact, stage cost (3) is a particular case of functional stage cost, where it linearly depends on the stage function. Using cost functionals $\mathcal{L}$ that do not necessarily take the form (3) is key in this paper to discuss the functional stability of closed-loop Markov Chains. We then denote the optimal steady-state measure by $\rho^\star$ and the corresponding stage cost by $\mathcal{L}_0$. Without loss of generality, we can assume that $\mathcal{L}_0 = 0$ in order to have a well-posed value functional. Clearly, if this does not hold, one can shift the stage cost to achieve $\mathcal{L}_0 = 0$. Let us consider the following discounted infinite-horizon OCP:

$$V^\star[\rho_0] = \min_{\boldsymbol{\pi}} \sum_{k=0}^{\infty} \gamma^k \mathcal{L}[\rho_k^{\boldsymbol{\pi}}, \boldsymbol{\pi}] \tag{4a}$$

$$\text{s.t. } \rho_{k+1}^{\boldsymbol{\pi}} = \mathcal{T}_{\boldsymbol{\pi}} \rho_k^{\boldsymbol{\pi}}, \quad \rho_0^{\boldsymbol{\pi}} = \rho_0 \ , \tag{4b}$$

where $\gamma \in (0, 1)$ is the discount factor and $V^\star : \Xi \to \mathbb{R}$ is the optimal value functional. We denote the optimal policy by $\boldsymbol{\pi}^\star$, solution of (4). In the following, we make a standard assumption on the stage cost functional and $V^\star$.

Assumption 1. *1) We assume that $\mathcal{L}[\rho, \boldsymbol{\pi}]$ is bounded, $\forall \rho \in \Xi, \forall \boldsymbol{\pi} \in \mathcal{P}$*

2) There exists a non-empty set of measures, denoted by Ξ_0, such that for all $\rho_0 \in \Xi_0$, $V^\star[\rho_k^{\boldsymbol{\pi}^\star}]$ remains bounded, $\forall k$.

The optimal action-value functional $Q^\star$ and advantage functional $A^\star$ associated to (4) are defined as follows:

$$Q^\star[\rho, \boldsymbol{\pi}] := \mathcal{L}[\rho, \boldsymbol{\pi}] + \gamma V^\star[\mathcal{T}_{\boldsymbol{\pi}} \rho], \quad \forall \rho \in \Xi, \forall \boldsymbol{\pi} \in \mathcal{P} \, , \tag{5a}$$

$$A^\star[\rho, \boldsymbol{\pi}] := Q^\star[\rho, \boldsymbol{\pi}] - V^\star[\rho], \quad \forall \rho \in \Xi, \forall \boldsymbol{\pi} \in \mathcal{P} \, . \tag{5b}$$

Then from the Bellman equation, we have:

$$V^\star[\rho] = Q^\star[\rho, \boldsymbol{\pi}^\star] = \min_{\boldsymbol{\pi}} Q^\star[\rho, \boldsymbol{\pi}], \quad \forall \rho \in \Xi \tag{6}$$

One can verify the following , $\forall \rho \in \Xi$:

$$0 = A^\star[\rho, \boldsymbol{\pi}^\star] = \min_{\boldsymbol{\pi}} A^\star[\rho, \boldsymbol{\pi}], \;\; \boldsymbol{\pi}^\star \in \arg\min_{\boldsymbol{\pi}} A^\star[\rho, \boldsymbol{\pi}] \; . \tag{7}$$

We will use these results in Section 4. The next section presents the conditions on the MDP (1) such that the sequence of measures under optimal policy converges to the optimal steady-state measure in some sense.

3 Stability of MDPs

In this section, we will detail the stability of MDPs in the sense of probability measures. We extend the dissipativity theory to propose a Lyapunov functional establishing the MDP stability in the sense of $\lim_{k\to\infty} \rho_k^{\boldsymbol{\pi}^\star}$. In order to discuss this limit formally, we first define the following concept.

Definition 1. ***(Dissimilarity measure)*** *For any $\rho, \rho' \in \Xi$, we define $D(\rho||\rho')$ as a dissimilarity measure on measure space, that maps any two measures ρ and ρ' to the real non-negative numbers, and $D(\rho||\rho) = 0$, $\forall \rho$.*

One can show that the Kullback-Leibler divergence, the Wasserstein metric, and the total variation distance are Dissimilarity measures. Using the Dissimilarity measure concept, we can define D-stability of Markov Chains [9].

Definition 2. ***(D-stability)*** *The closed-loop Markov Chain $\xi(\boldsymbol{s}^+|\boldsymbol{s}, \boldsymbol{\pi}(\boldsymbol{s}))$ with policy $\boldsymbol{\pi}$ is D-stable with respect to the optimal steady probability measure $\rho^\star$ and dissimilarity measure D if, for any $\epsilon > 0$ there exists a $\delta(\epsilon) > 0$ and a $K \in \mathbb{I}_{\geq 0}$ such that $D(\rho_0\|\rho^\star) < \delta(\epsilon)$ implies $D(\rho_k^{\boldsymbol{\pi}}\|\rho^\star) < \epsilon$, $\forall k \geq K$. Moreover, if $\lim_{k\to\infty} D(\rho_k^{\boldsymbol{\pi}}\|\rho^\star) = 0$ holds almost everywhere, then the closed-loop Markov Chain is D-asymptotically stable.*

The concept of D-stability provides a framework to argue about $\lim_{k\to\infty} \rho_k^{\boldsymbol{\pi}^\star}$ in the sense of dissimilarity measures. The next lemma connects the functional stability to the existence of a Lyapunov functional V, satisfying proper conditions.

Lemma 1. *The closed-loop Markov Chain $\xi(\boldsymbol{s}^+|\boldsymbol{s}, \boldsymbol{\pi}^\star(\boldsymbol{s}))$ is D-asymptotically stable with respect to the optimal steady probability measure $\rho_\star$ and dissimilarity measure D, if there exists a Lyapunov functional $V : \Xi \to \mathbb{R}^{\geq 0}$, satisfying:*

$$\beta_0(D(\rho_0\|\rho_\star)) \leq V[\rho_0] \leq \beta_1(D(\rho_0\|\rho_\star)) \; , \tag{8a}$$

$$V[\mathcal{T}_{\boldsymbol{\pi}^\star}\rho_0] - V[\rho_0] \leq -\beta_2(D(\rho_0\|\rho_\star)) \; , \tag{8b}$$

for some $\beta_0, \beta_1, \beta_2 \in \mathcal{K}_\infty$.

Proof. The proof can be found in [9]. ■

In the following, we will connect the Lyapunov functional in Lemma 1 with the value functional under some conditions. The next definition develops the SDSD conditions for undiscounted MDPs, where the stage cost is a generic functional.

Definition 3. ***(Functional Strong Discounted Strict Dissipativity (FSDSD))*** *MDP* (1) *with functional stage cost $\mathcal{L}$ and discount factor γ is Functional Strong Discounted Strict Dissipative (FSDSD), If there exists a bounded "storage" functional λ such that $\lambda[\rho^\star] = 0$, satisfying:*

$$\mathcal{L}[\rho, \boldsymbol{\pi}] - \gamma\lambda[\mathcal{T}_{\boldsymbol{\pi}}\rho] + \lambda[\rho] \geq \alpha(D(\rho\|\rho_\star)) \ , \tag{9a}$$

$$\mathcal{L}[\rho, \boldsymbol{\pi}] - \lambda[\mathcal{T}_{\boldsymbol{\pi}}\rho] + \lambda[\rho] + (\gamma - 1)V^\star[\mathcal{T}_{\boldsymbol{\pi}}\rho] \geq \alpha(D(\rho\|\rho_\star)) \, . \tag{9b}$$

for some $\alpha(\cdot) \in \mathcal{K}_\infty$[1] and $\forall \rho \in \Xi, \forall \boldsymbol{\pi} \in \mathcal{P}$, where ρ is the probability measure of state $\boldsymbol{s}$ and $\rho^\star$ is the optimal steady measure.

Note that condition (9a) corresponds to the commonly discounted dissipativity condition [10], but is generalized to a functional space [9]. Condition (9b) has been introduced in [8] in a non-functional form to show the stability of deterministic nonlinear systems with discounted cost. For an undiscounted setting with $\gamma \to 1$, two conditions in (9) coincide, and correspond to the condition proposed in [6]. For an FSDSD problem, we define the *rotated functional stage cost* $\bar{\mathcal{L}} : \Xi \to \mathbb{R}$ as follows:

$$\bar{\mathcal{L}}[\rho, \boldsymbol{\pi}] = \mathcal{L}[\rho, \boldsymbol{\pi}] - \gamma\lambda[\mathcal{T}_{\boldsymbol{\pi}}\rho] + \lambda[\rho] \ . \tag{10}$$

Then if (9a) holds, we have:

$$\bar{\mathcal{L}}[\rho, \boldsymbol{\pi}] \geq \alpha(D(\rho\|\rho_\star)) \ . \tag{11}$$

Indeed, condition (9a) allows us to convert the original general stage cost $\mathcal{L}$ to the rotated stage cost $\bar{\mathcal{L}}$. For any measure ρ, the rotated stage cost functional $\bar{\mathcal{L}}$ is lower bounded by a $\mathcal{K}_\infty$ function applied on the selected dissimilarity measure, even if the original stage cost $\mathcal{L}$ has not such property. The next theorem relates the optimal value functional and optimal policy resulting from $\mathcal{L}$ to the optimal value functional and optimal policy resulting from $\bar{\mathcal{L}}$.

Theorem 1. *If MDP* (1) *is FSDSD, then the following discounted OCP:*

$$\bar{V}^\star[\rho_0] := \min_{\boldsymbol{\pi}} \ \sum_{k=0}^{\infty} \gamma^k \bar{\mathcal{L}}[\rho_k^{\boldsymbol{\pi}}, \boldsymbol{\pi}] \ , \tag{12a}$$

$$\text{s.t. } \rho_{k+1}^{\boldsymbol{\pi}} = \mathcal{T}_{\boldsymbol{\pi}}\rho_k^{\boldsymbol{\pi}}, \quad \rho_0^{\boldsymbol{\pi}} = \rho_0 \ , \tag{12b}$$

yields the same optimal policy $\boldsymbol{\pi}^\star$ as (4)*, $\forall \rho_0 \in \Xi_0$, and:*

$$\bar{V}^\star[\rho_0] = V^\star[\rho_0] + \lambda[\rho_0] \ . \tag{13}$$

[1] A function $\alpha : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ is said to belong to class $\mathcal{K}_\infty$, if α is continuous, strictly increasing, unbounded and $\alpha(0) = 0$.

Proof. For an FSDSD problem, $\bar{\mathcal{L}}$ exists. Substitution of (10) into the cost of (4) and using a telescopic sum argument, one observes that:

$$\sum_{k=0}^{\infty} \gamma^k \mathcal{L}[\rho_k^{\boldsymbol{\pi}}, \boldsymbol{\pi}] = \sum_{k=0}^{\infty} \gamma^k \left(\bar{\mathcal{L}}[\rho_k^{\boldsymbol{\pi}}, \boldsymbol{\pi}] + \gamma \lambda[\rho_{k+1}^{\boldsymbol{\pi}}] - \lambda[\rho_k^{\boldsymbol{\pi}}] \right) =$$

$$- \lambda[\rho_0] + \lim_{N \to \infty} \gamma^N \bar{\mathcal{L}}[\rho_N^{\boldsymbol{\pi}}, \boldsymbol{\pi}] + \sum_{k=0}^{\infty} \gamma^k \bar{\mathcal{L}}[\rho_k^{\boldsymbol{\pi}}, \boldsymbol{\pi}] = -\lambda[\rho_0] + \sum_{k=0}^{\infty} \gamma^k \bar{\mathcal{L}}[\rho_k^{\boldsymbol{\pi}}, \boldsymbol{\pi}] \ . \quad (14)$$

Note that under assumption 1, all terms in (14) remain bounded and $\lim_{N\to\infty} \gamma^N \bar{\mathcal{L}}[\rho_N^{\boldsymbol{\pi}}, \boldsymbol{\pi}] = 0$. Taking $\min_{\boldsymbol{\pi}}$ on both sides of (14) results in (13) and the optimal policy $\boldsymbol{\pi}^\star$ from (4), minimizing the right-hand side, minimizes the left-hand side as well. ■

Theorem 1 states that for an MDP that satisfies the FSDSD conditions, we can find an equivalent OCP that yields the same optimal policy and the value functional that is shifted by λ. In the next, we assume that, for any measure, ρ, the optimal value functional $\bar{V}^\star[\rho]$ is upper bounded by a $\mathcal{K}_\infty$ function applied on the selected dissimilarity measure. This will be useful in showing Lyapunov stability.

Assumption 2. *We assume following for some $\alpha_1(\cdot) \in \mathcal{K}_\infty$:*

$$\bar{V}^\star[\rho] \leq \alpha_1(D(\rho \| \rho_\star)), \qquad \forall \rho \in \Xi_0 \quad (15)$$

The next theorem states that for an FSDSD MDP, $\bar{V}^\star$, defined in (12), is a Lyapunov functional in order to prove the D-stability of the closed-loop Markov Chain $\xi(\boldsymbol{s}^+|\boldsymbol{s}, \boldsymbol{\pi}^\star(\boldsymbol{s}))$ with respect to the optimal steady measure $\rho^\star$.

Theorem 2. *Under assumption 1, if the MDP with transition probability $\xi(\boldsymbol{s}^+|\boldsymbol{s}, \boldsymbol{a})$, stage cost $\mathcal{L}$ and discount factor γ is functional SDSD, then $\bar{V}^\star$, defined in* (12)*, is a Lyapunov functional for the closed-loop Markov Chain $\xi(\boldsymbol{s}^+|\boldsymbol{s}, \boldsymbol{\pi}^\star(\boldsymbol{s}))$ with optimal policy $\boldsymbol{\pi}^\star$, solution of* (4)*.*

Proof. Condition (15) directly implies the upper bound of (8a). Using (11), we have:

$$\alpha(D(\rho \| \rho_\star)) \leq \bar{V}^\star[\rho] \ , \quad (16)$$

which results in the lower bound of (8a). From the Bellman equation for OCP (4), we have:

$$\mathcal{L}[\rho_k^{\boldsymbol{\pi}^\star}, \boldsymbol{\pi}^\star] - V^\star[\rho_k^{\boldsymbol{\pi}^\star}] + \gamma V^\star[\rho_{k+1}^{\boldsymbol{\pi}^\star}] = 0 \ , \quad (17)$$

Rearranging (9b) and subtracting $V^\star[\rho_k^{\boldsymbol{\pi}^\star}]$ from both sides yields:

$$V^\star[\rho_{k+1}^{\boldsymbol{\pi}^\star}] + \lambda[\rho_{k+1}^{\boldsymbol{\pi}^\star}] - V^\star[\rho_k^{\boldsymbol{\pi}^\star}] - \lambda[\rho_k^{\boldsymbol{\pi}^\star}] \leq$$

$$- \alpha(D(\rho_k^{\boldsymbol{\pi}^\star} \| \rho_\star)) + \mathcal{L}[\rho_k^{\boldsymbol{\pi}^\star}, \boldsymbol{\pi}^\star] - V^\star[\rho_k^{\boldsymbol{\pi}^\star}] + \gamma V^\star[\rho_{k+1}^{\boldsymbol{\pi}^\star}] \overset{(17)}{=} -\alpha(D(\rho_k^{\boldsymbol{\pi}^\star} \| \rho_\star)) \ , \quad (18)$$

where we replace ρ, and $\mathcal{T}_{\boldsymbol{\pi}}\rho$ in (9b) by $\rho_k^{\boldsymbol{\pi}^\star}$, $\boldsymbol{\pi}^\star$ and $\rho_{k+1}^{\boldsymbol{\pi}^\star}$, respectively, and we have used (17) in the last equality. Then from (13) and (18), we have:

$$\bar{V}^\star[\rho_{k+1}^{\boldsymbol{\pi}^\star}] - \bar{V}^\star[\rho_k^{\boldsymbol{\pi}^\star}] = V^\star[\rho_{k+1}^{\boldsymbol{\pi}^\star}] + \lambda[\rho_{k+1}^{\boldsymbol{\pi}^\star}] - V^\star[\rho_k^{\boldsymbol{\pi}^\star}] - \lambda[\rho_k^{\boldsymbol{\pi}^\star}] \leq -\alpha(D(\rho_k^{\boldsymbol{\pi}^\star}\|\rho_\star)) \ ,$$

which concludes (8b). Then $\bar{V}^\star$ satisfies the conditions of Lemma 1 and the closed-loop Markov Chain $\xi(\boldsymbol{s}^+|\boldsymbol{s}, \boldsymbol{\pi}^\star(\boldsymbol{s}))$ is D-asymptotically stable with respect to the optimal steady probability measure $\rho_\star$ and dissimilarity measure D. ■

Theorem 2 states the conditions that imply a D-asymptotically stabilizing policy for the FSDSD MDPs. However, finding the optimal policy under dissipativity conditions of theorem 2 and the storage functional that satisfies (9) are very difficult. In the next section, we address this problem by using a parameterized finite-horizon OCP scheme.

4 Stabilizing Functional Approximator

Reinforcement Learning (RL) provides powerful tools to solve MDPs in practice. For instance, Q-learning is based on capturing the optimal action-value function of a given MDP, from which an optimal policy can be extracted. In this method, a parameterized action-value function is provided, and Q-learning attempts to find the optimal parameters that result in the best estimation of the optimal action-value. Deep Neural Network (DNN) is a common choice to provide a generic parameterization [11]. However, formal analysis of the stability properties of closed-loop systems is very challenging for DNNs-based function approximators. Therefore, using a more structured approximator such as the MPC scheme can be beneficial. The idea of using the function approximator based on a finite-horizon OCP has been introduced and justified in [12], where an EMPC was used as an approximator for RL algorithms. In fact, it has been shown that modifying stage cost and terminal cost in a parameterized MPC can capture the optimal value functions of MDPs even if an inaccurate model is used in the MPC scheme [12]. Moreover, this approximator has great capability to satisfy system constraints and safety [13]. Recent research have developed further in using such approximators in the RL context [14].

This section extends this parameterization to the functional space, where the arguments are on the measure space underlying the MDP. We use an OCP-based approximator for the optimal action-value functional to capture valid storage functional and verify the FSDSD conditions. The next theorem expresses that an undiscounted finite-horizon OCP is able to capture the optimal value functionals and policy of (4). Note that using the undiscounted OCP will be key to establishing stabilizing approximator results.

Theorem 3. *Under assumption 1, there exists a terminal cost functional $\hat{T} : \Xi \to \mathbb{R}$ and a stage cost functional $\hat{L} : \Xi \times \mathcal{P} \to \mathbb{R}$ such that the following undiscounted finite-horizon*

OCP:

$$\hat{V}^\star[\rho] = \min_{\boldsymbol{\pi}} \hat{V}^{\boldsymbol{\pi}}[\rho] := \hat{T}[\rho_N^{\boldsymbol{\pi}}] + \sum_{k=0}^{N-1} \hat{L}[\rho_k^{\boldsymbol{\pi}}, \boldsymbol{\pi}], \tag{19a}$$

$$\text{s.t. } \rho_{k+1}^{\boldsymbol{\pi}} = \mathcal{T}_{\boldsymbol{\pi}} \rho_k^{\boldsymbol{\pi}}, \quad \rho_0^{\boldsymbol{\pi}} = \rho \ , \tag{19b}$$

for all $\rho \in \Xi_0$, results in the following:

1. $\hat{\boldsymbol{\pi}}^\star = \boldsymbol{\pi}^\star$,
2. $\hat{V}^\star[\rho] = V^\star[\rho]$,
3. $\hat{Q}^\star[\rho, \boldsymbol{\pi}] = Q^\star[\rho, \boldsymbol{\pi}]$,

where $\hat{\boldsymbol{\pi}}^\star$ is the optimal policy resulting from (19) *and:*

$$\hat{Q}^\star[\rho, \boldsymbol{\pi}] := \hat{L}[\rho, \boldsymbol{\pi}] + \hat{V}^\star[\mathcal{T}_{\boldsymbol{\pi}} \rho] \ . \tag{20}$$

Proof. We select the terminal cost functional $\hat{T}$ and the stage cost functional $\hat{L}$ as follows:

$$\hat{T}[\rho] = V^\star[\rho] \ , \tag{21a}$$

$$\hat{L}[\rho, \boldsymbol{\pi}] = Q^\star[\rho, \boldsymbol{\pi}] - V^\star[\mathcal{T}_{\boldsymbol{\pi}} \rho] \ . \tag{21b}$$

Under assumption 1, the terminal cost and stage costs have finite values on Ξ_0. Substitution of (21) into (19) and using telescopic sum, we have:

$$\begin{aligned} \hat{V}^{\boldsymbol{\pi}}[\rho] &= \hat{T}[\rho_N^{\boldsymbol{\pi}}] + \sum_{k=0}^{N-1} \hat{L}[\rho_k^{\boldsymbol{\pi}}, \boldsymbol{\pi}] = V^\star[\rho_N^{\boldsymbol{\pi}}] + \sum_{k=0}^{N-1} Q^\star[\rho_k^{\boldsymbol{\pi}}, \boldsymbol{\pi}] - V^\star[\rho_{k+1}^{\boldsymbol{\pi}}] \\ &= Q^\star[\rho, \boldsymbol{\pi}] + \sum_{k=1}^{N-1} Q^\star[\rho_k^{\boldsymbol{\pi}}, \boldsymbol{\pi}] - V^\star[\rho_k^{\boldsymbol{\pi}}] = Q^\star[\rho, \boldsymbol{\pi}] + \sum_{k=1}^{N-1} A^\star[\rho_k^{\boldsymbol{\pi}}, \boldsymbol{\pi}] \ . \end{aligned} \tag{22}$$

From (7), we know that $\boldsymbol{\pi}^\star$ minimizes $A^\star[\rho_k^{\boldsymbol{\pi}}, \boldsymbol{\pi}]$ and $Q^\star[\rho, \boldsymbol{\pi}]$, hence it minimizes $\hat{V}^{\boldsymbol{\pi}}[\rho]$, i.e.,:

$$\hat{\boldsymbol{\pi}}^\star = \arg\min_{\boldsymbol{\pi}} \hat{V}^{\boldsymbol{\pi}}[\rho] = \arg\min_{\boldsymbol{\pi}} Q^\star[\rho, \boldsymbol{\pi}] + \sum_{k=1}^{N-1} A^\star[\rho_k^{\boldsymbol{\pi}}, \boldsymbol{\pi}] = \boldsymbol{\pi}^\star$$

and it yields (1). Then substitution of the optimal policy $\boldsymbol{\pi}^\star$ in the cost function of (19) reads:

$$\hat{V}^\star[\rho] = \hat{V}^{\boldsymbol{\pi}^\star}[\rho] = Q^\star[\rho, \boldsymbol{\pi}^\star] + \sum_{k=1}^{N-1} \underbrace{A^\star[\rho_k^{\boldsymbol{\pi}^\star}, \boldsymbol{\pi}^\star]}_{\overset{(7)}{=} 0} \overset{(6)}{=} V^\star[\rho] \, , \tag{23}$$

which results in (2). Moreover, from (20) and (21b) we have:

$$\hat{Q}^\star[\rho, \boldsymbol{\pi}]) = \hat{L}[\rho^{\boldsymbol{\pi}}, \boldsymbol{\pi}] + \hat{V}^\star[\mathcal{T}_{\boldsymbol{\pi}} \rho] = Q^\star[\rho, \boldsymbol{\pi}] - V^\star[\mathcal{T}_{\boldsymbol{\pi}} \rho] + \hat{V}^\star[\mathcal{T}_{\boldsymbol{\pi}} \rho] \overset{(23)}{=} Q^\star[\rho, \boldsymbol{\pi}]$$

which yields (3). ■

Theorem 3 states that an undiscounted finite-horizon OCP can estimate the value functional, action-value functional, and optimal policy of a discounted infinite horizon OCP. Then this allows us to use a function approximator based on the undiscounted finite-horizon OCP for the discounted MDP. Similar results can be found in [15] for value functions of classic MDPs. More specifically, let us consider the following finite-horizon undiscounted OCP as an approximator for the value functional, parameterized by $\boldsymbol{\theta}$:

$$V_{\boldsymbol{\theta}}[\rho_0] = \min_{\boldsymbol{\pi}} \; -\lambda_{\boldsymbol{\theta}}[\rho_0] + T_{\boldsymbol{\theta}}[\rho_N^{\boldsymbol{\pi}}] + \sum_{k=0}^{N-1} \mathcal{L}_{\boldsymbol{\theta}}[\rho_k^{\boldsymbol{\pi}}, \boldsymbol{\pi}] \tag{24a}$$

$$\text{s.t. } \rho_{k+1}^{\boldsymbol{\pi}} = \mathcal{T}_{\boldsymbol{\pi}} \rho_k^{\boldsymbol{\pi}}, \quad \rho_0^{\boldsymbol{\pi}} = \rho_0 \ , \tag{24b}$$

where $V_{\boldsymbol{\theta}}$, $\lambda_{\boldsymbol{\theta}}$, $T_{\boldsymbol{\theta}}$ and $\mathcal{L}_{\boldsymbol{\theta}}$ are the parameterized value functional, storage functional, terminal cost and stage cost, respectively. Note that the term $-\lambda_{\boldsymbol{\theta}}[\rho_0]$ only depends on the first measure sequence and does not affect on the optimal policy resulting from (24). The term $-\lambda_{\boldsymbol{\theta}}[\rho_0]$ is added to the cost to have consistency with the EMPC context [4]. We denote the parameterized policy by $\boldsymbol{\pi}_{\boldsymbol{\theta}}$, solution of (24).

The parameterized action-value functional associated with (24) is defined as follows:

$$Q_{\boldsymbol{\theta}}[\rho, \boldsymbol{\pi}] := -\lambda_{\boldsymbol{\theta}}[\rho] + \mathcal{L}_{\boldsymbol{\theta}}[\rho, \boldsymbol{\pi}] + \Psi_{\boldsymbol{\theta}}[\mathcal{T}_{\boldsymbol{\pi}} \rho] \tag{25}$$

where

$$\Psi_{\boldsymbol{\theta}}[\rho] := \lambda_{\boldsymbol{\theta}}[\rho] + V_{\boldsymbol{\theta}}[\rho] \tag{26}$$

In fact, one can verify that the action-value functional $Q_{\boldsymbol{\theta}}[\rho, \boldsymbol{\pi}]$, value functional $V_{\boldsymbol{\theta}}[\rho]$ and policy $\boldsymbol{\pi}_{\boldsymbol{\theta}}$ satisfy the fundamental Bellman equations. We next make a standard assumption on the terminal cost functional $T_{\boldsymbol{\theta}}$ and the parameterization of OCP (24).

Assumption 3. *We assume that the terminal cost functional $T_{\boldsymbol{\theta}}$ satisfies $T_{\boldsymbol{\theta}}[\rho] \geq 0$, $\forall \rho \in \Xi$.*

Assumption 4. *We assume that the parameterization of* (24) *is rich enough to capture the optimal action-value functional, i.e., there exists an optimal parameters vector $\boldsymbol{\theta}^\star$ such that:*

$$Q_{\boldsymbol{\theta}^\star}[\rho, \boldsymbol{\pi}] = Q^\star[\rho, \boldsymbol{\pi}], \tag{27a}$$

$$V_{\boldsymbol{\theta}^\star}[\rho] = V^\star[\rho] \ , \tag{27b}$$

Assumption 4 requires a universal approximator in the functional space. Note that this assumption may not hold in practice. In the next section, we detail Q-learning as a practical way to approach this assumption asymptotically.

Theorem 4. *Under assumptions 1, 3 and 4, $\lambda_{\boldsymbol{\theta}^\star}[\rho]$ satisfies* (9)*, if the following holds for some $\alpha_0(\cdot) \in \mathcal{K}_\infty$:*

$$\mathcal{L}_{\boldsymbol{\theta}}[\rho, \boldsymbol{\pi}] \geq \alpha_0(D(\rho \| \rho_\star)), \qquad \forall \rho \in \Xi, \, \forall \boldsymbol{\pi} \in \mathcal{P} \tag{28}$$

Proof. From assumption 4, we have:

$$\begin{aligned}\mathcal{L}[\rho,\boldsymbol{\pi}]+\gamma V^{\star}[\mathcal{T}_{\boldsymbol{\pi}}\rho] &\overset{(5a)}{=} Q^{\star}[\rho,\boldsymbol{\pi}] \overset{(27a)}{=} Q_{\boldsymbol{\theta}^{\star}}[\rho,\boldsymbol{\pi}] \overset{(25)}{=} -\lambda_{\boldsymbol{\theta}^{\star}}[\rho]+\mathcal{L}_{\boldsymbol{\theta}^{\star}}[\rho,\boldsymbol{\pi}]+\Psi_{\boldsymbol{\theta}^{\star}}[\mathcal{T}_{\boldsymbol{\pi}}\rho]\\ &\overset{(26)}{=} -\lambda_{\boldsymbol{\theta}^{\star}}[\rho]+\mathcal{L}_{\boldsymbol{\theta}^{\star}}[\rho,\boldsymbol{\pi}]+\lambda_{\boldsymbol{\theta}^{\star}}[\mathcal{T}_{\boldsymbol{\pi}}\rho]+V_{\boldsymbol{\theta}^{\star}}[\mathcal{T}_{\boldsymbol{\pi}}\rho]\\ &\overset{(28)}{\geq} -\lambda_{\boldsymbol{\theta}^{\star}}[\rho]+\alpha_0(D(\rho_0\|\rho_{\star}))+\lambda_{\boldsymbol{\theta}^{\star}}[\mathcal{T}_{\boldsymbol{\pi}}\rho]+V^{\star}[\mathcal{T}_{\boldsymbol{\pi}}\rho]\,. \end{aligned} \tag{29}$$

Rearranging (29) results in (9b). Moreover, from (26) we can write $\Psi_{\boldsymbol{\theta}}$ as the following OCP:

$$\Psi_{\boldsymbol{\theta}}[\rho_0]=\min_{\boldsymbol{\pi}}\ T_{\boldsymbol{\theta}}[\rho_N^{\boldsymbol{\pi}}]+\sum_{k=0}^{N-1}\mathcal{L}_{\boldsymbol{\theta}}[\rho_k^{\boldsymbol{\pi}},\boldsymbol{\pi}] \tag{30a}$$

$$\text{s.t. } \rho_{k+1}^{\boldsymbol{\pi}}=\mathcal{T}_{\boldsymbol{\pi}}\rho_k^{\boldsymbol{\pi}},\qquad \rho_0^{\boldsymbol{\pi}}=\rho_0\ , \tag{30b}$$

Then using (28) and assumption 3, the cost of (30) is non-negative and we have $0\leq\Psi_{\boldsymbol{\theta}}[\rho]$, $\forall\rho$. Then:

$$0\leq\Psi_{\boldsymbol{\theta}^{\star}}[\mathcal{T}_{\boldsymbol{\pi}}\rho]\overset{(26)}{=}\lambda_{\boldsymbol{\theta}^{\star}}[\mathcal{T}_{\boldsymbol{\pi}}\rho]+V_{\boldsymbol{\theta}^{\star}}[\mathcal{T}_{\boldsymbol{\pi}}\rho]\overset{(27b)}{=}\lambda_{\boldsymbol{\theta}^{\star}}[\mathcal{T}_{\boldsymbol{\pi}}\rho]+V^{\star}[\mathcal{T}_{\boldsymbol{\pi}}\rho]\ . \tag{31}$$

By rearranging and multiplying both sides of (31) by the positive factor $1-\gamma$:

$$-(1-\gamma)V^{\star}[\mathcal{T}_{\boldsymbol{\pi}}\rho]\leq(1-\gamma)\lambda_{\boldsymbol{\theta}^{\star}}[\mathcal{T}_{\boldsymbol{\pi}}\rho]\ , \tag{32}$$

or equivalently

$$(\gamma-1)V^{\star}[\mathcal{T}_{\boldsymbol{\pi}}\rho]-\lambda_{\boldsymbol{\theta}^{\star}}[\mathcal{T}_{\boldsymbol{\pi}}\rho]\leq-\gamma\lambda_{\boldsymbol{\theta}^{\star}}[\mathcal{T}_{\boldsymbol{\pi}}\rho]\ . \tag{33}$$

By adding $\mathcal{L}[\rho,\boldsymbol{a}]+\lambda_{\boldsymbol{\theta}^{\star}}[\rho]$ to both sides of (33), we have:

$$\begin{aligned}\mathcal{L}[\rho,\boldsymbol{\pi}]+\lambda_{\boldsymbol{\theta}^{\star}}[\rho]-\gamma\lambda_{\boldsymbol{\theta}^{\star}}[\mathcal{T}_{\boldsymbol{\pi}}\rho]\geq&\mathcal{L}[\rho,\boldsymbol{\pi}]+\lambda_{\boldsymbol{\theta}^{\star}}[\rho]-\lambda_{\boldsymbol{\theta}^{\star}}[\mathcal{T}_{\boldsymbol{\pi}}\rho]+\\ &(\gamma-1)V^{\star}[\mathcal{T}_{\boldsymbol{\pi}}\rho]\overset{(9b)}{\geq}\alpha_0(D(\rho\|\rho_{\star}))\ , \end{aligned} \tag{34}$$

and it results in (9a). ■

Assumption 4 is valid only for the FSDSD problems. In fact, for a non-FSDSD problem it is not possible to find $\boldsymbol{\theta}^{\star}$ that satisfies conditions of assumption 4. This approach enforces D-stability conditions for a given MDP, and if it is not stabilizable (non-FSDSD), then assumption 4 is invalid. Therefore theorem 4 implicitly assumes that the given problem is FSDSD and states that using undiscounted finite-horizon approximator (24) yields a valid storage functional that satisfies FSDSD conditions (9). The stage cost condition (28) can be satisfied using constrained steps in the learning algorithm or providing a positive functional by construction. The details of these methods for deterministic systems can be found in [16]. However, a detailed discussion on functional space is out of our scope. In general, finding $\boldsymbol{\theta}^{\star}$ that satisfies the conditions of assumption 4 is very difficult. However, Q-learning is a practical way to fulfill assumption 4. Q-learning uses a Least-Square (LS) optimization and approach assumption 4 asymptotically for a large number of data. Next section details this approach.

5 Practical Implementation

In this section, we focus on the classic MDPs with stage cost in the form of (3) and a given deterministic initial state, i.e., $\rho_0 = \delta_{\boldsymbol{s}_0}(\cdot)$, where $\delta_{\boldsymbol{s}_0}(\cdot)$ is the Dirac measure centered on the fixed point $\boldsymbol{s}_0$. This assumption is appropriate for fully observable MDPs since the current state is deterministic and available. We ought to stress here that we are still using functional stage cost functional as an important concept in the current work. Partially Observable MDPs (POMDPs) are the class of MDPs that the current state is estimated based on historical data of the system. Recently, Moving Horizon Estimation has been used in order to tackle POMDPs in combination with RL and MPC [17]. Note that all the results in the previous sections are valid when the current state distribution is a Dirac measure. Using $\rho_0 = \delta_{\boldsymbol{s}_0}(\cdot)$, (4) reads:

$$\begin{aligned} v^\star(\boldsymbol{s}_0) :=& V^\star[\delta_{\boldsymbol{s}_0}(\cdot)] = \min_{\boldsymbol{\pi}} \sum_{k=0}^{\infty} \gamma^k \mathbb{E}_{\boldsymbol{s}\sim\rho_k^{\boldsymbol{\pi}}} \left[\ell(\boldsymbol{s}, \boldsymbol{\pi}(\boldsymbol{s}))\right] \\ & \text{s.t. } \rho_{k+1}^{\boldsymbol{\pi}} = \mathcal{T}_{\boldsymbol{\pi}} \rho_k^{\boldsymbol{\pi}},\ \rho_0 = \delta_{\boldsymbol{s}_0}(\cdot) \end{aligned} \tag{35}$$

where $v^\star : \mathcal{X} \to \mathbb{R}$ is the classic optimal value function. The classic Bellman equation reads:

$$v^\star(\boldsymbol{s}_0) = \min_{\boldsymbol{\pi}} \ell(\boldsymbol{s}_0, \boldsymbol{\pi}(\boldsymbol{s}_0)) + \gamma \mathbb{E}_{\boldsymbol{s}_1\sim\rho_1^{\boldsymbol{\pi}}} \left[v^\star(\boldsymbol{s}_1)\right] \tag{36}$$

where $\rho_1^{\boldsymbol{\pi}} = \xi(\cdot|\boldsymbol{s}_0, \boldsymbol{\pi}(\boldsymbol{s}_0))$. Moreover, from the Bellman equation associate to (35), we have:

$$v^\star(\boldsymbol{s}_0) = \min_{\boldsymbol{\pi}} \ell(\boldsymbol{s}_0, \boldsymbol{\pi}(\boldsymbol{s}_0)) + \gamma V^\star[\rho_1^{\boldsymbol{\pi}}] \tag{37}$$

Comparing (36) and (37), we have the following relation between the classic optimal value function $v^\star(\boldsymbol{s})$ and the optimal value functional $V^\star[\rho]$:

$$V^\star[\rho_1^{\boldsymbol{\pi}}] = \mathbb{E}_{\boldsymbol{s}_1\sim\rho_1^{\boldsymbol{\pi}}} \left[v^\star(\boldsymbol{s}_1)\right] \tag{38}$$

The classic optimal action-value function $q^\star : \mathcal{X} \times \mathcal{U} \to \mathbb{R}$ can be defined as follows:

$$q^\star(\boldsymbol{s}, \boldsymbol{\pi}(\boldsymbol{s})) := \ell(\boldsymbol{s}, \boldsymbol{\pi}(\boldsymbol{s})) + \gamma \mathbb{E}_{\boldsymbol{s}^+\sim\rho_1} \left[v^\star(\boldsymbol{s}^+)\right] \tag{39}$$

where $\rho_1 = \xi(\cdot|\boldsymbol{s}, \boldsymbol{\pi}(\boldsymbol{s}))$. Substituting $\rho(\cdot) = \delta_s(\cdot)$ in (5a), we have:

$$\begin{aligned} & Q^\star[\delta_{\boldsymbol{s}}(\cdot), \boldsymbol{\pi}] = \ell(\boldsymbol{s}, \boldsymbol{\pi}(\boldsymbol{s})) + \gamma V^\star[\xi(\cdot|\boldsymbol{s}, \boldsymbol{\pi}(\boldsymbol{s}))] \\ & \overset{(38)}{=} \ell(\boldsymbol{s}, \boldsymbol{\pi}(\boldsymbol{s})) + \gamma \mathbb{E}_{\boldsymbol{s}^+\sim\xi(\cdot|\boldsymbol{s}, \boldsymbol{\pi}(\boldsymbol{s}))} \left[v^\star(\boldsymbol{s}^+)\right] \overset{(39)}{=} q^\star(\boldsymbol{s}, \boldsymbol{\pi}(\boldsymbol{s})) \end{aligned} \tag{40}$$

This equation shows that the optimal action-value function of a classic MDPs $q^\star(\boldsymbol{s}, \boldsymbol{\pi}(\boldsymbol{s}))$ can be seen as a function action-value function $Q^\star[\rho, \boldsymbol{\pi}]$ where the argument of measure ρ is a Dirac measure $\delta_{\boldsymbol{s}}(\cdot)$. Similarly, for the parametric action-value functional one can show that:

$$Q_{\boldsymbol{\theta}^\star}[\delta_{\boldsymbol{s}}(\cdot), \boldsymbol{\pi}] = q_{\boldsymbol{\theta}}(\boldsymbol{s}, \boldsymbol{\pi}(\boldsymbol{s})); \tag{41}$$

where $q_{\boldsymbol{\theta}}(\boldsymbol{s}, \boldsymbol{\pi}(\boldsymbol{s}))$ is a classic parameterized action-value function. Moreover, we denote the parameterized value function by $v_{\boldsymbol{\theta}}$. For $\rho_0 = \delta_{\boldsymbol{s}}(\cdot)$, assumption 4 reads:

$$Q_{\boldsymbol{\theta}^\star}[\delta_{\boldsymbol{s}}(\cdot), \boldsymbol{\pi}] = Q^\star[\delta_{\boldsymbol{s}}(\cdot), \boldsymbol{\pi}] \tag{42}$$

Then using (40) and (41), (42) reads:

$$q_{\boldsymbol{\theta}^\star}(\boldsymbol{s}, \boldsymbol{\pi}(\boldsymbol{s})) = q^\star(\boldsymbol{s}, \boldsymbol{\pi}(\boldsymbol{s})) \tag{43}$$

Fortunately, condition (43) is a well-known problem in RL context, especially in the value-based algorithms. Q-learning is a common method to approach (43) in practice. More specifically, Q-learning uses the following LS optimization problem:

$$\min_{\boldsymbol{\theta}} \mathbb{E}\left[\left(q_{\boldsymbol{\theta}}(\boldsymbol{s}, \boldsymbol{\pi}(\boldsymbol{s})) - q^\star(\boldsymbol{s}, \boldsymbol{\pi}(\boldsymbol{s}))\right)^2\right], \tag{44}$$

In fact, LS (44) tries to find the optimal parameters vector $\boldsymbol{\theta}^\star$ that has the best approximation of the exact optimal action-value function $q^\star$. A richer parameterization and a larger number of data increase the accuracy of the method.

6 Conclusion

This paper provided a framework to analyze the functional stability of the closed-loop Markov Chains under the optimal policy resulting from minimizing the expected value of the discounted sum of stage costs for the associated MDPs. We used the dissipativity theory of EMPC in order to characterize the stability properties of discounted MDPs that require a storage functional satisfying FSDSD conditions. We showed that using a function approximator based on a finite-horizon OCP allows us to obtain valid storage functional under some conditions. We focused on the Dirac measure to use the theorems in practice and addressed the use of Q-learning as a powerful RL technique to update the parameters. Considering an inaccurate model in the function approximator and providing more theoretical tools for learning and/or computing of the storage functional in a numerical example can be the direction of future works.

References

[1] Martin L Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

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

[3] Sean P Meyn and Richard L Tweedie. *Markov chains and stochastic stability*. Springer Science & Business Media, 2012.

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

[5] James B Rawlings and Rishi Amrit. "Optimizing process economic performance using model predictive control". In: *Nonlinear model predictive control*. Springer, 2009, pp. 119–138.

[6] Rishi Amrit, James B Rawlings, and David Angeli. “Economic optimization using model predictive control with a terminal cost”. In: *Annual Reviews in Control* 35.2 (2011), pp. 178–186.

[7] Romain Postoyan et al. “Stability analysis of discrete-time infinite-horizon optimal control with discounted cost”. In: *IEEE Transactions on Automatic Control* 62.6 (2016), pp. 2736–2749.

[8] Mario Zanon and Sébastien Gros. “A new dissipativity condition for asymptotic stability of discounted economic MPC”. In: *Automatica* 141 (2022), p. 110287.

[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] Lars Grüne, Christopher M Kellett, and Steven R Weller. “On a discounted notion of strict dissipativity”. In: *IFAC-PapersOnLine* 49.18 (2016), pp. 247–252.

[11] Volodymyr Mnih et al. “Human-level control through deep reinforcement learning”. In: *nature* 518.7540 (2015), pp. 529–533.

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

[13] Mario Zanon and Sébastien Gros. “Safe reinforcement learning using robust MPC”. In: *IEEE Transactions on Automatic Control* 66.8 (2020), pp. 3638–3652.

[14] 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.

[15] Mario Zanon, Sébastien Gros, and Michele Palladino. “Stability-constrained Markov decision processes using MPC”. In: *Automatica* 143 (2022), p. 110399.

[16] Arash Bahari Kordabad and Sebastien Gros. “Verification of Dissipativity and Evaluation of Storage Function in Economic Nonlinear MPC using Q-Learning”. In: *IFAC-PapersOnLine* 54.6 (2021). 7th IFAC Conference on Nonlinear Model Predictive Control NMPC 2021, pp. 308–313.

[17] Hossein Nejatbakhsh Esfahani, Arash Bahari Kordabad, and Sébastien Gros. “Reinforcement Learning based on MPC/MHE for Unmodeled and Partially Observable Dynamics”. In: *2021 American Control Conference (ACC)*. 2021, pp. 2121–2126.

G Q-learning of the storage function in Economic Nonlinear Model Predictive Control

Postprint of [79] **Arash Bahari Kordabad** and Sebastien Gros. "Q-learning of the storage function in Economic Nonlinear Model Predictive Control". In: *Engineering Applications of Artificial Intelligence* 116 (2022), p. 105343. DOI: 0.1016/j.engappai.2022.105343

Q-Learning of the Storage Function in Economic Nonlinear Model Predictive Control

Arash Bahari Kordabad[1] and Sebastien Gros[1]

[1]Department of Engineering Cybernetics, Norwegian University of Science and Technology (NTNU), Trondheim, Norway.

Abstract: The closed-loop stability of an optimal policy provided by an Economic Nonlinear Model Predictive Control (ENMPC) scheme requires the existence of a storage function satisfying dissipativity conditions. Unfortunately, finding such a storage function is difficult in general. In contrast, tracking NMPC scheme uses a stage cost that is lower-bounded by a class-$\mathcal{K}_\infty$ function and the closed-loop stability is fairly straightforward to establish. Under the dissipativity conditions, ENMPC has an equivalent tracking MPC that delivers the same optimal policy. In this paper, we use this idea and parameterize the stage cost and terminal cost of a tracking MPC with an additional parameterized storage function. We show that, if the parameterization of the tracking MPC scheme is rich enough to capture the exact optimal action-value function of the ENMPC scheme, then the parameterized storage function for the optimal parameters satisfies the dissipativity conditions for both discounted and undiscounted ENMPC schemes. In fact, we show that these conditions are met for dissipative problems. We propose to use Q-learning as a practical way of adjusting the parameters of the tracking MPC. Different numerical examples are provided to illustrate the efficiency of the proposed method, including LQR, non-dissipative, non-polynomial and a nonlinear chemical case studies. For instance, in the provided non-polynomial case study, the learning method can improve the storage function estimation by about 60% and 99.5% after 10 and 50 learning steps, respectively, compared with the Sum-of-Square method.

Keywords: Economic Nonlinear Model Predictive Control, Reinforcement Learning, Q-learning, Dissipativity, Storage Function

1 Introduction

Tracking Nonlinear Model Predictive Control (NMPC) refers to NMPC schemes that are formulated with a cost function penalizing the deviations of the current state and input from a desired steady-state reference [1]. More formally, the stage cost of a tracking NMPC scheme is lower-bounded by a class-$\mathcal{K}_\infty$ function, usually selected as convex, often quadratic. In contrast, the cost function used in Economic NMPC

(ENMPC) does not satisfy such requirement [2–5]. The cost function used in ENMPC is typically an economic cost, often corresponding to the energy, the time, or the financial cost of running a system [6]. Thus an ENMPC employs a cost function that is not necessarily lower-bounded by a class-$\mathcal{K}_\infty$ function with respect to any setpoint.

The stability of undiscounted tracking MPC schemes is fairly straightforward to establish [7], as the optimal value function can typically be used as a Lyapunov function for the closed-loop system. Indeed, under mild controllability assumptions and additional conditions on the terminal cost and constraints used in the MPC scheme, the system is asymptotically stable in closed-loop with a tracking MPC scheme [8]. However, these properties do not necessarily hold when an economic stage cost is used as an objective function [9] and an optimal economic policy may not lead to the closed-loop stability of the system with respect to the optimal steady-state point [3, 10].

In [11], a Lyapunov function was proposed to establish asymptotic stability of the closed-loop nonlinear systems satisfying a strong duality assumption for ENMPC. Then a generalization of this result has been proposed to address generic systems, for which stability requires that dissipation inequality is satisfied [3]. Recently, a stable Sontag controller and corresponding region have been designed to ensure stability and feasibility of EMPC in [12] for the boiler-turbine system. For a given ENMPC, if the problem is dissipative, then there exists a corresponding tracking MPC that yields the same policy as the ENMPC scheme. This observation allows one to use the well-established stability conditions on the equivalent tracking MPC to discuss the stability of the original ENMPC scheme and form that equivalent tracking controller. Dissipativity is then a fundamental concept in ENMPC to argue about the closed-loop stability of the resulting scheme [13]. Dissipativity was first discussed in [14, 15]. It is shown in [16] that strict dissipativity with the addition of a suitable controllability assumption yields the turnpike property in optimal control [17].

In order to establish dissipativity, the existence of a storage function satisfying the dissipation inequality is required [18]. Finding the storage function for a given problem can be very demanding for nonlinear dynamics and non-quadratic stage costs [19]. Linear Matrix Inequality (LMI) techniques are used in [20] to compute the storage function for linear systems with a quadratic stage cost. A similar method is used in [21] to verify the dissipativity properties based on noisy data for linear systems. Furthermore, the Sum-of-Squares (SOS) method is used in [19] for polynomials dynamics and stage cost. In [22], conditions have been provided to establish the local existence of the storage function, based on solving a Semi-Definite Program (SDP).

Recently, there has been an increasing interest in using data from the system trajectories to verify dissipativity. This approach has been investigated for linear systems [23] and certain classes of nonlinear systems [24]. Some conditions in [25] have been provided to establish the dissipativity based on observed trajectories for linear systems. Our contribution is to use Reinforcement Learning (RL) techniques to capture a valid storage function for general nonlinear systems.

RL offers powerful tools to find the optimal policy and associated optimal value functions that minimize the expected value of the discounted infinite-horizon sum of a stage cost [26, 27]. RL uses a parameterized function approximator of the optimal policy or optimal action-value function of a problem, and provides data-driven techniques to find the optimal function parameters. Recent research have focused on MPC-based approximation for RL [28–30]. A parameterized MPC scheme can be used as a function approximator for RL, providing a formal framework to analyze the stability of the closed-loop system. In [31] it has been shown that adjusting the MPC model, cost and constraints allows the MPC to capture the optimal policy for the system even if using an inaccurate model in the MPC scheme.

In this paper, we leverage on RL technique to compute a data-based storage function. In order to capture the storage function and verify dissipativity, we first parameterize the storage function, stage cost, and terminal cost in an undiscounted tracking MPC scheme. We then use the parameterized tracking MPC in order to capture the optimal action-value function resulting from a specified infinite-horizon sum of economic stage costs. The undiscounted tracking MPC then provides a stabilizing policy for the closed-loop system regardless of whether the original ENMPC scheme is dissipative or not, and discounted or not. We show that, for dissipative problems, if the parameterization is rich enough, then the resulting storage function satisfies the dissipativity conditions for the parameters that capture the optimal action-value function accurately. We use an undiscounted tracking MPC-scheme function approximator for both the discounted and undiscounted ENMPC settings and show that the proposed method works in both cases.

We then propose to use Q-learning to adjust the parameters of the tracking MPC scheme. For a non-dissipative problem, Q-learning converges to sub-optimal parameters that can not capture the optimal action-value function of the original ENMPC scheme if the tracking MPC scheme is used as a function approximator of the action-value function. As a result, the learned storage function does not satisfy the dissipation inequality. Then we can characterize dissipative or non-dissipative problems by whether the parameterized tracking MPC scheme provides an action-value function that captures the optimal action-value function resulting from the ENMPC scheme or not. Using different examples, we show that this method can be used for general problems, i.e. nonlinear dynamics and non-quadratic stage cost, to deliver the storage function with high accuracy.

Contributions. The approach, discussed in this paper was first suggested in [32] but was limited to the undiscounted setting. In this paper, we aim at extending this early work with the following additional novelties:

- We show that an undiscounted MPC scheme can be used as a function approximator of the optimal action-value function for the discounted setting regardless of the discount factor (Theorem 3).
- The undiscounted tracking MPC is able to deliver a valid storage function in both the discounted and the undiscounted setting (Theorem 4).

- A detailed explanation of the stage cost parameterization and practical implementation (Section 5.2) was missing in [32]. The current work formally provides a universal function approximator for the tracking stage cost parameterization (Theorem 5).
- The case studies were limited in [32]. Therefore we present a wider range of case studies (discounted setting of example 6.3 and example 6.4).

Outline. The paper is structured as follows. Section 2 recalls the undiscounted ENMPC formulation and the standard dissipativity conditions, and details how one can find an equivalent tracking MPC that yields the same optimal policy and value function. Section 3 details the parameterization of the tracking MPC scheme in order to obtain a parameterized action-value function. We then show that if the tracking MPC-based function approximator can capture the optimal action-value function, then under some conditions, it yields a valid storage function. Section 4 provides the dissipativity conditions for the discounted ENMPC formulation and extends the theorem of undiscounted setting to the discounted ENMPC scheme. Section 5 details the practical implementation of the proposed method. We introduce possible parameterizations of the stage cost and the use of Q-learning to approach the optimal parameters of the tracking MPC scheme that have the best approximation of the optimal action-value function of the ENMPC scheme. Moreover, we introduce constrained learning steps in order to ensure that the stage cost of the tracking MPC is lower-bounded by a $\mathcal{K}_\infty$ function. Section 6 illustrates the simulation results for the different case studies. Section 7 delivers a discussion and 8 provides a conclusion.

Notation. a is a scalar while $\mathbf{a}$ is a vector. We denote the set of non-negative real numbers, non-negative integers, and natural numbers by $\mathbb{R}_{\geq 0}$, $\mathbb{I}_{\geq 0}$ and $\mathbb{N}$, respectively, while $\mathbb{I}_{i:j}$ refers to the set $\{i, i+1, \ldots, j\}$. A function $\alpha : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ is said to belong to class $\mathcal{K}_\infty$, if α is continuous, strictly increasing, unbounded and $\alpha(0) = 0$. The operator $\|\cdot\|$ indicates an Euclidean norm and $\|\mathbf{x}\|_Q = \sqrt{\mathbf{x}^\top Q \mathbf{x}}$ is the weighted Euclidean norm of vector $\mathbf{x}$ with respect to the positive definite matrix Q, while $\|\cdot\|_p$ is the L^p function norm. For n vectors $\mathbf{x}_1, \ldots, \mathbf{x}_n$ we define $\mathrm{col}(\mathbf{x}_1, \ldots, \mathbf{x}_n) := [\mathbf{x}_1^\top, \ldots, \mathbf{x}_n^\top]^\top$.

2 Economic Nonlinear MPC

In this section, we formulate the concept of Economic Nonlinear Model Predictive Control (ENMPC) and the associated dissipativity condition. Consider the following discrete-time, constrained nonlinear dynamical system:

$$\mathbf{s}_{k+1} = \mathbf{f}(\mathbf{s}_k, \mathbf{a}_k), \quad \mathbf{h}(\mathbf{s}_k, \mathbf{a}_k) \leq 0, \tag{1}$$

where $k \in \mathbb{I}_{\geq 0}$ is the physical time index, $\mathbf{s}_k \in \mathbb{X} \subset \mathbb{R}^n$ is the state, $\mathbf{a}_k \in \mathbb{U} \subset \mathbb{R}^m$ is the input, and $\mathbb{X}$ and $\mathbb{U}$ are state and input set, respectively. Vector field $\mathbf{f} : \mathbb{R}^{n+m} \to \mathbb{R}^n$

expresses the state transition. Function $\mathbf{h} : \mathbb{R}^{n+m} \to \mathbb{R}^d$ is a vector of mixed input-state constraints. The set of feasible state-input pairs is defined as follows:

$$\mathbb{Z} := \{(\mathbf{s}, \mathbf{a}) \in \mathbb{X} \times \mathbb{U} \,|\, \mathbf{f}(\mathbf{s}, \mathbf{a}) \in \mathbb{X},\, \mathbf{h}(\mathbf{s}, \mathbf{a}) \leq 0\} \tag{2}$$

In the ENMPC context, the selected stage cost, denoted by $L : \mathbb{X} \times \mathbb{U} \to \mathbb{R}$, typically expresses the economic cost of operating the system (1) and is not necessarily lower-bounded by a $\mathcal{K}_\infty$ function. Note that $\mathbb{X}$ is the state set without system constraints, e.g., the work area of a robot, while $\mathbf{h}$ is the system constraint, e.g, obstacles. Obviously, one could take into account $\mathbf{h}$ to redefine state set $\mathbb{X}$, but for the sake of clarity, we distinguish these two in this paper. The following standard assumption is essential in the ENMPC context and will be used in the rest of the paper.

Assumption 1 *The set $\mathbb{Z}$ is non-empty and compact and the cost $L(\cdot)$ and function $\mathbf{f}(\cdot)$ are continuous on $\mathbb{Z}$.*

An important consequence of this assumption is the boundedness of the stage cost L on the compact set $\mathbb{Z}$. An optimal steady-state pair $(\mathbf{s}_e, \mathbf{a}_e)$ with respect to the economic stage cost L is defined as follows:

$$(\mathbf{s}_e, \mathbf{a}_e) \in \underset{(\mathbf{s},\mathbf{a})\in\mathbb{Z}}{\arg\min}\; L(\mathbf{s}, \mathbf{a}) \tag{3a}$$

$$\text{s.t.} \quad \mathbf{s} = \mathbf{f}(\mathbf{s}, \mathbf{a}) \tag{3b}$$

Note that under Assumption 1, an optimal steady-state pair $(\mathbf{s}_e, \mathbf{a}_e)$ exists [8]. We then define the *shifted* stage cost ℓ, as follows:

$$\ell(\mathbf{s}, \mathbf{a}) := L(\mathbf{s}, \mathbf{a}) - L(\mathbf{s}_e, \mathbf{a}_e). \tag{4}$$

One can readily observe that $\ell(\mathbf{s}_e, \mathbf{a}_e) = 0$. In the following, we define the concept of *dissipativity*, which is crucial in the ENMPC context.

Definition 1 *System* (1) *is strictly dissipative with respect to supply rate ℓ if there exists a continuous* storage *function $\lambda : \mathbb{X} \to \mathbb{R}$ satisfying:*

$$\lambda(\mathbf{f}(\mathbf{s}, \mathbf{a})) - \lambda(\mathbf{s}) \leq -\rho(\|\mathbf{s} - \mathbf{s}_e\|) + \ell(\mathbf{s}, \mathbf{a}) \tag{5}$$

for all $(\mathbf{s}, \mathbf{a}) \in \mathbb{Z}$ and some $\rho(\cdot) \in \mathcal{K}_\infty$.

Sometimes we drop the word *strict* for the sake of simplicity. One can interpret the storage function λ as a generalized energy that is stored in the system. Then the property of dissipativity means that along any trajectory of the system, energy is dissipated, i.e., the difference in stored energy is not larger than the supplied energy to the system from the outside. Note that, adding a constant c in the storage function in the form of $\lambda + c$ does not invalidate (5). Hence, we can assume that $\lambda(\mathbf{s}_e) = 0$ without loss of generality. The strict dissipativity is a critical concept in the ENMPC

context, and the strict dissipativity with respect to supply rate ℓ yields the stability of the closed-loop system in the ENMPC scheme (see e.g., [2]). If the system is strictly dissipative with respect to supply rate ℓ, the *rotated* stage cost $\bar{\ell}$ is defined as follows:

$$\bar{\ell}(\mathbf{s}, \mathbf{a}) = \ell(\mathbf{s}, \mathbf{a}) + \lambda(\mathbf{s}) - \lambda(\mathbf{f}(\mathbf{s}, \mathbf{a})) \tag{6}$$

From (5) and (6), we have the following property for the rotated stage cost $\bar{\ell}$:

$$\bar{\ell}(\mathbf{s}_e, \mathbf{a}_e) = 0, \quad \rho(\|\mathbf{s} - \mathbf{s}_e\|) \leq \bar{\ell}(\mathbf{s}, \mathbf{a}), \tag{7}$$

Then the rotated stage cost $\bar{\ell}$ can be seen as a *tracking* stage cost with respect to the optimal steady-state point $(\mathbf{s}_e, \mathbf{a}_e)$.

A deterministic policy $\boldsymbol{\pi} : \mathbb{X} \to \mathbb{U}$ maps the state space to the input space, and determines how to choose input $\mathbf{a}_k$ at each state $\mathbf{s}_k$. Let us consider the following infinite-horizon undiscounted optimal control problem:

$$V^\star(\mathbf{s}) = \min_{\boldsymbol{\pi}} \quad \sum_{j=0}^{\infty} \ell(\mathbf{x}_j, \boldsymbol{\pi}(\mathbf{x}_j)) \tag{8a}$$

$$\text{s.t.} \quad \forall j \in \mathbb{I}_{\geq 0} \quad \mathbf{x}_{j+1} = \mathbf{f}\left(\mathbf{x}_j, \boldsymbol{\pi}(\mathbf{x}_j)\right) \tag{8b}$$

$$(\mathbf{x}_j, \boldsymbol{\pi}(\mathbf{x}_j)) \in \mathbb{Z}, \quad \mathbf{x}_0 = \mathbf{s}, \tag{8c}$$

where $V^\star : \mathbb{X} \to \mathbb{R}$ is the optimal value function and sequence $(\mathbf{x}_j)_{j=1}^{\infty}$ is the state trajectory under policy $\boldsymbol{\pi}$ starting from an arbitrary (possibly random) state $\mathbf{x}_0 = \mathbf{s}$. We denote the optimal policy solution of (8) by $\boldsymbol{\pi}^\star$.

The optimal action-value function $Q^\star : \mathbb{X} \times \mathbb{U} \to \mathbb{R}$ associated to (8) is defined as follows:

$$Q^\star(\mathbf{s}, \mathbf{a}) := \ell(\mathbf{s}, \mathbf{a}) + V^\star(\mathbf{f}(\mathbf{s}, \mathbf{a})) \tag{9}$$

This equation will play a key role in this paper. The Bellman equations read as follows:

$$V^\star(\mathbf{s}) = \min_{\mathbf{a}} Q^\star(\mathbf{s}, \mathbf{a}), \quad \boldsymbol{\pi}^\star(\mathbf{s}) \in \arg\min_{\mathbf{a}} Q^\star(\mathbf{s}, \mathbf{a}) \tag{10}$$

Under Assumption 1, the stage cost ℓ and storage function λ will remain bounded over the set $\mathbb{Z}$. Then, using a telescoping sum, the cost (8a) can be expressed based on the rotated stage cost $\bar{\ell}$ as follows:

$$\begin{aligned} \sum_{k=0}^{\infty} \ell(\mathbf{x}_k, \boldsymbol{\pi}(\mathbf{x}_k)) &= \sum_{k=0}^{\infty} \bar{\ell}(\mathbf{x}_k, \boldsymbol{\pi}(\mathbf{x}_k)) - \lambda(\mathbf{x}_k) + \lambda(\mathbf{x}_{k+1}) \\ &= -\lambda(\mathbf{x}_0) + \lambda(\mathbf{x}_\infty) + \sum_{k=0}^{\infty} \bar{\ell}(\mathbf{x}_k, \boldsymbol{\pi}(\mathbf{x}_k)) \end{aligned} \tag{11}$$

Then using (11), (8) can be written as follows:

$$V^\star(\mathbf{s}) = \min_{\boldsymbol{\pi}} \quad -\lambda(\mathbf{s}) + \lambda(\mathbf{x}_\infty) + \sum_{k=0}^{\infty} \bar{\ell}(\mathbf{x}_k, \boldsymbol{\pi}(\mathbf{x}_k)), \quad \text{s.t.} \quad (8b), (8c) \tag{12}$$

where $\mathbf{x}_\infty := \lim_{k\to\infty} \mathbf{x}_k$. Since the term $-\lambda(\mathbf{s})$ is independent of the policy $\boldsymbol{\pi}$, it does not modify the optimal policy solution of (8c). For a strictly dissipative problem, an infinite-horizon ENMPC (8) has the same optimal policy as a corresponding tracking MPC (12) scheme using the rotated stage cost $\bar{\ell}$. This rotated cost is zero at the optimal steady-state and lower-bounded by a $\mathcal{K}_\infty$ function (see (7)). Thus, the dynamics in closed-loop with the optimal ENMPC policy will be stable for a strictly dissipative problem [7]. Therefore the closed-loop state trajectories converge to the optimal steady-state, i.e. $\lim_{k\to\infty} \mathbf{x}_k = \mathbf{s}_e$. From the continuity of the storage function, it results in $\lim_{k\to\infty} \lambda(\mathbf{x}_k) = \lambda(\mathbf{s}_e) = 0$. In fact, tracking NMPC (12) delivers the same input/state solution and same policy as ENMPC (8) and the closed-loop behaviour of the system under the optimal policy resulting from (8) and (12) are equivalent.

In this paper, we use the above idea stating that under the dissipativity condition, we can find an equivalent tracking MPC for a given ENMPC scheme. Indeed, we are interested in doing the above procedure in reverse. We provide a parameterized tracking MPC scheme and claim that if this problem can capture the optimal action-value of the ENMPC scheme, then the problem is dissipative. Moreover, the resulting storage function satisfies the dissipativity inequality. We will detail this idea in the next section.

3 Tracking MPC-based function approximator

In general, verifying that the given problem is dissipative or not and finding the storage function $\lambda(\mathbf{s})$ that satisfies (5) is not trivial. In this section, we parameterize the storage function as well as the stage cost and terminal cost of a tracking MPC scheme. We use the parameterized tracking MPC scheme as a function approximator for the optimal action-value function (9). Then we will show that if the resulting action-value function from the tracking MPC-scheme is able to capture the optimal action-value function of the ENMPC-scheme (9) for some parameters, then the storage function will satisfy the strict dissipativity inequality.

Let us consider the following finite-horizon MPC scheme parameterized by $\boldsymbol{\theta}$:

$$V_{\boldsymbol{\theta}}(\mathbf{s}) = \min_{\mathbf{u}} -\lambda_{\boldsymbol{\theta}}(\mathbf{s}) + T_{\boldsymbol{\theta}}(\mathbf{x}_N) + \sum_{j=0}^{N-1} \hat{\ell}_{\boldsymbol{\theta}}(\mathbf{x}_j, \mathbf{u}_j) \tag{13a}$$

$$\text{s.t.} \quad \forall j \in \mathbb{I}_{0:N-1} : \mathbf{x}_{j+1} = \mathbf{f}(\mathbf{x}_j, \mathbf{u}_j) \tag{13b}$$

$$(\mathbf{x}_j, \mathbf{u}_j) \in \mathbb{Z}, \ \mathbf{x}_0 = \mathbf{s}, \ \mathbf{x}_N \in \mathbb{X}_\mathrm{f} \tag{13c}$$

where $\lambda_{\boldsymbol{\theta}}$ is the approximated storage function, $\hat{\ell}_{\boldsymbol{\theta}}$ is the parameterized stage cost, $T_{\boldsymbol{\theta}}$ is the parameterized terminal cost and $\boldsymbol{\theta} \in \Theta \subseteq \mathbb{R}^{n_{\boldsymbol{\theta}}}$ is the parameters vector. Function $V_{\boldsymbol{\theta}}$ is the parameterized value function, N is the horizon length, $\mathbb{X}_{\mathrm{f}} \subseteq \mathbb{X}$ is the terminal set, containing $\mathbf{s}_e$, and $(\mathbf{x}_j)_{j=0}^{N}$ and $\mathbf{u} := \mathrm{col}(\mathbf{u}_0, \ldots, \mathbf{u}_{N-1})$ are the predicted state and input profile, respectively. We assume that $\lambda_{\boldsymbol{\theta}}$, $\hat{\ell}_{\boldsymbol{\theta}}$ and $T_{\boldsymbol{\theta}}$ are continuous functions and $\mathbb{X}_f$ is a control invariant set. Moreover, we assume that the MPC scheme (13) is a tracking MPC, i.e., the stage cost $\hat{\ell}_{\boldsymbol{\theta}}$ is lower-bounded by a $\mathcal{K}_\infty$ function for all $\theta \in \Theta$. More specifically, we will make the following assumption held by construction.

Assumption 2 *The stage cost $\hat{\ell}_{\boldsymbol{\theta}}$ satisfies:*

$$\hat{\ell}_{\boldsymbol{\theta}}(\mathbf{s}_e, \mathbf{a}_e) = 0, \quad \alpha_1(\|\mathbf{s} - \mathbf{s}_e\|) \leq \hat{\ell}_{\boldsymbol{\theta}}(\mathbf{s}, \mathbf{a}) \,, \tag{14}$$

for all $(\mathbf{s}, \mathbf{a}) \in \mathbb{Z}$, all $\boldsymbol{\theta} \in \Theta$ and some $\alpha_1 \in \mathcal{K}_\infty$.

We will further discuss in Section 5.2 how to satisfy Assumption 2 in the learning context.

For the MPC scheme (13), the parameterized deterministic policy $\boldsymbol{\pi}_{\boldsymbol{\theta}}$ can be obtained as follows:

$$\boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathbf{s}) = \mathbf{u}_0^\star(\mathbf{s}, \boldsymbol{\theta}) \,, \tag{15}$$

where $\mathbf{u}_0^\star$ is the optimal solution of (13) corresponding to the first input $\mathbf{u}_0$. We introduce next the parameterized action-value function. MPC-based action-value function $Q_{\boldsymbol{\theta}}$ associated to (13) can be defined as follows (see [31]):

$$\begin{aligned} Q_{\boldsymbol{\theta}}(\mathbf{s}, \mathbf{a}) := \min_{\mathbf{u}} \quad & \text{(13a)} && \text{(16a)} \\ \text{s.t.} \quad & \text{(13b), (13c)} && \text{(16b)} \\ & \mathbf{u}_0 = \mathbf{a} && \text{(16c)} \end{aligned}$$

Constraint (16c) is added to the MPC scheme (13) in order to enforce the first input $\mathbf{u}_0$ to have a specific value $\mathbf{a}$. One can verify that the value function $V_{\boldsymbol{\theta}}$ in (13), the action-value function $Q_{\boldsymbol{\theta}}$ in (16) and the policy $\boldsymbol{\pi}_{\boldsymbol{\theta}}$ in (15) satisfy the following fundamental Bellman equations:

$$V_{\boldsymbol{\theta}}(\mathbf{s}) = \min_{\mathbf{a}} Q_{\boldsymbol{\theta}}(\mathbf{s}, \mathbf{a}), \quad \boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathbf{s}) \in \arg\min_{\mathbf{a}} Q_{\boldsymbol{\theta}}(\mathbf{s}, \mathbf{a}) \tag{17}$$

In the following, we make a basic stability assumption on the parameterized terminal cost $T_{\boldsymbol{\theta}}$ and the terminal set $\mathbb{X}_{\mathrm{f}}$.

Assumption 3 *For all $\mathbf{s} \in \mathbb{X}_{\mathrm{f}}$, there exists an input $\mathbf{a} \in \mathbb{U}$, satisfying $(\mathbf{s}, \mathbf{a}) \in \mathbb{Z}$, $\mathbf{f}(\mathbf{s}, \mathbf{a}) \in \mathbb{X}_{\mathrm{f}}$, and:*

$$T_{\boldsymbol{\theta}}(\mathbf{f}(\mathbf{s}, \mathbf{a})) - T_{\boldsymbol{\theta}}(\mathbf{s}) \leq -\hat{\ell}_{\boldsymbol{\theta}}(\mathbf{s}, \mathbf{a}), \qquad \forall \theta \in \Theta \tag{18}$$

Moreover, for all $\boldsymbol{\theta} \in \Theta$, $T_{\boldsymbol{\theta}}(\mathbf{s}_e) = 0$ and $T_{\boldsymbol{\theta}}(\mathbf{s}) > 0$ for all $\mathbf{s} \in \mathbb{X}_{\mathrm{f}} \setminus \{\mathbf{s}_e\}$.

Assumption 3 is a basic assumption in the MPC context, which allows one to discuss the closed-loop stability of the resulting optimal policy when a finite horizon is used. To satisfy (18) in the learning context, one needs to provide a generic non-negative function approximator for the terminal cost $T_{\boldsymbol{\theta}}$. This assumption can be relaxed completely by choosing the terminal set $\mathbb{X}_{\mathrm{f}} = \{\mathbf{s}_e\}$. Such a terminal set, however, reduces the feasibility domain of the solution in (13) [33]. Moreover, in [34] it has been shown that for sufficiently large horizon N, the tracking MPC results in a stabilizing policy for the closed-loop system without the terminal cost $T_{\boldsymbol{\theta}}$ and the terminal set $\mathbb{X}_{\mathrm{f}}$.

The following two functions will be used in the rest of the paper. We define $\Psi^N_{\boldsymbol{\theta}} : \mathbb{X} \to \mathbb{R}$ and $\Lambda^N_{\boldsymbol{\theta}} : \mathbb{X} \times \mathbb{U} \to \mathbb{R}$ as:

$$\Psi^N_{\boldsymbol{\theta}}(\mathbf{s}) := \min_{\mathbf{u}} \quad T_{\boldsymbol{\theta}}(\mathbf{x}_N) + \sum_{k=0}^{N-1} \hat{\ell}_{\boldsymbol{\theta}}(\mathbf{x}_k, \mathbf{u}_k)$$
$$\text{s.t.} \qquad (13\text{b}), (13\text{c}) \tag{19a}$$
$$\Lambda^N_{\boldsymbol{\theta}}(\mathbf{s}, \mathbf{a}) := \hat{\ell}_{\boldsymbol{\theta}}(\mathbf{s}, \mathbf{a}) + \Psi^{N-1}_{\boldsymbol{\theta}}(\mathbf{f}(\mathbf{s}, \mathbf{a})) \tag{19b}$$

From (13) and (16), one can observe that:

$$V_{\boldsymbol{\theta}}(\mathbf{s}) = -\lambda_{\boldsymbol{\theta}}(\mathbf{s}) + \Psi^N_{\boldsymbol{\theta}}(\mathbf{s}) \tag{20a}$$
$$Q_{\boldsymbol{\theta}}(\mathbf{s}) = -\lambda_{\boldsymbol{\theta}}(\mathbf{s}) + \Lambda^N_{\boldsymbol{\theta}}(\mathbf{s}, \mathbf{a}) \tag{20b}$$

The following Lemma on the monotonicity property of $\Psi^N_{\boldsymbol{\theta}}$ with respect to the horizon N will be useful in the next theorem.

Lemma 1 *Under Assumptions 1-3, the following inequality holds:*

$$\Psi^{N+1}_{\boldsymbol{\theta}}(\mathbf{s}) \leq \Psi^N_{\boldsymbol{\theta}}(\mathbf{s}), \quad \forall N \in \mathbb{I}_{\geq 0}\ , \forall \theta \in \Theta,\ \forall s \in \mathbb{X}. \tag{21}$$

Proof: Let us assume that $\mathbf{u}^\star_0$, $\mathbf{x}^\star_0 = \mathbf{s}$, ..., $\mathbf{u}^\star_{N-1}$, $\mathbf{x}^\star_{N-1}$, $\mathbf{x}^\star_N$ is the solution of (19a), then the sequence $\mathbf{u}^\star_0$, $\mathbf{x}^\star_0 = \mathbf{s}$, ..., $\mathbf{u}^\star_{N-1}$, $\mathbf{x}^\star_{N-1}$, $\mathbf{x}^\star_N$, $\mathbf{u}_N$, $\mathbf{x}_{N+1}$ is a feasible candidate solution for $\Psi^{N+1}_{\boldsymbol{\theta}}$, where $\mathbf{u}_N$ is a control input such that $\mathbf{f}(\mathbf{x}^\star_N, \mathbf{u}_N) \in \mathbb{X}_f$ and $(\mathbf{x}^\star_N, \mathbf{u}_N) \in \mathbb{Z}$ and $\mathbf{x}_{N+1} := \mathbf{f}(\mathbf{x}^\star_N, \mathbf{u}_N)$. Note that such an input $\mathbf{u}_N$ exists under Assumption 3 because $\mathbf{x}^\star_N \in \mathbb{X}_{\mathrm{f}}$. Then we have:

$$\begin{aligned}
\Psi^{N+1}_{\boldsymbol{\theta}}(\mathbf{s}) &\leq T_{\boldsymbol{\theta}}(\mathbf{x}_{N+1}) + \hat{\ell}_{\boldsymbol{\theta}}(\mathbf{x}^\star_N, \mathbf{u}_N) + \sum_{k=0}^{N-1} \hat{\ell}_{\boldsymbol{\theta}}(\mathbf{x}^\star_k, \mathbf{u}^\star_k) \\
&= T_{\boldsymbol{\theta}}(\mathbf{x}_{N+1}) + \hat{\ell}_{\boldsymbol{\theta}}(\mathbf{x}^\star_N, \mathbf{u}_N) - T_{\boldsymbol{\theta}}(\mathbf{x}^\star_N) + T_{\boldsymbol{\theta}}(\mathbf{x}^\star_N) + \sum_{k=0}^{N-1} \hat{\ell}_{\boldsymbol{\theta}}(\mathbf{x}^\star_k, \mathbf{u}^\star_k) \\
&= \Psi^N_{\boldsymbol{\theta}}(\mathbf{s}) + T_{\boldsymbol{\theta}}(\mathbf{x}_{N+1}) + \hat{\ell}_{\boldsymbol{\theta}}(\mathbf{x}^\star_N, \mathbf{u}_N) - T_{\boldsymbol{\theta}}(\mathbf{x}^\star_N) \leq \Psi^N_{\boldsymbol{\theta}}(\mathbf{s})
\end{aligned} \tag{22}$$

Note that we used Assumption 3 in the last inequality since $\mathbf{u}_N$ has been selected according to Assumption 3. ■

The following theorem provides one of the main results of the paper and states that if $Q_{\boldsymbol{\theta}}$ captures the optimal action-value function $Q^\star$ for some parameters $\boldsymbol{\theta}^\star$ then the storage function $\lambda_{\boldsymbol{\theta}^\star}$ satisfies the strict dissipativity inequality with respect to supply rate ℓ.

Theorem 1 *Under Assumptions 1-3, if there exists a* $\boldsymbol{\theta}^\star \in \Theta$ *such that:*

$$Q_{\boldsymbol{\theta}^\star}(\mathbf{s},\mathbf{a}) = Q^\star(\mathbf{s},\mathbf{a})\,,\ \forall (\mathbf{s},\mathbf{a}) \in \mathbb{Z}. \tag{23}$$

Then system (1) *is strictly dissipative with respect to supply rate* ℓ *and storage function* $\lambda_{\boldsymbol{\theta}^\star}$ *satisfies strict dissipativity inequality* (5).

Proof: If (23) holds, we have:

$$V_{\boldsymbol{\theta}^\star}(\mathbf{s}) = \min_{\mathbf{a}} Q_{\boldsymbol{\theta}^\star}(\mathbf{s},\mathbf{a}) = \min_{\mathbf{a}} Q^\star(\mathbf{s},\mathbf{a}) = V^\star(\mathbf{s}) \tag{24}$$

and:

$$\begin{aligned} Q^\star(\mathbf{s},\mathbf{a}) = Q_{\boldsymbol{\theta}^\star}(\mathbf{s},\mathbf{a}) &\overset{(20b)}{=} -\lambda_{\boldsymbol{\theta}^\star}(\mathbf{s}) + \Lambda^N_{\boldsymbol{\theta}^\star}(\mathbf{s},\mathbf{a}) \\ &\overset{(19b)}{=} -\lambda_{\boldsymbol{\theta}^\star}(\mathbf{s}) + \hat{\ell}_{\boldsymbol{\theta}^\star}(\mathbf{s},\mathbf{a}) + \Psi^{N-1}_{\boldsymbol{\theta}^\star}(\mathbf{f}(\mathbf{s},\mathbf{a})) \end{aligned} \tag{25}$$

Moreover, from (9), (23) and (13), we have:

$$\begin{aligned} Q^\star(\mathbf{s},\mathbf{a}) &= \ell(\mathbf{s},\mathbf{a}) + V^\star(\mathbf{f}(\mathbf{s},\mathbf{a})) = \ell(\mathbf{s},\mathbf{a}) + V_{\boldsymbol{\theta}^\star}(\mathbf{f}(\mathbf{s},\mathbf{a})) \\ &= \ell(\mathbf{s},\mathbf{a}) - \lambda_{\boldsymbol{\theta}^\star}(\mathbf{f}(\mathbf{s},\mathbf{a})) + \Psi^N_{\boldsymbol{\theta}^\star}(\mathbf{f}(\mathbf{s},\mathbf{a})) \\ &\leq \ell(\mathbf{s},\mathbf{a}) - \lambda_{\boldsymbol{\theta}^\star}(\mathbf{f}(\mathbf{s},\mathbf{a})) + \Psi^{N-1}_{\boldsymbol{\theta}^\star}(\mathbf{f}(\mathbf{s},\mathbf{a})) \end{aligned} \tag{26}$$

where we used (21) in the last inequality. From (25) and (26), we have:

$$\lambda_{\boldsymbol{\theta}^\star}(\mathbf{f}(\mathbf{s},\mathbf{a})) - \lambda_{\boldsymbol{\theta}^\star}(\mathbf{s}) \leq -\hat{\ell}_{\boldsymbol{\theta}^\star}(\mathbf{s},\mathbf{a}) + \ell(\mathbf{s},\mathbf{a}) \leq -\alpha_1(\|\mathbf{s}-\mathbf{s}_e\|) + \ell(\mathbf{s},\mathbf{a}) \tag{27}$$

for all $(\mathbf{s},\mathbf{a}) \in \mathbb{Z}$. Then system (1) is strictly dissipative with respect to supply rate ℓ. ■

Theorem 1 requires a universal function approximator for the continuous action-value function $Q_{\boldsymbol{\theta}^\star}$ in order to satisfy (23). In the tracking MPC-based context, we can achieve a universal action-value approximator by choosing a generic function for the storage function, the terminal cost, and the stage cost satisfying Assumptions 1-3. In this paper, we will use RL techniques to find the optimal parameters $\boldsymbol{\theta}^\star$ fulfilling (23). In Section 5.1, we will detail Q-learning as a classic tool to attain the optimal parameters $\boldsymbol{\theta}^\star$ that best estimate the optimal action-value function $Q^\star$. In practice, we may not be able to provide such an approximator for the MPC scheme. In this case, the learning algorithm will find the optimal parameters that have the best

approximation of the optimal action-value function $Q^\star$. We will detail this case in Section 5.

Condition (23) is valid only for strictly dissipative problems. However, tracking MPC (13) always results in a stabilizing policy, regardless of whether the problem is strictly dissipative or strictly non-dissipative. The next proposition states the stability of the closed-loop system under policy (15).

Proposition 1 *Under Assumptions 1-3, the closed-loop system* $\mathbf{s}_{k+1} = \mathbf{f}(\mathbf{s}_k, \boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathbf{s}_k))$*, with policy* $\boldsymbol{\pi}_{\boldsymbol{\theta}}$*, resulting from tracking MPC scheme* (13)*, is asymptotically stable for all* $\boldsymbol{\theta} \in \Theta$*, with respect to the optimal steady state* $\mathbf{s}_e$*.*

Proof: This is a standard result given in, e.g., [7]. ■

As a consequence of Proposition 1, under Assumptions 1-3, parameterized policy $\boldsymbol{\pi}_{\boldsymbol{\theta}}$ resulting from tracking MPC scheme (13) is stabilizing in closed-loop with system (1) for both dissipative and non-dissipative problems. However, for a non-dissipative problem, there are no parameters $\boldsymbol{\theta}^\star \in \Theta$ that satisfy (23). The next corollary formalizes this statement.

Corollary 1 *For a strictly non-dissipative problem, there exists no* $\boldsymbol{\theta}^\star \in \Theta$ *such that* (23) *holds.*

Proof: Let us assume that there exists a $\boldsymbol{\theta}^\star \in \Theta$ such that (23) holds. Then from Theorem 1, system 1 is strictly dissipative with respect to supply rate ℓ. By contradiction, we conclude that there exists no such $\boldsymbol{\theta}^\star \in \Theta$ for a strictly non-dissipative problem. ■

Note that, for non-dissipative problems, the best estimation of the optimal action-value function of the ENMPC will have an error. Therefore, this will be the key to characterizing non-dissipative problems.

In Section 5, we will introduce Q-learning as a method to find the best approximation of the exact optimal action-value function in the Least-Square (LS) sense. In the next section, we will detail the discounted setting of the ENMPC scheme.

4 Discounted ENMPC

Discounted optimal control has attracted wide attention in e.g. economic application [35], and social science [36]. In the discounted setting, the stage costs are weighted by a factor γ^k, where $\gamma \in (0, 1)$ is labelled discount factor, and k is the physical time index in discrete-time systems. A discounted infinite-horizon objective function is often the preferred setting in both Dynamic Programming (DP) and RL to formulate

well-posed Markov Decision Processes (MDPs). The contraction property of the DP operator has been shown in [37] for discounted optimal control, while it needs more requirements for the undiscounted setting.

Let us consider the discounted optimal policy $\boldsymbol{\pi}_\gamma^\star$ as the solution of the following discounted infinite-horizon problem:

$$V_\gamma^\star(\mathbf{s}) = \min_{\boldsymbol{\pi}} \quad \sum_{j=0}^{\infty} \gamma^j \ell(\mathbf{x}_j, \boldsymbol{\pi}(\mathbf{x}_j)) \tag{28a}$$

$$\text{s.t.} \quad \forall j \in \mathbb{I}_{\geq 0} \quad \mathbf{x}_{j+1} = \mathbf{f}\left(\mathbf{x}_j, \boldsymbol{\pi}(\mathbf{x}_j)\right) \tag{28b}$$

$$(\mathbf{x}_j, \boldsymbol{\pi}(\mathbf{x}_j)) \in \mathbb{Z}, \ \mathbf{x}_0 = \mathbf{s}\,, \tag{28c}$$

where $V_\gamma^\star$ is the discounted optimal value function. Note that under Assumption (1) the stage cost is bounded on the compact set $\mathbb{Z}$ and a discounted sum of bounded stage costs results in a bounded value function.

The dissipativity theory for discounted formulations is more involved than for the undiscounted setting. In the former case, the discount factor γ has a central role to establish the closed-loop stability of the policy. It is shown in [38] that, unlike the undiscounted setting, discounted strictly dissipativity with respect to supply rate ℓ does not necessarily yield the stability of the closed-loop system under the optimal policy $\boldsymbol{\pi}_\gamma^\star$. It was shown that under mild assumptions on the value function, the controllability of the system, and the detectability with respect to the stage cost, there exists a $\gamma^\star < 1$ such that (1) is practically asymptotically stable for any $\gamma \in (\gamma^\star, 1]$. Asymptotic stability requires an additional condition to the discounted strict dissipativity conditions. Recently the dissipativity condition has been extended to the discounted setting [39]. The resulting conditions on the tuple $(\mathbf{f}, \ell, \gamma)$ are called *Strong Discounted Strict Dissipativity (SDSD)*. The SDSD conditions guarantee asymptotic stability of the closed-loop dynamics $\mathbf{f}$ with the discounted optimal policy $\boldsymbol{\pi}_\gamma^\star$.

In this section, we address the discounted optimal control problem and recall the dissipativity condition in the discounted setting. We will show that a finite-horizon undiscounted MPC scheme is still able to capture the optimal policy and optimal value functions of the discounted setting. Then we will show that the storage function resulting from the undiscounted tracking MPC parameterization (13) satisfies the SDSD conditions if the parameterized action-value function based on the tracking MPC (16) captures the optimal action-value function resulting from the discounted ENMPC-scheme.

For the discounted setting (28), the action-value function $Q_\gamma^\star$ is defined as follows:

$$Q_\gamma^\star(\mathbf{s}, \mathbf{a}) = \ell(\mathbf{s}, \mathbf{a}) + \gamma V_\gamma^\star(\mathbf{f}(\mathbf{s}, \mathbf{a})) \tag{29}$$

Then the Bellman equations read as:

$$V_\gamma^\star(\mathbf{s}) = \min_{\boldsymbol{\pi}} Q_\gamma^\star(\mathbf{s}, \boldsymbol{\pi}(\mathbf{s})) \tag{30a}$$

$$\boldsymbol{\pi}_\gamma^\star(\mathbf{s}) \in \arg\min_{\boldsymbol{\pi}} Q_\gamma^\star(\mathbf{s}, \boldsymbol{\pi}(\mathbf{s})) \tag{30b}$$

We next define the discounted dissipativity analogous to definition 1 (see e.g., [40]).

Definition 2 *System* (1) *is discounted strictly dissipative with respect to supply rate* ℓ *if there exists a continuous* storage *function* $\lambda : \mathbb{X} \to \mathbb{R}$ *satisfying:*

$$\gamma\lambda(\mathbf{f}(\mathbf{s},\mathbf{a})) - \lambda(\mathbf{s}) \leq -\rho(\|\mathbf{s}-\mathbf{s}_e\|) + \ell(\mathbf{s},\mathbf{a}) \tag{31}$$

for the discount factor $\gamma \in (0,1)$*, all* $(\mathbf{s},\mathbf{a}) \in \mathbb{Z}$ *and some* $\rho(\cdot) \in \mathcal{K}_\infty$*.*

We next recall Theorem 2 from [39] stating the SDSD conditions that result in asymptotic stability of the closed-loop system under the optimal policy $\boldsymbol{\pi}^\star_\gamma$, resulting from the discounted setting.

Theorem 2 *The closed-loop system* (1) *with policy* $\boldsymbol{\pi}^\star_\gamma$ *is asymptotically stable if:*

1. *System* (1) *is discounted strictly dissipative with respect to supply rate* ℓ*.*
2. *The storage function satisfies:*

$$\ell(\mathbf{s},\mathbf{a}) + \lambda(\mathbf{s}) - \lambda(\mathbf{f}(\mathbf{s},\mathbf{a})) + (\gamma-1)V^\star_\gamma(\mathbf{f}(\mathbf{s},\mathbf{a})) \geq \rho(\|\mathbf{s}-\mathbf{s}_e\|) \tag{32}$$

for all $(\mathbf{s},\mathbf{a}) \in \mathbb{Z}$ *and for some* $\rho \in \mathcal{K}_\infty$*.*

Proof: See Theorem 2 in [39]. ■

Conditions 1 and 2 in Theorem 2 are called SDSD. In fact, condition (32) is added to the discounted strictly dissipativity to ensure asymptotic stability. For $\gamma = 1$, the two conditions in Theorem 2 coincide and yield the undiscounted strictly dissipativity condition (5) with respect to supply rate ℓ.

In the following, we show that the resulting storage function from the tracking MPC-based parameterization of action-value function (16) is a valid storage function that satisfies SDSD conditions, even if the optimal action-value function is defined in the discounted setting. To this end, first, we show that an undiscounted finite-horizon optimal control problem is able to capture the optimal policy and optimal value functions of a discounted infinite-horizon problem. I.e., there exists an undiscounted finite-horizon MPC scheme that can capture the optimal policy and optimal value functions of a discounted infinite-horizon setting. Next theorem, states this claim formally.

Theorem 3 *There exists a terminal cost* $\hat{T} : \mathbb{X} \to \mathbb{R}$ *and a stage cost* $\hat{L} : \mathbb{X} \times \mathbb{U} \to \mathbb{R}$ *such that the following MPC-based value function* $\hat{V}^N$*:*

$$\hat{V}^N(\mathbf{s}) := \min_{\mathbf{u}} \ \hat{J}^N(\mathbf{s},\mathbf{u}) := \hat{T}(\mathbf{x}_N) + \sum_{j=0}^{N-1} \hat{L}(\mathbf{x}_j,\mathbf{u}_j) \tag{33a}$$

$$\text{s.t.} \quad (13b), (13c) \tag{33b}$$

and action-value function $\hat{Q}^N$*:*

$$\hat{Q}^N(\mathbf{s},\mathbf{a}) := \min_{\mathbf{u}} \text{(33a)} \tag{34a}$$

$$\text{s.t.} \quad \text{(16b)}, \text{(16c)} \tag{34b}$$

yield the following identities, $\forall\gamma$*:*

1. $\hat{\boldsymbol{\pi}}^N(\mathbf{s}) = \boldsymbol{\pi}^\star_\gamma(\mathbf{s})$, $\forall N > 0$
2. $\hat{V}^N(\mathbf{s}) = V^\star_\gamma(\mathbf{s})$, $\forall N \geq 0$
3. $\hat{Q}^N(\mathbf{s},\mathbf{a}) = Q^\star_\gamma(\mathbf{s},\mathbf{a})$, $\forall N > 0$

where $\hat{\boldsymbol{\pi}}(\mathbf{s}) = \mathbf{u}^\star_0(\mathbf{s})$ *is the solution of* (33)*, associated to the first input.*

Proof: First, one can show that the following Bellman equations hold for the undiscounted MPC:

$$\hat{V}^N(\mathbf{s}) = \hat{Q}^N(\mathbf{s}, \hat{\boldsymbol{\pi}}^N(\mathbf{s})) = \min_{\mathbf{a}} \hat{Q}^N(\mathbf{s},\mathbf{a}), \tag{35a}$$

$$\hat{Q}^N(\mathbf{s},\mathbf{a}) = \hat{L}(\mathbf{s},\mathbf{a}) + \hat{V}^{N-1}(\mathbf{f}(\mathbf{s},\mathbf{a})) \tag{35b}$$

We choose the terminal cost $\hat{T}$ and stage cost $\hat{L}$ as follows:

$$\hat{T}(\mathbf{s}) = V^\star_\gamma(\mathbf{s}) \tag{36a}$$

$$\hat{L}(\mathbf{s},\mathbf{a}) = \ell(\mathbf{s},\mathbf{a}) + (\gamma-1)V^\star_\gamma(\mathbf{f}(\mathbf{s},\mathbf{a})) = Q^\star_\gamma(\mathbf{s},\mathbf{a}) - V^\star_\gamma(\mathbf{f}(\mathbf{s},\mathbf{a})) \tag{36b}$$

For $N = 0$, we know that:

$$\hat{V}^1(\mathbf{s}) = \hat{T}(\mathbf{s}) = V^\star_\gamma(\mathbf{s}) \tag{37}$$

Then (2) is trivial in this case. Let us consider $N > 0$. Using a telescopic sum, we can rewrite the cost of (33) as follows:

$$\hat{J}^N(\mathbf{s},\mathbf{u}) = \hat{T}(\mathbf{x}_N) + \sum_{j=0}^{N-1} \hat{L}(\mathbf{x}_j,\mathbf{u}_j) = \tag{38}$$

$$Q^\star_\gamma(\mathbf{s},\mathbf{u}_0) + \sum_{j=1}^{N-1} Q^\star_\gamma(\mathbf{x}_j,\mathbf{u}_j) - V^\star_\gamma(\mathbf{x}_j)$$

Note that from Assumptions 1, the optimal value function $V^\star_\gamma$ and the optimal action-value function $Q^\star_\gamma$ are bounded, because the stage cost ℓ is bounded on the compact set $\mathbb{Z}$ and discounted infinite-horizon of bounded functions remains bounded. Therefore,

there are input sequences $(\mathbf{u}_j)_{j=0}^{N-1}$ such that all terms in (38) are bounded on $\mathbb{Z}$. From (30), we have:

$$0 \leq Q_\gamma^\star(\mathbf{x}_j, \mathbf{u}_j) - V_\gamma^\star(\mathbf{x}_j)\,, \forall j = 1, \ldots, N-1 \tag{39a}$$

$$V_\gamma^\star(\mathbf{s}) \leq Q_\gamma^\star(\mathbf{s}, \mathbf{u}_0) \tag{39b}$$

for all input sequence $\mathbf{u}_0, \ldots, \mathbf{u}_{N-1}$. A substitution of (39) into (38) yields:

$$V_\gamma^\star(\mathbf{s}) \leq \hat{J}^N(\mathbf{s}, \mathbf{u}), \quad \forall \mathbf{u} \tag{40}$$

Note that (40) is a tight inequality and the equality holds when:

$$\mathbf{u}^\star = \mathrm{col}(\boldsymbol{\pi}_\gamma^\star(\mathbf{x}_0), \ldots, \boldsymbol{\pi}_\gamma^\star(\mathbf{x}_{N-1})) \tag{41}$$

because this choice turns all inequalities in (39) into equalities. Indeed (41) is the optimal solution of (33) and it reads:

$$V_\gamma^\star(\mathbf{s}) = \min_{\mathbf{u}} \ \hat{J}^N(\mathbf{s}, \mathbf{u}) \overset{(33)}{=} \hat{V}^N(\mathbf{s}) \tag{42}$$

and it concludes (2). Moreover, from (41) we have:

$$\hat{\boldsymbol{\pi}}^N(\mathbf{s}) = \mathbf{u}_0^\star(\mathbf{s}) = \boldsymbol{\pi}_\gamma^\star(\mathbf{s}) \tag{43}$$

it results in (1), and:

$$\begin{aligned} \hat{Q}^N(\mathbf{s}, \mathbf{a}) =& \hat{L}(\mathbf{s}, \mathbf{a}) + \hat{V}^{N-1}(\mathbf{f}(\mathbf{s}, \mathbf{a})) = Q_\gamma^\star(\mathbf{s}, \mathbf{a}) - \\ & V_\gamma^\star(\mathbf{f}(\mathbf{s}, \mathbf{a})) + \hat{V}^{N-1}(\mathbf{f}(\mathbf{s}, \mathbf{a})) \overset{(42)}{=} Q_\gamma^\star(\mathbf{s}, \mathbf{a})\,, \end{aligned} \tag{44}$$

completes the proof. ■

Note that condition (32) can be seen as a standard (undiscounted) dissipativity condition when the stage cost $\hat{L}$ in (36b) is used. In fact, the dissipativity condition of undiscounted MPC-scheme (33) with stage cost (36b) is equal to condition (32). Then, similar to Section 2, it can be shown that if a problem satisfies SDSD, then undiscounted MPC (33) can be reformulated as an equivalent tracking MPC with the same optimal policy and optimal value functions. Therefore, a discounted ENMPC has an equivalent undiscounted tracking MPC for the problems that satisfy SDSD conditions.

Theorem 3 allows us to use an undiscounted MPC-scheme (13) and (16) as a function approximator for the value function $V_\gamma^\star$ and action-value function $Q_\gamma^\star$, respectively, independently of the discount factor γ. The next theorem extends theorem 1 to the undiscounted setting and SDSD conditions. Indeed, using undiscounted tracking MPC (13) to capture the optimal action-value function $Q_\gamma^\star$ is key to establishing the SDSD conditions.

Theorem 4 *Under Assumptions 1-3, if there exists a* $\boldsymbol{\theta}^\star \in \Theta$ *such that:*

$$Q_{\boldsymbol{\theta}^\star}(\mathbf{s},\mathbf{a}) = Q^\star_\gamma(\mathbf{s},\mathbf{a}),\ \forall(\mathbf{s},\mathbf{a}) \in \mathbb{Z}, \tag{45}$$

then the storage function $\lambda_{\boldsymbol{\theta}^\star}$ *satisfies the SDSD conditions.*

Proof: First, one can observe that:

$$V_{\boldsymbol{\theta}^\star}(\mathbf{s}) = \min_{\mathbf{a}} Q_{\boldsymbol{\theta}^\star}(\mathbf{s},\mathbf{a}) = \min_{\mathbf{a}} Q^\star_\gamma(\mathbf{s},\mathbf{a}) = V^\star_\gamma(\mathbf{s}) \tag{46}$$

Then we show that the storage function $\lambda_{\boldsymbol{\theta}^\star}$ satisfies (32). Using (45):

$$\begin{aligned} & Q^\star_\gamma(\mathbf{s},\mathbf{a}) \\ & = Q_{\boldsymbol{\theta}^\star}(\mathbf{s},\mathbf{a}) \overset{(20b)}{=} -\lambda_{\boldsymbol{\theta}^\star}(\mathbf{s}) + \Lambda^N_{\boldsymbol{\theta}^\star}(\mathbf{s},\mathbf{a}) \\ & \overset{(19b)}{=} -\lambda_{\boldsymbol{\theta}^\star}(\mathbf{s}) + \hat{\ell}_{\boldsymbol{\theta}^\star}(\mathbf{s},\mathbf{a}) + \Psi^{N-1}_{\boldsymbol{\theta}^\star}(\mathbf{f}(\mathbf{s},\mathbf{a})) \\ & \overset{(21)}{\geq} -\lambda_{\boldsymbol{\theta}^\star}(\mathbf{s}) + \hat{\ell}_{\boldsymbol{\theta}^\star}(\mathbf{s},\mathbf{a}) + \Psi^{N}_{\boldsymbol{\theta}^\star}(\mathbf{f}(\mathbf{s},\mathbf{a})) \\ & \overset{(20a)}{=} -\lambda_{\boldsymbol{\theta}^\star}(\mathbf{s}) + \hat{\ell}_{\boldsymbol{\theta}^\star}(\mathbf{s},\mathbf{a}) + \lambda_{\boldsymbol{\theta}^\star}(\mathbf{f}(\mathbf{s},\mathbf{a})) + V_{\boldsymbol{\theta}^\star}(\mathbf{f}(\mathbf{s},\mathbf{a})) \\ & = -\lambda_{\boldsymbol{\theta}^\star}(\mathbf{s}) + \hat{\ell}_{\boldsymbol{\theta}^\star}(\mathbf{s},\mathbf{a}) + \lambda_{\boldsymbol{\theta}^\star}(\mathbf{f}(\mathbf{s},\mathbf{a})) + V^\star_\gamma(\mathbf{f}(\mathbf{s},\mathbf{a})) \end{aligned} \tag{47}$$

then we have:

$$\begin{aligned} & \ell(\mathbf{s},\mathbf{a}) + \gamma V^\star_\gamma(\mathbf{f}(\mathbf{s},\mathbf{a})) \overset{(29)}{=} Q^\star_\gamma(\mathbf{s},\mathbf{a}) \\ & \overset{(47)}{\geq} -\lambda_{\boldsymbol{\theta}^\star}(\mathbf{s}) + \hat{\ell}_{\boldsymbol{\theta}^\star}(\mathbf{s},\mathbf{a}) + \lambda_{\boldsymbol{\theta}^\star}(\mathbf{f}(\mathbf{s},\mathbf{a})) + V^\star_\gamma(\mathbf{f}(\mathbf{s},\mathbf{a})) \\ & \overset{(14)}{\geq} -\lambda_{\boldsymbol{\theta}^\star}(\mathbf{s}) + \alpha_1(\|\mathbf{s}-\mathbf{s}_e\|) + \lambda_{\boldsymbol{\theta}^\star}(\mathbf{f}(\mathbf{s},\mathbf{a})) + V^\star_\gamma(\mathbf{f}(\mathbf{s},\mathbf{a})) \end{aligned} \tag{48}$$

By rearranging the terms in (48), we have:

$$\ell(\mathbf{s},\mathbf{a}) + \lambda_{\boldsymbol{\theta}^\star}(\mathbf{s}) - \lambda_{\boldsymbol{\theta}^\star}(\mathbf{f}(\mathbf{s},\mathbf{a})) + (\gamma-1)V^\star_\gamma(\mathbf{f}(\mathbf{s},\mathbf{a})) \geq \alpha_1(\|\mathbf{s}-\mathbf{s}_e\|) \tag{49}$$

which results in (32). In order to show the discounted strict dissipation inequality, we use that $T_{\boldsymbol{\theta}}(\cdot) \geq 0$ in Assumption 3. Then Assumption 2 holds, cost function of $\Psi^N_{\boldsymbol{\theta}^\star}$ in (19a) will be non-negative, i.e., $\Psi^N_{\boldsymbol{\theta}^\star}(\cdot) \geq 0$ and :

$$\begin{aligned} 0 \leq \Psi^N_{\boldsymbol{\theta}^\star}(\mathbf{f}(\mathbf{s},\mathbf{a})) &= \lambda_{\boldsymbol{\theta}^\star}(\mathbf{f}(\mathbf{s},\mathbf{a})) + V_{\boldsymbol{\theta}^\star}(\mathbf{f}(\mathbf{s},\mathbf{a})) \\ &= \lambda_{\boldsymbol{\theta}^\star}(\mathbf{f}(\mathbf{s},\mathbf{a})) + V^\star_\gamma(\mathbf{f}(\mathbf{s},\mathbf{a})) \end{aligned} \tag{50}$$

By rearranging and multiplying both sides by the positive factor $1-\gamma$:

$$-(1-\gamma)V^\star(\mathbf{f}(\mathbf{s},\mathbf{a})) \leq (1-\gamma)\lambda_{\boldsymbol{\theta}^\star}(\mathbf{f}(\mathbf{s},\mathbf{a})) \tag{51}$$

or equivalently

$$(\gamma-1)V^\star(\mathbf{f}(\mathbf{s},\mathbf{a})) - \lambda_{\boldsymbol{\theta}^\star}(\mathbf{f}(\mathbf{s},\mathbf{a})) \leq -\gamma\lambda_{\boldsymbol{\theta}^\star}(f(\mathbf{s},\mathbf{a})) \tag{52}$$

By adding $\ell(\mathbf{s},\mathbf{a}) + \lambda_{\boldsymbol{\theta}^\star}(\mathbf{s})$ to the both sides of (52), we have:

$$\ell(\mathbf{s},\mathbf{a}) + \lambda_{\boldsymbol{\theta}^\star}(\mathbf{s}) - \gamma\lambda_{\boldsymbol{\theta}^\star}(\mathbf{f}(\mathbf{s},\mathbf{a})) \overset{(52)}{\geq} \ell(\mathbf{s},\mathbf{a}) + \lambda_{\boldsymbol{\theta}^\star}(\mathbf{s}) - \lambda_{\boldsymbol{\theta}^\star}(\mathbf{f}(\mathbf{s},\mathbf{a}))+$$
$$(\gamma-1)V^\star_\gamma(\mathbf{f}(\mathbf{s},\mathbf{a})) \overset{(49)}{\geq} \alpha_1(\|\mathbf{s}-\mathbf{s}_e\|) \tag{53}$$

which shows the discounted strictly dissipativity with respect to supply rate ℓ. ■

Theorem 4 states that for a discounted optimal control problem, a parameterized undiscounted tracking MPC-scheme can be used to capture the optimal-action value function $Q^\star_\gamma$, independent of the discount factor γ. Then if the parameterization is rich enough, for the optimal parameters $\boldsymbol{\theta}^\star$, the storage function $\lambda_{\boldsymbol{\theta}^\star}$ satisfies SDSD. In fact, the use of undiscounted parameterization (16) is the key to obtaining a storage function satisfying SDSD conditions.

The next section details the parameterization of stage costs $\hat{\ell}_{\boldsymbol{\theta}}$ to satisfy Assumption 2. We will introduce Q-learning as a practical way to attain the optimal parameters $\boldsymbol{\theta}^\star$ that satisfy conditions (23) and (45) asymptotically.

5 Practical Implementation

In this section, we first recall Q-learning, and detail how it can be used to find the optimal parameters $\boldsymbol{\theta}^\star$ that fulfill conditions (23) and (45). Then we present two methods to ensure that the parameterized stage cost $\hat{\ell}_{\boldsymbol{\theta}}$ is lower-bounded by a $\mathcal{K}_\infty$ function (see (14)) in the learning context.

5.1 Q-learning

In this section, we detail the use of Q-learning in order to approach conditions (23) and (45) in practice. Q-learning is a well-known RL method that attempts to capture optimal action-value function $Q^\star_\gamma$ (or $Q^\star$) via tuning the vector of parameters $\boldsymbol{\theta}$ of the function approximator $Q_{\boldsymbol{\theta}}$. For deterministic systems, Q-learning uses the following Least-Square (LS) problem for the parameters $\boldsymbol{\theta}$ (see e.g. [26]) in the discounted setting (and the undiscounted setting when $Q^\star_\gamma \leftarrow Q^\star$):

$$\min_{\boldsymbol{\theta}\in\Theta} \quad \mathbb{E}_{\mathbf{s}_0}\left[\frac{1}{M}\sum_{k=0}^{M-1}\left[Q_{\boldsymbol{\theta}}(\mathbf{s}_k,\mathbf{a}_k) - Q^\star_\gamma(\mathbf{s}_k,\mathbf{a}_k)\right]^2\right], \tag{54}$$

where M is the episode length at each RL-step and the expectation $\mathbb{E}_{\mathbf{s}_0}$ is taken over the initial conditions $\mathbf{s}_0$ if they are randomly distributed, or can represent fixed initial conditions. the input $\mathbf{a}_k$ is selected according to the corresponding parametric policy, defined in (30b), with the possible addition of small random exploration. Note that the random initial condition increases the visited state-input pair domain and results

in a better approximation in the Q-learning [41]. The solution of $\boldsymbol{\theta}$ in (54) yields the optimal parameters $\boldsymbol{\theta}^\star$ that asymptotically capture the optimal action-value function $Q_\gamma^\star$ (or $Q^\star$) in LS sense. For dissipative problems, the LS fitting in (54) will result in a zero cost (satisfying (23) and (45)) if the parameterization is rich enough, provided by universal costs in $Q_{\boldsymbol{\theta}}$, and the number of data M is large enough. Moreover, the data must cover the state input space sufficiently, e.g., using a random initial state. However, these conditions may not hold in practice. We will detail the case $Q_{\boldsymbol{\theta}^\star} \neq Q_\gamma^\star$ (or $Q_{\boldsymbol{\theta}^\star} \neq Q^\star$) later.

Temporal-Difference (TD) learning is a common way to tackle (54) [26]. More specifically, a basic TD-based learning step uses the following update rule for the parameters $\boldsymbol{\theta}$ at time instance k in the discounted setting (and the undiscounted setting when $\gamma = 1$):

$$\delta_k = \ell(\mathbf{s}_k, \mathbf{a}_k) + \gamma V_{\boldsymbol{\theta}}(\mathbf{s}_{k+1}) - Q_{\boldsymbol{\theta}}(\mathbf{s}_k, \mathbf{a}_k) \tag{55a}$$

$$\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k + \xi \delta_k \nabla_{\boldsymbol{\theta}} Q_{\boldsymbol{\theta}}(\mathbf{s}_k, \mathbf{a}_k) \tag{55b}$$

where scalar $\xi > 0$ is the learning step-size, δ_k is labelled the TD error and the gradient $\nabla_{\boldsymbol{\theta}}^\top Q_{\boldsymbol{\theta}}(\mathbf{s}_k, \mathbf{a}_k)$ is calculated at $\boldsymbol{\theta}_k$. This algorithm generates a sequence of the parameters $\boldsymbol{\theta}_k$ that converge to the parameters that have the best estimation of the exact optimal action-value function. From (29) and (45), one can easily verify that $\boldsymbol{\theta}^\star$ in Theorems 1 and 4 is a fixed point of (55), i.e., $\delta_k = 0$ for $\boldsymbol{\theta} = \boldsymbol{\theta}^\star$. The convergence conditions for the Q-learning method can be found in e.g., [42]. Note that there are more advanced methods to tackle (54) in the literature, but this aspect is not the focus of this paper [26].

5.2 Satisfaction of Assumption 2

In this section, we detail how to satisfy Assumption 2 while using the Q-learning method. In the following, we provide two techniques for the parameterization of tracking MPC (13) in the Q-learning context in order to ensure that the stage cost $\hat{\ell}_{\boldsymbol{\theta}}$ remains lower-bounded by a $\mathcal{K}_\infty$ function.

I. Satisfaction of Assumption 2 by construction

In this section, we propose a generic parameterization of the stage cost $\hat{\ell}_{\boldsymbol{\theta}}$ that satisfies Assumption 2 by construction for all parameters $\boldsymbol{\theta}$.

A well-known method to provide a universal function approximation is to use Neural Networks (NNs) and Deep Neural Networks (DNNs). We parameterize the stage cost $\hat{\ell}_{\boldsymbol{\theta}}$ such that it is zero at steady-state and strictly positive otherwise for all $\boldsymbol{\theta} \in \Theta$. More specifically, let us consider the following parameterization:

$$\hat{\ell}_{\boldsymbol{\theta}}(\mathbf{s}, \mathbf{a}) = N_{\boldsymbol{\theta}}^2(\mathbf{s} - \mathbf{s}_e, \mathbf{a} - \mathbf{a}_e) + \epsilon \|\mathbf{s} - \mathbf{s}_e\| \tag{56}$$

where ϵ is a small enough positive constant and $N_{\boldsymbol{\theta}}$ is an NN-based function with weights $\boldsymbol{\theta}$ such that:

$$N_{\boldsymbol{\theta}}(\mathbf{0}, \mathbf{0}) = 0 \tag{57}$$

holds. In order to satisfy (57) by construction, one can use an activation function that has zero output for zero inputs without bias neurons in the hidden layers.

The next lemma expresses the universal function approximator theory that we will use in the next theorem.

Lemma 2 ***(Universal function approximator)*** *A standard multilayer feedforward network $N_{\boldsymbol{\theta}}(\mathbf{r})$ can approximate any continuous function $g(\mathbf{r})$ on the compact set $\mathbb{Z}$ arbitrarily accurately with respect to the uniform distance, provided that sufficiently many hidden units are available, and if a continuous, bounded and non-constant activation function is used. More specifically, there exists an NN-based function $N_{\boldsymbol{\theta}}(\mathbf{r})$ with weights $\boldsymbol{\theta}$ such that:*

$$\Big\| g(\mathbf{r}) - N_{\boldsymbol{\theta}}(\mathbf{r}) \Big\|_p := \left[\int_{\mathbb{Z}} \Big| g(\mathbf{r}) - N_{\boldsymbol{\theta}}(\mathbf{r}) \Big|^p \mathrm{d}\mathbf{r} \right]^{\frac{1}{p}} \leq \epsilon_1 \tag{58}$$

for all $1 \leq p < \infty$ and all $\epsilon_1 > 0$, where $\mathbf{r}$ is the input of the functions.

Proof: The proof can be found e.g. in Theorem 2 of [43]. ■

The following Lemma will be used in the next theorem.

Lemma 3 *For every functions f and g, the following inequality holds for all $1 \leq p < \infty$:*

$$\|f^2 - g^2\|_p \leq \|f - g\|_{2p}^2 + 2\|f\|_{2p}\|f - g\|_{2p} \tag{59}$$

Proof: From the Hölder's inequality, we have:

$$\|f^2 - g^2\|_s \leq \|f - g\|_q \|f + g\|_r \tag{60}$$

for $1/s = 1/q + 1/r$. Using (60) for $q = r = 2p$ and $s = p$, we have:

$$\|f^2 - g^2\|_p \leq \|f - g\|_{2p} \|f + g\|_{2p} \tag{61}$$

then from the Minkowski inequality, (61) reads as:

$$\begin{aligned} \|f - g\|_{2p}\|f + g\|_{2p} &= \|f - g\|_{2p}\|g - f + 2f\|_{2p} \\ &\leq \|f - g\|_{2p}(\|g - f\|_{2p} + 2\|f\|_{2p}) = \|f - g\|_{2p}^2 + 2\|f\|_{2p}\|f - g\|_{2p} \end{aligned} \tag{62}$$

Note that we have used that $\|\alpha f\|_p = |\alpha| \|f\|_p$ for $\alpha \in \mathbb{R}$. ■

Next theorem states that the stage cost $\hat{\ell}_{\boldsymbol{\theta}}$, defined in (56), provides a universal approximation of the stage costs that satisfy Assumption 2.

Theorem 5 *1. If* (57) *holds, the parameterized stage cost* $\hat{\ell}_{\boldsymbol{\theta}}(\mathbf{s},\mathbf{a})$ *in the form of* (56) *satisfies Assumption 2.*

2. For every continuous stage costs $\hat{\ell}:\mathbb{X}\times\mathbb{U}\to\mathbb{R}$ *that satisfy the following conditions:*

$$\alpha_1(\|\mathbf{s}-\mathbf{s}_e\|)\leq\hat{\ell}(\mathbf{s},\mathbf{a}),\quad 0=\hat{\ell}(\mathbf{s}_e,\mathbf{a}_e),\qquad\forall(\mathbf{s},\mathbf{a})\in\mathbb{Z},\tag{63}$$

for some $\alpha_1\in\mathcal{K}_\infty$ *with compact set* $\mathbb{Z}$ *and a given* $(\mathbf{s}_e,\mathbf{a}_e)$*, there exists an NN-based function* $N_{\boldsymbol{\theta}}$*, parameters* $\boldsymbol{\theta}$ *and* ϵ *such that:*

$$\Big\|\hat{\ell}(\mathbf{s},\mathbf{a})-\underbrace{(N^2_{\boldsymbol{\theta}}(\mathbf{s}-\mathbf{s}_e,\mathbf{a}-\mathbf{a}_e)+\epsilon\|\mathbf{s}-\mathbf{s}_e\|)}_{\overset{(56)}{=}\hat{\ell}_{\boldsymbol{\theta}}(\mathbf{s},\mathbf{a})}\Big\|_p\leq\epsilon_0\tag{64}$$

for all $\epsilon_0>0$, $\forall(\mathbf{s},\mathbf{a})\in\mathbb{Z}$ *and* $1\leq p\leq\infty$.

Proof: First statement can be easily verified by choosing $\alpha_1(\|\mathbf{s}-\mathbf{s}_e\|)=\epsilon\|\mathbf{s}-\mathbf{s}_e\|$ and using $N^2_{\boldsymbol{\theta}}(\cdot,\cdot)\geq 0$ and $N^2_{\boldsymbol{\theta}}(\mathbf{0},\mathbf{0})=0$. For the second statement, we define function $g(\mathbf{s},\mathbf{a}):=\sqrt{\hat{\ell}(\mathbf{s},\mathbf{a})}$. Note that this function exists and it is continuous because the function $\hat{\ell}(\mathbf{s},\mathbf{a})$ is non-negative and continuous. One can observe that $g(\mathbf{s}_e,\mathbf{a}_e)=0$. Then using Lemma 2, there exists an NN-based function $N_{\boldsymbol{\theta}}$ that satisfies:

$$\Big\|g(\mathbf{s},\mathbf{a})-N_{\boldsymbol{\theta}}(\mathbf{s}-\mathbf{s}_e,\mathbf{a}-\mathbf{a}_e)\Big\|_{2p}\leq\epsilon_1\tag{65}$$

for some parameters $\boldsymbol{\theta}$, all $(\mathbf{s},\mathbf{a})\in\mathbb{Z}$ and all $\epsilon_1>0$, where $N_{\boldsymbol{\theta}}(\mathbf{0},\mathbf{0})=0$. By substitution of (56) in (64), we have:

$$\begin{aligned}&\Big\|g^2(\mathbf{s},\mathbf{a})-N^2_{\boldsymbol{\theta}}(\mathbf{s}-\mathbf{s}_e,\mathbf{a}-\mathbf{a}_e)-\epsilon\|\mathbf{s}-\mathbf{s}_e\|\Big\|_p\leq\\&\Big\|g^2(\mathbf{s},\mathbf{a})-N^2_{\boldsymbol{\theta}}(\mathbf{s}-\mathbf{s}_e,\mathbf{a}-\mathbf{a}_e)\Big\|_p+\epsilon\Big\|\|\mathbf{s}-\mathbf{s}_e\|\Big\|_p\leq\\&\Big\|g(\mathbf{s},\mathbf{a})-N_{\boldsymbol{\theta}}(\mathbf{s}-\mathbf{s}_e,\mathbf{a}-\mathbf{a}_e)\Big\|^2_{2p}+\epsilon\Big\|\|\mathbf{s}-\mathbf{s}_e\|\Big\|_p+\\&2\Big\|g(\mathbf{s},\mathbf{a})\Big\|_{2p}\Big\|g(\mathbf{s},\mathbf{a})-N_{\boldsymbol{\theta}}(\mathbf{s}-\mathbf{s}_e,\mathbf{a}-\mathbf{a}_e)\Big\|_{2p}\end{aligned}\tag{66}$$

Note that we have used the Minkowski inequality in the first inequality and Lemma 3 in the second inequality. From Assumption (1), there exist positive constants M_0 and L_0 such that:

$$\Big\|g(\mathbf{s},\mathbf{a})\Big\|_{2p}\leq M_0,\qquad\Big\|\|\mathbf{s}-\mathbf{s}_e\|\Big\|_p\leq L_0,\tag{67}$$

for all $(\mathbf{s},\mathbf{a})\in\mathbb{Z}$. From (65) and (67), (66) reads:

$$\begin{aligned}&\Big\|g(\mathbf{s},\mathbf{a})-N_{\boldsymbol{\theta}}(\mathbf{s}-\mathbf{s}_e,\mathbf{a}-\mathbf{a}_e)\Big\|^2_{2p}+\epsilon\Big\|\|\mathbf{s}-\mathbf{s}_e\|\Big\|_p+\\&2\Big\|g(\mathbf{s},\mathbf{a})\Big\|_{2p}\Big\|g(\mathbf{s},\mathbf{a})-N_{\boldsymbol{\theta}}(\mathbf{s}-\mathbf{s}_e,\mathbf{a}-\mathbf{a}_e)\Big\|_{2p}\leq\epsilon_p^2+\epsilon L_0+2\epsilon_1M_0\end{aligned}\tag{68}$$

For a given ϵ_0, we select ϵ_1 and ϵ such that the following inequalities hold:

$$\epsilon_1 \leq \sqrt{\frac{\epsilon_0}{2} + M_0^2} - M_0, \qquad \epsilon \leq \frac{\epsilon_0}{2L_0} \tag{69}$$

Then (69) reads the following inequalities:

$$\epsilon_1 + M_0 \leq \sqrt{\frac{\epsilon_0}{2} + M_0^2} \Rightarrow \epsilon_1^2 + 2\epsilon_1 M_0 \leq \frac{\epsilon_0}{2} \tag{70a}$$

$$\epsilon L_0 \leq \frac{\epsilon_0}{2} \tag{70b}$$

Summing inequalities (70a) and (70b) yields:

$$\epsilon_1^2 + \epsilon L_0 + 2\epsilon_1 M_0 \leq \epsilon_0 \tag{71}$$

Then (66), (68) and (71) result in (64). ■

Note that (56) is one way of representing the functions that are lower-bounded by a $\mathcal{K}_\infty$ function. A more generic representation can be written as follows:

$$\hat{\ell}_{\boldsymbol{\theta}}(\mathbf{s}, \mathbf{a}) = \rho(|N_{\boldsymbol{\theta}}(\mathbf{s} - \mathbf{s}_e, \mathbf{a} - \mathbf{a}_e)|) + \epsilon \|\mathbf{s} - \mathbf{s}_e\| \tag{72}$$

where ρ is an arbitrary class $\mathcal{K}_\infty$ function.

II. Ensuring Assumption 2 by constrained RL steps

In this method, we assume that the parameterized stage cost $\hat{\ell}_{\boldsymbol{\theta}}$ is polynomial, and in order to enforce Assumption 2, at each learning step, we use Sum-of-Squares (SOS) programming. More specifically, for a polynomial parameterized stage cost $\hat{\ell}_{\boldsymbol{\theta}}$, the positivity of the stage cost can be represented as the following linear matrix inequality:

$$0 \prec R(\boldsymbol{\theta}_k), \quad \forall k \in \mathbb{I}_{\geq 0} \tag{73}$$

where $\hat{\ell}_{\boldsymbol{\theta}}(\mathbf{s}, \mathbf{a})$ has the following form:

$$\hat{\ell}_{\boldsymbol{\theta}}(\mathbf{s}, \mathbf{a}) = \mathbf{m}^\top(\mathbf{s} - \mathbf{s}_e, \mathbf{a} - \mathbf{a}_e) R(\boldsymbol{\theta}_k) \mathbf{m}(\mathbf{s} - \mathbf{s}_e, \mathbf{a} - \mathbf{a}_e) \tag{74}$$

and where $\mathbf{m}$ is a vector of monomials of $\mathbf{s} - \mathbf{s}_e$ and $\mathbf{a} - \mathbf{a}_e$ without bias (constant) value to satisfy $\hat{\ell}_{\boldsymbol{\theta}}(\mathbf{s}_e, \mathbf{a}_e)$. In this case, Q-learning (54) can be seen as minimizing an LS cost subject to constraint (73) in the parameters, more specifically:

$$\min_{\boldsymbol{\theta} \in \Theta} \quad \mathbb{E}_{\mathbf{s}_0} \left[\frac{1}{M} \sum_{k=0}^{M-1} \left[Q_{\boldsymbol{\theta}}(\mathbf{s}_k, \mathbf{a}_k) - Q^\star(\mathbf{s}_k, \mathbf{a}_k) \right]^2 \right] \tag{75a}$$

$$\text{s.t.} \quad 0 \prec R(\boldsymbol{\theta}) \tag{75b}$$

Note that (75) provides the Q-learning steps that satisfy the requirement (73). In the TD-learning case, the update rule (55) can be seen as minimizing a quadratic cost subject to some constraints in the parameters taking the form of the following Semi-Definite Program (SDP):

$$\min_{\Delta\boldsymbol{\theta}} \quad \frac{1}{2}\|\Delta\boldsymbol{\theta}\|^2 - \xi\delta_k\nabla_{\boldsymbol{\theta}}^\top Q_{\boldsymbol{\theta}}(\mathbf{s}_k,\mathbf{a}_k)\Delta\boldsymbol{\theta} \tag{76a}$$

$$\text{s.t.} \quad 0 \prec R(\Delta\boldsymbol{\theta}+\boldsymbol{\theta}_k) \tag{76b}$$

where the gradient $\nabla_{\boldsymbol{\theta}}^\top Q_{\boldsymbol{\theta}}(\mathbf{s}_k,\mathbf{a}_k)$ is calculated at $\boldsymbol{\theta}_k$. Then the parameter updates are then obtained from $\boldsymbol{\theta}_{k+1} = \Delta\boldsymbol{\theta}+\boldsymbol{\theta}_k$. In fact, if constraint (76b) is inactive, then (76) will be equivalent to (55), otherwise, SDP (76) delivers parameter steps that enforce constraint (73).

Note that using an SOS method in the Q-learning steps has less complexity and more accuracy in the storage function than when solving (5) directly using the SOS method as is proposed in, e.g., [19]. Indeed, to solve dissipativity (5) using SOS one needs to approximate the exact stage cost ℓ and the dynamics $\mathbf{f}$ by polynomials, and the obtained storage function is based of these approximations and may have an error compared to the exact storage function [19]. However, to solve (75), one only needs to provide a polynomial MPC stage cost $\hat{\ell}_{\boldsymbol{\theta}}$ in the function approximator, while the terminal cost and the storage function have not such requirement. Using a generic approximator for the storage function and terminal cost, Q-learning will be able to find the best polynomial stage cost and capture the optimal action-value function. Indeed, Unlike SOS method, in this method, the exact stage cost ℓ and dynamics $\mathbf{f}$ do not need to be approximated by polynomials.

We ought to stress here that in this paper, we have not enforced convexity for the parameterized action-value function resulting from MPC scheme (13) with respect to the parameter $\boldsymbol{\theta}$. Then Q-learning may converge to the local optimal parameters in set Θ. Providing a convex parameterized MPC scheme with respect to the parameters could be a direction of future investigations.

5.3 The case that (23) and (45) do not hold

In practice, if the parameterization is not rich or the problem is not dissipative (conditions (23) and (45) do not hold for any $\boldsymbol{\theta}$), $Q_{\boldsymbol{\theta}}$ will not necessarily converge to $Q^\star$. In this case, LS (54) does not yield a perfect fitting, and Q-learning will find the best parameters among the set of functions provided by the parameterization selected. Then one can check the dissipativity inequality (5) (or (31)) for the optimal learned parameters. If it holds, then the problem is dissipative, otherwise, the algorithm is inconclusive, i.e. either the parameterization is not rich enough or the problem is not dissipative. In order to check the dissipativity, we define the following auxiliary functions at the optimal parameters $\boldsymbol{\theta}^\star$ as follows:

$$\bar{\ell}^{\gamma}_{\boldsymbol{\theta}^\star}(\mathbf{s},\mathbf{a}) := \ell(\mathbf{s},\mathbf{a}) + \lambda_{\boldsymbol{\theta}^\star}(\mathbf{s}) - \gamma\lambda_{\boldsymbol{\theta}^\star}(\mathbf{f}(\mathbf{s},\mathbf{a})) \tag{77a}$$

$$H^{\gamma}_{\boldsymbol{\theta}^\star}(\mathbf{s},\mathbf{a}) := \ell(\mathbf{s},\mathbf{a}) + \lambda_{\boldsymbol{\theta}^\star}(\mathbf{s}) - \lambda_{\boldsymbol{\theta}^\star}(\mathbf{f}(\mathbf{s},\mathbf{a})) + (\gamma-1)V_{\boldsymbol{\theta}^\star}(\mathbf{f}(\mathbf{s},\mathbf{a})) \tag{77b}$$

$$\bar{\ell}_{\boldsymbol{\theta}^\star}(\mathbf{s},\mathbf{a}) := \ell(\mathbf{s},\mathbf{a}) + \lambda_{\boldsymbol{\theta}^\star}(\mathbf{s}) - \lambda_{\boldsymbol{\theta}^\star}(\mathbf{f}(\mathbf{s},\mathbf{a})) \tag{77c}$$

Then one can check if $\bar{\ell}^{\gamma}_{\boldsymbol{\theta}^\star}$ and $H^{\gamma}_{\boldsymbol{\theta}^\star}$ are lower-bounded by a $\mathcal{K}_\infty$ function, then the discounted problem is SDSD. Moreover, if $\bar{\ell}_{\boldsymbol{\theta}^\star}$ is lower-bounded by a $\mathcal{K}_\infty$ function, then the undiscounted problem is strictly dissipative. In order to check whether a function, defined in a compact set, is lower-bounded by a $\mathcal{K}_\infty$ function or not, one can grid the entire state-input space. If the function is continuous, zero at the steady state and strictly positive otherwise, then it is lower-bounded by a $\mathcal{K}_\infty$ function (see See Lemma A.1.2 of [44]).

Note that the method still delivers a stabilizing policy regardless of the dissipativity of the problem or the way of parameterization if Assumptions 1-3 hold.

Remark 1 *One of the best advantages of using the tracking MPC-scheme* (13) *as a function approximator is the closed-loop stability under the extracted parameterized policy* $\boldsymbol{\pi}_{\boldsymbol{\theta}}$*. In fact,* $\boldsymbol{\pi}_{\boldsymbol{\theta}}$*, defined in* (15)*, is a stabilizing policy for the closed-loop system* $\mathbf{f}(\mathbf{s},\boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathbf{s}))$ *for either discounted and undiscounted setting and for both dissipative and non-dissipative problem. For dissipative problems if* $\boldsymbol{\theta}^\star$ *satisfies* (45) *then* $\boldsymbol{\pi}_{\boldsymbol{\theta}^\star}(\mathbf{s}) = \boldsymbol{\pi}^\star_\gamma(\mathbf{s})$ *for discounted problem, because:*

$$\boldsymbol{\pi}_{\boldsymbol{\theta}^\star}(\mathbf{s}) = \arg\min_{\mathbf{a}} Q_{\boldsymbol{\theta}^\star}(\mathbf{s},\mathbf{a}) = \arg\min_{\mathbf{a}} Q^\star_\gamma(\mathbf{s},\mathbf{a}) = \boldsymbol{\pi}^\star_\gamma(\mathbf{s}) \tag{78}$$

and similar statements are valid for the undiscounted setting. Note that other function approximators, e.g., DNNs, for the optimal action-value function do not provide a stabilizing policy necessarily.

Remark 2 *For non-dissipative problems, although there exists no* $\boldsymbol{\theta}^\star$ *that captures the exact optimal-value function and optimal policy of the ENMPC-scheme, the resulting policy* $\boldsymbol{\pi}_{\boldsymbol{\theta}}$ *from the tracking MPC-scheme is a stabilizing policy for the closed-loop system* $\mathbf{f}(\mathbf{s},\boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathbf{s}))$ *(see Proposition 1). In fact, for non-dissipative problems the Q-learning algorithm is not able to deliver a perfect fitting of the optimal action-value function of the ENMPC scheme, and the resulting policy, at the convergence, is a sub-optimal policy* $\boldsymbol{\pi}_{\boldsymbol{\theta}^\star}(\mathbf{s}) \neq \boldsymbol{\pi}^\star_\gamma(\mathbf{s})$ *(or* $\boldsymbol{\pi}_{\boldsymbol{\theta}^\star}(\mathbf{s}) \neq \boldsymbol{\pi}^\star(\mathbf{s})$*) under the closed-loop stability constraints.*

The proposed approach for the discounted setting is summarized in the Algorithm 1. A similar algorithm can be used for the undiscounted setting. Note that for an undiscounted problem, we do not need to check SDSD, and the strict dissipativity property leads to asymptotic stability directly.

Algorithm 1: Q-learning to evaluate storage function and verify the dissipativity.

Input: Parameterize $\lambda_{\boldsymbol{\theta}}$, $T_{\boldsymbol{\theta}}$, and $\hat{\ell}_{\boldsymbol{\theta}}$ Initialize $\boldsymbol{\theta}_0$

while *converge* **do**
  Update $\boldsymbol{\theta}$ from (75) (or (76)) if $\hat{\ell}_{\boldsymbol{\theta}}$ if has a SOS form, or update from (54) (or (55)) if $\hat{\ell}_{\boldsymbol{\theta}}$ is lower-bounded by $\mathcal{K}_\infty$ by construction e.g. in form of (72)

if *Converge* **then**
  Compute rotated storage function $\bar{\ell}^{\gamma}_{\boldsymbol{\theta}^\star}$ and $H^{\gamma}_{\boldsymbol{\theta}^\star}$ from the learned storage function $\lambda_{\boldsymbol{\theta}^\star}$
  if $\bar{\ell}^{\gamma}_{\boldsymbol{\theta}^\star} > \epsilon\|\mathbf{s}-\mathbf{s}_e\|^2$ *for small enough* $\epsilon > 0$ **then**
    System is (discounted) strictly dissipative and $\lambda_{\boldsymbol{\theta}^\star}$ is a valid storage function
    if $H^{\gamma}_{\boldsymbol{\theta}^\star} > \epsilon\|\mathbf{s}-\mathbf{s}_e\|^2$ *for small enough* $\epsilon > 0$ **then**
      System (1) is SDSD and the closed-loop is asymptotically stable under optimal policy $\boldsymbol{\pi}^\star_\gamma$.
  else
    Inconclusive
else
  Inconclusive

6 Simulation

In this section, we provide four numerical examples in order to illustrate the efficiency of the proposed method.

6.1 The LQR case

First, we look at a trivial Linear dynamics, Quadratic stage cost, regulator (LQR) problem with an indefinite stage cost. LQR is a well-known problem because the exact optimal policy and the optimal value functions are obtainable using other techniques, e.g., the Riccati equation. Consider the following linear dynamics with a quadratic economic stage cost:

$$s_{k+1} = 2s_k - a_k\,, \quad L(s,a) = s^2 + a^2 + 4sa \not\geqslant 0 \tag{79}$$

The optimal steady state is $(s_s, a_s) = (0,0)$. We consider $\mathbb{X} = \mathbb{U} = [-\zeta, \zeta]$ with a large enough constant ζ. Note that these constraints satisfy assumption 1. The storage function $\lambda_{\boldsymbol{\theta}}$, terminal cost $T_{\boldsymbol{\theta}}$ and stage cost approximations $\hat{\ell}_{\boldsymbol{\theta}}$ are selected

as follows:

$$\lambda_{\boldsymbol{\theta}}(s) = \theta_1 s^2\,, \quad T_{\boldsymbol{\theta}}(s) = \theta_2 s^2 \tag{80a}$$

$$\hat{\ell}_{\boldsymbol{\theta}}(s) = \begin{bmatrix} s \\ a \end{bmatrix}^\top \begin{bmatrix} \theta_3 & \theta_4 \\ \theta_5 & \theta_6 \end{bmatrix} \begin{bmatrix} s \\ a \end{bmatrix} \tag{80b}$$

where $\boldsymbol{\theta} = \{\theta_1, \ldots, \theta_6\}$ is the set of parameters adjusted by Q-learning. The RL steps are restricted to making $\hat{\ell}_{\boldsymbol{\theta}}$ lower-bounded by a $\mathcal{K}_\infty$ function. Moreover, $T_{\boldsymbol{\theta}}$ is restricted to provide positive values, i.e., $\theta_2 > 0$.

We initialize $\boldsymbol{\theta}_0 = [0.1, 1, 1, 0, 0, 0]^\top$. Fig. 1 shows the convergence of the parameters resulting from Q-learning during 50 episodes. As can be seen in Fig. 2, after 50 iterations Q-learning is able to capture the optimal value and optimal policy functions. The learned storage function is $\lambda_{\boldsymbol{\theta}^\star}(s) = -0.4456 s^2$. From (77c) the learned rotated

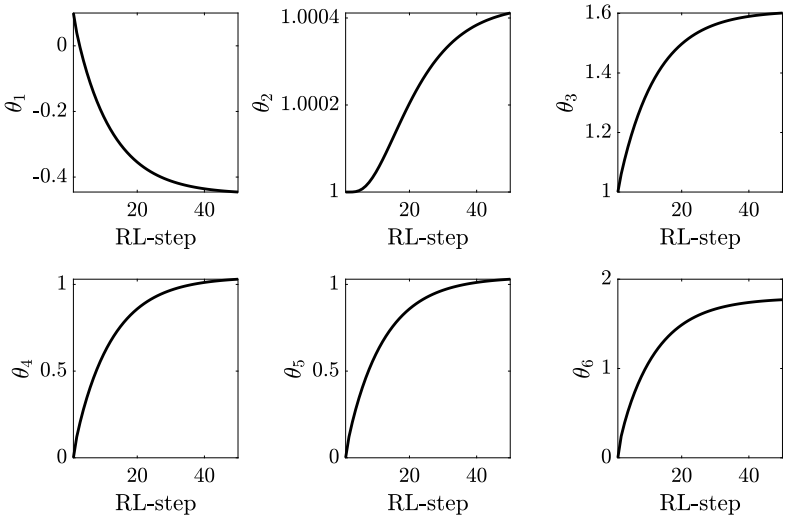


Figure 1: Parameters update using Q-learning.

stage cost satisfies the strict dissipativity inequality:

$$\bar{\ell}_{\boldsymbol{\theta}^\star}(s,a) = s^2 + a^2 + 4sa - 0.4456 s^2 + 0.4456(2s - a)^2 \geq \rho s^2 \tag{81}$$

for $0 < \rho \leq 2.3286$.

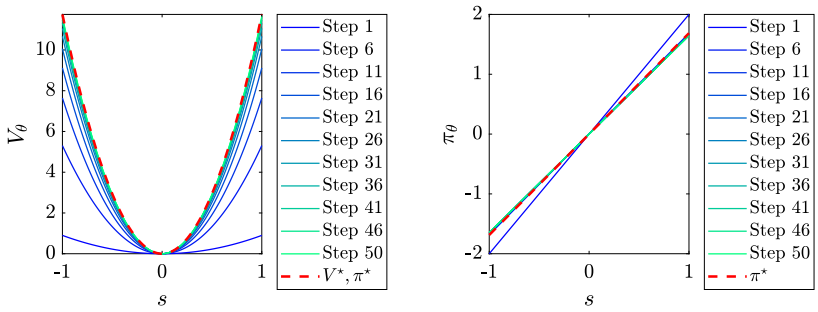


Figure 2: (Left) Value function and (Right) Policy function during the learning.

6.2 Non-dissipative dynamics

We provide next an example of non-dissipative dynamics and stage cost. Consider the following dynamics and economic stage cost:

$$s_{k+1} = a_k\,, \quad L(s,a) = -2s^2 + a^2 + sa + s^2a^2 \tag{82}$$

the optimal steady-state is $(s_e, a_e) = (0,0)$. In the Appendix, it is shown that there is no storage function for this example, that satisfies (5) with respect to supply rate L, i.e., (82) is non-dissipative. Similar to the previous example, a quadratic stage cost, terminal cost, and storage function with adjustable parameters are used for the simulation. Fig. 3 shows the learned rotated stage cost after convergence. It can be seen that Q-learning does not manage to learn a positive definite rotated stage cost. Note that since the tracking MPC scheme satisfies the stability conditions, then the policy provided by the MPC scheme is stabilizing for all parameters $\boldsymbol{\theta}$. Q-learning will find the best local parameters such that the resulting action-value function has the minimum error with respect to the optimal action-value function in the LS sense [31].

6.3 Non-polynomial case

In this example, we consider the non-polynomial case. Consider the following dynamics with a non-polynomial economic stage cost:

$$s_{k+1} = a_k\,, L(s,a) = -\ln(5s^{0.34} - a), \tag{83}$$

and the following compact state and input spaces:

$$\mathbb{X} = [0, 10],\ \mathbb{U} = [0.01, 10]\ . \tag{84}$$

This model is a benchmark optimal investment problem, where s denotes the investment in a company and the term $5s^{0.34}$ is the return from this investment after one

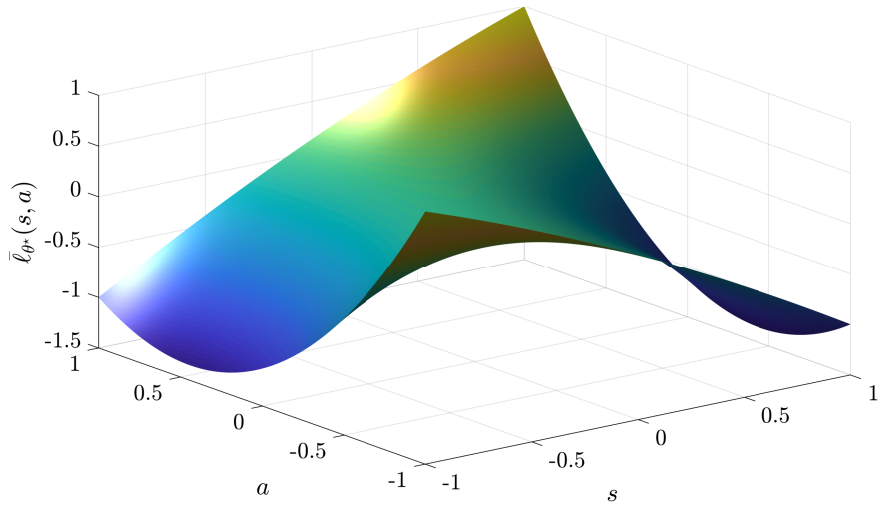


Figure 3: Rotated stage cost according to the learned storage function.

period. Then $5s^{0.34} - a$ is the amount of money that can be used for consumption in the current time period. Then the objective is to maximize the sum of the logarithmic utility function. The optimal steady-state point is $s_e = a_e = 2.2344$. In [19], it is shown that a storage function in the form

$$\lambda_\theta(s) = \theta(s - s_e), \tag{85}$$

is valid for this problem and the SOS approach delivers an approximated storage function ($\theta = 0.23$) using an order-3 Taylor approximation for the stage cost. The analytical solution can be obtained for θ (see [19]). We use the approximated storage function obtained from SOS as an initial guess for $\boldsymbol{\lambda_\theta}$ and apply the proposed learning method. The following stage cost is used in the simulation:

$$\hat{\ell}_\theta(s,a) = L(s,a) - L(s_e,a_e) + \theta(s-a)\,, \tag{86}$$

and we use a long horizon $N = 100$ without terminal cost. Fig. 4 shows the update of the parameter θ over the 100 episodes. As can be seen, the learning-based storage function converges to the analytical solution, while the parameter θ resulting from the SOS method has a bias. As it can be seen, the learning method can improve the storage function estimation by about 60% after 10 learning steps with respect to the Sum-of-Square method, while the improvement is 99.5% after 50 learning steps. Note that this example is simple and the bias issue is not significant, but for a more complex problem in practice, the SOS method may lead to a storage function that has a significant bias with the true storage function. Fig. 5 illustrates the rotated stage cost from the learned storage function.

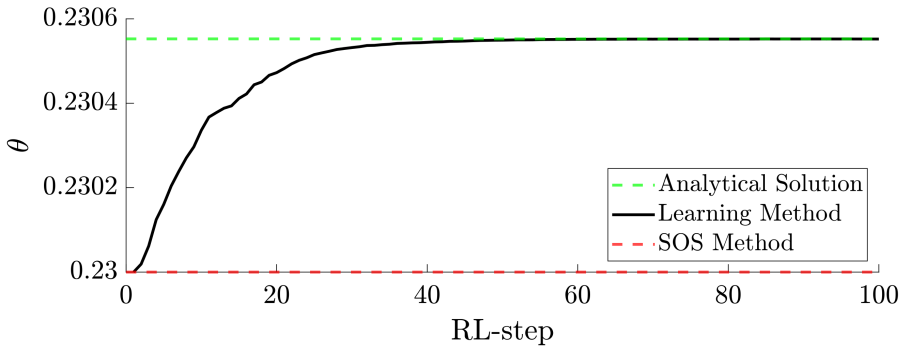


Figure 4: Learning of the storage function parameter.

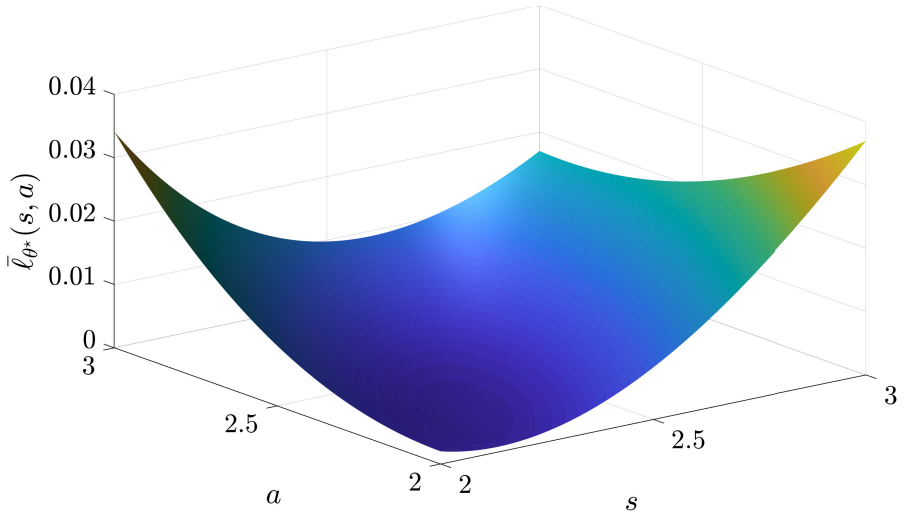


Figure 5: Rotated stage cost according to the learned storage function.

The discounted setting for (83) with discount factor γ has the following analytical storage function solution [40]:

$$\lambda(s) = \frac{0.34\ {}^{-0.66}\sqrt{1.7\gamma}}{1 - 0.34\gamma}(s - s_e) \tag{87}$$

where:

$$s_e = a_e = {}^{0.66}\sqrt{1.7\gamma} \tag{88}$$

is the optimal steady-state state. Fig. 6 shows the convergence of the parameter of

storage function $\lambda_\theta(s) = \theta(s - s_e)$ for discounted setting with $\gamma = 0.99$. Fig. 7 (right)

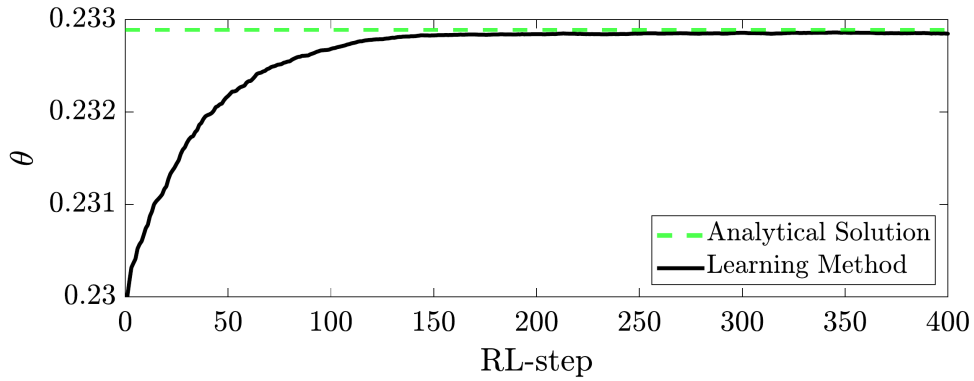


Figure 6: Learning of the storage function parameter for the discounted setting.

shows the rotated stage cost and verifies the discounted strictly dissipativity. Fig. 7 (left) verifies the second SDSD condition.

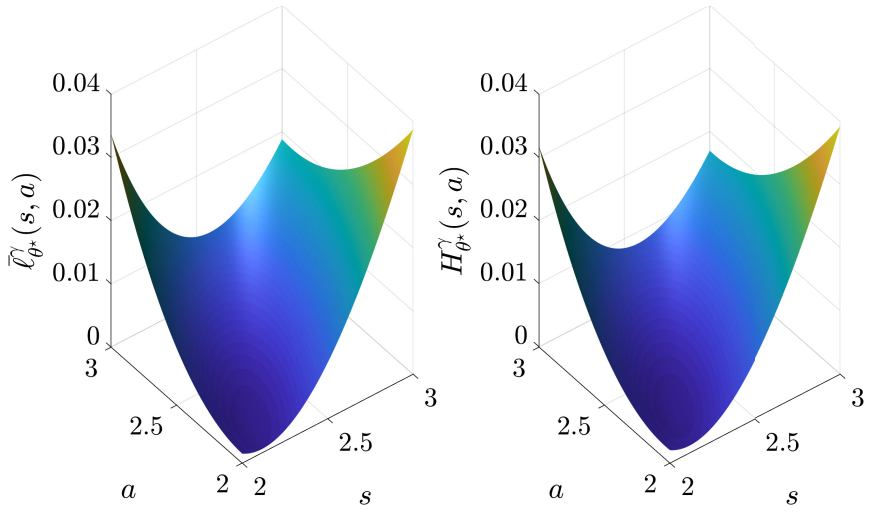


Figure 7: Right: Discounted rotated stage cost, Left: SDSD condition.

6.4 CSTR process

We next provide a nonlinear numerical example in the chemical engineering context. Continuously Stirred Tank Reactor (CSTR) is a common ideal reactor in chemical engineering, usually used for liquid-phase or multiphase reactions with fairly high reaction rates. We consider a non-isothermal reactor where an elementary, exothermic second-order reaction takes place that, converting the reactant A to the desired product B (see Fig. 8).

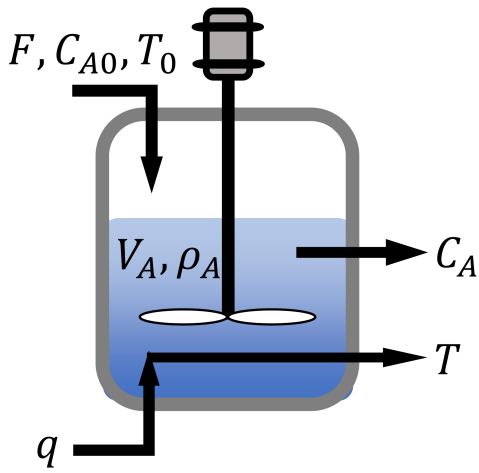


Figure 8: The Continuously Stirred Tank Reactor.

The CSTR nonlinear dynamics can be written as follows [45]:

$$\begin{aligned} \dot{C}_A &= \frac{F}{V_R}(C_{A0} - C_A) - k_0 e^{-E/RT} C_A^2 \\ \dot{T} &= \frac{F}{V_R}(T_0 - T) - \frac{\Delta H k_0}{\rho_R C_p} e^{-E/RT} C_A^2 + \frac{q}{\rho_R C_p V_R}, \end{aligned} \tag{89}$$

where T denotes the temperature of the reactor contents, C_A is the concentration of A in the reactor, F is the flow rate, and q is the heat rate. The remaining notation definitions and process parameter values are given in Table 1.

Then $\mathbf{s} = [C_A\,,\, T]^\top$ and $\mathbf{a} = [F\,,\, q]^\top$ is the state and the input of the system, respectively. The input $\mathbf{a}$ satisfies the following inequality:

$$[0\,,\, -2\text{e}5]^\top \leq \mathbf{a} \leq [10\,,\, 2\text{e}5]^\top \tag{90}$$

An economic stage cost is defined as follows:

$$L = -\alpha \underbrace{F(C_{A0} - C_A)}_{:=r} + \beta q \tag{91}$$

Table 1: Parameter definitions and values of CSTR.

Symbol	Description	Value
C_{A0}	Feed concentration of A	$3.5kmol/m^3$
T_0	Feedstock temperature	$300K$
V_R	Reactor fluid volume	$1.0m^3$
E	Activation energy	$5.0e4kJ/kmol$
k_0	Pre-exponential rate factor	$8.46e6m^3/kmolh$
ΔH	Reaction enthalpy change	$-1.16e4kJ/kmol$
C_p	Heat capacity	$0.231kJ/kgK$
ρ_R	Density	$1000kg/m^3$
R	Gas constant	$8.314kJ/kmolK$

where α and β are positive constant, and r is the production rate. This cost maximizes the production rate and minimizes the energy consumption of the production (the second term). For $\alpha = 1.7e4$ and $\beta = 1$ the production rate and energy consumption will be almost balanced. Sampling time 0.02h is used to discretize the system (89) to the form (1). Using (3), the optimal steady-state pair is:

$$\mathbf{s}_e = [0.7572\,,\,497.71]^\top\,,\,\mathbf{a}_e = [10.00\,,\,1.38557e5]^\top \tag{92}$$

A Deep Neural Network (DNN) is used to approximate the storage function $\lambda_{\boldsymbol{\theta}}$, as shown in Fig. 9. This DNN is a fully connected Multi-Layer Perceptron (MLP) with tanh(.) as the activation functions.

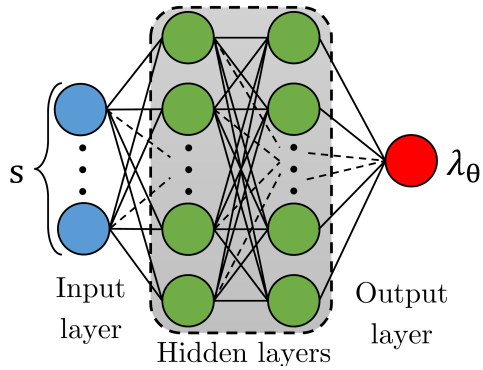


Figure 9: Approximation of the storage function.

Then, the input-output relation for the two hidden layers MLP can be written as follows:

$$\lambda_{\boldsymbol{\theta}}(\mathbf{s}) = W_2\left(\tanh\left(W_1\left(\tanh\left(W_0(\mathbf{s}-\mathbf{s}_e)\right)\right)\right)\right) \tag{93}$$

where W_0, W_1 and W_2 are the weight matrices with appropriate dimension and tanh is applied element-wise. Note that we used this activation function without bias

parameters to preserve $\lambda_{\boldsymbol{\theta}}(\mathbf{s}_e) = 0$. For the simulation, an MLP with two hidden layers and 16 neurons per hidden layer is used. The following stage and terminal cost are used in the simulation:

$$\hat{\ell}_{\boldsymbol{\theta}}(\mathbf{s}, \mathbf{a}) = \left\| \text{col}(\mathbf{s} - \mathbf{s}_e, \mathbf{a} - \mathbf{a}_e) \right\|_{W_\ell} + \epsilon \|\mathbf{s} - \mathbf{s}_e\| \tag{94a}$$

$$T_{\boldsymbol{\theta}}(\mathbf{s}) = \|\mathbf{s} - \mathbf{s}_e\|_{W_T} \tag{94b}$$

where ϵ is a small positive constant and matrices W_ℓ and W_T are the weights of the vector norm and MPC horizon is $N = 10$. Q-learning steps are restricted to deliver a positive definite W_ℓ and W_T using SDP. Then the parameters vector $\boldsymbol{\theta}$ read as:

$$\boldsymbol{\theta} = \{W_0, W_1, W_2, W_\ell, W_T\} \tag{95}$$

Note that in order to use the SOS method for this example, one needs to approximate the dynamics with polynomials. Then the dissipativity can be discussed locally only in a neighborhood of the optimal steady state. Moreover, it results in a polynomial storage function that satisfies dissipativity *approximately*. In contrast, our approach can deliver a DNN-based general storage function without approximating the dynamics.

Fig. 10 shows the trajectories of the states and inputs for the different episodes, while the initial state is chosen randomly. As can be seen, the states and inputs trajectories are stable and they converge to their steady-state point.

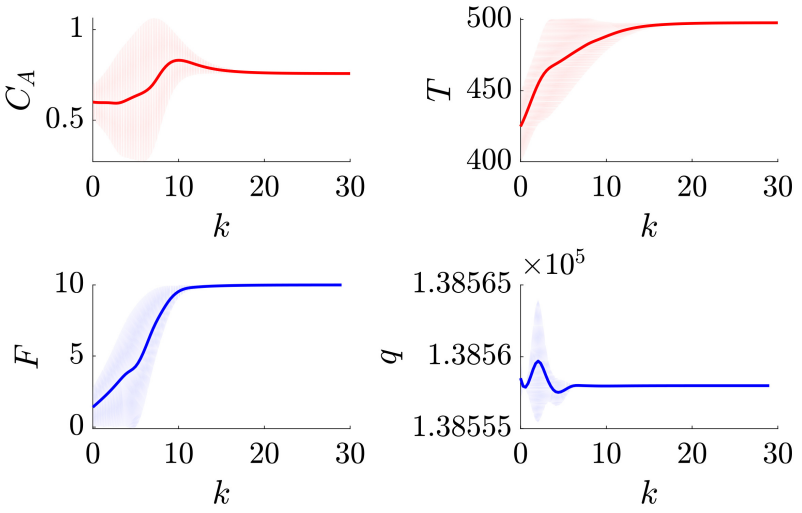


Figure 10: Time response of state-input for the different episodes starting from a random initial state.

Fig. 11 illustrates the closed-loop performance over RL steps. The closed-loop performance refers to the summation of stage costs over episodes. As the initial state

is selected randomly, the closed-loop performance is noisy. The moving average of the performance is improving during the learning and Q-learning tries to find the optimal parameters among the provided functions. Fig. 12 illustrates $\min \bar{\ell}_{\boldsymbol{\theta}}(\mathbf{s}, \mathbf{a})$ along the closed-loop trajectories over the learning steps. Note that min here is taken with respect to time at each episode, i.e., the minimum of observed stage cost at each episode, or RL-step, is stored. Then we plot these minimums over RL steps. As it can be seen, RL tries to push $\bar{\ell}_{\boldsymbol{\theta}}$ to be positive definite while in the first few steps $\bar{\ell}_{\boldsymbol{\theta}}$ is negative in some points. Note that, since the initial state is selected randomly, then these noises and fluctuations in $\min \bar{\ell}_{\boldsymbol{\theta}}$ are expected.

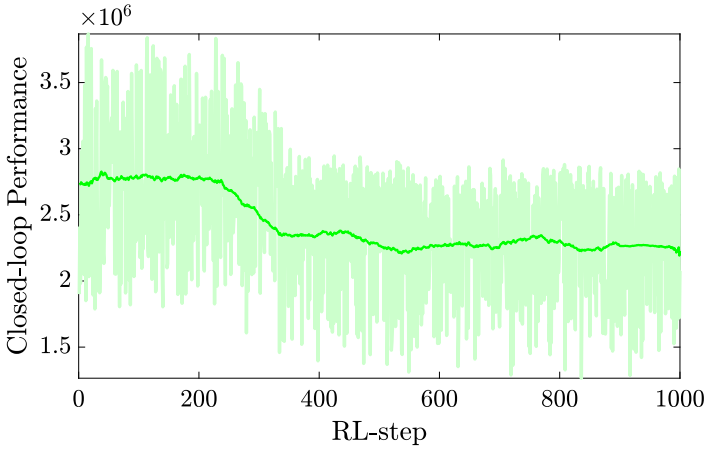


Figure 11: Closed-loop Performance over RL-steps.

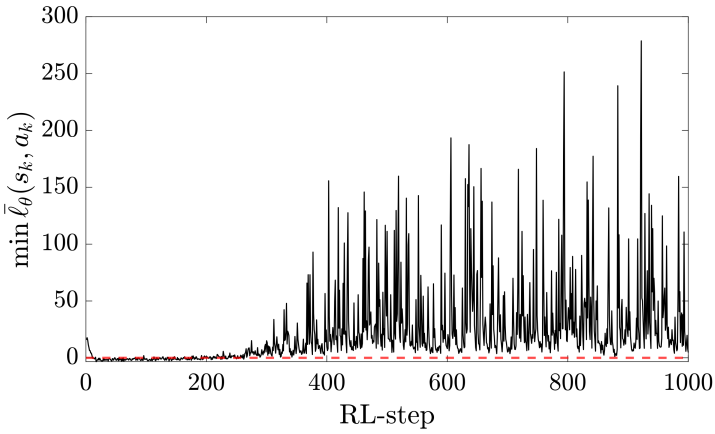


Figure 12: Minimum of the rotated stage cost along the closed-loop trajectories.

7 Discussion

In this paper, we presented the use of parameterized tracking MPC scheme with an additional parametrized storage function as a function approximator of the optimal action-value function for the ENMPC scheme. We enforced the closed-loop stability of the system for all sets of parameters by Assumptions 2 and 3. Therefore, this scheme provides a stabilizing policy for the closed-loop system regardless of whether the problem is dissipative or not and discounted or not. We used Q-learning to adjust the parameters toward the parameters that have the best approximation of the optimal action-value function. We applied the proposed method in the previous section.

First, an LQR example was considered. A parameterized quadratic stage cost, terminal cost, and storage function have been selected. The parameters are restricted to fulfill the stability assumptions. Therefore we solved a constrained Q-learning as detailed in (75) and (76). For an LQR problem, this parameterization is rich enough, and we showed that the Q-learning method is able to learn the optimal policy and action-value function. Moreover, the resulting storage function satisfies the strict dissipativity inequality.

Next, a non-dissipative problem was investigated. In this example, we showed that the proposed method delivers a rotated storage function that is not lower-bounded by a $\mathcal{K}_\infty$ function after the convergence in the learning.

Then we detailed a non-polynomial case where the SOS method struggles to deliver an accurate result. We showed that our method is able to learn the storage function more accurately than the SOS method in both discounted and undiscounted cases. In fact, in this method, we do not need to approximate system and stage costs by polynomials. One only may need to provide a parameterized stage cost in the function approximator to use the SOS method in the learning steps. Then a generic storage function and terminal cost would accurately approximate the optimal action-value function.

Finally, we considered a nonlinear chemical reactor system. This system was non-polynomial with an economic stage cost. We used neural networks in the stage cost, terminal cost, and storage function to approximate the optimal action-value function. We showed that the method is able to improve the closed-loop economic performance of the system. Moreover, the learning steps push the rotated stage cost to be lower-bounded by a $\mathcal{K}_\infty$ function. Note that the MPC scheme delivers a stabilizing policy during the learning.

8 Conclusion

This paper presented the use of tracking MPC-based function approximator of the optimal action-value function to evaluate the storage function and verify the dissipativity in the ENMPC for general discrete-time dynamics and cost for both discounted and undiscounted optimal control problems. We showed that, under some conditions, the MPC-based function approximator will be able to deliver a valid storage function for a dissipative problem at the optimal parameters. For non-dissipative problems, however, the resulting policy is stabilizing, but the tracking MPC scheme can not capture the optimal action-value function. Then we proposed the use of Q-learning to tune the parameters of the parameterized MPC. This method tries to estimate the optimal action-value function in the Least-Square sense. Moreover, we proposed two methods to ensure the tracking stage cost remains lower-bounded by a $\mathcal{K}_\infty$ function, in the learning context. The need for complex neural networks for general nonlinear problems and heavy computations, local optimum point issues for non-convex problems, and the need for an exact deterministic model of the system for theoretical analysis can be considered as limitations of the present work. The efficiency of the method was illustrated in the different case studies. Combining this method with SOS, addressing the convexity in the ENMPC scheme and global optimality of the Q-learning method, and applying it for the noisy data and stochastic systems can be considered in future research.

Acknowledgment

This work was funded by the Research Council of Norway (RCN) project “Safe Reinforcement Learning using MPC" (SARLEM).

References

[1] James B Rawlings and Rishi Amrit. “Optimizing process economic performance using model predictive control”. In: *Nonlinear model predictive control*. Springer, 2009, pp. 119–138.

[2] David Angeli, Rishi Amrit, and James B Rawlings. “On average performance and stability of economic model predictive control”. In: *IEEE transactions on automatic control* 57.7 (2011), pp. 1615–1626.

[3] Rishi Amrit, James B Rawlings, and David Angeli. “Economic optimization using model predictive control with a terminal cost”. In: *Annual Reviews in Control* 35.2 (2011), pp. 178–186.

[4] Mario Zanon, Sébastien Gros, and Moritz Diehl. "A Lyapunov function for periodic economic optimizing model predictive control". In: *52nd IEEE Conference on Decision and control*. IEEE. 2013, pp. 5107–5112.

[5] Timm Faulwasser, Lars Grüne, Matthias A Müller, et al. "Economic nonlinear model predictive control". In: *Foundations and Trends® in Systems and Control* 5.1 (2018), pp. 1–98.

[6] Gabriele Pozzato et al. "Economic MPC for online least costly energy management of hybrid electric vehicles". In: *Control Engineering Practice* 102 (2020), p. 104534.

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

[8] Lars Grüne and Jürgen Pannek. *Nonlinear model predictive control*. Springer, 2017.

[9] Matthew Ellis, Jinfeng Liu, and Panagiotis D Christofides. "Economic model predictive control". In: *Springer* 5.7 (2017), p. 65.

[10] MA Müller. "Dissipativity in economic model predictive control: beyond steady-state optimality". In: *Recent Advances in Model Predictive Control: Theory, Algorithms, and Applications* 485 (2021), p. 27.

[11] Moritz Diehl, Rishi Amrit, and James B Rawlings. "A Lyapunov function for economic optimizing model predictive control". In: *IEEE Transactions on Automatic Control* 56.3 (2010), pp. 703–707.

[12] Xiangjie Liu and Jinghan Cui. "Economic model predictive control of boiler-turbine system". In: *Journal of Process Control* 66 (2018), pp. 59–67.

[13] Lars Grüne and Marleen Stieler. "Asymptotic stability and transient optimality of economic MPC without terminal conditions". In: *Journal of Process Control* 24.8 (2014), pp. 1187–1196.

[14] Jan C Willems. "Dissipative dynamical systems part I: General theory". In: *Archive for rational mechanics and analysis* 45.5 (1972), pp. 321–351.

[15] Jan C Willems. "Dissipative dynamical systems part II: Linear systems with quadratic supply rates". In: *Archive for rational mechanics and analysis* 45.5 (1972), pp. 352–393.

[16] Lars Grüne and Matthias A Müller. "On the relation between strict dissipativity and turnpike properties". In: *Systems & Control Letters* 90 (2016), pp. 45–53.

[17] Timm Faulwasser et al. "Turnpike and dissipativity properties in dynamic real-time optimization and economic MPC". In: *53rd IEEE conference on decision and control*. IEEE. 2014, pp. 2734–2739.

[18] Julian Berberich et al. "Dissipativity properties in constrained optimal control: A computational approach". In: *Automatica* 114 (2020), p. 108840.

[19] Simon Pirkelmann, David Angeli, and Lars Grüne. "Approximate computation of storage functions for discrete-time systems using sum-of-squares techniques". In: *IFAC-PapersOnLine* 52.16 (2019), pp. 508–513.

[20] Carsten Scherer and Siep Weiland. "Linear matrix inequalities in control". In: *Lecture Notes, Dutch Institute for Systems and Control, Delft, The Netherlands* 3.2 (2015).

[21] Anne Koch, Julian Berberich, and Frank Allgöwer. "Provably robust verification of dissipativity properties from data". In: *IEEE Transactions on Automatic Control* 67.8 (2021), pp. 4248–4255.

[22] Mario Zanon, Sébastien Gros, and Moritz Diehl. "A tracking MPC formulation that is locally equivalent to economic MPC". In: *Journal of Process Control* 45 (2016), pp. 30–42.

[23] Anne Romer et al. "One-shot verification of dissipativity properties from input–output data". In: *IEEE Control Systems Letters* 3.3 (2019), pp. 709–714.

[24] Julian Berberich and Frank Allgöwer. "A trajectory-based framework for data-driven system analysis and control". In: *2020 European Control Conference (ECC)*. IEEE. 2020, pp. 1365–1370.

[25] TM Maupong, Jonathan C Mayo-Maldonado, and Paolo Rapisarda. "On Lyapunov functions and data-driven dissipativity". In: *IFAC-PapersOnLine* 50.1 (2017), pp. 7783–7788.

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

[27] Dimitri P Bertsekas. *Reinforcement learning and optimal control*. Athena Scientific Belmont, MA, 2019.

[28] Arash Bahari Kordabad et al. "Reinforcement learning based on scenario-tree MPC for ASVs". In: *2021 American Control Conference (ACC)*. IEEE. 2021, pp. 1985–1990.

[29] Hossein Nejatbakhsh Esfahani, Arash Bahari Kordabad, and Sébastien Gros. "Reinforcement learning based on MPC/MHE for unmodeled and partially observable dynamics". In: *2021 American Control Conference (ACC)*. IEEE. 2021, pp. 2121–2126.

[30] 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.

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

[32] Arash Bahari Kordabad and Sebastien Gros. "Verification of Dissipativity and Evaluation of Storage Function in Economic Nonlinear MPC using Q-Learning". In: *IFAC-PapersOnLine* 54.6 (2021). 7th IFAC Conference on Nonlinear Model Predictive Control NMPC 2021, pp. 308–313.

[33] D.Q. Mayne et al. "Constrained model predictive control: Stability and optimality". In: *Automatica* 36.6 (2000), pp. 789–814.

[34] Ali Jadbabaie and John Hauser. "On the stability of receding horizon control with a general terminal cost". In: *IEEE Transactions on Automatic Control* 50.5 (2005), pp. 674–678.

[35] John Rust. *Dynamic programming*. Vol. 1. London, UK, Palgrave Macmillan, Ltd, 2008.

[36] Dieter Grass et al. "Optimal control of nonlinear processes". In: *Berlino: Springer* (2008).

[37] Dimitri P Bertsekas et al. *Dynamic programming and optimal control: Vol. 1*. Athena scientific Belmont, 2000.

[38] Romain Postoyan et al. "Stability analysis of discrete-time infinite-horizon optimal control with discounted cost". In: *IEEE Transactions on Automatic Control* 62.6 (2016), pp. 2736–2749.

[39] Mario Zanon and Sébastien Gros. "A new dissipativity condition for asymptotic stability of discounted economic MPC". In: *Automatica* 141 (2022), p. 110287.

[40] Lars Grüne, Christopher M Kellett, and Steven R Weller. "On a discounted notion of strict dissipativity". In: *IFAC-PapersOnLine* 49.18 (2016), pp. 247–252.

[41] Feten Slimeni et al. "Jamming mitigation in cognitive radio networks using a modified Q-learning algorithm". In: *2015 International Conference on Military Communications and Information Systems (ICMCIS)*. IEEE. 2015, pp. 1–7.

[42] Christopher JCH Watkins and Peter Dayan. "Q-learning". In: *Machine learning* 8.3-4 (1992), pp. 279–292.

[43] Kurt Hornik. "Approximation capabilities of multilayer feedforward networks". In: *Neural networks* 4.2 (1991), pp. 251–257.

[44] Julian Berberich. "Indefinite linear quadratic optimal control: periodic dissipativity and turnpike properties". MA thesis. University of Stuttgart, 2018.

[45] Xinchun Li et al. "Application of economic MPC to a CSTR process". In: *2016 IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC)*. 2016, pp. 685–690.

Appendix. Non-dissipativity of the Example 6.2

Let us assume that the system-stage cost described in (82) is dissipative. Then there exists a storage function $\lambda(s)$ and $0 < \rho$ that satisfies the following inequality:

$$\lambda(a) - \lambda(s) \leq -2s^2 + a^2 + sa + s^2a^2 - \rho s^2, \tag{A.1}$$

and $\lambda(0) = 0$. For $s = 0$ we have:

$$\lambda(a) \leq a^2 \Rightarrow \lambda(s) \leq s^2, \tag{A.2}$$

where we changed the variable name $a \rightarrow s$ in the last inequality. For $a = 0$, (A.1) reads as:

$$-\lambda(s) \leq -2s^2 - \rho s^2 \xrightarrow{0<\rho} 2s^2 \leq \lambda(s), \tag{A.3}$$

the contradiction from (A.2) and (A.3) shows that the storage function does not exist and the system is non-dissipative.

H Equivalence of Optimality Criteria for Markov Decision Process and Model Predictive Control

Postprint of [101] **Arash Bahari Kordabad**, Mario Zanon, and Sebastien Gros. "Equivalence of Optimality Criteria for Markov Decision Process and Model Predictive Control". In: *arXiv preprint, Submitted* (2022). DOI: 10.48550/arXiv.2210.04302

Equivalence of Optimality Criteria for Markov Decision Process and Model Predictive Control

Arash Bahari Kordabad[1], Mario Zanon[2], and Sebastien Gros[1]

[1]Department of Engineering Cybernetics, Norwegian University of Science and Technology (NTNU), Trondheim, Norway.
[2]IMT School for Advanced Studies Lucca, Italy.

Abstract: This paper shows that the optimal policy and value functions of a Markov Decision Process (MDP), either discounted or not, can be captured by a finite-horizon undiscounted Optimal Control Problem (OCP), even if based on an inexact model. This can be achieved by selecting a proper stage cost and terminal cost for the OCP. A very useful particular case of OCP is a Model Predictive Control (MPC) scheme where a deterministic (possibly nonlinear) model is used to reduce the computational complexity. This observation leads us to parameterize an MPC scheme fully, including the cost function. In practice, Reinforcement Learning algorithms can then be used to tune the parameterized MPC scheme. We verify the developed theorems analytically in an LQR case and we investigate some other nonlinear examples in simulations.

Keywords: Markov Decision Process, Model Predictive Control, Reinforcement Learning, Optimality

1 Introduction

Markov Decision Processes (MDPs) provide a standard framework for the optimal control of discrete-time stochastic processes, where the stage cost and transition probability depend only on the current state and the current input of the system [1]. A control system, described by an MDP, receives an input at each time instance and proceeds to a new state with a given probability density, and in the meantime, it gets a stage cost at each transition. For an MDP, a policy is a mapping from the state space into the input space and determines how to select the input based on the observation of the current state. This policy can either be a deterministic mapping from the state space [2] or a conditional probability of the current state, describing the stochastic policy [3]. This paper focuses on deterministic policies. Solving an MDP refers to finding an optimal policy that minimizes the expected value of a total cumulative cost as a function of the current state. The cumulative cost can be either discounted or undiscounted with respect to the time instant. Therefore, different definitions for the cumulative cost yields different optimality criteria for the MDPs. Dynamic Programming (DP) techniques can be used to solve MDPs based on the Bellman equations. However, solving the Bellman equations

is typically intractable unless the problem is of very low dimension [4]. This issue is known as "curse of dimensionality" in the literature [5]. Besides, DP requires the exact transition probability of MDPs, while in most engineering applications, we do not have access to the exact probability transition of the real system.

Reinforcement Learning (RL) [6] and approximate DP [7] are two common techniques that tackle these difficulties. RL offers powerful tools for tackling MDP without having an accurate knowledge of the probability distribution underlying the state transition. In most cases, RL requires a function approximator to capture the optimal policy or the optimal value functions underlying the MDP. A common choice of function approximator in the RL community is to use a Deep Neural Network (DNN) [8]. DNNs can be used to capture either the optimal policy underlying the MDP directly or the action-value function from which the optimal policy can be indirectly extracted. However, the formal analysis of closed-loop stability and safety of the policies provided by approximators such as DNNs is challenging. Moreover, DNNs usually need a large number of tunable parameters and a pre-training is often required so that the initial values of the parameters are reasonable.

Model Predictive Control (MPC) is a well-known control strategy that employs a (possibly inaccurate) model of the real system dynamics to produce an input-state sequence over a given finite-horizon such that the resulting predicted state trajectory minimizes a given cost function while explicitly enforcing the input-state constraints imposed on the system trajectories [9]. For computational reasons, simple models are usually preferred in the MPC scheme. Hence, the MPC model often does not have the structure required to correctly capture the real system dynamics and stochasticity. The idea of using MPC as a function approximator for RL techniques was justified first in [10], where it was shown that the optimal policy of a discounted MDP can be captured by a discounted MPC scheme even if the model is inexact. Recently, MPC has been used in different systems to deliver a structured function approximator for MDPs (see e.g., [10–12]) and partially observable MDPs [13]. Stability for discounted MPC schemes is challenging, and for a finite-horizon problem, it is shown in [14] that even if the provided stage cost, terminal cost and terminal set satisfy the stability requirements, the closed-loop might be unstable for some discount factors. Indeed, the discount factor has a critical role in the stability of the closed-loop system under the optimal policy of the discounted cost. The conditions for the asymptotic stability for discounted optimal control problems have been recently developed in [15] for deterministic systems with the exact model. Therefore, an undiscounted MPC scheme is more desirable, where the closed-loop stability analysis is straightforward and well-developed [9].

The equivalence of MDPs criteria (discounted and undiscounted) has been recently discussed in [16] in the case an exact model of MDP is available. However, in practice, the exact probability transition of the MDP might not be available and we usually have a (possibly inaccurate) model of the real system. This work extends the results of [16] in the sense of the model mismatch and while extends also the results of [10] to the case of using undiscounted MPC scheme to capture a (possibly discounted) MDP. More specifically, we show that, under some conditions, an undiscounted finite-horizon Optimal Control Problem (OCP) can capture the optimal policy and the optimal value functions of a given MDP, either discounted or undiscounted, even if an inexact model is used in the undiscounted OCP. We then propose to

use a deterministic (possibly nonlinear) MPC scheme as a particular case of the theorem to formulate the undiscounted OCP as a common MPC scheme. By parameterizing the MPC scheme, and tuning the parameters via RL algorithms one can achieve the best approximation of the optimal policy and the optimal value functions of the original MDP within the adopted MPC structure.

The paper is structured as follows. Section 2 provides the formulation of MDPs under discounted and undiscounted optimality criteria. Section 3 provides formal statements showing that using cost modification in a finite-horizon undiscounted OCP one is able to capture the optimal value function and optimal policy function of the real system with discounted and undiscounted cost even with a wrong model. Section 4 presents a parameterized MPC scheme as a special case of the undiscounted OCP, where the model is deterministic (i.e. the probability transition is a Dirac measure). Then the parameters can be tuned using RL techniques. Section 5 provides an analytical LQR example. Section 6 illustrates different numerical simulation. Finally, section 7 delivers the conclusions.

2 Real System

In this section, we formulate the real system as Markov Decision Processes (MDPs). We consider an MDP on a continuous state and input spaces over $\mathbb{R}^n$ and $\mathbb{R}^m$, respectively, with stochastic states $\boldsymbol{s}_k \in \mathcal{X} \subseteq \mathbb{R}^n$ in the Lebesgue-measurable set $\mathcal{X}$ and inputs $a_k \in \mathcal{U} \in \mathbb{R}^m$. The triple $(\Omega, \mathcal{F}, \rho)$ defines the probability space associated with a Markov chain, where $\Omega = \Pi_{k=0}^{\infty}\mathcal{X}$, with associated σ-field $\mathcal{F}$ and ρ is the probability measure. We then consider stochastic dynamics defined by the following conditional probability measure:

$$\rho\left[\boldsymbol{s}_{k+1}|\boldsymbol{s}_k, \boldsymbol{a}_k\right] , \tag{1}$$

defining the conditional probability of observing a transition from a given state-action pair $\boldsymbol{s}_k$, $\boldsymbol{a}_k$ to a subsequent state $\boldsymbol{s}_{k+1}$. The input $\boldsymbol{a}$ applied to the system for a given state $\boldsymbol{s}$ is selected by a deterministic policy $\boldsymbol{\pi} : \mathcal{X} \to \mathcal{U}$. We denote $\boldsymbol{s}^{\boldsymbol{\pi}}_{0,1,\dots}$ the (possibly stochastic) trajectories of the system (1) under policy $\boldsymbol{\pi}$, i.e., $\boldsymbol{s}^{\boldsymbol{\pi}}_{k+1} \sim \rho\left[\cdot|\boldsymbol{s}^{\boldsymbol{\pi}}_k, \boldsymbol{\pi}(\boldsymbol{s}^{\boldsymbol{\pi}}_k)\right]$, starting from $\boldsymbol{s}^{\boldsymbol{\pi}}_0 = \boldsymbol{s}$, $\forall \boldsymbol{\pi}$. We further denote the measure associated with such trajectories as $\tau^{\boldsymbol{\pi}}_k$ in the same space as ρ. More specifically, $\tau^{\boldsymbol{\pi}}_0(\cdot) = \rho_0(\cdot)$, $\forall \boldsymbol{\pi}$, where $\rho_0(\cdot)$ is the initial state distribution and $\tau^{\boldsymbol{\pi}}_{k+1}(\cdot) := \int_{\mathcal{X}} \rho\left[\cdot|\boldsymbol{s}, \boldsymbol{\pi}(\boldsymbol{s})\right] \tau^{\boldsymbol{\pi}}_k(\mathrm{d}\boldsymbol{s}) , k > 0$.

2.1 Discounted MDPs

In the discounted setting, we aim to find the optimal policy $\boldsymbol{\pi}^\star$, solution of the following discounted infinite-horizon OCP:

$$V^\star(\boldsymbol{s}) := \min_{\boldsymbol{\pi}} V^{\boldsymbol{\pi}}(\boldsymbol{s}) := \mathbb{E}_{\tau^{\boldsymbol{\pi}}}\left[\sum_{k=0}^{\infty} \gamma^k \ell(\boldsymbol{s}^{\boldsymbol{\pi}}_k, \boldsymbol{\pi}\left(\boldsymbol{s}^{\boldsymbol{\pi}}_k\right))\right], \tag{2}$$

for all initial states $\boldsymbol{s}_0^{\boldsymbol{\pi}} = \boldsymbol{s}$, where $V^\star : \mathcal{X} \to \mathbb{R}$ is the optimal value function, $V^{\boldsymbol{\pi}}$ is the value function of the Markov Chain in closed-loop with policy $\boldsymbol{\pi}$, $\ell : \mathcal{X} \times \mathcal{U} \to \mathbb{R}$ is the stage cost function of the real system and $\gamma \in (0, 1]$ is the discount factor. The expectation $\mathbb{E}_{\tau^{\boldsymbol{\pi}}}$ is taken over the distribution underlying the Markov Chain (1) in closed-loop with policy $\boldsymbol{\pi}$, i.e., $\boldsymbol{s}_k \sim \tau_k^{\boldsymbol{\pi}}(\cdot)$ for $k > 0$. The action-value function $Q^\star(\boldsymbol{s}, \boldsymbol{a})$ and advantage function $A^\star(\boldsymbol{s}, \boldsymbol{a})$ associated to (2) are defined as follows:

$$Q^\star(\boldsymbol{s}, \boldsymbol{a}) := \ell(\boldsymbol{s}, \boldsymbol{a}) + \gamma \mathbb{E}_\rho \left[V^\star(\boldsymbol{s}^+) | \boldsymbol{s}, \boldsymbol{a} \right], \tag{3a}$$

$$A^\star(\boldsymbol{s}, \boldsymbol{a}) := Q^\star(\boldsymbol{s}, \boldsymbol{a}) - V^\star(\boldsymbol{s}). \tag{3b}$$

Then from the Bellman equation, we have the following identities:

$$V^\star(\boldsymbol{s}) = Q^\star(\boldsymbol{s}, \boldsymbol{\pi}^\star(\boldsymbol{s})) = \min_{\boldsymbol{a}} Q^\star(\boldsymbol{s}, \boldsymbol{a}), \quad \forall \boldsymbol{s} \in \mathcal{X}, \tag{4a}$$

$$0 = \min_{\boldsymbol{a}} A^\star(\boldsymbol{s}, \boldsymbol{a}), \quad \boldsymbol{\pi}^\star(\boldsymbol{s}) \in \arg\min_{\boldsymbol{a}} A^\star(\boldsymbol{s}, \boldsymbol{a}), \ \forall \boldsymbol{s} \in \mathcal{X}. \tag{4b}$$

2.2 Undiscounted MDPs

Undiscounted MDPs refer to MDPs when $\gamma = 1$. In this case $V^\star$ is in general unbounded and the MDP is ill-posed. In order to tackle this issue, alternative optimality criteria are needed. Gain optimality is one of the common criteria in the undiscounted setting. Gain optimality is defined based on the following average-cost problem:

$$\bar{V}^\star(\boldsymbol{s}) := \min_{\boldsymbol{\pi}} \lim_{N \to \infty} \frac{1}{N} \mathbb{E}_{\tau^{\boldsymbol{\pi}}} \left[\sum_{k=0}^{N-1} \ell(\boldsymbol{s}_k^{\boldsymbol{\pi}}, \boldsymbol{\pi}(\boldsymbol{s}_k^{\boldsymbol{\pi}})) \right], \tag{5}$$

for all initial states $\boldsymbol{s}_0^{\boldsymbol{\pi}} = \boldsymbol{s}$, $\forall \boldsymbol{\pi}$, where $\bar{V}^\star$ is the optimal average cost. We denote the optimal policy solution of (5) as $\bar{\boldsymbol{\pi}}^\star$. This optimal policy is called *gain optimal*. The gain optimal policy $\bar{\boldsymbol{\pi}}^\star$ may not be unique. Moreover, the optimal average cost $\bar{V}^\star$ is commonly assumed to be independent of the initial state $\boldsymbol{s}$ [17]. This assumption e.g. holds for *unichain* MDPs, in which under any policy any state can be reached in finite time from any other state. Unfortunately, the gain optimality criterion only considers the optimal steady-state distribution and it overlooks transients. As an alternative, *bias optimality* considers the optimality of the transients. Precisely, bias optimality can be formulated through the following OCP:

$$\tilde{V}^\star(\boldsymbol{s}) = \min_{\boldsymbol{\pi}} \mathbb{E}_{\tau^{\boldsymbol{\pi}}} \left[\sum_{k=0}^{\infty} (\ell(\boldsymbol{s}_k^{\boldsymbol{\pi}}, \boldsymbol{\pi}(\boldsymbol{s}_k^{\boldsymbol{\pi}})) - \bar{V}^\star) \right], \tag{6}$$

where $\tilde{V}^\star$ is the optimal value function associated to bias optimality. Note that (6) can be seen as a special case of the discounted setting in (2) when $\gamma = 1$ and the optimal average cost $\bar{V}^\star$ is subtracted from the stage cost in (2). Therefore, for the rest of the paper we will consider the discounted setting (2). Without loss of generality we assume that $\bar{V}^\star = 0$ in the case $\gamma = 1$. This choice yields a well-posed optimal value function in the undiscounted setting. Clearly, if this does not hold, one can shift the stage cost to achieve $\bar{V}^\star = 0$.

3 Model of the system

In general, we may not have full knowledge of the probability transition of the real MDP (1). One then typically considers an imperfect model of the real MDP (1), having the state transition:

$$\hat{\rho}\left[\boldsymbol{s}_{k+1}|\boldsymbol{s}_k, \boldsymbol{a}_k\right]. \tag{7}$$

in the same space as ρ. In order to distinguish it from the real system trajectory, let us denote $\hat{\boldsymbol{s}}^{\boldsymbol{\pi}}_{0,1,\ldots}$ the (possibly stochastic) trajectories of the state transition model (7) under policy $\boldsymbol{\pi}$, i.e., $\hat{\boldsymbol{s}}^{\boldsymbol{\pi}}_{k+1} \sim \hat{\rho}\left[\cdot|\hat{\boldsymbol{s}}^{\boldsymbol{\pi}}_k, \boldsymbol{\pi}(\hat{\boldsymbol{s}}^{\boldsymbol{\pi}}_k)\right]$, starting from $\hat{\boldsymbol{s}}^{\boldsymbol{\pi}}_0 = \boldsymbol{s}, \forall \boldsymbol{\pi}$. We further denote the measure associated with such trajectories as $\hat{\tau}^{\boldsymbol{\pi}}$. In general, $\tilde{\cdot}$ refers to the notations related to the imperfect model of the system in this paper. It has been shown in [18] that proving closed-loop stability of the Markov Chains with the optimal policy resulting from an undiscounted OCP is more straightforward than a discounted setting [16]. This observation is well-known in MPC of deterministic systems [19]. Therefore, in this paper, we are interested in using an undiscounted OCP for the model (7) in order to extract the optimal policy and optimal value functions of the real system (1), as this allows us to enforce stability guarantees.

3.1 Finite-horizon OCP

While MPC allows one to introduce stability and safety guarantees, it also requires a model of the real system which is bound to be imperfect, and it optimizes the cost over a finite horizon with unitary discount factor. In other words, MPC is an MDP based on the imperfect system model (7) which we will formulate in (8). In this section we will prove that these differences between the MPC formulation and the original MDP formulation do not hinder the ability to obtain the optimal policy and the optimal value functions of the real system through MPC. Consider the following undiscounted finite-horizon OCP associated to model (7):

$$\hat{V}^{\star}_N(\boldsymbol{s}) = \min_{\boldsymbol{\pi}} \hat{V}^{\boldsymbol{\pi}}_N(\boldsymbol{s}) := \mathbb{E}_{\hat{\tau}^{\boldsymbol{\pi}}}\left[\hat{T}(\hat{\boldsymbol{s}}^{\boldsymbol{\pi}}_N) + \sum_{k=0}^{N-1} \hat{L}(\hat{\boldsymbol{s}}^{\boldsymbol{\pi}}_k, \boldsymbol{\pi}\left(\hat{\boldsymbol{s}}^{\boldsymbol{\pi}}_k\right))\right], \tag{8}$$

with initial state $\hat{\boldsymbol{s}}^{\boldsymbol{\pi}}_0 = \boldsymbol{s}$, where $N \in \mathbb{N}$ is the horizon length, $\hat{T}$, $\hat{L}$, $\hat{V}^{\star}_N$ and $\hat{V}^{\boldsymbol{\pi}}_N$ are the terminal cost, the stage cost, the optimal value function and the value function of the policy $\boldsymbol{\pi}$ associated to model (7), respectively, and where $\mathbb{N}$ is the set of natural numbers. The expectation $\mathbb{E}_{\hat{\tau}^{\boldsymbol{\pi}}}$ in (8) is taken over undiscounted closed-loop Markov Chain (7) with policy $\boldsymbol{\pi}$. We denote $\hat{\boldsymbol{\pi}}^{\star}_N$ the optimal policy resulting from (8). Moreover, the action-value function $\hat{Q}^{\star}_N$ associated to (8) is defined as follows:

$$\hat{Q}^{\star}_N(\boldsymbol{s}, \boldsymbol{a}) := \hat{L}(\boldsymbol{s}, \boldsymbol{a}) + \mathbb{E}_{\hat{\rho}}\left[\hat{V}^{\star}_{N-1}(\boldsymbol{s}^+)|\boldsymbol{s}, \boldsymbol{a}\right], \qquad \hat{V}^{\star}_0(\boldsymbol{s}) := \hat{T}(\boldsymbol{s}) \tag{9}$$

The next assumption expresses a requirement on the boundedness of $V^{\star}$ under model trajectories $\hat{\boldsymbol{s}}^{\boldsymbol{\pi}}_{0,1,\ldots}$ with the optimal policy $\boldsymbol{\pi}^{\star}$ which allows us to develop the theoretical results of this paper.

Assumption 1. *The following set is non-empty for a given $\bar{N} \in \mathbb{N}$.*

$$\mathcal{S} =: \left\{ \boldsymbol{s} \in \mathcal{X} \;\middle|\; \left| \mathbb{E}_{\hat{\tau}^{\boldsymbol{\pi}^\star}} \left[V^\star(\hat{\boldsymbol{s}}_k^{\boldsymbol{\pi}^\star}) \right] \right| < \infty,\ \forall\, k \leq \bar{N} \right\} \tag{10}$$

Assumption 1 requires that there exists a non-empty set $\mathcal{S}$ such that for all trajectories starting in it, the expected value of $V^\star$ is bounded at all future times under the state distribution given by the model in finite time under the optimal policy. This assumption plays a vital role in the derivation of our main result. We will further detail this assumption in Section 5.1.

The next theorem provides theoretical support to the idea that one can recover the optimal policy and value functions by means of an MPC scheme which is based on an imperfect model and has an undiscounted formulation over a finite prediction horizon.

Theorem 1. *Suppose that Assumption 1 holds for $\bar{N} \geq N$. Then, there exist a terminal cost $\hat{T}$ and a stage cost $\hat{L}$ such that the following identities hold, $\forall\, \gamma$, $N \in \mathbb{N}$ and $\boldsymbol{s} \in \mathcal{S}$:*

(i) $\hat{\boldsymbol{\pi}}_N^\star(\boldsymbol{s}) = \boldsymbol{\pi}^\star(\boldsymbol{s})$,

(ii) $\hat{V}_N^\star(\boldsymbol{s}) = V^\star(\boldsymbol{s})$,

(iii) $\hat{Q}_N^\star(\boldsymbol{s}, \boldsymbol{a}) = Q^\star(\boldsymbol{s}, \boldsymbol{a})$, *for the inputs $\boldsymbol{a} \in \mathcal{U}$ such that $|\mathbb{E}_{\hat{\rho}}\left[V^\star(\boldsymbol{s}^+)|\boldsymbol{s}, \boldsymbol{a}\right]| < \infty$*

Proof. We select the terminal cost $\hat{T}$ and the stage cost $\hat{L}$ as follows:

$$\hat{T}(\boldsymbol{s}) = V^\star(\boldsymbol{s}) \tag{11a}$$

$$\hat{L}(\boldsymbol{s}, \boldsymbol{a}) = \begin{cases} Q^\star(\boldsymbol{s}, \boldsymbol{a}) - \mathbb{E}_{\hat{\rho}}\left[V^\star(\boldsymbol{s}^+)|\boldsymbol{s}, \boldsymbol{a}\right] & \text{If } |\mathbb{E}_{\hat{\rho}}\left[V^\star(\boldsymbol{s}^+)|\boldsymbol{s}, \boldsymbol{a}\right]| < \infty \\ \infty & \text{otherwise} \end{cases} \tag{11b}$$

Under Assumption 1, the terminal and stage costs in (8) have a finite expected value for all $\hat{\boldsymbol{s}}_0^{\boldsymbol{\pi}^\star} \in \mathcal{S}$. By substitution of (11) in (8) and using telescopic sum, we have:

$$\begin{aligned} \hat{V}_N^{\boldsymbol{\pi}}(\boldsymbol{s}) &= \mathbb{E}_{\hat{\tau}^{\boldsymbol{\pi}}}\left[\hat{T}(\hat{\boldsymbol{s}}_N^{\boldsymbol{\pi}}) + \sum_{k=0}^{N-1} \hat{L}(\hat{\boldsymbol{s}}_k^{\boldsymbol{\pi}}, \boldsymbol{\pi}\left(\hat{\boldsymbol{s}}_k^{\boldsymbol{\pi}}\right)) \right] \\ &\overset{(11)}{=} \mathbb{E}_{\hat{\tau}^{\boldsymbol{\pi}}}\left[V^\star(\hat{\boldsymbol{s}}_N^{\boldsymbol{\pi}}) + \sum_{k=0}^{N-1} \left(Q^\star(\hat{\boldsymbol{s}}_k^{\boldsymbol{\pi}}, \boldsymbol{\pi}\left(\hat{\boldsymbol{s}}_k^{\boldsymbol{\pi}}\right)) - V^\star(\hat{\boldsymbol{s}}_{k+1}^{\boldsymbol{\pi}}) \right) \right] \\ &= Q^\star(\boldsymbol{s}, \boldsymbol{\pi}(\boldsymbol{s})) + \mathbb{E}_{\hat{\tau}^{\boldsymbol{\pi}}}\left[\sum_{k=1}^{N-1} \left(Q^\star(\hat{\boldsymbol{s}}_k^{\boldsymbol{\pi}}, \boldsymbol{\pi}\left(\hat{\boldsymbol{s}}_k^{\boldsymbol{\pi}}\right)) - V^\star(\hat{\boldsymbol{s}}_k^{\boldsymbol{\pi}}) \right) \right] \\ &= Q^\star(\boldsymbol{s}, \boldsymbol{\pi}(\boldsymbol{s})) + \mathbb{E}_{\hat{\tau}^{\boldsymbol{\pi}}}\left[\sum_{k=1}^{N-1} A^\star(\hat{\boldsymbol{s}}_k^{\boldsymbol{\pi}}, \boldsymbol{\pi}\left(\hat{\boldsymbol{s}}_k^{\boldsymbol{\pi}}\right)) \right], \end{aligned} \tag{12}$$

where $\hat{\boldsymbol{s}}_0 = \boldsymbol{s}$. From (4a) and (4b), we know that:

$$\boldsymbol{\pi}^\star(\cdot) = \arg\min_{\boldsymbol{\pi}} A^\star\left(\cdot, \boldsymbol{\pi}\left(\cdot\right)\right) = \arg\min_{\boldsymbol{\pi}} Q^\star\left(\cdot, \boldsymbol{\pi}\left(\cdot\right)\right) \tag{13}$$

then from (12):

$$\begin{aligned}\boldsymbol{\pi}^\star(\boldsymbol{s}) &= \arg\min_{\boldsymbol{\pi}} \hat{V}_N^{\boldsymbol{\pi}}(\boldsymbol{s}) \\ &= \arg\min_{\boldsymbol{\pi}} Q^\star(\boldsymbol{s}, \boldsymbol{\pi}(\boldsymbol{s})) + \mathbb{E}_{\hat{\tau}^{\boldsymbol{\pi}}}\left[\sum_{k=1}^{N-1} A^\star(\hat{\boldsymbol{s}}_k^{\boldsymbol{\pi}}, \boldsymbol{\pi}\left(\hat{\boldsymbol{s}}_k^{\boldsymbol{\pi}}\right))\right]\end{aligned} \tag{14}$$

Note that $\boldsymbol{\pi}^\star$ minimizes all terms in the cost above, i.e., $A^\star$ and $Q^\star$, such that is must also minimize $\hat{V}_N^{\boldsymbol{\pi}}$. This proves (i), i.e.,

$$\boldsymbol{\pi}^\star(\boldsymbol{s}) = \hat{\boldsymbol{\pi}}_N^\star(\boldsymbol{s}).$$

In turn, this proves (ii), since

$$\begin{aligned}\hat{V}_N^\star(\boldsymbol{s}) = \hat{V}_N^{\boldsymbol{\pi}^\star}(\boldsymbol{s}) &= Q^\star(\boldsymbol{s}, \boldsymbol{\pi}^\star(\boldsymbol{s})) + \mathbb{E}_{\hat{\tau}^{\boldsymbol{\pi}}}\left[\sum_{k=1}^{N} \underbrace{A^\star(\hat{\boldsymbol{s}}_k^{\boldsymbol{\pi}^\star}, \boldsymbol{\pi}^\star\left(\hat{\boldsymbol{s}}_k^{\boldsymbol{\pi}^\star}\right))}_{\overset{(4b)}{=}0} \middle| \hat{\boldsymbol{s}}_0 = \boldsymbol{s}\right] \\ &= Q^\star(\boldsymbol{s}, \boldsymbol{\pi}^\star(\boldsymbol{s})) \overset{(4a)}{=} V^\star(\boldsymbol{s}).\end{aligned} \tag{15}$$

Moreover, from (9) and (11b), for any inputs $\boldsymbol{a} \in \mathcal{U}$ such that $|\mathbb{E}_{\hat{\rho}}\left[V^\star(\boldsymbol{s}^+)|\boldsymbol{s}, \boldsymbol{a}\right]| < \infty$, we have:

$$\begin{aligned}\hat{Q}_N^\star(\boldsymbol{s}, \boldsymbol{a}) &= \hat{L}(\boldsymbol{s}, \boldsymbol{a}) + \mathbb{E}_{\hat{\rho}}\left[\hat{V}_{N-1}^\star(\boldsymbol{s}^+)|\boldsymbol{s}, \boldsymbol{a}\right] \\ &\overset{(11b)}{=} Q^\star(\boldsymbol{s}, \boldsymbol{a}) + \mathbb{E}_{\hat{\rho}}\left[\hat{V}_{N-1}^\star(\boldsymbol{s}^+) - V^\star(\boldsymbol{s}^+)|\boldsymbol{s}, \boldsymbol{a}\right] = Q^\star(\boldsymbol{s}, \boldsymbol{a}),\end{aligned} \tag{16}$$

where the last inequality is obtained by noting that (ii) for $N > 1$ and $\hat{V}_0^\star(\boldsymbol{s}) = \hat{T}(\boldsymbol{s}) = V^\star(\boldsymbol{s})$ for $N = 1$. This directly yields (iii). ■

Theorem 1 states that, independent of the discount factor γ, it is possible to find a finite-horizon OCP cost function that provides the optimal policy and optimal value functions of a discounted MDP if an inexact model is used in the finite-horizon OCP. We observe that the setup of this paper has been analyzed in [16], under the assumption of a perfect model, i.e., $\hat{\rho}[\cdot|\boldsymbol{s}, \boldsymbol{a}] = \rho[\cdot|\boldsymbol{s}, \boldsymbol{a}]$. In that case (11b) reads:

$$\hat{L}(\boldsymbol{s}, \boldsymbol{a}) = \ell(\boldsymbol{s}, \boldsymbol{a}) + (\gamma - 1)\mathbb{E}_\rho[V^\star(\boldsymbol{s}^+)|\boldsymbol{s}, \boldsymbol{a}],\ \forall \boldsymbol{s} \in \mathcal{S}, \tag{17}$$

which corresponds to the cost modification discussed in [16].

3.2 Infinite-horizon OCP

In this section, we investigate the case $N \to \infty$ for which, under some conditions, the terminal cost can be dismissed. In this case, we first make the next additional assumption.

Assumption 2. *We assume that the optimal value function converges to a constant and finite value with model* (7) *under the optimal policy* $\boldsymbol{\pi}^\star$*. I.e.:*

$$-\infty < \lim_{N\to\infty} \mathbb{E}_{\hat{\tau}^{\boldsymbol{\pi}^\star}} \left[V^\star(\hat{\boldsymbol{s}}_N^{\boldsymbol{\pi}^\star}) \right] = \hat{v}_\infty < \infty \tag{18}$$

Assumption 2 can be interpreted as some forms of the stability condition on the model dynamics under the optimal policy $\boldsymbol{\pi}^\star$. We will explain this assumption in Section 5.1. In this section, we consider the following undiscounted value function without terminal cost:

$$\hat{V}_\infty^\star(\boldsymbol{s}) := \min_{\boldsymbol{\pi}} \hat{V}_\infty^{\boldsymbol{\pi}}(\boldsymbol{s}) := \lim_{N\to\infty} \mathbb{E}_{\hat{\tau}^{\boldsymbol{\pi}}} \left[\sum_{k=0}^{N-1} \hat{L}(\hat{\boldsymbol{s}}_k^{\boldsymbol{\pi}}, \boldsymbol{\pi}\left(\hat{\boldsymbol{s}}_k^{\boldsymbol{\pi}}\right)) \right] \tag{19}$$

with initial state $\hat{\boldsymbol{s}}_0^{\boldsymbol{\pi}} = \boldsymbol{s}$. We denote the optimal policy solution of (19) as $\hat{\boldsymbol{\pi}}_\infty^\star(\boldsymbol{s})$. We then define the optimal action-value function $\hat{Q}_\infty^\star$ associated to (19) as follows:

$$\hat{Q}_\infty^\star(\boldsymbol{s}, \boldsymbol{a}) = \hat{L}(\boldsymbol{s}, \boldsymbol{a}) + \mathbb{E}_{\hat{\rho}} \left[\hat{V}_\infty^\star(\boldsymbol{s}^+) | \boldsymbol{s}, \boldsymbol{a} \right] \ , \tag{20}$$

We are now ready to state the equivalent of Theorem 1 in case of an infinite horizon without a terminal cost.

Theorem 2. *Suppose that Assumptions 1 and 2 hold, then the following hold* $\forall \boldsymbol{s} \in \mathcal{S}, \forall \gamma$*:*

(i) $\hat{\boldsymbol{\pi}}_\infty^\star(\boldsymbol{s}) = \boldsymbol{\pi}^\star(\boldsymbol{s})$

(ii) $\hat{V}_\infty^\star(\boldsymbol{s}) = V^\star(\boldsymbol{s}) - \hat{v}_\infty$

(iii) $\hat{Q}_\infty^\star(\boldsymbol{s}, \boldsymbol{a}) = Q^\star(\boldsymbol{s}, \boldsymbol{a}) - \hat{v}_\infty$*, for the inputs* $\boldsymbol{a} \in \mathcal{U}$ *such that* $|\mathbb{E}_{\hat{\rho}} \left[V^\star(\boldsymbol{s}^+) | \boldsymbol{s}, \boldsymbol{a} \right]| < \infty$

if the stage cost $\hat{L}$ *is selected according Equation* (11b)*.*

Proof. Using stage cost $\hat{L}$ in (11b), we have:

$$\begin{aligned}
\hat{V}_\infty^{\boldsymbol{\pi}}(\boldsymbol{s}) &= \lim_{N\to\infty} \ \mathbb{E}_{\hat{\tau}^{\boldsymbol{\pi}}} \left[\sum_{k=0}^{N-1} Q^\star(\hat{\boldsymbol{s}}_k^{\boldsymbol{\pi}}, \boldsymbol{\pi}\left(\hat{\boldsymbol{s}}_k^{\boldsymbol{\pi}}\right)) - \mathbb{E}_{\hat{\rho}} \left[V^\star(\hat{\boldsymbol{s}}_{k+1}^{\boldsymbol{\pi}}) | \hat{\boldsymbol{s}}_k^{\boldsymbol{\pi}}, \boldsymbol{\pi}\left(\hat{\boldsymbol{s}}_k^{\boldsymbol{\pi}}\right) \right] \right] \\
&= \lim_{N\to\infty} \ \mathbb{E}_{\hat{\tau}^{\boldsymbol{\pi}}} \left[\sum_{k=0}^{N-1} Q^\star(\hat{\boldsymbol{s}}_k^{\boldsymbol{\pi}}, \boldsymbol{\pi}\left(\hat{\boldsymbol{s}}_k^{\boldsymbol{\pi}}\right)) - V^\star(\hat{\boldsymbol{s}}_{k+1}^{\boldsymbol{\pi}}) \right] \\
&= Q^\star(\boldsymbol{s}, \boldsymbol{\pi}(\boldsymbol{s})) + \lim_{N\to\infty} \mathbb{E}_{\hat{\tau}^{\boldsymbol{\pi}}} \left[-V^\star(\hat{\boldsymbol{s}}_N^{\boldsymbol{\pi}}) + \sum_{k=1}^{N-1} Q^\star(\hat{\boldsymbol{s}}_k^{\boldsymbol{\pi}}, \boldsymbol{\pi}(\hat{\boldsymbol{s}}_k^{\boldsymbol{\pi}})) - V^\star(\hat{\boldsymbol{s}}_k^{\boldsymbol{\pi}}) \right] \\
&= Q^\star(\boldsymbol{s}, \boldsymbol{\pi}(\boldsymbol{s})) + \lim_{N\to\infty} \mathbb{E}_{\hat{\tau}^{\boldsymbol{\pi}}} \left[-V^\star(\hat{\boldsymbol{s}}_N^{\boldsymbol{\pi}}) + \sum_{k=1}^{N-1} A^\star(\hat{\boldsymbol{s}}_k^{\boldsymbol{\pi}}, \boldsymbol{\pi}(\hat{\boldsymbol{s}}_k^{\boldsymbol{\pi}})) \right]
\end{aligned} \tag{21}$$

where $\hat{\boldsymbol{s}}_0^{\boldsymbol{\pi}} = \boldsymbol{s}$. By (4a) and (4b) we know that the policy $\boldsymbol{\pi}(\boldsymbol{s}) = \boldsymbol{\pi}^\star(\boldsymbol{s})$ minimizes all terms $A^\star(\cdot, \boldsymbol{\pi}(\cdot))$ and $Q^\star(\cdot, \boldsymbol{\pi}(\cdot))$, such that it also minimizes $\hat{V}_\infty^{\boldsymbol{\pi}}(\boldsymbol{s})$, i.e.,:

$$\hat{\boldsymbol{\pi}}_\infty^\star(\boldsymbol{s}) = \arg\min_{\boldsymbol{\pi}} \hat{V}_\infty^{\boldsymbol{\pi}}(\boldsymbol{s}) = \boldsymbol{\pi}^\star(\boldsymbol{s})\,, \tag{22}$$

which proves (i). Moreover:

$$\hat{V}_\infty^{\boldsymbol{\pi}^\star}(\boldsymbol{s}) = V^\star(\boldsymbol{s}) - \lim_{N\to\infty} \mathbb{E}\left[V^\star(\hat{\boldsymbol{s}}_N^{\boldsymbol{\pi}^\star})\right]. \tag{23}$$

Using (18) we have:

$$\hat{V}_\infty^\star(\boldsymbol{s}) = \hat{V}_\infty^{\boldsymbol{\pi}^\star}(\boldsymbol{s}) = V^\star(\boldsymbol{s}) - \hat{v}_\infty. \tag{24}$$

For the inputs $\boldsymbol{a} \in \mathcal{U}$ such that $|\mathbb{E}_{\hat{\rho}}\left[V^\star(\boldsymbol{s}^+)|\boldsymbol{s}, \boldsymbol{a}\right]| < \infty$:

$$\begin{aligned} \hat{Q}_\infty^\star(\boldsymbol{s}, \boldsymbol{a}) &= \hat{L}(\boldsymbol{s}, \boldsymbol{a}) + \mathbb{E}_{\hat{\rho}}\left[\hat{V}_\infty^\star(\boldsymbol{s}^+)|\boldsymbol{s}, \boldsymbol{a}\right] \\ &= Q^\star(\boldsymbol{s}, \boldsymbol{a}) - \mathbb{E}_{\hat{\rho}}\left[V^\star(\boldsymbol{s}^+)|\boldsymbol{s}, \boldsymbol{a}\right] + \mathbb{E}_{\hat{\rho}}\left[\hat{V}_\infty^\star(\boldsymbol{s}^+)|\boldsymbol{s}, \boldsymbol{a}\right] \\ &= Q^\star(\boldsymbol{s}, \boldsymbol{a}) - \mathbb{E}_{\hat{\rho}}\left[V_\infty^\star(\boldsymbol{s}^+) - \hat{V}^\star(\boldsymbol{s}^+)|\boldsymbol{s}, \boldsymbol{a}\right] = Q^\star(\boldsymbol{s}, \boldsymbol{a}) - \hat{v}_\infty, \end{aligned} \tag{25}$$

which completes the proof. ■

Theorem 2 extends Theorem 1 to the case of an infinite horizon with zero terminal cost. Assumption 2 is necessary in order to be able to remove the terminal cost. In the next section we will detail the use of the theorems in practice and reformulate OCP (8) as a Model Predictive Control (MPC) scheme.

4 MPC as a function approximator for RL

As it was shown in the previous section, the optimal policy and value functions of any MDP with either discounted or undiscounted criteria can be captured using a finite-horizon undiscounted OCP (8) even if the model is not accurate. Since the equivalence only holds at the initial state, if one is interested in recovering the optimal MDP policy, the finite-horizon OCP needs to be solved from scratch for each initial state. In practice, this amounts to deploying the finite-horizon OCP in an MPC framework, i.e., in a closed-loop.

As discussed above, the equivalence is only obtained if a properly modified stage and terminal costs are introduced for the finite-horizon undiscounted MPC scheme. However, finding such costs requires knowledge about the optimal value functions of the real MDP. In this section, we detail how the theorems we provided in the previous sections can be used in practice to exploit MPC as a structured function approximator of the optimal policy and value functions of the real MDP. One of the main advantages of MPC is that it allows us to straightforwardly introduce state and input constraints in the policy. We parameterize the MPC scheme with parameter

vector $\boldsymbol{\theta}$ such that RL methods can be deployed to tune $\boldsymbol{\theta}$ in order to achieve the equivalence yielding the optimal policy and value functions of the real system and, consequently, the best possible closed-loop performance.

As the MPC model is not required to capture the real system dynamics exactly, for the sake of reducing the computational burden, and due to the (relative) simplicity of the resulting MPC scheme, a popular choice of model $\hat{\rho}\left[\boldsymbol{s}^{+}|\boldsymbol{s},\boldsymbol{a}\right]$ is a deterministic model, i.e.:

$$\hat{\rho}\left[\boldsymbol{s}^{+}|\boldsymbol{s},\boldsymbol{a}\right] = \delta\left(\boldsymbol{s}^{+} - \boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{s},\boldsymbol{a})\right) \tag{26}$$

where $\delta(\cdot)$ is the Dirac measure and $\boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{s},\boldsymbol{a})$ is a parameterized deterministic (possibly nonlinear) model. We approximate the modified costs $\hat{L}$ and $\hat{T}$ by parametric functions $L_{\boldsymbol{\theta}}$ and $T_{\boldsymbol{\theta}}$, respectively. Due to the mismatch between the model and the real system, hard constraints in the MPC scheme could become infeasible. This is a well-known issue in the MPC community and one simple solution consists in formulating the state constraints as soft constraints [20]. We therefore formulate the MPC finite-horizon OCP as:

$$\hat{V}_N^{\boldsymbol{\theta}}(\boldsymbol{s}) = \min_{\hat{\boldsymbol{a}},\hat{\boldsymbol{s}},\boldsymbol{\sigma}} \quad -\lambda_{\boldsymbol{\theta}}(\hat{\boldsymbol{s}}_0) + T_{\boldsymbol{\theta}}(\hat{\boldsymbol{s}}_N) + \boldsymbol{\mu}_{\mathrm{f}}^{\top}\boldsymbol{\sigma}_N + \sum_{k=0}^{N-1} L_{\boldsymbol{\theta}}(\hat{\boldsymbol{s}}_k,\hat{\boldsymbol{a}}_k) + \boldsymbol{\mu}^{\top}\boldsymbol{\sigma}_k \tag{27a}$$

$$\text{s.t.} \quad \hat{\boldsymbol{s}}_{k+1} = \boldsymbol{f}_{\boldsymbol{\theta}}(\hat{\boldsymbol{s}}_k,\hat{\boldsymbol{a}}_k),\ \hat{\boldsymbol{s}}_0 = \boldsymbol{s}, \tag{27b}$$

$$\hat{\boldsymbol{a}}_k \in \mathcal{U},\ 0 \leq \boldsymbol{\sigma}_k,\ 0 \leq \boldsymbol{\sigma}_N, \tag{27c}$$

$$\boldsymbol{h}_{\boldsymbol{\theta}}(\hat{\boldsymbol{s}}_k,\hat{\boldsymbol{a}}_k) \leq \boldsymbol{\sigma}_k^{\star},\ \boldsymbol{h}_{\boldsymbol{\theta}}^{\mathrm{f}}(\hat{\boldsymbol{s}}_N) \leq \boldsymbol{\sigma}_N^{\star}, \tag{27d}$$

where $\hat{V}_N^{\boldsymbol{\theta}}$ is the MPC-based parameterized value function, $\boldsymbol{h}_{\boldsymbol{\theta}}(\boldsymbol{s},\boldsymbol{a})$ is a mixed input-state constraint, $\boldsymbol{h}_{\boldsymbol{\theta}}^{\mathrm{f}}(\boldsymbol{s})$ is the terminal constraint, $\boldsymbol{\sigma}_k$ and $\boldsymbol{\sigma}_N$ are slack variables guaranteeing the feasibility of the MPC scheme and $\boldsymbol{\mu}$ and $\boldsymbol{\mu}_{\mathrm{f}}$ are constant vectors that ought to be selected sufficiently large [20]. Note that these constants allow the MPC scheme to find a feasible solution, but penalize constraint violations enough to guarantee that a feasible solution is found whenever possible. While alternative feasibility-enforcing strategies, e.g., robust MPC, do exist, an exhaustive discussion on the topic is beyond the scope of this paper. Function $\lambda_{\boldsymbol{\theta}}$ parameterizes the so-called storage function, which has been added to the cost in order to enable the MPC scheme to tackle the case of so-called economic problems. Such situations arise when the MDP stage cost is not positive definite, while the MPC stage cost is forced to be positive definite in order to obtain a stabilizing feedback policy. Note that since the term $-\lambda_{\boldsymbol{\theta}}(\hat{\boldsymbol{s}}_0)$ only depends on the current state, it does not modify the optimal policy. For more details, we refer the interested readers to [10, 21].

While Theorem 1 states that one can find suitable stage and terminal costs for any given model, adjusting the model parameters is not essential from the theoretical perspective. However, in practice, the stage and the terminal cost parameterization may not capture $\hat{L}$ and $\hat{T}$ exactly. Since $\hat{L}$ and $\hat{T}$ are (implicitly) functions of the model, using a parameterized model $\boldsymbol{f}_{\boldsymbol{\theta}}$ introduces extra degrees of freedom to bring $\hat{L}$ and $\hat{T}$ closer to the functions that can be represented by $L_{\boldsymbol{\theta}}$ and $T_{\boldsymbol{\theta}}$. In turn, this can yield a better approximation of the optimal policy and value function. The MPC parameterized policy can be obtained from (27) as follows:

$$\hat{\boldsymbol{\pi}}_N^{\boldsymbol{\theta}}(\boldsymbol{s}) = \hat{\boldsymbol{a}}_0^{\star}(\boldsymbol{\theta},\boldsymbol{s}), \tag{28}$$

where $\hat{\boldsymbol{a}}_0^\star$ is the solution of (27), corresponding to the first input $\hat{\boldsymbol{a}}_0$. Moreover, the parameterized action-value function based on MPC scheme (27) can be formulated as follows:

$$\hat{Q}_N^{\boldsymbol{\theta}}(\boldsymbol{s},\boldsymbol{a}) := \min_{\hat{\boldsymbol{a}},\hat{\boldsymbol{s}},\boldsymbol{\sigma}} \ (27a)\ , \qquad \text{s.t. } (27b)-(27d)\,,\ \hat{\boldsymbol{a}}_0 = \boldsymbol{a}\,. \tag{29}$$

Then one obtains the following identities:

$$\hat{V}_N^{\boldsymbol{\theta}}(\boldsymbol{s}) = \min_{\boldsymbol{a}} \hat{Q}_N^{\boldsymbol{\theta}}(\boldsymbol{s},\boldsymbol{a}), \qquad \hat{\boldsymbol{\pi}}_N^{\boldsymbol{\theta}}(\boldsymbol{s}) \in \arg\min_{\boldsymbol{a}} \hat{Q}_N^{\boldsymbol{\theta}}(\boldsymbol{s},\boldsymbol{a})\,. \tag{30}$$

We can use RL techniques, such as Q-learning and policy gradient method to tune the parameters $\boldsymbol{\theta}$ of parameterized MPC scheme (27) and approach the *optimal* parameter $\boldsymbol{\theta}^\star$. For instance, at each learning step, Q-learning based on Temporal difference (TD) method uses the following update rule for θ:

$$\delta_k := \ell(\boldsymbol{s}_k,\boldsymbol{a}_k) + \gamma \hat{V}_N^{\boldsymbol{\theta}}(\boldsymbol{s}_{k+1}) - \hat{Q}_N^{\boldsymbol{\theta}}(\boldsymbol{s}_k,\boldsymbol{a}_k) \tag{31a}$$

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \zeta \delta_k \nabla_{\boldsymbol{\theta}} \hat{Q}_N^{\boldsymbol{\theta}}(\boldsymbol{s}_k,\boldsymbol{a}_k) \tag{31b}$$

in order to capture the optimal value function $\hat{Q}_N^{\boldsymbol{\theta}^\star} \approx Q^\star$ for the optimal parameters $\boldsymbol{\theta}^\star$, where the scalar $\zeta > 0$ is the learning step-size, δ_k is labelled the TD error. The use of RL for the tuning the MPC scheme can be found e.g., in [10, 22].

5 Analytical Case Study

We consider a Linear Quadratic Regulator (LQR) example in order to obtain the corresponding optimal value functions analytically and verify Theorem 2. The real system state transition and stage cost are given as follows:

$$\boldsymbol{s}^+ = A\boldsymbol{s} + B\boldsymbol{a} + \boldsymbol{e}, \quad \ell(\boldsymbol{s},\boldsymbol{a}) = \begin{bmatrix}\boldsymbol{s}\\ \boldsymbol{a}\end{bmatrix}^\top \begin{bmatrix} T & N \\ N^\top & R\end{bmatrix} \begin{bmatrix}\boldsymbol{s}\\ \boldsymbol{a}\end{bmatrix}, \tag{32}$$

where $\boldsymbol{e} \sim \mathcal{N}(0,\Sigma)$ with the discount factor γ. One can verify the following optimal value functions:

$$V^\star(\boldsymbol{s}) = \boldsymbol{s}^\top S \boldsymbol{s} + \hat{v}_\infty, \tag{33}$$

$$Q^\star(\boldsymbol{s},\boldsymbol{a}) = \hat{v}_\infty + \begin{bmatrix}\boldsymbol{s}\\ \boldsymbol{a}\end{bmatrix}^\top \begin{bmatrix} T+\gamma A^\top S A & N+\gamma A^\top S B \\ N^\top+\gamma B^\top S A & R+\gamma B^\top S B\end{bmatrix} \begin{bmatrix}\boldsymbol{s}\\ \boldsymbol{a}\end{bmatrix},$$

where $\hat{v}_\infty = \frac{\gamma}{1-\gamma}\mathrm{Tr}(S\Sigma)$ and S is obtained from the following Riccati equations:

$$T + \gamma A^\top S A = S + (N + \gamma A^\top S B)\left(K_\gamma^\star\right)^\top, \tag{34a}$$

$$(R + \gamma B^\top S B) K_\gamma^\star = N^\top + \gamma B^\top S A. \tag{34b}$$

Then $\boldsymbol{\pi}^\star(\boldsymbol{s}) = -K_\gamma^\star \boldsymbol{s}$ and $\bar{\boldsymbol{\pi}}^\star(\boldsymbol{s}) = \tilde{\boldsymbol{\pi}}^\star(\boldsymbol{s}) = -K_1^\star \boldsymbol{s}$, where $K_1^\star = \lim_{\gamma\to 1} K_\gamma^\star$. We then consider a linear deterministic model:

$$\boldsymbol{s}^+ = \hat{A}\boldsymbol{s} + \hat{B}\boldsymbol{a}, \tag{35}$$

and an undiscounted OCP with the following stage cost, defined according to Equation (11b) as:

$$\begin{aligned}\hat{L}(\boldsymbol{s},\boldsymbol{a}) &= Q^\star(\boldsymbol{s},\boldsymbol{a}) - V^\star(\hat{\boldsymbol{s}}^+) \\ &\overset{(33)}{=} \begin{bmatrix}\boldsymbol{s}\\ \boldsymbol{a}\end{bmatrix}^\top \begin{bmatrix} T+\gamma A^\top S A & N+\gamma A^\top S B \\ N^\top + \gamma B^\top S A & R + \gamma B^\top S B\end{bmatrix}\begin{bmatrix}\boldsymbol{s}\\ \boldsymbol{a}\end{bmatrix} \\ -(\hat{A}\boldsymbol{s}+\hat{B}\boldsymbol{a})^\top S(\hat{A}\boldsymbol{s}+\hat{B}\boldsymbol{a}) &:= \begin{bmatrix}\boldsymbol{s}\\ \boldsymbol{a}\end{bmatrix}^\top \begin{bmatrix}\hat{T} & \hat{N}\\ \hat{N}^\top & \hat{R}\end{bmatrix}\begin{bmatrix}\boldsymbol{s}\\ \boldsymbol{a}\end{bmatrix}.\end{aligned} \tag{36}$$

The Riccati equations for the undiscounted problem with the model (35) read as:

$$\hat{T} + \hat{A}^\top \hat{S}\hat{A} = \hat{S} + (\hat{N} + \hat{A}^\top \hat{S}\hat{B})\left(\hat{K}^\star\right)^\top, \tag{37a}$$

$$(\hat{R} + \hat{B}^\top \hat{S}\hat{B})\hat{K}^\star = \hat{N}^\top + \hat{B}^\top \hat{S}\hat{A}. \tag{37b}$$

with the optimal policy $\hat{\boldsymbol{\pi}}_\infty^\star(\boldsymbol{s}) = -\hat{K}^\star \boldsymbol{s}$ and the optimal value function $\hat{V}_\infty^\star(\boldsymbol{s}) = \boldsymbol{s}^\top \hat{S}\boldsymbol{s}$. From (36), we have:

$$T + \gamma A^\top S A - \hat{A}^\top S \hat{A} = \hat{T}, \tag{38a}$$

$$N + \gamma A^\top S B - \hat{A}^\top S \hat{B} = \hat{N}, \tag{38b}$$

$$R + \gamma B^\top S B - \hat{B}^\top S \hat{B} = \hat{R}. \tag{38c}$$

Equivalently, this entails that $\hat{T}$, $\hat{N}$ and $\hat{R}$ must satisfy

$$\hat{T} + \hat{A}^\top S \hat{A} = T + \gamma A^\top S A, \tag{39a}$$

$$\hat{N} + \hat{A}^\top S \hat{B} = N + \gamma A^\top S B, \tag{39b}$$

$$\hat{R} + \hat{B}^\top S \hat{B} = R + \gamma B^\top S B. \tag{39c}$$

Then:

$$\begin{aligned}\hat{T} + \hat{A}^\top S \hat{A} &\overset{(39a)}{=} T + \gamma A^\top S A \overset{(34a)}{=} S + \\ S(N + \gamma A^\top S B)\left(K_\gamma^\star\right)^\top &\overset{(39b)}{=} S + (\hat{N} + \hat{A}^\top S \hat{B})\left(K_\gamma^\star\right)^\top,\end{aligned} \tag{40}$$

and

$$\begin{aligned}(\hat{R} + \hat{B}^\top S \hat{B})K_\gamma^\star &\overset{(39c)}{=} (R + \gamma B^\top S B)K_\gamma^\star \\ &\overset{(34b)}{=} N^\top + \gamma B^\top S A \overset{(39b)}{=} \hat{N} + \hat{A}^\top S \hat{B}.\end{aligned} \tag{41}$$

Equations (40) and (41) show that $\hat{S} = S$ and $\hat{K}^\star = K_\gamma^\star$ satisfy the undiscounted Riccati equations (37). Then it reads that $\boldsymbol{\pi}^\star(\boldsymbol{s}) = \hat{\boldsymbol{\pi}}_\infty^\star(\boldsymbol{s})$ and $V^\star(\boldsymbol{s}) = \hat{V}_\infty^\star(\boldsymbol{s}) + \hat{v}_\infty$.

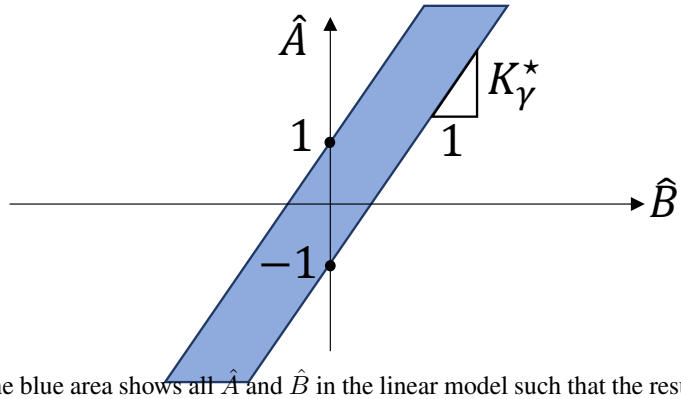


Figure 1: The blue area shows all $\hat{A}$ and $\hat{B}$ in the linear model such that the resulting trajectory and optimal value function remain bounded for the given optimal policy $\boldsymbol{\pi}^\star(\boldsymbol{s}) = -K_\gamma^\star \boldsymbol{s}$.

5.1 Satisfying the assumptions

Regarding Assumption 1, the value function will remain bounded in the finite horizon prediction for every bounded initial condition $\boldsymbol{s}_0$ and every linear model in form (35) for a given control policy $\boldsymbol{\pi}^\star(\boldsymbol{s}) = -K_\gamma^\star \boldsymbol{s}$ or $\bar{\boldsymbol{\pi}}^\star(\boldsymbol{s}) = \tilde{\boldsymbol{\pi}}^\star(\boldsymbol{s}) = -K_1^\star \boldsymbol{s}$. For Assumption 2, the linear model matrices $\hat{A}$ and $\hat{B}$ must be chosen such that $\rho(\hat{A} - \hat{B}K_\gamma^\star) \leq 1$ in order to guarantee boundedness of the optimal value function (33). For instance, for a scalar dynamics, the locus of $\hat{A}$ and $\hat{B}$ is shown in Figure 1. Inspired by this example, we ought to point out here that for linear systems Assumption 1 is automatically obtained if the model is stabilized by the optimal policy, though the converse might not be true (e.g., if the cost is 0). Note that, the systems without constraint satisfying Assumption 1 is fairly straightforward while in the presence of the system constraints, the model also must not violate those constraints. To satisfy Assumption 2, a model must be adopted whose trajectory does not diverge under the optimal policy of the real system and satisfy the system constraint. It is clear that the closer the model is to the real system the more likely it is to satisfy this assumption. This model can be obtained based on offline system identification. In [23], the authors proposed to use robust MPC in order to ensure constraint satisfaction. A deeper discussion of these assumptions can be found in [10] and [16].

6 Numerical Examples

6.1 Non-quadratic stage cost

In this example, we provide a benchmark optimal investment problem with a non-quadratic stage cost. Consider the following dynamics and stage cost [24]:

$$s_{k+1} = a_k\,, \qquad \ell(s,a) = -\ln(As^\alpha - a), \tag{42}$$

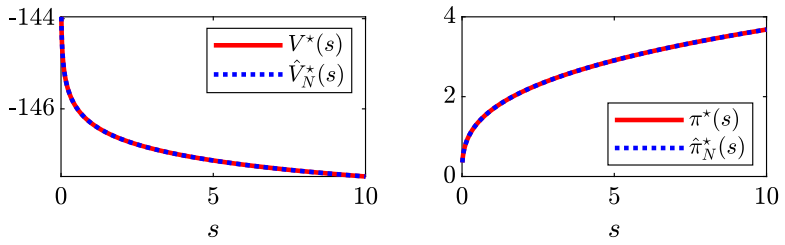


Figure 2: (Left:) Optimal value functions (Right:) and optimal policy resulting from the discounted real system and undiscounted MPC scheme with the wrong model.

where A and $0 < \alpha < 1$ are given constants. It is known that for the discount factor γ, the optimal value and policy functions are $V^\star(s) = B + C\ln(s)$ and $\pi^\star(s) = \gamma\alpha A s^\alpha$, where [25]:

$$B = \frac{\ln((1-\alpha\gamma)A) + \frac{\gamma\alpha}{1-\gamma\alpha}\ln(\alpha\gamma A)}{\gamma - 1}, \quad C = \frac{\alpha}{\alpha\gamma - 1}\,. \tag{43}$$

We then consider a model of the dynamics with $\hat{s}_{k+1} = \mu\hat{a}_k$ and, based on this model, we construct a finite-horizon undiscounted MPC with the costs according Equation (11) in Theorem 1 and $N = 10$. In this example we have considered $A = 5$, $\alpha = 0.34$, $\mu = 0.8$ and $\gamma = 0.9$. Figure 2 compares the optimal value and policy functions from the discounted real system (42) and from the MPC scheme with a wrong model. As predicted by Theorem 1, one can see that they match perfectly. Note that the results are valid for every discount factor $0 < \gamma < 1$, every horizon length and for other values of the constants A, α, and μ.

6.2 Inverted pendulum with process noise

We consider the following discrete-time stochastic dynamics, representing an inverted pendulum with a random support excitation:

$$\boldsymbol{s}_{k+1} = \boldsymbol{s}_k + \begin{bmatrix} s_k(2) \\ (\frac{g}{l} + \xi)\sin(s_k(1)) \end{bmatrix} \delta t + \begin{bmatrix} 0 \\ \frac{\delta t}{ml^2} \end{bmatrix} \boldsymbol{a}_k \tag{44}$$

where $g = 9.81$, $l = 0.3$, $m = 0.5$ and $\delta t = 0.1$ are constants representing the gravity, mass, length and the sampling time of the discrete dynamics. Disturbance $\xi \sim \mathcal{U}[-0.5, 0.5]$ has a uniform distribution and $\boldsymbol{s}_k := [s_k(1),\ s_k(2)]^\top$ is the system state and $\boldsymbol{a}_k$ is the system input. We consider $\ell(\boldsymbol{s}, \boldsymbol{a}) = \boldsymbol{s}^\top \boldsymbol{s} + \boldsymbol{a}^2$ as a stage cost with the discount factor $\gamma = 0.95$. We first aim to find an approximate solution for the optimal policy and the optimal value functions using Dynamic Programming (DP). We consider the state constraints $-1 \leq s_k(1) \leq 1$, $-1 \leq s_k(2) \leq 1$ and the input constraint $-0.8 \leq a_k \leq 0.8$. Figure 3 shows the optimal value function and the optimal policy function resulting from DP for the discounted infinite-horizon MDP.

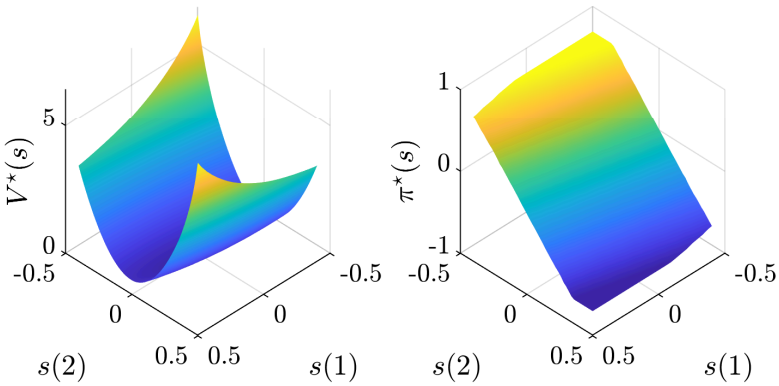


Figure 3: Optimal Value (left) and policy (right) functions resulting from ADP.

We build an undiscounted finite-horizon OCP with a wrong model in order to capture the optimal value and the optimal policy functions of the discounted infinite horizon MDP. To do this, we consider an MPC scheme with a deterministic linearized form of the dynamics as a model of the real system as follows:

$$\hat{\boldsymbol{s}}_{k+1} = \boldsymbol{f}_{\boldsymbol{\theta}}(\hat{\boldsymbol{s}}_k, \hat{\boldsymbol{a}}_k) = \hat{\boldsymbol{s}}_k + \begin{bmatrix} \hat{s}_k(2) \\ \frac{g}{\theta_l}\hat{s}_k(1) \end{bmatrix} \delta t + \begin{bmatrix} 0 \\ \frac{\delta t}{m\theta_l^2} \end{bmatrix} \hat{\boldsymbol{a}}_k \tag{45}$$

where $\hat{\boldsymbol{s}}_k := [\hat{s}_k(1),\ \hat{s}_k(2)]^\top$ and $\hat{\boldsymbol{a}}_k$ are the model state and input. Moreover, we consider an uncertain l with a adjustable parameter θ_l, with an initial value 0.25. We consider the parameterized MPC scheme with the horizon length $N = 10$ and the following parameterized quadratic stage and terminal cost:

$$T_{\boldsymbol{\theta}}(\boldsymbol{s}) = \boldsymbol{s}^\top G \boldsymbol{s}, \qquad L_{\boldsymbol{\theta}}(\boldsymbol{s}, \boldsymbol{a}) = \begin{bmatrix} \boldsymbol{s} \\ a \end{bmatrix}^\top H \begin{bmatrix} \boldsymbol{s} \\ a \end{bmatrix} \tag{46}$$

where G and H are parametric positive definite matrices. Then the parameters vector $\boldsymbol{\theta}$ gathers all the adjustable parameters as $\boldsymbol{\theta} = \{\theta_l,\ G,\ H\}$. We use the Q-learning method in order to update the parameters $\boldsymbol{\theta}$ to achieve the optimal solutions of the real system and improve the closed-loop performance. Figure 4 shows the difference between the MPC value $\hat{V}_N^{\boldsymbol{\theta}}$ and policy $\hat{\boldsymbol{\pi}}_N^{\boldsymbol{\theta}}$ functions with their optimal solutions computed by DP. The blue and red surfaces represent this difference at the beginning of the learning and after 500 learning steps, respectively. As it can be seen, the results are getting closer to zero as the learning proceeds. Note that the stage and terminal costs yielding a perfect match of $V^\star$ and $\boldsymbol{\pi}^\star$, as per Theorem 1, do not have a quadratic form, hence the selected MPC formulation cannot capture them exactly. The green surfaces in Figure 4 have been obtained by computing these stage and terminal costs numerically and shows the corresponding $\hat{V}_N^\star - V^\star$ and $\hat{\boldsymbol{\pi}}_N^\star - \boldsymbol{\pi}^\star$. As expected the difference is zero, modulo tiny numerical inaccuracies.

Finally, Figure 5 illustrates the closed-loop performance of the system under the MPC policy

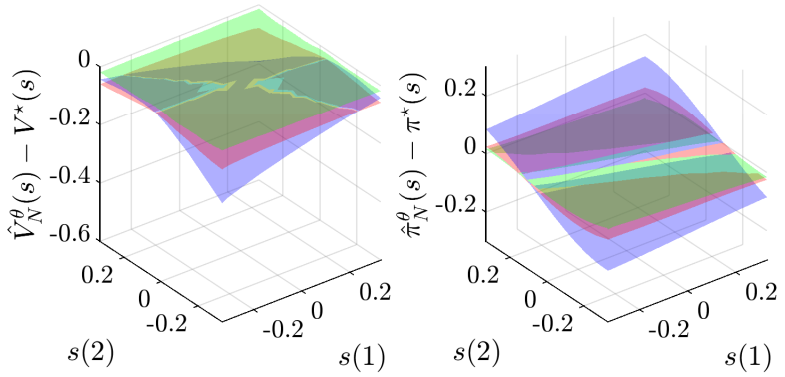


Figure 4: The difference between the MPC based parameterized value (left)\policy (right) and their optimal solutions for the beginning of the learning (blue) and after 500 learning steps (red) and the exact cost modification from theorem 1 (green).

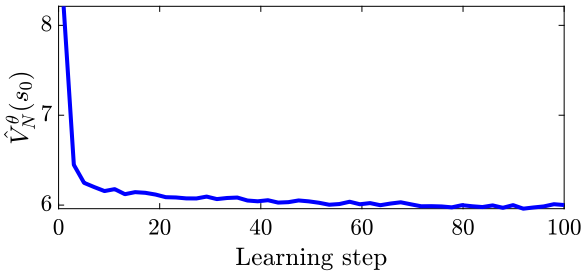


Figure 5: The MPC-based value function $\hat{V}_N^{\boldsymbol{\theta}}(\boldsymbol{s}_0)$ during the learning.

$\hat{\boldsymbol{\pi}}_N^{\boldsymbol{\theta}}$. As the closed loop cost decreases, this demonstrates that RL can be effective in tuning the MPC parameters so as to achieve the best closed-loop performance.

6.3 Learning based MPC: Tracking stage cost

In this section, we consider the cart-pendulum balancing problem shown in Figure 6 in order to illustrate the proposed method in a constrained tracking problem. The dynamics are given by:

$$(M+m)\ddot{x} + \frac{1}{2}ml\ddot{\phi}\cos\phi = \frac{1}{2}ml\dot{\phi}^2\sin\phi + u, \tag{47a}$$

$$\frac{1}{3}ml^2\ddot{\phi} + \frac{1}{2}ml\ddot{x}\cos\phi = -\frac{1}{2}mgl\sin\phi, \tag{47b}$$

where M and m are the cart mass and pendulum mass, respectively, l is the pendulum length and ϕ is its angle from the vertical axis. Force u is the control input, x is the cart displacement

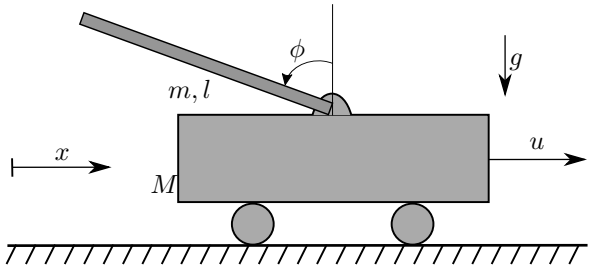


Figure 6: The cart-pendulum system. We use $M = 0.5\text{kg}$, $m = 0.2\text{kg}$, $l = 0.3\text{m}$ and $g = 9.8\text{m/s}^2$ for the simulation.

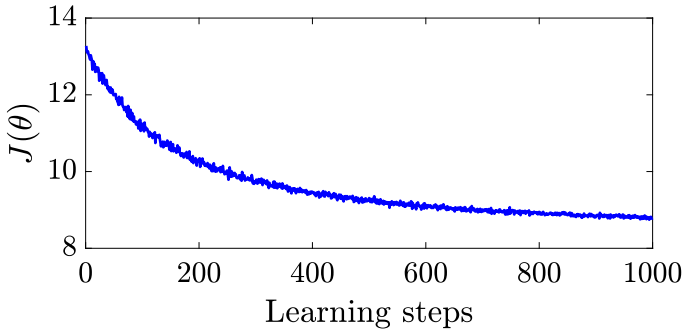


Figure 7: The closed-loop performance of the MPC scheme over RL-steps.

and g is gravity. We use the Runge-Kutta 4^{th}-order method to discretize (47) with a sampling time $\mathrm{d}t = 0.1\text{s}$ and cast it as $\boldsymbol{s}^+ = \boldsymbol{f}(\boldsymbol{s}, \boldsymbol{a}) + \boldsymbol{\xi}$, where $\boldsymbol{s} = [x, \dot{x}, \phi, \dot{\phi}]^\top$ is the state, $\boldsymbol{a} = u$ is the input, $\boldsymbol{\xi}$ is a Gaussian noise and $\boldsymbol{f}$ is a nonlinear function representing (47) in discrete time. We consider the state constraint $x \geq 0$, discount factor $\gamma = 0.95$ and the following MDP stage cost to stabilize the system at the origin while penalizing the system constraint:

$$\ell(\boldsymbol{s}, \boldsymbol{a}) = \begin{bmatrix} \boldsymbol{s} \\ \boldsymbol{a} \end{bmatrix}^\top \begin{bmatrix} I_4 & 0 \\ 0 & 0.01 \end{bmatrix} \begin{bmatrix} \boldsymbol{s} \\ \boldsymbol{a} \end{bmatrix} + \lambda \max(-x, 0), \tag{48}$$

where λ is a large constant value introduced to model the state constraint as a soft constraint. In the MPC scheme, we use the linear model $\boldsymbol{s}^+ = \hat{A}\boldsymbol{s} + \hat{B}\boldsymbol{a}$ obtained by linearizing $\boldsymbol{f}$ at the origin. We provide a parametrized quadratic stage and terminal cost and select prediction horizon $N = 20$. We use the deterministic policy gradient method to minimize the performance function $J(\boldsymbol{\theta}) := \mathbb{E}_{\boldsymbol{s}_0}[\hat{V}_N^{\boldsymbol{\theta}}(\boldsymbol{s}_0)]$, and we run a simulation for 1000 learning steps of the policy gradient method. Figure 7 shows the value function over the learning steps for a fixed initial state. This illustrates that RL successfully manages to reduce J throughout the iterates, therefore tuning MPC as desired.

Figure 8 shows the states and input trajectories of the real system corresponding to the 1000^{th} learning step of the policy gradient method. The MPC scheme with the positive definite stage

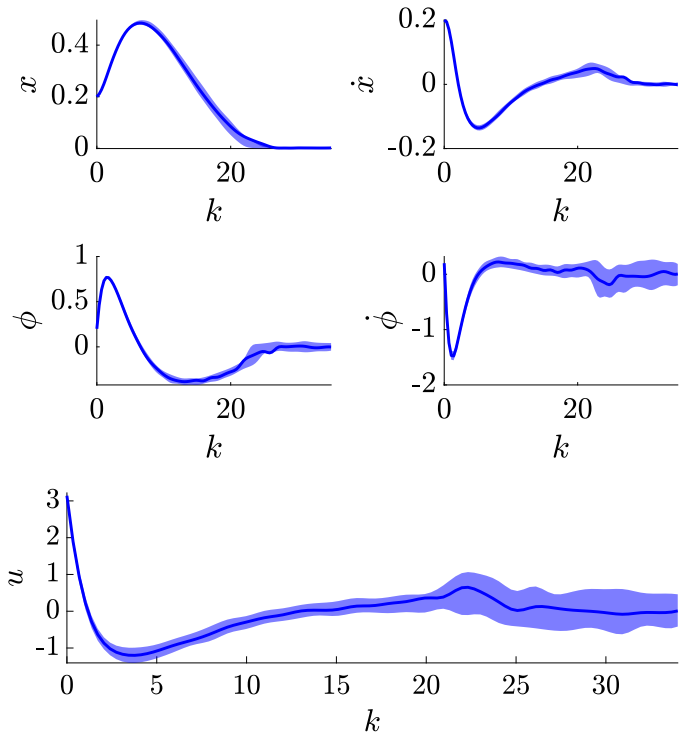


Figure 8: States and input trajectories of the real system for the last learning step.

cost and other stability conditions in the terminal cost, terminal constraint is able to deliver the stabilizing policy for the closed-loop system for the small enough model error [9]. Note that the terminal cost and constraint conditions can be relaxed for the large enough MPC horizon [26]. Figure 9 compares the state constraint violation for $x \geq 0$ in the first and the last (1000^{th}) learning step. As one can see, RL reduces the state constraint violation. Note that, we have used a common MPC formulation as (27) in this example. However, one can use robust MPC to avoid constraint violation as shown in [23].

6.4 Learning based MPC: Economic stage cost

In this example, we investigate an economic cost in the real system with bias optimality criterion. We use a parameterized MPC scheme with a parameterized storage function as a function approximator in the Q-learning algorithm. Continuously Stirred Tank Reactor (CSTR) is a common ideal reactor in chemical engineering, usually used for liquid-phase or multiphase reactions with fairly high reaction rates. The CSTR nonlinear dynamics can be

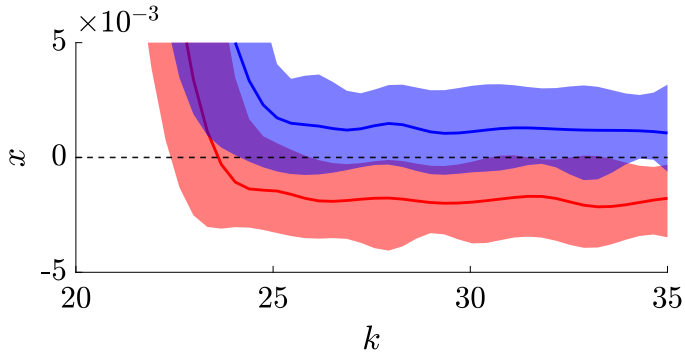


Figure 9: Violation of the state constraint $x \geq 0$ in the first step (red) and the last step (blue).

written as follows (see [27]):

$$\dot{C}_A = \frac{F}{V_R}(C_{A0} - C_A) - k_0 e^{-E/RT} C_A^2 \tag{49}$$
$$\dot{T} = \frac{F}{V_R}(T_0 - T) - \frac{\Delta H k_0}{\rho_R C_p} e^{-E/RT} C_A^2 + \frac{q}{\rho_R C_p V_R},$$

where T denotes the temperature of the reactor contents, C_A is the concentration of A in the reactor, F is the flow rate, and q is the heat rate. The remaining notation definitions and process parameter values are given in e.g., [28]. Then $\boldsymbol{s} = [C_A\,,\, T]^\top$ and $\boldsymbol{a} = [F\,,\, q]^\top$ are the state and input of the system, respectively. The input $\boldsymbol{a}$ must satisfy the following inequality:

$$[0\,,\, -2\text{e}5]^\top \leq \boldsymbol{a} \leq [10\,,\, 2\text{e}5]^\top \tag{50}$$

An economic stage cost is defined as follows:

$$\ell(\boldsymbol{s}, \boldsymbol{a}) = -\eta \underbrace{F(C_{A0} - C_A)}_{:=r} + \beta q \tag{51}$$

where η and β are positive constants, and r is the production rate. This cost maximizes the production rate and minimizes the energy consumption of the production (the second term). We consider $\eta = 1.7\text{e}4$ and $\beta = 1$ for the simulation. Sampling time 0.02h is used to discretize the system (49). We use an MPC scheme with a neural network-based storage function and parameterized stage cost and terminal cost and we denote the adjustable parameters by $\boldsymbol{\theta}$. Then we use Q-learning in order to update the parameters $\boldsymbol{\theta}$. Figure 10 (left) illustrates the value function $\hat{V}_N^{\boldsymbol{\theta}}(\boldsymbol{s}_0)$. It can be seen that the parameterized value function is decreasing during the learning. Figure 10 (right) shows the convergence of the parameters.

7 Conclusion

In this paper, we showed that a finite-horizon OCP can capture the optimal policy and value functions of any MDPs with either discounted or undiscounted cost even if we use an inexact

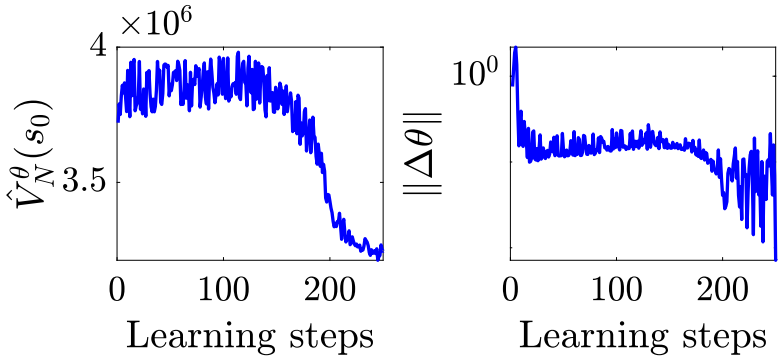


Figure 10: (Left:) The MPC-based value function $\hat{V}_N^{\boldsymbol{\theta}}(\boldsymbol{s}_0)$ during the learning .(Right:) Convergence of the norm of the parameters during the Q-learning steps.

model in the OCP. We showed that an MPC scheme can be interpreted as a particular case of the OCP where we use a deterministic model to avoid computational complexity. In practice, we proposed the use of a parameterized MPC scheme to provide a structured function approximator for the RL techniques. RL algorithms then can be used in order to tune the MPC parameters to achieve the best closed-loop performance. We verified the theorems in an LQR case and investigated some nonlinear examples to illustrate the efficiency of the method numerically.

References

[1] Martin L Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

[2] David Silver et al. "Deterministic Policy Gradient Algorithms". In: *Proceedings of the 31st International Conference on Machine Learning*. 2014, pp. 387–395.

[3] Richard S Sutton et al. "Policy gradient methods for reinforcement learning with function approximation". In: *Advances in neural information processing systems*. 2000, pp. 1057–1063.

[4] Dimitri P Bertsekas. *Dynamic programming and optimal control*. Vol. 1. 2. Athena scientific Belmont, MA, 1995.

[5] Warren B Powell. *Approximate Dynamic Programming: Solving the curses of dimensionality*. Vol. 703. John Wiley & Sons, 2007.

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

[7] Dimitri P Bertsekas. *Approximate dynamic programming*. Citeseer, 2008.

[8] Kai Arulkumaran et al. “Deep reinforcement learning: A brief survey”. In: *IEEE Signal Processing Magazine* 34.6 (2017), pp. 26–38.

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

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

[11] 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.

[12] 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.

[13] Hossein Nejatbakhsh Esfahani, Arash Bahari Kordabad, and Sébastien Gros. “Reinforcement Learning based on MPC/MHE for Unmodeled and Partially Observable Dynamics”. In: *2021 American Control Conference (ACC)*. 2021, pp. 2121–2126.

[14] Mathieu Granzotto et al. “Finite-horizon discounted optimal control: stability and performance”. In: *IEEE Transactions on Automatic Control* 66.2 (2020), pp. 550–565.

[15] Mario Zanon and Sébastien Gros. “A new dissipativity condition for asymptotic stability of discounted economic MPC”. In: *Automatica* 141 (2022), p. 110287.

[16] Mario Zanon, Sébastien Gros, and Michele Palladino. “Stability-Constrained Markov Decision Processes Using MPC”. In: *Automatica* 143 (2022), p. 110399.

[17] Sridhar Mahadevan. “Average reward reinforcement learning: Foundations, algorithms, and empirical results”. In: *Machine learning* 22.1 (1996), pp. 159–195.

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

[19] Romain Postoyan et al. “Stability analysis of discrete-time infinite-horizon optimal control with discounted cost”. In: *IEEE Transactions on Automatic Control* 62.6 (2016), pp. 2736–2749.

[20] Eric C Kerrigan and Jan M Maciejowski. “Soft constraints and exact penalty functions in model predictive control”. In: *Control 2000 Conference, Cambridge*. Citeseer. 2000, pp. 2319–2327.

[21] Arash Bahari Kordabad and Sebastien Gros. “Verification of Dissipativity and Evaluation of Storage Function in Economic Nonlinear MPC using Q-Learning”. In: *IFAC-PapersOnLine* 54.6 (2021). 7th IFAC Conference on Nonlinear Model Predictive Control NMPC 2021, pp. 308–313.

[22] Arash Bahari Kordabad et al. “Reinforcement learning based on scenario-tree MPC for ASVs”. In: *2021 American Control Conference (ACC)*. IEEE. 2021, pp. 1985–1990.

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

[24] Manuel S Santos and Jesus Vigo-Aguiar. "Analysis of a numerical dynamic programming algorithm applied to economic models". In: *Econometrica* (1998), pp. 409–426.

[25] Lars Grüne, Christopher M Kellett, and Steven R Weller. "On a discounted notion of strict dissipativity". In: *IFAC-PapersOnLine* 49.18 (2016), pp. 247–252.

[26] Ali Jadbabaie and John Hauser. "On the stability of receding horizon control with a general terminal cost". In: *IEEE Transactions on Automatic Control* 50.5 (2005), pp. 674–678.

[27] Xinchun Li et al. "Application of economic MPC to a CSTR process". In: *2016 IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC)*. 2016, pp. 685–690.

[28] Arash Bahari Kordabad and Sebastien Gros. "Q-learning of the storage function in economic nonlinear model predictive control". In: *Engineering Applications of Artificial Intelligence* 116 (2022), p. 105343.

I Safe Reinforcement Learning Using Wasserstein Distributionally Robust MPC and Chance Constraint

Postprint of [102] **Arash Bahari Kordabad**, Rafal Wisniewski, and Sebastien Gros. "Safe Reinforcement Learning Using Wasserstein Distributionally Robust MPC and Chance Constraint". In: *Submitted* (2022)

Safe Reinforcement Learning Using Wasserstein Distributionally Robust MPC and Chance Constraint

Arash Bahari Kordabad[1], Rafal Wisniewski[2], and Sebastien Gros[1]

[1]Department of Engineering Cybernetics, Norwegian University of Science and Technology (NTNU), Trondheim, Norway.
[2]Department of Electronical Systems, Aalborg University, Aalborg, Denmark.

Abstract: In this paper, we address the chance-constrained safe Reinforcement Learning (RL) problem using the function approximators based on Stochastic Model Predictive Control (SMPC) and Distributionally Robust Model Predictive Control (DRMPC). We use Conditional Value at Risk (CVaR) to measure the probability of constraint violation and safety. In order to provide a safe policy by construction, we first propose using parameterized nonlinear DRMPC at each time step. DRMPC optimizes a finite-horizon cost function subject to the worst-case constraint violation in an ambiguity set. We use a statistical ball around the empirical distribution with a radius measured by the Wasserstein metric as the ambiguity set. Unlike the sample average approximation SMPC, DRMPC provides a probabilistic guarantee of the out-of-sample risk and requires lower samples from the disturbance. Then the Q-learning method is used to optimize the parameters in the DRMPC to achieve the best closed-loop performance. Wheeled Mobile Robot (WMR) path planning with obstacle avoidance will be considered to illustrate the efficiency of the proposed method.

Keywords: Safe Reinforcement Learning, Model Predictive Control, Distributionally Robust Optimization, Chance constraint, Conditional Value at Risk, Q-learning

1 Introduction

Enforcing safety in the presence of uncertainty and stochasticity of nonlinear dynamical systems is a challenging task [1]. Chance constraints are a common way of mathematical modeling of safety that requires a user-specified upper bound for the probability of the constraint violation [2]. However, it is challenging to handle a chance constraint from the computational point of view due to its nonconvexity. Conditional Value at Risk (CVaR) [3] is a convex risk measure that has received considerable attention in decision-making problems, such as Markov Decision Processes (MDPs) [4, 5].

The theory of stochastic optimal control typically assumes that the probability distribution of the disturbance is fully known. However, this assumption may not hold in many real-world applications, and one needs to estimate the probability distribution. However, stochastic

optimization is challenging to solve, especially for non-convex problems [6]. In data-driven stochastic optimization, Sample Average Approximation (SAA) is a fundamental way to estimate the probability distribution of the random variables [7]. SAA typically needs quite an extensive data set to fulfill risk constraints accurately. Distributionally Robust Optimization (DRO) is an alternative that overcomes this problem. DRO tackles stochastic optimization by considering the worst-case distribution in an ambiguity set. There are several ways to construct ambiguity sets, e.g., moment ambiguity [8], Prohorov-based ball [9], Kullback–Leibler divergence-based ball [10] and Wasserstein-based ball [11]. The Wasserstein-based ball is a statistical ball in the space of probability distributions around the empirical distribution such that the radius of this ball is measured using Wasserstein distance. Then the radius of the ball represents the conservatism of the DRO problem. Unlike the SAA method, Wasserstein DRO provides a probabilistic guarantee based on finite samples in a tractable formulation [12].

Model Predictive Control (MPC) is an optimization-based control approach operating with a receding horizon [13]. MPC employs a (possibly inaccurate) model of the real system dynamics to produce an input-state sequence over a given finite horizon. The resulting trajectory optimizes a given cost function while explicitly enforcing the system constraints. The optimization problem is solved at each time instance based on the current system state, and the first input of the optimal solution is applied to the system. Due to the finite-horizon scheme and (possibly) model mismatch, MPC usually delivers a reasonable but suboptimal approximation of the optimal policy. This paper uses the DRO in the chance-constrained nonlinear MPC. This approach has been known as Distributionally Robust MPC (DRMPC) [14].

Reinforcement Learning (RL) is a technique for solving problems involving MDPs. RL typically requires a function approximator to approximate the optimal policy, value function, or action-value function. For instance, Q-learning has been used in [15] for unmanned vehicle applications. In [16], the comparison of MPC and RL has been studied in the distributed setting. Recently, MPC has been used as a structured function approximator for RL algorithms. In this method, a parameterized MPC scheme is used in order to generate policy and/or value functions of the real system. Then RL algorithms can be used to adjust the MPC parameters to achieve the best closed-loop performance. The combination of MPC and RL has been proposed and justified in [17], where it is shown that an MPC scheme can theoretically generate the optimal policy and value functions for a given system even if the MPC model is inaccurate. Recent research have further developed and demonstrated this approach [18, 19].

Related works. In [14], the authors have proposed to use DRMPC to utilize its benefits for motion control. A DRMPC has been applied to the multi-area dynamic optimal power flow in [20] to better hedge the uncertainties of distributed generation and loads. For the Gaussian processes, a learning-based DRMPC has been proposed in [21]. A learning-Based DRMPC has been developed for chance-constrained Markovian switching systems with unknown switching probabilities in [22]. The authors have shown that this framework provides mean-square stability of the system without requiring explicit knowledge of the transition probabilities. In [23], a DRO has been proposed for chance-constrained data-enabled predictive control with stochastic linear time-invariant systems. In [24], a DRMPC algorithm has been presented for spacecraft circular orbital rendezvous and docking problems. A soft-constrained DRMPC has been proposed for linear systems in [25].

A robust MPC scheme has been used as a function approximator for safe RL in [26]. Control Barrier Functions (CBF) have been used in the safe RL context in [27]. A safe RL-CBF framework has been developed to guarantee safety and improve exploration in [28]. Probabilistic safety in learning-based control methods has been provided in [29] based on probabilistic model predictive safety certification. In [30], the safe RL problem is formulated as a constrained MDP. Then a Lyapunov approach has been proposed to solve it.

Contributions. There are a limited number of data from uncertainties and disturbances available in many real stochastic systems. Therefore, traditional methods such as SAA cannot accurately estimate the distribution of these random variables. An accurate distribution may be more important for safety-critical systems to design a safe controller for the system. In this paper, we propose to use a parameterized nonlinear DRMPC based on the Wasserstein metric as a function approximator for RL in order to generate a family of policies that are safe by construction. DRMPC is subject to the chance constraint, approximated by the CVaR risk measure. We reformulate Wasserstein DRMPC as a tractable optimization. Then we use the Q-learning technique to optimize the parameters of the DRMPC scheme to achieve the best closed-loop performance among the safe policies.

Organization. The paper is structured as follows. Section 2 details safe RL and chance constraints. Section 3 provides safe policies based on the SMPC scheme, evaluated using the SAA method. Moreover, we formulate CVaR as a convex approximator of chance constraints. Section 4 formulates a tractable DRMPC scheme and provides out-of-sample guarantees. Section 5 details Q-learning as an efficient way to optimize the parameters of the DRMPC scheme. Section 6 provides a numerical simulation and section 7 delivers a conclusion.

Notation. We denote the set of real numbers, non-negative real numbers, extended real numbers, non-negative integers, and natural numbers by $\mathbb{R}$, $\mathbb{R}_{\geq 0}$, $\bar{\mathbb{R}} := \mathbb{R} \cup \{-\infty, \infty\}$, $\mathbb{Z}$ and $\mathbb{N}$, respectively, while $\mathbb{I}_{i:j}$ refers to the set $\{i, i+1, \ldots, j\}$. Vectors in $\mathbb{R}^n$ are denoted by the bold letters, e.g., $\boldsymbol{a}$. $\langle \boldsymbol{x}, \boldsymbol{y} \rangle := \boldsymbol{x}^\top \boldsymbol{y}$ denotes the usual inner product for given vectors $\boldsymbol{x}, \boldsymbol{y}$. A function $f : \mathbb{R}^n \to \bar{\mathbb{R}}$ is proper if $f(\boldsymbol{x}) < +\infty$ for at least one $\boldsymbol{x}$ and $f(\boldsymbol{x}) > -\infty$ for every $\boldsymbol{x}$ in $\mathbb{R}^n$. The conjugate function of a function $f : \mathbb{R}^n \to \mathbb{R}$ is denoted by $[f]^\star(\boldsymbol{y}) := \sup_{\boldsymbol{x} \in \mathbb{R}^n} \langle \boldsymbol{x}, \boldsymbol{y} \rangle - f(\boldsymbol{x})$. Support function of set $\mathbb{W}$ is defined as $\Xi_{\mathbb{W}}(\boldsymbol{x}) := \sup_{\boldsymbol{y} \in \mathbb{W}} \langle \boldsymbol{x}, \boldsymbol{y} \rangle$. For scalar a, we define $(a)_+ := \max\{a, 0\}$.

2 Safe Reinforcement Learning

In this section, we formulate safe Reinforcement Learning (RL) using chance constraints. Let us consider the following (possibly) nonlinear discrete-time stochastic dynamical system:

$$\boldsymbol{s}_{k+1} = \boldsymbol{f}(\boldsymbol{s}_k, \boldsymbol{a}_k, \boldsymbol{w}_k) \tag{1}$$

where $k \in \mathbb{Z}$ is the time index, $\boldsymbol{s}_k \in \mathbb{X} \subseteq \mathbb{R}^n$ is the system state, $\boldsymbol{a}_k \in \mathbb{U} \subseteq \mathbb{R}^m$ is the control input, $\boldsymbol{w}_k \in \mathbb{W} \subset \mathbb{R}^d$ is a random variable representing the stochastic disturbance of the system and $\boldsymbol{f} : \mathbb{R}^{n+m+d} \to \mathbb{R}^n$ is a Borel-measurable function. Note that the notation in (1) is standard in the literature of control, while the RL literature typically uses the conditional

probability notation $\mathbb{P}[\boldsymbol{s}_{k+1}|\boldsymbol{s}_k, \boldsymbol{a}_k]$ for the state transition. We then make the following assumption on $\mathbb{W}$.

Assumption 1. *The disturbance set $\mathbb{W}$ is convex and closed.*

We will use this assumption in the rest of the paper to reformulate DRO as finite convex programming.

A deterministic policy $\boldsymbol{\pi} : \mathbb{X} \to \mathbb{U}$ maps the state space to the input space and determines how to choose input $\boldsymbol{a}_k$ at each state $\boldsymbol{s}_k$. We aim to find the optimal safe policy $\boldsymbol{\pi}^\star$, given by the solution of:

$$\boldsymbol{\pi}^\star \in \arg\min_{\boldsymbol{\pi}} \mathbb{E}_{\boldsymbol{s}_0 \sim \mu_0} \left[V^{\boldsymbol{\pi}}(\boldsymbol{s}_0) \right] \tag{2}$$

where μ_0 is the probability distribution of the initial state $\boldsymbol{s}_0$ and $V^{\boldsymbol{\pi}} : \mathbb{X} \to \mathbb{R}$ is the value function associated with the policy $\boldsymbol{\pi}$, defined as follows:

$$V^{\boldsymbol{\pi}}(\boldsymbol{s}_0) := \mathbb{E}_{\boldsymbol{w}} \left[\sum_{k=0}^{\infty} \gamma^k L(\boldsymbol{s}_k, \boldsymbol{\pi}(\boldsymbol{s}_k)) \right], \tag{3a}$$

$$\text{s.t. } \boldsymbol{s}_{k+1} = \boldsymbol{f}(\boldsymbol{s}_k, \boldsymbol{\pi}(\boldsymbol{s}_k), \boldsymbol{w}_k), \ \forall k \in \mathbb{Z} \tag{3b}$$

$$\mathbb{P}[\boldsymbol{s}_{k+i} \in \mathcal{S} | \boldsymbol{s}_k] \geq \alpha, \ \forall i \in \mathbb{I}_{1:I}, \forall k \in \mathbb{Z} \tag{3c}$$

where $L : \mathbb{X} \times \mathbb{U} \to \mathbb{R}$ is the stage cost, $\gamma \in (0, 1]$ is the discount factor, $\mathcal{S} \subseteq \mathbb{X}$ is a safe set and $\alpha \in (0, 1)$ is a user-chosen confidence level. The chance-constraint (3c) guarantees probabilistic safety of state trajectories $\boldsymbol{s}_{k+i}$ for a finite-horizon with length $I \in \mathbb{N}$ given state $\boldsymbol{s}_k$ at each time instance k. In fact, we generalize the common chance constraint in the literature not only to be satisfied for one step ahead but also to be satisfied for a finite horizon ahead at every time instance. This paper provides such policies using both an SMPC scheme and a DRMPC scheme with horizon I.

The safe set $\mathcal{S}$ can be defined as follows:

$$\mathcal{S} = \{\boldsymbol{s} \in \mathbb{X} | h_j(\boldsymbol{s}) \leq 0, \forall j \in \mathbb{I}_{1:J}\} \tag{4}$$

where $h_j : \mathbb{X} \to \mathbb{R}$ specifies a state constraint and J is the number of constraints. For the sake of simplicity and in order to avoid the complexity of joint constraints, we consider the following individual constraint:

$$\mathbb{P}[\max_j h_j(\boldsymbol{s}_{k+i}) \leq 0 | \boldsymbol{s}_k] \geq \alpha, \quad \forall i \in \mathbb{I}_{1:I} \tag{5}$$

Then one can verify that using (4), (5) implies (3c).

Assumption 2. *Each function $-h_j$ is proper, convex and lower semi-continuous functions.*

In the next section, we will use an SMPC scheme in order to provide a family of safe policies.

3 Stochastic MPC-based Policy

In the RL context, we consider a family of the parameterized policy given by $\boldsymbol{\pi_\theta}$ with parameter vector $\boldsymbol{\theta} \in \mathbb{R}^p$ and seek the best parameters $\boldsymbol{\theta}^\star$ that provide the best closed-loop performance. More specifically, (2) is reformulated as:

$$\boldsymbol{\theta}^\star \in \arg\min_{\boldsymbol{\theta}} \mathbb{E}_{\boldsymbol{s}_0 \sim \mu_0} \left[V^{\boldsymbol{\pi_\theta}}(\boldsymbol{s}_0) \right] \tag{6}$$

Instead of solving (3) directly, we use a function approximator based on the MPC scheme to extract policy $\boldsymbol{\pi_\theta}$ that satisfies (3c) by construction for all parameters $\boldsymbol{\theta}$.

More specifically, consider the following parameterized SMPC at time instant k:

$$\min_{\boldsymbol{a},\boldsymbol{s}} \ \mathbb{E}\left[T_{\boldsymbol{\theta}}(\boldsymbol{s}_{I+k|k}) + \sum_{i=0}^{I-1} l_{\boldsymbol{\theta}}(\boldsymbol{s}_{i+k|k}, \boldsymbol{a}_{i+k|k}) \,\middle|\, \boldsymbol{s}_{k|k} \right], \tag{7a}$$

$$\text{s.t.} \ \ \boldsymbol{s}_{i+k+1|k} = \boldsymbol{f}(\boldsymbol{s}_{i+k|k}, \boldsymbol{a}_{i+k|k}, \boldsymbol{w}_i), \forall i \in \mathbb{I}_{0:I-1} \tag{7b}$$

$$\mathbb{P}[\max_j h_j(\boldsymbol{s}_{k+i|k}) \leq 0] \geq \alpha, \forall i \in \mathbb{I}_{1:I} \tag{7c}$$

$$\boldsymbol{a}_{i+k|k} \in \mathbb{U}, \qquad \boldsymbol{s}_{k|k} = \boldsymbol{s}_k, \tag{7d}$$

where $T_{\boldsymbol{\theta}} : \mathbb{X} \to \mathbb{R}$ and $l_{\boldsymbol{\theta}} : \mathbb{X} \times \mathbb{U} \to \mathbb{R}$ is the parameterized terminal cost and stage cost, respectively. This parameterization allows one to provide a family of policies that are safe for all $\boldsymbol{\theta} \in \mathbb{R}^p$. Then by tuning the parameters $\boldsymbol{\theta}$ and reshaping the cost function and MPC-scheme, one can achieve the best closed-loop performance. Decision variables $\boldsymbol{a} = \{\boldsymbol{a}_{k|k}, \ldots, \boldsymbol{a}_{I+k-1|k}\}$ and $\boldsymbol{s} = \{\boldsymbol{s}_{k|k}, \ldots, \boldsymbol{s}_{I+k|k}\}$ are the input and state sequence, respectively. Then the parameterized policy $\boldsymbol{\pi_\theta}$ at time instance k is extracted as follows:

$$\boldsymbol{\pi}_{\boldsymbol{\theta}}^{\mathrm{SMPC}}(\boldsymbol{s}_k) = \boldsymbol{a}_{k|k}^\star(\boldsymbol{\theta}, \boldsymbol{s}_k) \tag{8}$$

where $\boldsymbol{a}_{k|k}^\star$ is the solution of SMPC (7) corresponding to the first input $\boldsymbol{a}_{k|k}$.

The use of parameterized MPC scheme as a function approximator in order to capture the optimal policy and value function was proposed and justified in [17]. Moreover, the authors showed that RL methods such as Q-learning and policy gradient can be used in order to adjust the MPC scheme parameters and achieve the best closed-loop performance.

We ought to stress here that MPC scheme (7) provides a family of safe policy for all parameters $\boldsymbol{\theta}$ based on the best state-input sequence that minimizes a finite-horizon parameterized cost function of an MPC scheme. Obviously, a richer parameterization in the stage cost and terminal cost provides a more extensive set of policies. Then tuning the parameters $\boldsymbol{\theta}$ leads us to get the optimal policy among the provided policy families. We will detail Q-learning as a practical way of adjusting the parameters in Section 5.

In order to tackle the chance constraint (7c), a natural measure of risk is value-at-risk VaR. For a random variable r and confidence level α, VaR_α is defined as follows:

$$\mathrm{VaR}_\alpha(r) := \min\{\eta \in \mathbb{R} \,|\, \mathbb{P}(r \leq \eta) \geq \alpha\} \tag{9}$$

In fact, VaR represents the worst-case loss with probability α. Then one can show that:

$$\mathrm{VaR}_\alpha(r) \leq 0 \Leftrightarrow \mathbb{P}(r \leq 0) \geq \alpha \tag{10}$$

Unfortunately, VaR is, in general, non-convex, and optimizing models involving VaR are numerically intractable for high-dimensional, non-normal distributions.

An alternative measure of risk is conditional value-at-risk CVaR, defined as follows:

$$\mathrm{CVaR}_\alpha(r) := \min_{\eta \in \mathbb{R}} \mathbb{E}\left[\eta + \frac{(r-\eta)_+}{1-\alpha}\right] \tag{11}$$

Indeed, CVaR is a *coherent* risk measure that satisfies conditions such as convexity and monotonicity [3]. Risk management with CVaR functions can be done quite efficiently. CVaR can be formulated with convex and linear programming methods, while VaR is comparably complicated to optimize. Detailed benefits and concepts of CVaR can be found in, e.g., [31].

It can be shown that for $\alpha \to 1$, CVaR can approximate VaR more accurately. i.e.:

$$\lim_{\alpha \to 1} \mathrm{CVaR}_\alpha(r) - \mathrm{VaR}_\alpha(r) = 0\,. \tag{12}$$

Note that in engineering applications, we often are interested in a very low probability of failure ($\alpha \to 1$). Then using CVaR, with the numerical and mathematical benefits, imposes a very low conservative on the problem. Using CVaR, MPC (7) can be approximated as follows:

$$\min_{\boldsymbol{a},\boldsymbol{s}} \ \mathbb{E}\left[T_{\boldsymbol{\theta}}(\boldsymbol{s}_{I+k|k}) + \sum_{i=0}^{I-1} l_{\boldsymbol{\theta}}(\boldsymbol{s}_{i+k|k}, \boldsymbol{a}_{i+k|k}) \,\middle|\, \boldsymbol{s}_{k|k}\right], \tag{13a}$$

$$\text{s.t. } \boldsymbol{s}_{i+k+1|k} = \boldsymbol{f}(\boldsymbol{s}_{i+k|k}, \boldsymbol{a}_{i+k|k}, \boldsymbol{w}_i), \forall i \in \mathbb{I}_{0:I-1} \tag{13b}$$

$$\mathrm{CVaR}_\alpha(\max_j h_j(\boldsymbol{s}_{k+i|k})) \leq 0, \forall i \in \mathbb{I}_{1:I} \tag{13c}$$

$$\boldsymbol{a}_{i+k|k} \in \mathbb{U}, \qquad \boldsymbol{s}_{k|k} = \boldsymbol{s}_k, \tag{13d}$$

At each time k we first consider N_s , independent and identically distributed (i.i.d.) samples of the disturbance $\boldsymbol{w}_i$ and we denote these samples by $\boldsymbol{w}_i^m$, $i \in \mathbb{I}_{1:I}$ $m \in \mathbb{I}_{1:N_s}$. Then N_s scenarios are described as follows:

$$\boldsymbol{s}_{k+i|k}^m = \boldsymbol{f}(\boldsymbol{s}_{k+i-1|k}^m, \boldsymbol{a}_{k+i-1|k}^m, \boldsymbol{w}_i^m) \tag{14}$$

where $\boldsymbol{s}_{k+i|k}^m$ and $\boldsymbol{a}_{k+i|k}^m$ are the predicted state and input for m^{th} scenario at time $k+i$ given time k. We then define auxiliary variables x_i^m for $i \in \mathbb{I}_{1:I}$, $m \in \mathbb{I}_{1:N_s}$ in order to approximate CVaR, in (13c), in the following tractable Linear Programming (LP), $\forall i \in \mathbb{I}_{1:I}$:

$$\mathrm{CVaR}_\alpha(\max_j h_j(\boldsymbol{s}_{k+i|k})) \approx \tag{15a}$$

$$\min_{\eta_i, \boldsymbol{x}_i} \ \eta_i + \frac{1}{(1-\alpha)N_s} \sum_{m=1}^{N_s} x_i^m \tag{15b}$$

$$\text{s.t. } \max_j h_j(\boldsymbol{s}_{k+i|k}^m) - \eta_i \leq x_i^m, \ \forall m \in \mathbb{I}_{1:N_s} \tag{15c}$$

$$0 \leq x_i^m, \ \forall m \in \mathbb{I}_{1:N_s} \tag{15d}$$

where $\boldsymbol{x}_i = \{x_i^m\}_{m=1}^{N_s}$ and $\eta_i \in \mathbb{R}$. In [32], it has been shown that for $N_s \to \infty$ the approximation in (15) will converge to its exact value with probability one.

Substitution of (15) into (13) and using SAA, SMPC (13) reads as:

$$\min_{\boldsymbol{s},\boldsymbol{a},\boldsymbol{\eta},\boldsymbol{x}} \ \frac{1}{N_s}\sum_{m=1}^{N_s}\Big(T_{\boldsymbol{\theta}}(\boldsymbol{s}_{k+I|k}^m) + \sum_{i=0}^{I-1} l_{\boldsymbol{\theta}}(\boldsymbol{s}_{k+i|k}^m, \boldsymbol{a}_{k+i|k}^m)\Big) \tag{16a}$$

$$\text{s.t.} \quad \boldsymbol{s}_{k+i|k}^m = \boldsymbol{f}(\boldsymbol{s}_{k+i-1|k}^m, \boldsymbol{a}_{k+i-1|k}^m, \boldsymbol{w}_i^m), \quad \forall m \in \mathbb{I}_{1:N_s}, \forall i \in \mathbb{I}_{1:I} \tag{16b}$$

$$\eta_i + \frac{1}{(1-\alpha)N_s}\sum_{m=1}^{N_s} x_i^m \le 0, \forall i \in \mathbb{I}_{1:I} \tag{16c}$$

$$\max_j h_j(\boldsymbol{s}_{k+i|k}^m) - \eta_i \le x_i^m, \ \forall m \in \mathbb{I}_{1:N_s}, \forall i \in \mathbb{I}_{1:I} \tag{16d}$$

$$\boldsymbol{a}_{i+k|k}^m \in \mathbb{U}, \ 0 \le x_i^m, \ \boldsymbol{s}_{k|k}^m = \boldsymbol{s}_k, \quad \forall m \in \mathbb{I}_{1:N_s}, \forall i \in \mathbb{I}_{1:I} \tag{16e}$$

where $\boldsymbol{\eta} = \{\eta_i\}_{i=1}^I$, $\boldsymbol{x} = \{\boldsymbol{x}_i\}_{i=1}^I$.

From a theoretical point of view, SMPC (16) requires $N_s \to \infty$ in order to provide an accurate approximation of the original MPC (13). In the next section, we will introduce DRMPC scheme to overcome this problem.

4 Distributionally Robust MPC-based Policy

In order to tackle the limited distributional information issue with finite-many sampling, we use Distributionally Robust Optimization (DRO) in the chance constraint of the MPC scheme. In this section, we suppress the subscript i, denoting the horizon index, to simplify the notations.

The core idea of the theoretical developments in this section was proposed in [12] for general optimization problems. For the sake of clarity, in the context of learning-based MPC, we detail these developments in this section.

We use the Wasserstein metric to define an ambiguity set as a ball around the empirical distribution $\hat{\mathcal{P}}$. Then the optimization will be solved with respect to the worst-case distribution in the ambiguity set. Empirical distribution $\hat{\mathcal{P}}$, evaluated from N_s i.i.d. samples $\{\boldsymbol{w}^m\}_{m=1}^{N_s}$, is defined as follows:

$$\hat{\mathcal{P}} = \frac{1}{N_s}\sum_{m=1}^{N_s} \delta_{\boldsymbol{w}^m} \tag{17}$$

where $\delta_{\boldsymbol{w}}$ is the Dirac measure concentrated at $\boldsymbol{w}$. Then we define the Wasserstein ball $\mathbb{D}$ around the empirical distribution $\hat{\mathcal{P}}$ as the ambiguity set as follows:

$$\mathbb{D} := \{\mathcal{P} \in \mathcal{P}(\mathbb{W}) \,|\, d_{\mathrm{W}}(\mathcal{P}, \hat{\mathcal{P}}) \le \epsilon\} \tag{18}$$

where $\mathcal{P}(\mathbb{W})$ denotes the set of Borel probability measures on the support $\mathbb{W}$, $\epsilon \geq 0$ is the radius of the ball and $d_{\mathrm{W}} : \mathcal{P}(\mathbb{W}) \times \mathcal{P}(\mathbb{W}) \to \mathbb{R}_{\geq 0}$ is the Wasserstein metric, defined as follows:

$$d_{\mathrm{W}}(\mathcal{P}_1, \mathcal{P}_2) := \min_{\kappa \in \mathcal{P}(\mathbb{W}^2)} \left\{ \int_{\mathbb{W}^2} \|\boldsymbol{w}_1 - \boldsymbol{w}_2\| \mathrm{d}\kappa(\boldsymbol{w}_1, \boldsymbol{w}_2) \right.$$
$$\left. \Big| \, \Pi^l \kappa = \mathcal{P}_l, l = 1, 2 \right\} \tag{19}$$

for all distributions $\mathcal{P}_1, \mathcal{P}_2 \in \mathcal{P}(\mathbb{W})$ where $\Pi^l \kappa$ denotes the lth marginal of the transportation plan κ for $l = 1, 2$ [33]. Indeed, the Wasserstein distance of $\mathcal{P}_1$ and $\mathcal{P}_2$ can be interpreted as the minimum transportation cost for moving the probability mass from $\mathcal{P}_1$ to $\mathcal{P}_2$. Then distributionally robust optimization minimizes the worst-case cost over all the distributions in the ambiguity set. Distributionally robust constraint (13c) can be written as follows:

$$\sup_{\mathcal{P} \in \mathbb{D}} \mathrm{CVaR}_{\alpha}^{\mathcal{P}}(\max_j h_j(\boldsymbol{s})) \leq 0 \tag{20}$$

For the sake of simplicity, we define a new variable $c := \max_j h_j(\boldsymbol{s})$. We then recall the definition of CVaR:

$$\mathrm{CVaR}_{\alpha}^{\mathcal{P}}(c) = \min_{\eta} \mathbb{E}^{\mathcal{P}} \left[\eta + \frac{1}{1-\alpha}(c - \eta)_+ \right] \tag{21}$$

We then use the minimax inequality for (20):

$$\begin{aligned} \sup_{\mathcal{P} \in \mathbb{D}} \mathrm{CVaR}_{\alpha}^{\mathcal{P}}(c) \leq & \min_{\eta} \sup_{\mathcal{P} \in \mathbb{D}} \mathbb{E}^{\mathcal{P}} \left[\eta + \frac{1}{1-\alpha}(c - \eta)_+ \right] \\ = & \min_{\eta} \eta + \frac{1}{1-\alpha} \sup_{\mathcal{P} \in \mathbb{D}} \mathbb{E}^{\mathcal{P}} \left[(c - \eta)_+ \right] \end{aligned} \tag{22}$$

on the other hand:

$$\begin{aligned} \sup_{\mathcal{P} \in \mathbb{D}} \mathbb{E}^{\mathcal{P}} \left[(c - \eta)_+ \right] = & \sup_{\mathcal{P} \in \mathcal{P}(\mathbb{W})} \mathbb{E}^{\mathcal{P}} \left[(c - \eta)_+ \right] \\ & \text{s.t.} \quad d_{\mathrm{W}}(\mathcal{P}, \hat{\mathcal{P}}) \leq \epsilon \end{aligned} \tag{23}$$

then using the Lagrangian function for the constrained optimization (23):

$$\begin{aligned} & \sup_{\mathcal{P} \in \mathbb{D}} \mathbb{E}^{\mathcal{P}} \left[(c - \eta)_+ \right] = \\ & \quad \sup_{\mathcal{P} \in \mathcal{P}(\mathbb{W})} \inf_{\lambda \geq 0} \left\{ \mathbb{E}^{\mathcal{P}} \left[(c - \eta)_+ \right] + \lambda(\epsilon - d_{\mathrm{W}}(\mathcal{P}, \hat{\mathcal{P}})) \right\} \end{aligned} \tag{24}$$

where $\lambda \in \mathbb{R}$ is the Lagrange multiplier. Using Theorem 1 in [34]:

$$\begin{aligned}
&\sup_{\mathcal{P}\in\mathcal{P}(\mathbb{W})} \inf_{\lambda\geq 0} \left\{ \mathbb{E}^{\mathcal{P}}\left[(c-\eta)_+\right] + \lambda(\epsilon - d_{\mathrm{W}}(\mathcal{P},\hat{\mathcal{P}})) \right\} \\
&= \inf_{\lambda\geq 0} \left\{ \lambda\epsilon + \sup_{\mathcal{P}\in\mathcal{P}(\mathbb{W})} \{\mathbb{E}^{\mathcal{P}}[(c-\eta)_+] - \lambda d_{\mathrm{W}}(\mathcal{P},\hat{\mathcal{P}})\} \right\} \\
&= \inf_{\lambda\geq 0} \left\{ \lambda\epsilon + \frac{1}{N_s}\sum_{m=1}^{N_s} \sup_{\boldsymbol{w}\in\mathbb{W}} \left\{ (c-\eta)_+ - \lambda\|\boldsymbol{w}-\boldsymbol{w}^m\| \right\} \right\}
\end{aligned} \tag{25}$$

In fact, the first equality in (25) follows from the strong duality that has been shown in [34], and the second equality holds because $\mathcal{P}(\mathbb{W})$ contains all the Dirac distributions supported on $\mathbb{W}$.

Introducing a new auxiliary variable y^m, we can rewrite (25) as follows:

$$\inf_{\lambda,\boldsymbol{y}} \quad \lambda\epsilon + \frac{1}{N_s}\sum_{m=1}^{N_s} y^m \tag{26a}$$

$$\text{s.t.} \quad \sup_{\boldsymbol{w}\in\mathbb{W}} \{(c-\eta)_+ - \lambda\|\boldsymbol{w}-\boldsymbol{w}^m\|\} \leq y^m,\ \forall m \in \mathbb{I}_{1:N_s} \tag{26b}$$

$$0 \leq \lambda \tag{26c}$$

where $\boldsymbol{y} = \{y^m\}_{m=1}^{N_s}$. From the definition of dual norm, we decompose the expression inside $(\cdot)_+$ in constraint (26b) as follows [12]:

$$\sup_{\boldsymbol{w}\in\mathbb{W}} \left\{ \min_{\|\boldsymbol{\xi}_1^m\|_\star\leq\lambda} -\langle \boldsymbol{\xi}_1^m, \boldsymbol{w}-\boldsymbol{w}^m\rangle \right\} \leq y^m \tag{27a}$$

$$\sup_{\boldsymbol{w}\in\mathbb{W}} \left\{ \min_{\|\boldsymbol{\xi}_2^m\|_\star\leq\lambda} -\langle \boldsymbol{\xi}_2^m, \boldsymbol{w}-\boldsymbol{w}^m\rangle + c \right\} - \eta \leq y^m \tag{27b}$$

where $\|\cdot\|_\star := \sup_{\|\boldsymbol{\xi}\|\leq 1} \langle \cdot, \boldsymbol{\xi}\rangle$ is the dual norm. Since $\{\boldsymbol{\xi} \mid \|\boldsymbol{\xi}\|_\star \leq \lambda\}$ is a convex set, we use the minimax inequality, and (27) reads:

$$\min_{\|\boldsymbol{\xi}_1^m\|_\star\leq\lambda} \sup_{\boldsymbol{w}\in\mathbb{W}} \{-\langle \boldsymbol{\xi}_1^m, \boldsymbol{w}-\boldsymbol{w}^m\rangle\} \leq y^m \tag{28a}$$

$$\min_{\|\boldsymbol{\xi}_2^m\|_\star\leq\lambda} \sup_{\boldsymbol{w}\in\mathbb{W}} \{-\langle \boldsymbol{\xi}_2^m, \boldsymbol{w}-\boldsymbol{w}^m\rangle + c\} - \eta \leq y^m \tag{28b}$$

Then optimization (26) can be written as follows:

$$\begin{aligned}
\inf_{\lambda,\boldsymbol{y},\boldsymbol{\xi}_1,\boldsymbol{\xi}_2} \quad & \lambda\epsilon + \frac{1}{N_s}\sum_{m=1}^{N_s} y^m \\
\text{s.t.} \quad & \sup_{\boldsymbol{w}\in\mathbb{W}} \{-\langle \boldsymbol{\xi}_1^m, \boldsymbol{w}-\boldsymbol{w}^m\rangle\} \leq y^m,\ \forall m \in \mathbb{I}_{1:N_s} && \text{(29a)} \\
& \sup_{\boldsymbol{w}\in\mathbb{W}} \{-\langle \boldsymbol{\xi}_2^m, \boldsymbol{w}-\boldsymbol{w}^m\rangle + c\} - \eta \leq y^m,\ \forall m \in \mathbb{I}_{1:N_s} && \text{(29b)} \\
& \|\boldsymbol{\xi}_1^m\|_\star \leq \lambda, \quad \|\boldsymbol{\xi}_2^m\|_\star \leq \lambda,\ \forall m \in \mathbb{I}_{1:N_s} && \text{(29c)}
\end{aligned}$$

where $\boldsymbol{\xi}_l = \{\boldsymbol{\xi}_l^m\}_{m=1}^{N_s}$ for $l = 1, 2$. Changing $\boldsymbol{\xi}_l^m$ to $-\boldsymbol{\xi}_l^m$, we have:

$$\inf_{\lambda, y, \boldsymbol{\xi}_1, \boldsymbol{\xi}_2, \boldsymbol{v}} \quad \lambda\epsilon + \frac{1}{N_s}\sum_{m=1}^{N_s} y^m \tag{30a}$$

$$\text{s.t.} \quad -\langle \boldsymbol{\xi}_1^m, \boldsymbol{w}^m \rangle + \Xi_{\mathbb{W}}(\boldsymbol{\xi}_1^m) \leq y^m, \ \forall m \in \mathbb{I}_{1:N_s} \tag{30b}$$

$$[-c]^\star(\boldsymbol{\xi}_2^m - \boldsymbol{v}^m) + \Xi_{\mathbb{W}}(\boldsymbol{v}^m) - \langle \boldsymbol{\xi}_2^m, \boldsymbol{w}^m \rangle - \eta \leq y^m, \ \forall m \in \mathbb{I}_{1:N_s} \tag{30c}$$

$$\|\boldsymbol{\xi}_1^m\|_\star \leq \lambda, \quad \|\boldsymbol{\xi}_2^m\|_\star \leq \lambda, \ \forall m \in \mathbb{I}_{1:N_s} \tag{30d}$$

where $[-c]^\star$ is the conjugate of $-c$ that is calculated at $\boldsymbol{\xi}_2^m - \boldsymbol{v}^m$. Note that under assumptions 1 and 2, (30) is a finite convex program [12]. Restoring the index i, the DRMPC scheme based on the Wasserstein metric reads as follows:

$$\min_{\substack{\boldsymbol{s}, \boldsymbol{a}, \boldsymbol{\eta}, \\ \boldsymbol{\lambda}, \boldsymbol{y}, \boldsymbol{\xi}, \boldsymbol{v}}} \quad \frac{1}{N_s}\sum_{m=1}^{N_s}\Big(T_{\boldsymbol{\theta}}(\boldsymbol{s}_{k+I|k}^m) + \sum_{i=0}^{I-1} l_{\boldsymbol{\theta}}(\boldsymbol{s}_{k+i|k}^m, \boldsymbol{a}_{k+i|k}^m)\Big) \tag{31a}$$

$$\text{s.t.} \quad \boldsymbol{s}_{k+i|k}^m = \boldsymbol{f}(\boldsymbol{s}_{k+i-1|k}^m, \boldsymbol{a}_{k+i-1|k}^m, \boldsymbol{w}_i^m), \quad \forall m \in \mathbb{I}_{1:N_s}, \forall i \in \mathbb{I}_{1:I} \tag{31b}$$

$$\eta_i + \frac{1}{1-\alpha}\left[\lambda_i\epsilon_i + \frac{1}{N_s}\sum_{m=1}^{N_s} y_i^m\right] \leq 0, \ \forall i \in \mathbb{I}_{1:I} \tag{31c}$$

$$-\left\langle \boldsymbol{\xi}_{i,1}^m, \boldsymbol{w}_i^m \right\rangle + \Xi_{\mathbb{W}}(\boldsymbol{\xi}_{i,1}) \leq y_i^m, \quad \forall m \in \mathbb{I}_{1:N_s}, \forall i \in \mathbb{I}_{1:I} \tag{31d}$$

$$\left[-\max_j h_j\right]^\star (\boldsymbol{\xi}_{i,2}^m - \boldsymbol{v}_i^m) + \Xi_{\mathbb{W}}(\boldsymbol{v}_i^m) - \left\langle \boldsymbol{\xi}_{i,2}^m, \boldsymbol{w}_i^m \right\rangle - \eta_i \leq y_i^m, \ \forall m \in \mathbb{I}_{1:N_s}, \forall i \in \mathbb{I}_{1:I} \tag{31e}$$

$$\|\boldsymbol{\xi}_{i,1}^m\|_\star \leq \lambda_i, \quad \|\boldsymbol{\xi}_{i,2}^m\|_\star \leq \lambda_i, \ \forall m \in \mathbb{I}_{1:N_s}, \forall i \in \mathbb{I}_{1:I} \tag{31f}$$

$$\boldsymbol{a}_{i+k|k}^m \in \mathbb{U}, \ \boldsymbol{s}_{k|k}^m = \boldsymbol{s}_k, \ \forall m \in \mathbb{I}_{1:N_s}, \forall i \in \mathbb{I}_{1:I} \tag{31g}$$

where $\boldsymbol{\xi} = \{\boldsymbol{\xi}_{i,1}, \boldsymbol{\xi}_{i,2}\}_{i=0}^I$. Then the parameterized safe policy $\boldsymbol{\pi}_{\boldsymbol{\theta}}^{\mathrm{DRMPC}}$ based on DRMPC scheme at time instance k is extracted as follows:

$$\boldsymbol{\pi}_{\boldsymbol{\theta}}^{\mathrm{DRMPC}}(\boldsymbol{s}_k) = \boldsymbol{a}_{k|k}^\star(\boldsymbol{\theta}, \boldsymbol{s}_k) \tag{32}$$

where $\boldsymbol{a}_{k|k}^\star$ is solution of DRMPC (31) corresponding to the first input $\boldsymbol{a}_{k|k}$. Note that all the optimal solutions of $\boldsymbol{a}_{k|k}^m$s are identical since the random samples are generated based on the first given state $\boldsymbol{s}_{k|k}^m = \boldsymbol{s}_k$ and the uncertainty cannot be anticipated [35]. Then we select one of the optimal solutions of $\boldsymbol{a}_{k|k}^m$s as $\boldsymbol{a}_{k|k}$.

4.1 out-of-sample guarantee

Unlike the SAA method, Wasserstein DRMPC provides a probabilistic guarantee on the out-of-sample performance with finitely many samples. More specifically, let us consider the following inequality:

$$\mathrm{CVaR}_{\alpha}^{\mathcal{P}}(\max_{j} h_j(\boldsymbol{s}^{\star}_{k+i|k})) \leq 0 \tag{33}$$

where $\boldsymbol{s}^{\star}_{k+i|k}$ is the optimal solution of (31) and $\mathcal{P}$ is an unknown arbitrary distribution. Then it is worth fulfilling inequality (33) with high probability, i.e.:

$$\mathbb{P}\left\{\mathrm{CVaR}_{\alpha}^{\mathcal{P}}(\max_{j} h_j(\boldsymbol{s}^{\star}_{k+i|k})) \leq 0\right\} \geq 1-\beta \tag{34}$$

where β is a user-specified confidence level. It has been shown in [12], if the Wasserstein radius ϵ_i is chosen as follows:

$$\epsilon_i = \begin{cases} \left(\frac{\log(c_1\beta^{-1})}{c_2 N_s}\right)^{\frac{1}{\max\{d,2\}}} & \text{if} \quad N_s \geq \frac{\log(c_1\beta^{-1}))}{c_2} \\ \left(\frac{\log(c_1\beta^{-1})}{c_2 N_s}\right)^{\frac{1}{a}} & \text{if} \quad N_s < \frac{\log(c_1\beta^{-1}))}{c_2} \end{cases} \tag{35}$$

then (34) holds, where c_1, c_2 are positive constants. In fact, we have assumed that the measure concentration inequality holds [36], i.e., $B = \mathbb{E}^{\mathcal{P}}[\exp \|\boldsymbol{w}\|^a] \leq \infty$ for $a > 1$ (Light-tailed distribution), then c_1, c_2 depend on a, B and the disturbance dimension.

We must emphasize here that in practice, analysis and (probabilistic) finite sampling guarantees are essential in the context of RL and stochastic optimization because, in practice, there is typically limited access to real system data. This analysis can include various criteria in the context of RL, such as convergence rate [37], regret analysis [38], and performance [39].

The next Proposition summarizes the theoretical development of this section.

Proposition 1. *Under assumptions 1 and 2, DRMPC has a tractable reformulation as* (31) *and the extracted policy* $\boldsymbol{\pi}_{\boldsymbol{\theta}}^{\mathrm{DRMPC}}$*, based on finite* N_s *i.i.d. samples, satisfies* (34) $\forall k \in \mathbb{Z}, \forall i \in \mathbb{I}_{1:I}, \forall \boldsymbol{\theta} \in \mathbb{R}^p$*, for a user-specified* β *and* α *and any distributions* $\mathcal{P}$*, if* ϵ_i *is selected from* (35)*.*

4.2 Feasibility pre-filtration

As discussed, satisfying a state-depend hard constraint with $\alpha = 1$ is generally impossible. The same problem exists when the required α is higher than the problem nature requirement. This problem arises in solving (31) when no solution is found. This problem is known as the infeasibility of optimization. A common way to solve the feasibility issue is to soften the constraints using slack variables. The slack variables are positive decision variables that allow inequalities to violate. However, the violation is penalized in the cost function.

A common way to use slack variables is by adding them to the original cost function. However, in this case, finding proper positive coefficients is still challenging. Another way to use slack

variables is to build an optimization as a pre-filtration to find the feasible slack variables and then apply them to the original optimization problem. More specifically, we consider the following optimization problem:

$$\min_{\boldsymbol{s},\boldsymbol{a},\boldsymbol{\eta},\boldsymbol{\lambda},\boldsymbol{y},\boldsymbol{\xi},\boldsymbol{v},\boldsymbol{\sigma}} \quad \sum_{i=0}^{I} \sigma_i^2 \tag{36a}$$

$$\text{s.t.} \quad (31b)$$

$$\eta_i + \frac{1}{1-\alpha}\left[\lambda_i \epsilon_i + \frac{1}{N_s}\sum_{m=1}^{N_s} y_i^m\right] \leq \sigma_i,$$

$$0 \leq \sigma_i,\ \forall i \in \mathbb{I}_{1:I} \tag{36b}$$

$$(31d) - (31g) \tag{36c}$$

with the optimal solutions $\sigma_i^\star$. We then replace constraint (31c) with the following constraint $\forall i \in \mathbb{I}_{1:I}$:

$$\eta_i + \frac{1}{1-\alpha}\left[\lambda_i \epsilon_i + \frac{1}{N_s}\sum_{m=1}^{N_s} y_i^m\right] \leq \sigma_i^\star \tag{37}$$

Then the DRMPC scheme always has a feasible solution. Note that inverting the procedure of obtaining DRMPC (31), we can see DRMPC (31) with the feasible constraint (37), equivalent to the following robust constraint:

$$\sup_{\mathcal{P}\in\mathbb{D}} \mathrm{CVaR}_\alpha^{\mathcal{P}}(\max_j h_j(\boldsymbol{s})) \leq \sigma_i^\star \tag{38}$$

while (20) may yield an infeasible solution. Note that DRMPC scheme provides a family of safe policies $\boldsymbol{\pi}_{\boldsymbol{\theta}}^{\mathrm{DRMPC}}$ for all tuning parameters $\boldsymbol{\theta}$. Therefore, in the next stage, it is necessary to update the parameters to achieve the best performance. The next section details the Q-learning method as a practical way of updating the parameters $\boldsymbol{\theta}$ to achieve the best closed-loop performance.

5 Q-learning based on DRMPC scheme

Q-learning is a powerful, well-known, and popular method in the field of RL, whose use is practical due to relatively low-cost computational efforts, especially in engineering and economic applications [40]. In fact, Q-learning is a classical model-free RL algorithm that tries to capture the optimal action-value function $Q_{\boldsymbol{\theta}} \approx Q^\star$ via tuning the parameters vector $\boldsymbol{\theta}$ where $Q_{\boldsymbol{\theta}}$ is the parameterized action-value function, and $Q^\star$ is the optimal action-value function [41]. The optimal action-value function $Q^\star$ is defined as follows:

$$Q^\star(\boldsymbol{s}_k, \boldsymbol{a}_k) = L(\boldsymbol{s}_k, \boldsymbol{a}_k) + \gamma \min_{\boldsymbol{\pi}} \mathbb{E}[V^{\boldsymbol{\pi}}(\boldsymbol{s}_{k+1})|\boldsymbol{s}_k, \boldsymbol{a}_k] \tag{39}$$

The parameterized action-value function $Q_{\boldsymbol{\theta}}(\boldsymbol{s}_k, \boldsymbol{a}_k)$ based on DRMPC scheme (31) can be formulated as follows:

$$Q_{\boldsymbol{\theta}}(\boldsymbol{s}_k, \boldsymbol{a}_k) = \min_{\boldsymbol{s}, \boldsymbol{a}, \boldsymbol{\eta}, \boldsymbol{\lambda}, \boldsymbol{y}, \boldsymbol{\xi}, \boldsymbol{v}} \quad \text{(31a)} \tag{40a}$$

$$\text{s.t.} \quad \text{(31b), (37), (31d)} - \text{(31g)} \tag{40b}$$

$$\boldsymbol{a}_{k|k} = \boldsymbol{a}_k, \tag{40c}$$

while the approximation of the value function $V_{\boldsymbol{\theta}}$ can be extracted from (40) when constraint (40c) is removed. Then one can verify that the MPC-based action-value function and value function satisfy the fundamental Bellman equations [17]. Q-learning solves the following Least Square (LS) problem:

$$\min_{\boldsymbol{\theta}} \mathbb{E}\left[\left(Q_{\boldsymbol{\theta}}(\boldsymbol{s}_k, \boldsymbol{a}_k) - Q^{\star}(\boldsymbol{s}_k, \boldsymbol{a}_k)\right)^2\right]. \tag{41}$$

In order to solve (41), Temporal-Difference (TD) method uses the following update rule for the parameters $\boldsymbol{\theta}$ at state $\boldsymbol{s}_k$ [42]:

$$\delta_k = L(\boldsymbol{s}_k, \boldsymbol{a}_k) + \gamma V_{\boldsymbol{\theta}}(\boldsymbol{s}_{k+1}) - Q_{\boldsymbol{\theta}}(\boldsymbol{s}_k, \boldsymbol{a}_k) \tag{42a}$$

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \zeta \delta_k \nabla_{\boldsymbol{\theta}} Q_{\boldsymbol{\theta}}(\boldsymbol{s}_k, \boldsymbol{a}_k) \tag{42b}$$

where the scalar $\zeta > 0$ is the learning step-size, δ_k is labelled the TD error, and the input $\boldsymbol{a}_k$ is selected according to the corresponding parametric policy $\boldsymbol{\pi}_{\boldsymbol{\theta}}^{\mathrm{DRMPC}}(\boldsymbol{s}_k)$ with the possible addition of small random exploration such that it preserves the safety. The gradient $\nabla_{\boldsymbol{\theta}} Q_{\boldsymbol{\theta}}$ required in (42) can be obtained by a sensitivity analysis on the DRMPC scheme (40) as detailed in [17] for generic MPC schemes.

In order to generate $\boldsymbol{a}_k$, we first add a small exploration noise to the policy, i.e.:

$$\boldsymbol{a}_k^{\mathrm{e}}(\boldsymbol{\theta}, \boldsymbol{s}_k, \boldsymbol{\rho}_k) = \boldsymbol{\pi}_{\boldsymbol{\theta}}^{\mathrm{DRMPC}}(\boldsymbol{s}_k) + \boldsymbol{\rho}_k \tag{43}$$

where $\boldsymbol{\rho}_k \in \mathcal{E}$ is a random variable providing the exploration. One can easily observe that $\boldsymbol{a}_k^{\mathrm{e}}$ may not deliver a safe input. Therefore a safety filtration based on the DRMPC scheme is needed to provide safe exploration, more specifically consider the following parametric DRMPC scheme with the additional parameter $\boldsymbol{\rho}_k$:

$$\min_{\boldsymbol{s}, \boldsymbol{a}, \boldsymbol{\eta}, \boldsymbol{\lambda}, \boldsymbol{y}, \boldsymbol{\xi}, \boldsymbol{v}} \quad \|\boldsymbol{a}_{k|k} - \boldsymbol{a}_k^{\mathrm{e}}(\boldsymbol{\theta}, \boldsymbol{s}_k, \boldsymbol{\rho}_k)\| \tag{44a}$$

$$\text{s.t.} \quad \text{(40b)} \tag{44b}$$

Then $\boldsymbol{a}_k(\boldsymbol{\theta}, \boldsymbol{s}_k, \boldsymbol{\rho}_k) = \boldsymbol{a}_{k|k}^{\star}(\boldsymbol{\theta}, \boldsymbol{s}_k, \boldsymbol{\rho}_k)$ delivers a safe input after exploration where $\boldsymbol{a}_{k|k}^{\star}$ is the optimal solution of (44) for the first input. Fig. 1 illustrates the safe exploration based on the DRMPC scheme. DRMPC safe set is defined as follows:

$$\mathrm{DRMPC\,safe\,set} := \{\boldsymbol{a}_{k|k} \mid \exists\, \boldsymbol{s}, \boldsymbol{a}, \boldsymbol{\eta}, \boldsymbol{\lambda}, \boldsymbol{y}, \boldsymbol{\xi}, \boldsymbol{v} : \text{(40b)}\}$$

In the policy gradient method, the projection technique results in a bias in the gradient of the performance function [43]. Roughly speaking, this is because the safe exploration set may not be a centered ball, as shown in fig. 1. We have proposed a robust MPC scheme in [44] to solve the bias issue. The proposed method can be easily applied to the DRMPC scheme for the policy gradient method, but applying it is out of the focus of the current work.

Fig.2 illustrates the proposed safe learning method using the DRMPC scheme.

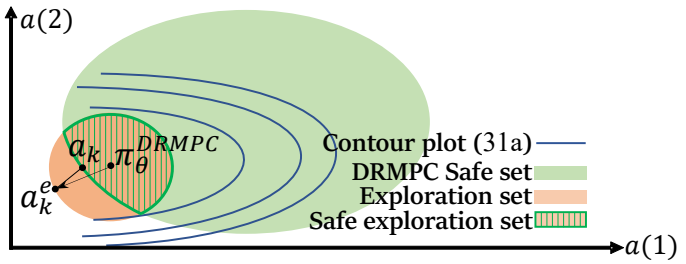


Figure 1: The illustration of the safe exploration for the Q-learning method in a 2-input system. Safe exploration input $\boldsymbol{a}_k \in$ *safe exploration set*, while $\boldsymbol{a}_k^{\mathrm{e}} \in$ *Exploration set* and $\boldsymbol{\pi}_{\boldsymbol{\theta}}^{\mathrm{DRMPC}}(\boldsymbol{s}_k) \in$ *DRMPC Safe set*.

Remark 1. *The proposed method can be applied to the general nonlinear stochastic dynamics with an unknown distribution of stochasticity. Obviously, the computational efforts are increased as the dimension and complexity of the dynamics grow.*

Remark 2. *In this paper, we do not focus on the convergence of the learning method. It is well-known that under the mild assumptions, the Q-learning technique generates a sequence of the parameters $\boldsymbol{\theta}$ that converge to the parameters that best estimate the exact optimal action-value function. Then the extracted policy is an optimal policy among the provided safe policies. The convergence conditions for the Q-learning method can be found in, e.g., [45].*

Remark 3. *Closed-loop stability of the policy for the nominal model used in the MPC scheme resulting from an MPC scheme is straightforward under some mild assumptions on the stage cost and terminal cost and constraints. However, these conditions are not painless for general stochastic systems and stochastic and robust MPC. This aspect is not the main scope of*

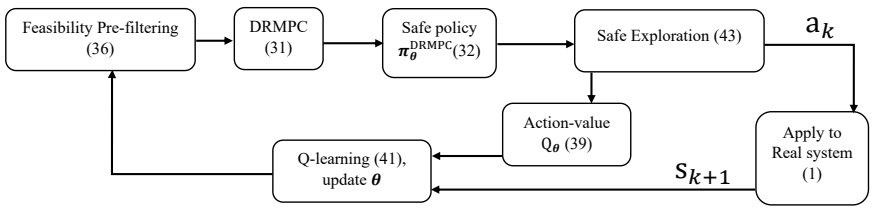


Figure 2: Schematics of the proposed safe RL using DRMPC scheme.

ontour plot (31a)
DRMPC Safe set
Exploration set
e exploration set
$a(1)$

oration for the Q-learning method in a
$\boldsymbol{a}_k \in$ *safe exploration set*, while $\boldsymbol{a}_k^{\mathrm{e}} \in$
DRMPC Safe set.

PC scheme with the additional

$\|\boldsymbol{a}_{k|k} - \boldsymbol{a}_k^{\mathrm{e}}(\boldsymbol{\theta}, \boldsymbol{s}_k, \boldsymbol{\rho}_k)\|$ (44a)

a_1

40b) (44b)

$\boldsymbol{s}_k, \boldsymbol{\rho}_k)$ delivers a safe input
the optimal solution of (44) for
the safe exploration based on
safe set is defined as follows:

$\exists \boldsymbol{s}, \boldsymbol{a}, \boldsymbol{\eta}, \boldsymbol{\lambda}, \boldsymbol{y}, \boldsymbol{\xi}, \boldsymbol{v}$: (40b)}

hod, the projection technique
f the performance function [43].
se the safe exploration set may
n in fig. 1. We have proposed
to solve the bias issue. The
applied to the DRMPC scheme
, but applying it is out of the

d safe learning method using

od can be applied to the general
with an unknown distribution of
putational efforts are increased
ty of the dynamics grow.

o not focus on the convergence
ell-known that under the mild
chnique generates a sequence
rge to the parameters that best
tion-value function. Then the
olicy among the provided safe
tions for the Q-learning method

ty of the policy for the nominal
eme resulting from an MPC
er some mild assumptions on

Algorithm 1: Using DRMPC based Q-learning to provide optimal safe policy.

Input : α, β, I, γ, parameterize $l_{\boldsymbol{\theta}}$, $T_{\boldsymbol{\theta}}$
Initialize : $\boldsymbol{s}_0, \boldsymbol{\theta}_0$
while $\boldsymbol{\theta}$ *converges* **do**
  for *k=0,..., K (end of the mission)* **do**
    Initialize : $\boldsymbol{s}_{k|k}^m = \boldsymbol{s}_k$,
    run feasibility pre-filtration (36) to get $\sigma_i^\star$,
    run DRMPC (31) with the parameters $\boldsymbol{\theta}_k$ and relaxed constraint (37) to get safe policy $\boldsymbol{\pi}_{\boldsymbol{\theta}}^{\mathrm{DRMPC}}(\boldsymbol{s}_k)$,
    apply the safe exploration using (43) and (44) to get the input $\boldsymbol{a}_k$,
    apply the input $\boldsymbol{a}_k$ to the dynamics (1) to get $\boldsymbol{s}_{k+1}$,
    update parameters $\boldsymbol{\theta}_{k+1} \leftarrow \boldsymbol{\theta}_k$ using Q-learning technique, e.g., (42) (ϵ_is are among the parameters),
  Save the last parameters $\boldsymbol{\theta}_0 = \boldsymbol{\theta}_{K+1}$,
end

the current work. However, in the finite-dimensional space, the closed-loop stability properties are recently addressed in [46] for general stochastic systems (MDP).

The next section provides a numerical case study for the proposed method.

The proposed approach has been summarized in Algorithm 1.

The next section provides a numerical case study for the proposed method.

VI. Numerical Simulation

In this section, we consider Wheeled Mobile Robot (WMR) path planning while avoiding static obstacles. The stochastic nonlinear dynamics can be considered as follows:

$$\boldsymbol{s}_{k+1} = \begin{bmatrix} t_e \cos(\phi_k) & 0 \\ t_e \sin(\phi_k) & 0 \\ 0 & t_e \end{bmatrix} \boldsymbol{a}_k + \boldsymbol{s}_k + \boldsymbol{w}_k \tag{45}$$

where $\boldsymbol{s}_k = [x_k, y_k, \phi_k]^\top$, $\boldsymbol{a}_k = [v_k, \psi_k]^\top$ and $\|\boldsymbol{w}_k\|_\infty \leq 0.1$ are the system state, input and disturbance, respectively. x_k and y_k are the position of the robot in two dimensions and $\phi_k \in [-\pi, \pi]$ is the orientation angle. Sampling time t_e is selected 0.2sec for the simulation. The control inputs v_k and ψ_k are the linear and angular velocities, respectively. The control input is restricted as follows:

$$\begin{bmatrix} 0 \\ -1 \end{bmatrix} \leq \boldsymbol{a}_k \leq \begin{bmatrix} 0.5 \\ 1 \end{bmatrix} \tag{46}$$

For simplicity, we consider obstacles of elliptic shape. Hence, the condition for obstacles avoidance can be seen as the following inequality:

$$h_j(\boldsymbol{s}) = 1 - \left(\frac{x - o_{x,j}}{r_{x,j}}\right)^2 - \left(\frac{y - o_{y,j}}{r_{y,j}}\right)^2 \tag{47}$$

input is restricted as follows:

$$\begin{bmatrix} 0 \\ -1 \end{bmatrix} \leq \boldsymbol{a}_k \leq \begin{bmatrix} 0.5 \\ 1 \end{bmatrix} \tag{46}$$

For simplicity, we consider obstacles of elliptic shape. Hence, the condition for obstacles avoidance can be seen as the following inequality:

$$h_j(\boldsymbol{s}) = 1 - \left(\frac{x - o_{x,j}}{r_{x,j}}\right)^2 - \left(\frac{y - o_{y,j}}{r_{y,j}}\right)^2 \tag{47}$$

where $(o_{x,j}, o_{y,j})$ and $(r_{x,j}, r_{y,j})$ are the center and radii of the j^{th} ellipse ($j = 1, \ldots, J$), respectively, and J is number of obstacles.

First, we simulate SMPC with CVaR constraints based on Sample average approximation and DRMPC, and we compare them with deterministic MPC. As shown in figure 3, there are some constraint violations in the MPC scheme. As the probability level α increases, the distance from the path and obstacle increases in SMPC. As mentioned, this method usually requires a large number of data to capture the chance constraint accurately. Moreover, as shown in figure 3, the planned path using DRMPC is farther from the obstacle. We then consider the following stage cost for the RL:

$$\begin{aligned} L(\boldsymbol{s}, \boldsymbol{a}) &= \|\boldsymbol{a}\| + |\phi| + \\ &\underbrace{|x - 8| + |y| - \frac{1}{\tau} \log\left(\frac{1}{2}\Big(|\max_j h_j(\boldsymbol{s})| - \max_j h_j(\boldsymbol{s})\Big) + \omega\right)}_{r(x,y)} \end{aligned} \tag{48}$$

where τ and ω are small positive constants. Since h_j only depends on x, y, function r also depends on x, y. Note that the logarithmic barrier function has been inspired by the constrained optimization context [47]. Moreover, this function allows us to compute the logarithm for every $\boldsymbol{s}$, while it has a large value when the constraints violate. Figure 4 illustrates $r(x, y)$. We include the radius of the Wasserstein ball in the DRMPC parameters to tune it using Q-learning. Figure 5 shows the average stage costs during each mission. As can be seen, the average stage costs are decreasing in five missions in both SMPC and DRMPC. However, DRMPC has lower average costs, and Q-learning is more effective in the DRMPC scheme than in the SMPC scheme. The better improvement in the DRMPC scheme is due to the more freedom and parameters in the provided policies, such as the radius of the Wasserstein ball around the empirical distribution, whereas in the standard SMPC scheme, there is no such parameter. Obviously, tuning the radius of the Wasserstein ball and, consequently, adjusting the conservatism of the safe policy positively impacts the improvement of the closed-loop performance.

7 Conclusion

In this paper, we proposed to use a tractable Distributionally Robust MPC (DRMPC) scheme in order to provide safe policy for Reinforcement Learning (RL) by construction. DRMPC

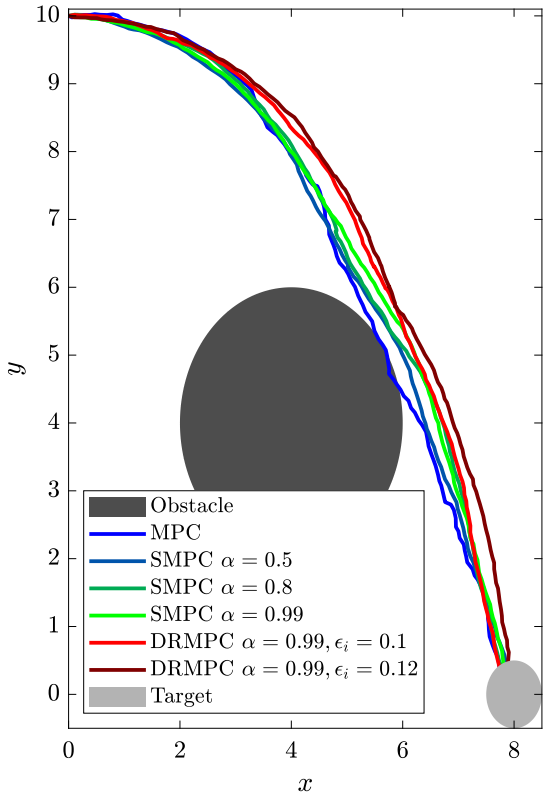


Figure 3: Sample average approximation of SMPC and DRMPC with CVaR constraints.

optimized the cost function subject to the worst-case distribution in a given statistical ball around the empirical distribution. The radius of this ball was measured using the Wasserstein metric. Moreover, Conditional Value at Risk (CVaR) was used as a convex approximator of chance constraints in the DRMPC scheme. We used Q-learning to update the parameters of the DRMPC scheme. We showed the efficiency of the method in the path planning of a Wheeled Mobile Robot (WMR). Considering model mismatch, joint chance-constrained and Neural Network based cost functions in the DRMPC scheme will be the directions of future works.

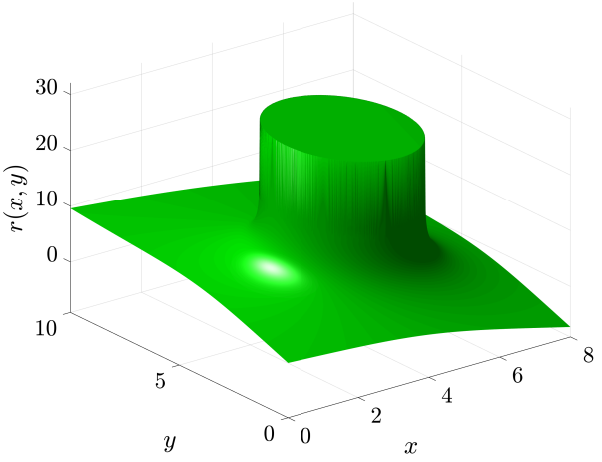


Figure 4: The function $r(x, y)$ for $\tau = 0.2$ and $\omega = 10^{-4}$.

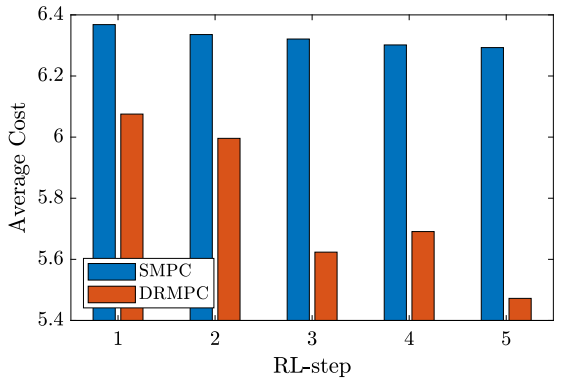


Figure 5: Average costs of five missions during Q-learning from SMPC and DRMPC.

References

[1] Björn Lütjens, Michael Everett, and Jonathan P How. "Safe reinforcement learning with model uncertainty estimates". In: *2019 International Conference on Robotics and Automation (ICRA)*. IEEE. 2019, pp. 8662–8668.

[2] Alexander T Schwarm and Michael Nikolaou. “Chance-constrained model predictive control”. In: *AIChE Journal* 45.8 (1999), pp. 1743–1752.

[3] R Tyrrell Rockafellar, Stanislav Uryasev, et al. “Optimization of conditional value-at-risk”. In: *Journal of risk* 2 (2000), pp. 21–42.

[4] Yinlam Chow and Mohammad Ghavamzadeh. “Algorithms for CVaR optimization in MDPs”. In: *Advances in neural information processing systems* 27 (2014).

[5] Yinlam Chow et al. “Risk-sensitive and robust decision-making: a cvar optimization approach”. In: *Advances in neural information processing systems* 28 (2015).

[6] Zhengru Fang et al. “Stochastic Optimization-Aided Energy-Efficient Information Collection in Internet of Underwater Things Networks”. In: *IEEE Internet of Things Journal* 9.3 (2022), pp. 1775–1789.

[7] Anton J Kleywegt, Alexander Shapiro, and Tito Homem-de-Mello. “The sample average approximation method for stochastic discrete optimization”. In: *SIAM Journal on Optimization* 12.2 (2002), pp. 479–502.

[8] Erick Delage and Yinyu Ye. “Distributionally robust optimization under moment uncertainty with application to data-driven problems”. In: *Operations research* 58.3 (2010), pp. 595–612.

[9] Emre Erdoğan and Garud Iyengar. “Ambiguous chance constrained problems and robust optimization”. In: *Mathematical Programming* 107.1 (2006), pp. 37–61.

[10] Zhaolin Hu and L Jeff Hong. “Kullback-Leibler divergence constrained distributionally robust optimization”. In: *Available at Optimization Online* (2013), pp. 1695–1724.

[11] Georg Pflug and David Wozabal. “Ambiguity in portfolio selection”. In: *Quantitative Finance* 7.4 (2007), pp. 435–442.

[12] Peyman Mohajerin Esfahani and Daniel Kuhn. “Data-driven distributionally robust optimization using the Wasserstein metric: Performance guarantees and tractable reformulations”. In: *Mathematical Programming* 171.1 (2018), pp. 115–166.

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

[14] Astghik Hakobyan and Insoon Yang. “Wasserstein distributionally robust motion control for collision avoidance using conditional value-at-risk”. In: *IEEE Transactions on Robotics* (2021).

[15] Wei Wei et al. “3U: Joint Design of UAV-USV-UUV Networks for Cooperative Target Hunting”. In: *IEEE Transactions on Vehicular Technology* (2022).

[16] Ifrah Saeed et al. “Distributed Nonlinear Model Predictive Control and Reinforcement Learning”. In: *2019 Australian & New Zealand Control Conference (ANZCC)*. 2019, pp. 1–3.

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

[18] 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. 2020, pp. 2573–2578.

[19] 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.

[20] Wanjun Huang, Weiye Zheng, and David J Hill. "Distributionally robust optimal power flow in multi-microgrids with decomposition and guaranteed convergence". In: *IEEE Transactions on Smart Grid* 12.1 (2020), pp. 43–55.

[21] Astghik Hakobyan and Insoon Yang. "Learning-based distributionally robust motion control with Gaussian processes". In: *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2020, pp. 7667–7674.

[22] Mathijs Schuurmans and Panagiotis Patrinos. "Learning-based distributionally robust model predictive control of markovian switching systems with guaranteed stability and recursive feasibility". In: *2020 59th IEEE Conference on Decision and Control (CDC)*. IEEE. 2020, pp. 4287–4292.

[23] Jeremy Coulson, John Lygeros, and Florian Dorfler. "Distributionally robust chance constrained data-enabled predictive control". In: *IEEE Transactions on Automatic Control* (2021).

[24] Bin Li, Zuo Xun Li, and Kai Zhang. "Distributionally Model Predictive Control for Spacecraft Rendezvous and Docking". In: *Advances in Guidance, Navigation and Control*. Springer, 2022, pp. 4447–4457.

[25] Shuwen Lu, Jay H Lee, and Fengqi You. "Soft-constrained model predictive control based on data-driven distributionally robust optimization". In: *AIChE Journal* 66.10 (2020), e16546.

[26] Mario Zanon and Sébastien Gros. "Safe reinforcement learning using robust mpc". In: *IEEE Transactions on Automatic Control* 66.8 (2020), pp. 3638–3652.

[27] Zahra Marvi and Bahare Kiumarsi. "Safe reinforcement learning: A control barrier function optimization approach". In: *International Journal of Robust and Nonlinear Control* 31.6 (2021), pp. 1923–1940.

[28] Richard Cheng et al. "End-to-end safe reinforcement learning through barrier functions for safety-critical continuous control tasks". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 33. 2019, pp. 3387–3395.

[29] Kim P Wabersich et al. "Probabilistic model predictive safety certification for learning-based control". In: *IEEE Transactions on Automatic Control* 67.1 (2021), pp. 176–188.

[30] Yinlam Chow et al. "A lyapunov-based approach to safe reinforcement learning". In: *Advances in neural information processing systems* 31 (2018).

[31] R Tyrrell Rockafellar and Stanislav Uryasev. "Conditional value-at-risk for general loss distributions". In: *Journal of banking & finance* 26.7 (2002), pp. 1443–1471.

[32] Alexander Shapiro, Darinka Dentcheva, and Andrzej Ruszczynski. *Lectures on stochastic programming: modeling and theory*. SIAM, 2021.

[33] Leonid Vasilevich Kantorovich and SG Rubinshtein. “On a space of totally additive functions”. In: *Vestnik of the St. Petersburg University: Mathematics* 13.7 (1958), pp. 52–59.

[34] Rui Gao and Anton J Kleywegt. “Distributionally robust stochastic optimization with Wasserstein distance”. In: *arXiv preprint arXiv:1604.02199* (2016).

[35] E. Klintberg et al. “An improved dual Newton strategy for scenario-tree MPC”. In: *2016 IEEE 55th Conference on Decision and Control (CDC)*. 2016, pp. 3675–3681.

[36] Nicolas Fournier and Arnaud Guillin. “On the rate of convergence in Wasserstein distance of the empirical measure”. In: *Probability Theory and Related Fields* 162.3 (2015), pp. 707–738.

[37] Gal Dalal et al. “Finite sample analysis of two-timescale stochastic approximation with applications to reinforcement learning”. In: *Conference On Learning Theory*. PMLR. 2018, pp. 1199–1233.

[38] Zhengqing Zhou et al. “Finite-sample regret bound for distributionally robust offline tabular reinforcement learning”. In: *International Conference on Artificial Intelligence and Statistics*. PMLR. 2021, pp. 3331–3339.

[39] Zhuoran Yang et al. “A finite sample analysis of the actor-critic algorithm”. In: *2018 IEEE conference on decision and control (CDC)*. IEEE. 2018, pp. 2759–2764.

[40] Arash Bahari Kordabad and Sebastien Gros. “Q-learning of the storage function in Economic Nonlinear Model Predictive Control”. In: *Engineering Applications of Artificial Intelligence* 116 (2022), p. 105343.

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

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

[43] 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)*. 2021.

[44] Arash Bahari Kordabad, Hossein Nejatbakhsh Esfahani, and Sebastien Gros. “Bias Correction in Deterministic Policy Gradient Using Robust MPC”. In: *2021 European Control Conference (ECC)*. IEEE. 2021, pp. 1086–1091.

[45] Christopher JCH Watkins and Peter Dayan. “Q-learning”. In: *Machine learning* 8.3-4 (1992), pp. 279–292.

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

[47] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

J Reinforcement Learning for MPC: Fundamentals and Current Challenges

Preprint of [103] **Arash Bahari Kordabad**, Dirk Reinhardt, Akhil S Anand, and Sebastien Gros. "Reinforcement Learning for MPC: Fundamentals and Current Challenges". In: *Submitted* (2022)

Reinforcement Learning for MPC: Fundamentals and Current Challenges

Arash Bahari Kordabad[1], Dirk Reinhardt[1], Akhil S Anand[1], and Sebastien Gros[1]

[1]Department of Engineering Cybernetics, Norwegian University of Science and Technology (NTNU), Trondheim, Norway.

Abstract: Recent publications have laid a solid theoretical foundation for the combination of Reinforcement Learning and Model Predictive Control, in view of obtaining high-performance data-driven MPC policies. Early practical results, both in simulation and in experiments have shown the potential of this combination, but also revealed certain challenges. In addition, the technical complexity of these results makes it difficult for interested readers to gather the fundamental ideas and principles behind this combination. This paper aims at providing a coherent and more accessible picture of these results, but also significantly deeper and more mature insights into their meaning than has been proposed before. It also aims at identifying the current challenges in the field.

Keywords: MPC, Reinforcement Learning, Learning for MPC, Learning for MPC, Stability & Safety

1 Introduction

Model Predictive Control (MPC) is a successful control strategy that employs a (possibly inaccurate) model of the real system dynamics to generate input-state sequences that minimize a certain cost, possibly under some constraints [1]. The MPC problem is solved at every time instant, in a receding-horizon fashion, delivering a policy for the real system. For many applications, the building an MPC model able to capture the real system dynamics accurately is very difficult, especially if the real system is stochastic. For these applications, the performance of the MPC scheme can be severely affected by this lack of accuracy. This is especially the case if the objective of the MPC scheme is not to bring the real system to a specific reference state (a.k.a. tracking objective), but rather to minimize a generic cost (a.k.a. economic objective).

Recent research focus on alleviating this issue by integrating techniques from Machine Learning (ML). The most classic and obvious approach is to use ML techniques to develop more accurate data-driven MPC models [2, 3]. While this paradigm has a clear value, it does not circumvent the issues related to model inaccuracies. Indeed, the performance of a policy delivered by an MPC scheme integrating an ML-based model is still only as good as the ML-based model is, and therefore limited by the structure and choices made in the ML tools.

Besides, for many applications, a higher model accuracy can only be achieved through a higher model complexity. Because complex MPC models tend to yield complex MPC schemes, the ML-based MPC paradigm tends to bind the MPC performance to its complexity.

A core issue with ML-based MPC is that the modelling is not directly related to the control objectives. Indeed, the ML-based model is constructed to deliver the best possible predictions, in the hope that this will turn into the best possible MPC performances. But the development of the ML-based model is not easily tied to the MPC closed-loop performance they will result in. Model-free MPC techniques are sometimes described as building MPC predictions that are tailored to the MPC objectives. However, this description is not formally supported. Indeed, if using the ideal class of regularization, these techniques are equivalent to building an MPC model using subspace system identification, and are therefore very classic. If using different regularizations, they can produce significantly suboptimal MPC policies. It is easy to produce examples where this effect occurs [4]. RL for MPC can be used in the context Model-Free

Reinforcement Learning (RL) focuses on optimizing the closed-loop performance of policies using data obtained on the real system, without necessarily relying directly on a model of that system. In the RL context, MPC can be construed as a tool generating highly structured policies. RL then offers a rich toolbox for adjusting the MPC scheme from data, in view of improving its closed-loop performance. The combination of RL and MPC is therefore unique in the field of learning-based MPC, insofar as it does not focus on improving the MPC model for more accurate predictions, but ties the MPC tuning directly to the closed-loop optimality of the resulting policy. This uniqueness is, e.g., illustrated in examples where RL sacrifices the MPC model accuracy to improve the MPC closed-loop performance [5].

This paper provides critical insights on the fundamentals of RL and MPC that have been recently detailed in the literature, as well as on the known challenges. Section 2 provides some background. Section 3 some fundamental theoretical results on RL and MPC, and insights on their consequences. Section 4 provides a discussion on RL methods for MPC, and the associated challenges. Section 5 and 6 discuss the use of RL for MPC with stability and safety requirements.

2 Background

Markov Decision Processes (MDP) provide a generic framework for the class of problems at the center of MPC. An MDP operates over given state and action (aka input) spaces S, A, respectively. These spaces can be both discrete (i.e. integer sets), continuous, or mixed. An MDP is defined by the triplet (L, γ, ρ), where L is a stage cost, $\gamma \in (0, 1]$ a discount factor and ρ a conditional probability (measure) defining the dynamics of the system considered, i.e. for a given state-action pair $\boldsymbol{s}, \boldsymbol{a} \in S \times A$, the successive state $\boldsymbol{s}_+$ is distributed according to

$$\boldsymbol{s}_+ \sim \rho(\cdot|\boldsymbol{s}, \boldsymbol{a}) \tag{1}$$

Note that (1) is a generalization of the classic dynamics, deterministic or not, often considered in MPC, usually cast as

$$\boldsymbol{s}_+ = \boldsymbol{F}\left(\boldsymbol{s}, \boldsymbol{a}, \boldsymbol{w}\right), \quad \boldsymbol{w} \sim W \tag{2}$$

where $\boldsymbol{w}$ is a random disturbance from distribution W. In the special case $\boldsymbol{w} = 0$, (2) simply yields deterministic dynamics. An MDP is then the problem of finding the optimal policy $\boldsymbol{\pi}^\star : S \to A$ solution of:

$$\boldsymbol{\pi}^\star = \arg\min_{\boldsymbol{\pi}} J(\boldsymbol{\pi}) \quad \text{where} \tag{3a}$$

$$J(\boldsymbol{\pi}) = \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k L\left(\boldsymbol{s}_k, \boldsymbol{a}_k\right) \,\middle|\, \boldsymbol{a}_k = \boldsymbol{\pi}\left(\boldsymbol{s}_k\right)\right], \tag{3b}$$

and the expected value operator $\mathbb{E}[.]$ is taken over the (possibly) stochastic closed loop trajectories of the system. Discussing the solution of MDPs is often best done via the Bellman equations defining implicitly the optimal value function $V^\star : S \to \mathbb{R}$ and the optimal action-value function $Q^\star : S \times A \to \mathbb{R}$ as

$$V^\star\left(\boldsymbol{s}\right) = \min_{\boldsymbol{a}} Q^\star\left(\boldsymbol{s}, \boldsymbol{a}\right) \tag{4a}$$

$$Q^\star\left(\boldsymbol{s}, \boldsymbol{a}\right) = L\left(\boldsymbol{s}, \boldsymbol{a}\right) + \gamma \mathbb{E}\left[V^\star\left(\boldsymbol{s}_+\right) \mid \boldsymbol{s}, \boldsymbol{a}\right] \tag{4b}$$

The optimal policy then reads as:

$$\boldsymbol{\pi}^\star\left(\boldsymbol{s}\right) = \arg\min_{\boldsymbol{a}} Q^\star\left(\boldsymbol{s}, \boldsymbol{a}\right) \tag{5}$$

2.1 Reinforcement Learning

The fundamental goal of Reinforcement Learning (RL) is to use data to deliver an approximation of the optimal policy $\boldsymbol{\pi}^\star$. The field can be coarsely divided in two large classes of approaches. The first class, generically labelled Q-learning, approximates the optimal action-value function $Q^\star$ via a parametrized function approximator $Q_{\boldsymbol{\theta}}$. The parameters $\boldsymbol{\theta}$ are then adjusted using data such that $Q_{\boldsymbol{\theta}} \approx Q^\star$ is some sense. An approximation of the optimal policy $\boldsymbol{\pi}^\star$ can then be obtained using:

$$\hat{\boldsymbol{\pi}}^\star\left(\boldsymbol{s}\right) = \arg\min_{\boldsymbol{a}} Q_{\boldsymbol{\theta}}\left(\boldsymbol{s}, \boldsymbol{a}\right) \tag{6}$$

The second class approximates $\boldsymbol{\pi}^\star$ directly via a parametrized policy $\boldsymbol{\pi}_{\boldsymbol{\theta}}$, and adjust the parameters $\boldsymbol{\theta}$ from data so as to minimize $J(\boldsymbol{\pi}_{\boldsymbol{\theta}})$. This can, e.g., be done by estimating policy gradients $\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\pi}_{\boldsymbol{\theta}})$, or by building surrogate models of $J(\boldsymbol{\pi}_{\boldsymbol{\theta}})$, used to adjust $\boldsymbol{\theta}$.

2.2 Model Predictive Control

For a given system state $\boldsymbol{s}$, MPC produces control policies based on repeatedly solving an optimal control problem on a finite, receding horizon, often cast as:

$$\min_{\boldsymbol{x},\boldsymbol{u}} \quad T\left(\boldsymbol{x}_N\right) + \sum_{k=0}^{N-1} L\left(\boldsymbol{x}_k, \boldsymbol{u}_k\right) \tag{7a}$$

$$\text{s.t.} \quad \boldsymbol{x}_{k+1} = \boldsymbol{f}\left(\boldsymbol{x}_k, \boldsymbol{u}_k\right), \quad \boldsymbol{x}_0 = \boldsymbol{s} \tag{7b}$$

$$\boldsymbol{h}\left(\boldsymbol{x}_k, \boldsymbol{u}_k\right) \leq 0, \quad \boldsymbol{u}_k \in A, \tag{7c}$$

for a given system state $\boldsymbol{s}$, problem (7) produces a complete profile of control inputs $\boldsymbol{u}^\star = \{\boldsymbol{u}_0^\star, \ldots, \boldsymbol{u}_{N-1}^\star\}$ and corresponding state predictions $\boldsymbol{x} = \{\boldsymbol{x}_0^\star, \ldots, \boldsymbol{x}_N^\star\}$. Only the first element $\boldsymbol{u}_0^\star$ of the input sequence $\boldsymbol{u}^\star$ is applied to the system. At the next physical sampling time, a new state $\boldsymbol{s}$ is received, and problem (7) is solved again, producing a new $\boldsymbol{u}^\star$ and a new $\boldsymbol{u}_0^\star$. MPC hence yields a policy:

$$\boldsymbol{\pi}_{\mathrm{MPC}}\left(\boldsymbol{s}\right) = \boldsymbol{u}_0^\star, \tag{8}$$

with $\boldsymbol{u}_0^\star$ solution of (7) for $\boldsymbol{s}$ given. For $\gamma \approx 1$, policy (8) can provide a good approximation of the optimal policy $\boldsymbol{\pi}^\star$ for an adequate choice of prediction horizon N, terminal cost T and if the MPC model $\boldsymbol{f}$ approximates the true dynamics (1) sufficiently well. In that context, the latter is arguably the major weakness. Indeed, many systems are difficult to model accurately. Furthermore, within a modelling structure, selecting the model $\boldsymbol{f}$ that yields the best closed-loop performance $J(\boldsymbol{\pi}_{\mathrm{MPC}})$ is very difficult. Indeed, there is in general no guarantee that the model $\boldsymbol{f}$ that best fits the data collected from the real system is the best model in terms of $J(\boldsymbol{\pi}_{\mathrm{MPC}})$.

3 Fundamentals of RL and MPC

The combination of RL and MPC can address the issues raised above. In this section, we provide the central result supporting that statement. To that end, it is useful to construe MPC as a (possibly local) model of the action-value function $Q^\star$. Indeed, consider an MPC-based policy

$$\boldsymbol{\pi}_{\boldsymbol{\theta}}(\boldsymbol{s}) = \boldsymbol{u}_0^\star \tag{9}$$

where $\boldsymbol{u}_0^\star$ is part of the solution of:

$$\boldsymbol{x}^\star, \boldsymbol{u}^\star = \arg\min_{\boldsymbol{x},\boldsymbol{u}} \quad T_{\boldsymbol{\theta}}\left(\boldsymbol{x}_N\right) + \sum_{k=0}^{N-1} L_{\boldsymbol{\theta}}\left(\boldsymbol{x}_k, \boldsymbol{u}_k\right), \tag{10a}$$

$$\text{s.t.} \quad \boldsymbol{x}_{k+1} = \boldsymbol{f}_{\boldsymbol{\theta}}\left(\boldsymbol{x}_k, \boldsymbol{u}_k\right), \quad \boldsymbol{x}_0 = \boldsymbol{s}, \tag{10b}$$

$$\boldsymbol{h}_{\boldsymbol{\theta}}\left(\boldsymbol{x}_k, \boldsymbol{u}_k\right) \leq 0, \quad \boldsymbol{u}_k \in A. \tag{10c}$$

This MPC formulation is identical to (7), but the cost, constraints and dynamics underlying the MPC scheme are now all parametrized in $\boldsymbol{\theta}$, to the exception of the input constraint $\boldsymbol{u}_k \in U$. This choice is motivated below. An MPC-based model of $Q^\star$ is then provided by:

$$Q_{\boldsymbol{\theta}}(\boldsymbol{s}, \boldsymbol{a}) = \min_{\boldsymbol{x}, \boldsymbol{u}} \quad \text{(10a)}, \tag{11a}$$

$$\text{s.t.} \quad \text{(10b)} - \text{(10c)}, \quad \boldsymbol{u}_0 = \boldsymbol{a}, \tag{11b}$$

where a constraint $\boldsymbol{u}_0 = \boldsymbol{a}$ on the initial input has been added to (10). MPC (11) is a valid model of $Q^\star$ in the sense that it satisfies the relationships (4) and (5), i.e.:

$$\boldsymbol{\pi}_{\boldsymbol{\theta}}(\boldsymbol{s}) = \arg\min_{\boldsymbol{a}} Q_{\boldsymbol{\theta}}(\boldsymbol{s}, \boldsymbol{a}), \quad V_{\boldsymbol{\theta}}(\boldsymbol{s}) = \min_{\boldsymbol{a}} Q_{\boldsymbol{\theta}}(\boldsymbol{s}, \boldsymbol{a}), \tag{12}$$

where $V_{\boldsymbol{\theta}}(\boldsymbol{s})$ is the optimal cost resulting from solving MPC (10). One can then readily verify that if the MPC parameters $\boldsymbol{\theta}$ are such that $Q_{\boldsymbol{\theta}} = Q^\star$, then MPC scheme (10) delivers the optimal policy $\boldsymbol{\pi}^\star$ through (9), i.e. $\boldsymbol{\pi}_{\boldsymbol{\theta}} = \boldsymbol{\pi}^\star$. An important question, then, is how effective can an MPC scheme be at approximating $Q^\star$, at least in a neighborhood of $\boldsymbol{a} = \boldsymbol{\pi}^\star(\boldsymbol{s})$. The main concern here is arguably the MPC model $\boldsymbol{f}_{\boldsymbol{\theta}}$ again, for the reasons already raised in Sec. 2.2. In addition, $Q^\star$ is typically built from a discounted sum of the stage costs L, while undiscounted MPC formulations are typically preferred.

The Theorem reported below addresses these concerns and provides the central justification for considering the MPC parametrization (10) in learning-based MPC. It establishes that under some mild conditions, (11) is able to provide an exact model of $Q^\star$ even if its predictive model (10b) is inaccurate. This in turn entails that MPC (10) can achieve optimal closed-loop performances even if the MPC model is inaccurate.

Theorem 1. *Suppose that the parameterized stage cost, terminal cost and constraints in* (10) *are universal function approximators with adjustable parameters* $\boldsymbol{\theta}$*. Then there exist parameters* $\boldsymbol{\theta}^\star$ *such that the following identities hold,* $\forall \gamma$*:*

1. $V_{\boldsymbol{\theta}^\star}(\boldsymbol{s}) = V^\star(\boldsymbol{s}),\ \forall \boldsymbol{s} \in \mathcal{S}$
2. $\boldsymbol{\pi}_{\boldsymbol{\theta}^\star}(\boldsymbol{s}) = \boldsymbol{\pi}^\star(\boldsymbol{s}),\ \forall \boldsymbol{s} \in \mathcal{S}$
3. $Q_{\boldsymbol{\theta}^\star}(\boldsymbol{s}, \boldsymbol{a}) = Q^\star(\boldsymbol{s}, \boldsymbol{a}),\ \forall \boldsymbol{s} \in \mathcal{S}$*, for the inputs* $\boldsymbol{a} \in A$ *such that* $|V^\star(\boldsymbol{f}_{\boldsymbol{\theta}^\star}(\boldsymbol{s}, \boldsymbol{a}))| < \infty$

if the set

$$\mathcal{S} =: \left\{ \boldsymbol{s} \in S \;\middle|\; |[V^\star(\boldsymbol{x}_k^\star)]| < \infty,\ \forall\, k \leq N \right\} \tag{13}$$

is non-empty.

Proof. We select the parameters such that the following holds:

$$T_{\boldsymbol{\theta}^\star}(\boldsymbol{s}) = V^\star(\boldsymbol{s}) \tag{14a}$$

$$L_{\boldsymbol{\theta}^\star}(\boldsymbol{s}, \boldsymbol{a}) = \tag{14b}$$

$$\begin{cases} Q^\star(\boldsymbol{s}, \boldsymbol{a}) - V^\star(\boldsymbol{f}_{\boldsymbol{\theta}^\star}(\boldsymbol{s}, \boldsymbol{a})) & \text{If } |V^\star(\boldsymbol{f}_{\boldsymbol{\theta}^\star}(\boldsymbol{s}, \boldsymbol{a}))| < \infty \\ \infty & \text{otherwise} \end{cases}$$

The proof then follows from [5, 6]. ■

Theorem 1 states that, for a given MDP, an MPC scheme with a possible inaccurate model can deliver the optimal value functions and the optimal policy of the original MDP. This can be achieved by selecting the proper stage cost, terminal cost, and constraints. Theorem 1 extends to robust MPC, stochastic MPC, and Economic MPC (EMPC), all discounted or not. The assumption in (13) can be interpreted as some form of the stability condition on $\boldsymbol{f}_{\boldsymbol{\theta}^\star}$ under the optimal trajectory $\boldsymbol{x}^\star$. More specifically, this assumption requires the existence of a non-empty set such that the optimal value function $V^\star$ of the predicted optimal trajectories $\boldsymbol{x}^\star$ based on the system model is finite with a unitary probability for all initial states starting from this set.

3.1 Role of RL in Learning-based MPC

Many recent learning-based MPC methods focus on learning a predictive model for the MPC scheme from the data, using Machine Learning (ML) or other data-driven techniques. In these methods, only (10b) is parametrized in the MPC scheme (10), and adjusted in view of providing as accurate predictions as possible. It is then important to clarify why an approach centred purely on adjusting the model is not necessarily sufficient for achieving the highest possible performances. While adjusting the MPC model alone from data has a high practical value, two issues stand in the way of obtaining optimal policies from that alone.

First, if the objective of the MPC scheme is optimality in the sense of $J(\boldsymbol{\pi}_{\boldsymbol{\theta}})$, then adjusting the MPC model $\boldsymbol{f}_{\boldsymbol{\theta}}$ for delivering better predictions is a very indirect proxy for minimizing $J(\boldsymbol{\pi}_{\boldsymbol{\theta}})$. Indeed, if the true system dynamics (1) do not belong to the set of dynamics that $\boldsymbol{f}_{\boldsymbol{\theta}}$ can represent, then there is no guarantee that adjusting the MPC model $\boldsymbol{f}_{\boldsymbol{\theta}}$ to better fit (1) will reduce $J(\boldsymbol{\pi}_{\boldsymbol{\theta}})$. In fact, it is straightforward to propose trivial counter-examples where model fitting worsens the closed-loop performance, see [5].

Second, Theorem 1 shows that a modification of not only the MPC model but also of the cost and constraints in the MPC formulation is conducive of obtaining the optimal policy $\boldsymbol{\pi}^\star$ from the MPC scheme, even if the MPC model cannot predict the real system dynamics accurately. However, learning techniques focusing on fitting the MPC model to the real system provide no indication as to how one ought to adjust the MPC cost and constraints for performance. This issue stems from the fact that there is no simple relationship between the predictive performance of the MPC model and the closed-loop performance $J(\boldsymbol{\pi}_{\boldsymbol{\theta}})$.

In the light of these observations, while fitting the MPC model to the real system trajectories has practical value, it is not sufficient if closed-loop performance is targeted.

Indeed, performance-oriented learning-based MPC ought to consider a full parametrizations of the MPC scheme as per (10), and integrate learning tools that aim at minimizing $J(\boldsymbol{\pi}_{\boldsymbol{\theta}})$, or at achieving $Q_{\boldsymbol{\theta}} \approx Q^\star$ from the data. The role of RL in that context is to provide these learning tools.

3.2 Role of the Model in RL for MPC

Theorem 1 suggests that if using the complete parametrization (10a), the MPC model is less important than normally thought. Indeed, it suggests that under some mild assumptions, cost and constraints modifications can compensate for the model error and produce an optimal policy and value functions.

This observation ought to trigger the natural question as to what is then the role of the MPC model if it does not need to be accurate. This central question has not been properly discussed in the literature so far, we propose four central insights next.

The first and most obvious insight lies in the core assumption of Theorem 1. This assumption, while arguably mild, forbids the use of *any model* in the MPC scheme, and requires that it satisfies a requirement akin to—but less demanding than–stability of the model under the optimal policy. This assumption is clearly very impractical to verify in practice. Fortunately, RL for MPC can be deployed without verifying this assumption, through the methods presented in Sec. 5.

The second less obvious insight stems from the observation that the cost and constraints modifications (14) required by Theorem 1 can be difficult to approximate in practice, and difficult to use in an MPC scheme. Indeed, these modifications can require fairly complex and possibly very non-convex functions. They may require very rich function approximations (e.g. large DNNs), and their complexity and non-convexity can make their use in an MPC scheme impractical. In that context, one can readily observe from (14b) that being able to adjust the MPC model introduces extra degrees of freedom in how the cost and constraints modifications can be shaped, allowing one, in turn, to impose certain restrictions on these modifications. These restrictions can be related to the simplicity of the function approximations used, and/or in imposing convexity in the resulting cost and constraints. This approach is used in [7], and further elaborated in Sec. 5.2.

A third insight is in the use of Robust MPC to build safe policies in RL, see Sec. 6. In that context, the role of the model is to predict the worst-case scenario with respect to safety-critical constraints that the real system ought not to violate. The model must then "fit" the real system in the sense of predicting these worst cases, where the "fitting" is performed via set-membership identification [8].

The last insight lies in the explainability of the policy delivered by MPC. Because MPC proposes a full prediction of the actions that it plans to take on the system, and of the expected system response, it can be considered a more explainable policy than generic function approximations, such as DNN. In that context, if the MPC model is a very poor match for the real system, that explainability is lost.

4 RL methods for MPC

RL offers two main classes of methods, which either target a direct minimization of $J(\boldsymbol{\pi}_{\boldsymbol{\theta}})$ over the parameters $\boldsymbol{\theta}$, or a fitting of the action-value function model $Q_{\boldsymbol{\theta}}$ to the optimal one $Q^{\star}$, see Sec. 2.1. We aim at providing next further insights as to how these two different families operate in the context of RL for MPC.

4.1 Q learning

Basic Q learning aims at achieving $Q_{\boldsymbol{\theta}} \approx Q^{\star}$ via solving a fitting problem, typically in the form:

$$\min_{\boldsymbol{\theta}} \mathbb{E}\left[\left(Q^{\star}\left(\boldsymbol{s},\boldsymbol{a}\right)-Q_{\boldsymbol{\theta}}(\boldsymbol{s},\boldsymbol{a})\right)^{2}\right] \tag{15}$$

where the expected value $\mathbb{E}[.]$ is taken over the system trajectories and actions $\boldsymbol{a}$. Because $Q^{\star}$ is unknown, (15) is commonly replaced by an approximation e.g. based on iterating the temporal difference problem:

$$\min_{\boldsymbol{\theta}_{+}} \mathbb{E}\left[\left(L\left(\boldsymbol{s},\boldsymbol{a}\right)+\gamma \min_{\boldsymbol{a}'} Q_{\boldsymbol{\theta}}(\boldsymbol{s}_{+},\boldsymbol{a}')-Q_{\boldsymbol{\theta}_{+}}(\boldsymbol{s},\boldsymbol{a})\right)^{2}\right] \tag{16}$$

until $\boldsymbol{\theta}_{+}=\boldsymbol{\theta}$. In the RL for MPC context, $Q_{\boldsymbol{\theta}}$ is delivered by (11) and $\min_{\boldsymbol{a}'} Q_{\boldsymbol{\theta}}(\boldsymbol{s},\boldsymbol{a}')=V_{\boldsymbol{\theta}}(\boldsymbol{s})$ is delivered by (12), i.e. by the optimal cost of MPC (10) solved at state $\boldsymbol{s}$.

In Q learning, two MPC schemes need to be solved for each state transition, i.e at each time instant if RL is performed "online". This can be done in parallel. The action $\boldsymbol{a}$ used in (11) ought to (at least regularly) differ from the MPC policy $\boldsymbol{\pi}_{\boldsymbol{\theta}}$, so as to introduce exploration. Exploration satisfying the MPC constraints by construction is discussed in Sec. 6.1.

4.2 Policy Gradient Methods & Direct Policy Search

An alternative to Q learning is to treat the MPC as a policy (9) whose parameters $\boldsymbol{\theta}$ ought to be adjusted to minimize $J(\boldsymbol{\pi}_{\boldsymbol{\theta}})$ directly. Two broad classes of approaches can be distinguished here. Policy gradient methods estimate the "policy gradient" $\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\pi}_{\boldsymbol{\theta}})$ from data and use it to update the parameters $\boldsymbol{\theta}$ in a gradient descent fashion. Alternatively, a direct policy searches builds a surrogate model of $J(\boldsymbol{\pi}_{\boldsymbol{\theta}})$ from data, and uses it to propose new policy parameters $\boldsymbol{\theta}$. However, direct policy search tends to scale poorly with the size of the policy parameters.

Policy gradient methods can e.g. use the deterministic approach, whereby the policy gradient is evaluated via:

$$\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\pi}_{\boldsymbol{\theta}})=\mathbb{E}\left[\nabla_{\boldsymbol{\theta}} \boldsymbol{\pi}_{\boldsymbol{\theta}}\left(\boldsymbol{s}\right) \nabla_{\boldsymbol{a}} Q_{\boldsymbol{\pi}_{\boldsymbol{\theta}}}\left(\boldsymbol{s},\boldsymbol{\pi}_{\boldsymbol{\theta}}(\boldsymbol{s})\right)\right] \tag{17}$$

Here π_θ is delivered by MPC (10). The "critic" Q_{π_θ} is typically built separately using a generic function approximator and policy evaluation techniques.

An inconvenient of the policy gradient approach is that an approximation structure must be selected for the critic Q_{π_θ}, e.g. as a DNN. The design of that approximator is not obvious, making the method less straightforward than Q learning, see Sec. 4.1. This difficulty has been recently alleviated in [9].

4.3 NLP sensitivities & Smoothness

Many RL methods, including Q learning and policy gradient methods, require the sensitivities of the function approximators π_θ, Q_θ, and V_θ. When provided by an MPC scheme, these approximators are typically continuous but only piecewise smooth.However, when the MPC achieves Linear Independence Constraint Qualification (LICQ) and Second Order Sufficient Condition (SOSC), then non-smooth points correspond to weakly active constraints in the MPC. For a well-formulated MPC scheme, these points form a set of zero measures. Because RL methods always use the sensitivities inside expected value operators, their contribution to the learning then disappears.

Hence, while the non-smoothness of MPC schemes may superficially appear as an issue, in most cases it is fortunately not. In practice, this question can be simply ignored when the MPC response is continuous.

The arguments above do not necessarily hold anymore if the MPC can lose SOSC, possibly producing discontinuous policies, e.g. having a "bang-bang" response. This situation can occur if, e.g., the MPC is a Linear Program. In that context, while Q_θ and V_θ typically remain continuous and piecewise smooth, allowing Q learning to be used, the deterministic policy method briefly discussed in Sec. 4.2 becomes problematic. This issue has been discussed in [10], and an early solution has been proposed. However, it deserves further attention.

4.4 Feasible exploration

RL methods require exploration, i.e. actions $a \neq \pi_\theta(s)$ must be (regularly) applied to the real system in order to gather information to improve the policy. In the presence of constraints (10c) on the system evolution, a natural question is how to generate exploration that does not jeopardize the MPC feasibility.

In the context of MPC-based policies, feasible exploration can be trivially achieved, without adding complexity in the RL methods nor in the MPC scheme.

Indeed, in the MPC context, one can simply add a disturbance in the MPC cost, e.g. in the form of a gradient over the initial action u_0, in order for the MPC action to differ from the

policy. More specifically, we can consider:

$$\phi_{\boldsymbol{\theta}}(\boldsymbol{s}, \boldsymbol{d}) = \min_{\boldsymbol{x},\boldsymbol{u}} \boldsymbol{d}^\top \boldsymbol{u}_0 + T_{\boldsymbol{\theta}}(\boldsymbol{x}_N) + \sum_{k=0}^{N-1} L_{\boldsymbol{\theta}}(\boldsymbol{x}_k, \boldsymbol{u}_k) \tag{18a}$$

$$\text{s.t.} \quad (10b), (10c) \tag{18b}$$

where $\boldsymbol{d} \in \mathbb{R}^m$ is a vector of the size of the action, possibly selected randomly. Because only the MPC cost is modified, the solution of (18) is feasible for (10). MPC (18) can be used to produce feasible exploration for policy gradient methods, but it can also generate action-value functions for Q learning, see Sec. 4.1. In that context, (18) produces a feasible action with exploration $\boldsymbol{a} = \boldsymbol{u}_0^\star$ where $\boldsymbol{u}_0^\star$ is solution of (18) and depends on $\boldsymbol{d}$. The optimal cost $\phi_{\boldsymbol{\theta}}$ of (18) delivers the action-value function:

$$Q_{\boldsymbol{\theta}}(\boldsymbol{s}, \boldsymbol{a}) = \phi_{\boldsymbol{\theta}}(\boldsymbol{s}, \boldsymbol{d}) - \boldsymbol{d}^\top \boldsymbol{u}_0^\star \tag{19}$$

This principle has been further detailed in [11]. However, while the exploration generated by (18) respects the MPC constraints (10c), the real system dynamics may not, due to stochasticity and model error. This issue can be addressed via safe exploration, see Sec. 6.1.

4.5 Current Challenges

A challenge identified in using RL methods on MPC is related to performing the learning on existing “big data". Learning from existing data is performed via taking numerous “sweeps" (a.k.a. experience replay) through the data using the methods detailed above. This requires a large number of evaluations of $\boldsymbol{\pi}_{\boldsymbol{\theta}}$, $Q_{\boldsymbol{\theta}}$, $V_{\boldsymbol{\theta}}$ and of their sensitivities. Classical RL function approximators such as DNNs have dedicated computational tools such as GPUs for fast evaluation and differentiation. Hence, big data can be efficiently processed via DNNs. Function approximations from MPC schemes are inexpensive to differentiate, but often expensive to evaluate because they require solving the MPC problem. Excellent tools exist to solve MPC schemes in real-time such that performing RL “online", i.e. while the system is running, is not an issue. However, performing RL for MPC on *existing* big data can be impractical due to the amount of computational time required. This issue has received some attention in [12], but more work is required to fully address this question.

5 Learning Stable Policies via MPC

A benefit of MPC as a function approximator in RL is the existence of a strong theory establishing properties such as safety and stability. The nominal stability of the closed-loop system with a policy is crucial in the control context. The stability of MPC schemes for deterministic systems is relatively straightforward to establish if the MPC stage cost is lower-bounded by a class-$\mathcal{K}_\infty$ function, see [1]. If a generic (a.k.a. *economic*) stage cost is considered, asymptotic stability requires the extra *dissipativity* conditions:

$$L(\boldsymbol{s}, \boldsymbol{a}) - \lambda(\boldsymbol{s}) + \lambda(\boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{s}, \boldsymbol{a})) \geq \alpha(\|\boldsymbol{s} - \bar{\boldsymbol{s}}\|) \tag{20}$$

to hold for some bounded storage function λ and some $\alpha \in \mathcal{K}_\infty$, where $\bar{\boldsymbol{s}}$ is a steady state point. Condition (20) is difficult to interpret and verify. However, it simply entails the existence of an MPC scheme

$$\min_{\boldsymbol{x},\boldsymbol{u}} \quad -\lambda\left(\boldsymbol{s}\right) + \tilde{T}\left(\boldsymbol{x}_N\right) + \sum_{k=0}^{N-1} \tilde{L}\left(\boldsymbol{x}_k, \boldsymbol{u}_k\right) \tag{21a}$$

$$\text{s.t.} \quad \text{(7b)}, \text{(7c)} \tag{21b}$$

with $\tilde{L}$ lower-bounded by a class-$\mathcal{K}_\infty$ function, which delivers the same policy and value function as (7). This simple observation opens the door for enforcing (nominal) stability by design in RL for MPC. We detail that next.

5.1 Stability by verification vs. stability in learning

In the literature, dissipativity is typically considered a property to verify rather than to enforce. This verification is unfortunately fairly complex to perform, see e.g. [13]. When adjusting an MPC scheme (10) via the MPC model (10b) only, one has to incur that complexity.

> However, when combining RL and MPC in the fully parametrized form (10), the stability question can be approached as a fairly simple a priori design requirement rather than a complex a posteriori verification.

Indeed, using a parametrized form for (21):

$$\min_{\boldsymbol{x},\boldsymbol{u}} \quad -\lambda_{\boldsymbol{\theta}}\left(\boldsymbol{s}\right) + T_{\boldsymbol{\theta}}\left(\boldsymbol{x}_N\right) + \sum_{k=0}^{N-1} L_{\boldsymbol{\theta}}\left(\boldsymbol{x}_k, \boldsymbol{u}_k\right) \tag{22a}$$

$$\text{s.t.} \quad \text{(10b)}, \text{(10c)} \tag{22b}$$

and requiring the modified stage cost $L_{\boldsymbol{\theta}}$ to fulfil:

$$L_{\boldsymbol{\theta}}\left(\boldsymbol{s}, \boldsymbol{a}\right) \geq \alpha\left(\left\|\boldsymbol{s} - \bar{\boldsymbol{s}}_{\boldsymbol{\theta}}\right\|\right), \quad \forall \boldsymbol{s}, \boldsymbol{a} = \boldsymbol{\pi}_{\boldsymbol{\theta}}(\boldsymbol{s}) \tag{23}$$

for some steady state $\bar{\boldsymbol{s}}_{\boldsymbol{\theta}}$, ensures that the parametrized MPC scheme (22) is (nominally) dissipative. The added storage function in (22) is required if one wants the MPC scheme (10) to be able to approximate correctly the value functions in addition to the optimal policy. MPC (22) can then be treated via the RL methods detailed in Sec. 4, to the addition of restriction (23) on the MPC parameters proposed by RL. In [14], it has been shown that treating the parameterized MPC scheme in (22) yields a valid storage function that satisfies the dissipativity condition.

5.2 What cost function parametrization?

Two points are then useful to clarify here. First, if the baseline cost L does not satisfy condition (23), then it cannot be used as an initial guess for $L_{\boldsymbol{\theta}}$, i.e. the initial MPC parameters $\boldsymbol{\theta}$ need to

yield an initial $L_{\boldsymbol{\theta}}$ that is possibly very different than L. It is then not obvious how to select a meaningful initial MPC stage cost $L_{\boldsymbol{\theta}}$ to the learning tools. Fortunately, a simple approximate approach can be used here. Indeed, in a learning context where the MPC model (10b) is not accurate and where a fully parametrized MPC (10) is considered, it is arguably not necessarily productive to compute a modified cost $L_{\boldsymbol{\theta}}$ and storage function $\lambda_{\boldsymbol{\theta}}$ that leaves the MPC policy and value functions perfectly unchanged. Instead, one can e.g. provide a quadratic stage cost as an initial guess for $L_{\boldsymbol{\theta}}$, which produces the same policy and value function as the original MPC scheme in a neighborhood of the closed-loop steady state, see [15]. RL can then improve on that guess without jeopardizing stability.

The second point to clarify here is that requirement (23) is not necessarily easy to satisfy in practice because it yields a semi-infinite constraint on the parameters $\boldsymbol{\theta}$.

A simpler approach is to adopt a parametrization of the cost $L_{\boldsymbol{\theta}}$ that satisfies (23) by construction, e.g. using a strictly convex cost parametrization, see [7]. Such a choice has the additional advantage of making the MPC scheme significantly easier to solve than if using a non-convex stage cost.

Unfortunately, choosing a convex parametrization of the cost $L_{\boldsymbol{\theta}}$ is more restrictive than the original requirement (23), which can then prevent MPC (22) from reaching the optimal policy and value functions. It can then become important to be able to adjust the MPC model (10b) to alleviate this potential issue. A currently open question is how rich the model parametrization ought to be in order for MPC (22) to reach the optimal policy and value functions with a convex cost parametrization.

5.3 Current Challenges

The observations provided above apply to the nominal stability of the MPC scheme, i.e. to the stability of the MPC if applied in closed-loop to its own model. While nominal stability is clearly a highly desirable property, it does not discuss closed-loop stability in the presence of stochasticity in the real dynamics (1) and model error. These issues can arguably be addressed through Robust MPC techniques, see Sec. 6. However, Robust MPC adopts conservative approaches, which can degrade the closed-loop performance. Another intriguing approach is the use of the functional dissipativity theory presented in [16], which extends dissipativity to MDPs and hence to stochastic problems. However, this concept has not been explored yet in the context of learning.

6 Learning Safe Policies via MPC

The application of RL in safety-critical applications is drawing research attention. Safety is easiest defined through a set of critical constraints on the state of the real system, which should not be violated. A classic approach to safe RL is to learn the policy in silico, on a “pessimistic" model of the real system, in the sense that the model ought to overestimate the probability of

violating critical constraints. The in silico learning can then be performed by assigning high penalties to violations of the critical constraints, e.g. through the use of barrier functions [17]. RL will then naturally adjust the policy to avoid these penalties.

In the context of MPC, the use of a pessimistic model of the system naturally finds its place in the context of Robust MPC (RMPC). From a pessimistic model, RMPC builds a safe policy by ensuring that the worst-case predictions satisfy the critical constraints at all future time. The deterministic model (10b) is then replaced by a model that describes the evolution of sets enclosing all possible future trajectories, e.g. in the form:

$$\mathbb{X}_{k+1} = \boldsymbol{f}_{\boldsymbol{\theta}}(\mathbb{X}_k, \boldsymbol{\pi}_{\mathrm{c}}(\mathbb{X}_k, \boldsymbol{x}_k, \boldsymbol{u}_k)) \oplus \mathbb{W}_{\boldsymbol{\theta}} \tag{24}$$

where $\boldsymbol{\pi}_{\mathrm{c}}$ is a policy "managing" the growth of the sets $\mathbb{X}_k$, usually operating on their deviation from a reference trajectory $\boldsymbol{x}_k$. Set $\mathbb{W}_{\boldsymbol{\theta}}$ accounts for the possible process noise aimed at capturing the prediction uncertainties. Learning (24) from is done through *set-membership identification*. The satisfaction of the constraints (10c) is then enforced for all points in these sets, either explicitly of implicitly. An explicit construction can take the form:

$$\begin{aligned}
\min_{\boldsymbol{x},\boldsymbol{u},\mathbb{X}} \quad & T_{\boldsymbol{\theta}}\left(\boldsymbol{x}_N\right) + \sum_{k=0}^{N-1} L_{\boldsymbol{\theta}}\left(\boldsymbol{x}_k, \boldsymbol{u}_k\right) && \text{(25a)}\\
\text{s.t.} \quad & \boldsymbol{x}_{k+1} = \boldsymbol{f}_{\boldsymbol{\theta}}\left(\boldsymbol{x}_k, \boldsymbol{u}_k\right), \quad (24) && \text{(25b)}\\
& \boldsymbol{h}_{\boldsymbol{\theta}}\left(\mathbb{X}_k, \boldsymbol{\pi}_{\mathrm{c}}(\mathbb{X}_k, \boldsymbol{x}_k, \boldsymbol{u}_k)\right) \leq 0, && \text{(25c)}\\
& \boldsymbol{\pi}_{\mathrm{c}}(\mathbb{X}_k, \boldsymbol{x}_k, \boldsymbol{u}_k) \in \mathcal{A} && \text{(25d)}\\
& \mathbb{X}_N \in \mathbb{T}_{\boldsymbol{\theta}}, \quad \boldsymbol{x}_0 = \boldsymbol{s}, \quad \mathbb{X}_0 = \boldsymbol{s} && \text{(25e)}
\end{aligned}$$

MPC (25) generates a safe policy $\boldsymbol{\pi}_{\boldsymbol{\theta}} = \boldsymbol{u}_0^\star$ by construction for $\mathbb{T}_{\boldsymbol{\theta}}$ adequately chosen and if (24) accounts for the worst case situations observed in the data, see [1]. The use of RL to adjust RMPC schemes has been proposed in [18]. In the context of RL for RMPC-based policies, it is useful to stress that there is a "hard" separation between learning for safety and learning for closed-loop performance. Indeed, safety is learned via set-membership identification on (24) and then enforced in the RMPC scheme by construction. Closed-loop performance is optimized using RL in parallel [18]. This echoes the remarks proposed in Sec. 3.2.

In safe RL based on RMPC, the role of the model becomes clear and plain again: it consists in ensuring the safety of the policies generated by MPC, and the "fitting" of the model to the real system ought to be seen in the sense of set-membership identification.

For the sake of clarity, we ought to underline that the adjustment of the constraints (25c) in the RMPC scheme and of set $\mathbb{W}_{\boldsymbol{\theta}}$ ought to be done with care in order to preserve safety. In particular, the adjustment of set $\mathbb{W}_{\boldsymbol{\theta}}$ must ensure that model (24) accounts for all past data points, in the set-membership sense. Arguably, safety-critical constraints in (25c) ought not to be modified.

6.1 Safe Exploration

Safe exploration is difficult to produce without a model of the system in the form (24), which can predict the worst-case evolution of the system, and assess the impact of the exploration on the system safety. However, even with a model (24) of the system, it can be expensive to verify the safety of an input differing from the safe policy, and even more expensive to build the set of safe inputs.

Fortunately, the use of RMPC as a tool to generate a safe policy offers a straightforward way to generate safe exploration, which does not require more computations than solving the RMPC itself.

Indeed, the feasible exploration approach detailed in Sec. 4.4 can be readily applied to the RMPC formulation (25). Then, because only the RMPC cost is modified, the resulting solution is feasible for (25), and therefore if (25) yields safe policy, then a safe exploration is produced. This principle has been further detailed in [18].

6.2 Safe Policies and Safe Learning

An important question when performing safe learning online on a running system, either in a batch fashion (i.e. by collecting a certain amount of data before computing a parameter update) or not (i.e. performing a parameter update at every time step), is whether safety is preserved through the parameter updates or not. Indeed, while taking actions from a safe and stable policy ensures the stability and safety of the system, taking actions from a sequence of safe and stable, but changing policies may not. That is because a sequence of policies does not necessarily inherit the properties of the individual policies.

Hence if stability and safety are of importance for the system at hand, and the parameter updates are performed while the system is running, specific conditions ought to be satisfied for a parameter update to be implemented.

These conditions are detailed in [8].

6.3 Current challenges

RMPC provides a solid methodology to provide policies that are formally safe. RMPC is fairly straightforward to use if the selected MPC model f_θ is linear in the states and inputs, and if set $\mathbb{W}$ is “simple" (e.g. polytopic or ellipsoidal). However, RMPC remains difficult to implement formally if the MPC model is nonlinear. This restricts the use of RMPC for generating formally safe policies for specific classes or problems, i.e. those where a linear MPC model can perform reasonably well. RMPC for generic problems can be deployed, e.g. using scenario-based approaches [19], or set integrators [20]. However, while effective in practice, the former approach does not provide formal guarantees of safety. The latter approach can be fairly complex to use. Even in the case of RMPC using a linear model, adjusting (24)

for closed-loop performance while ensuring that it captures the worst-case situations observed in the data is difficult on big data sets. This difficulty is further discussed in [18].

7 Conclusion

This paper made a general and coherent review of the foundations, theories, and essential results that recently have developed in the context of RL based on MPC. We also expressed the applications and challenges ahead. We reviewed how a parameterized MPC scheme can learn the optimal policies and the value functions of a given MDP, even if the model used in the MPC scheme cannot capture the real system. We showed how RL algorithms and concepts such as exploration and sensitivity could be formulated in the context of MPC. Some advantages of the method, such as the nominal stability of the closed-loop system and safety, were summarized.

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] Sergio Lucia and Benjamin Karg. "A deep learning-based approach to robust nonlinear model predictive control". In: *IFAC-PapersOnLine* 51.20 (2018), pp. 511–516.

[3] Zhe Wu et al. "Machine learning-based predictive control of nonlinear processes. Part I: theory". In: *AIChE Journal* 65.11 (2019), e16729.

[4] Florian Dorfler, Jeremy Coulson, and Ivan Markovsky. "Bridging direct & indirect data-driven control formulations via regularizations and relaxations". In: *IEEE Transactions on Automatic Control* (2022).

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

[6] Arash Bahari Kordabad, Mario Zanon, and Sebastien Gros. "Equivalency of Optimality Criteria of Markov Decision Process and Model Predictive Control". In: *arXiv preprint arXiv:2210.04302* (2022).

[7] Katrine Seel et al. "Convex neural network-based cost modifications for learning model predictive control". In: *IEEE Open Journal of Control Systems* 1 (2022), pp. 366–379.

[8] Sebastien Gros and Mario Zanon. "Learning for MPC with stability & safety guarantees". In: *Automatica* 146 (2022), p. 110598.

[9] S Anand Akhil et al. "A Painless Deterministic Policy Gradient Method For MPC". In: *[submitted]* (2022).

[10] 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.

[11] Sebastien Gros and Mario Zanon. "Towards safe reinforcement learning using NMPC and policy gradients: Part II-deterministic case". In: *arXiv preprint arXiv:1906.04034* (2019).

[12] Shambhuraj Sawant et al. "Offline-Model Predictive Control using Reinforcement Learning". In: *[submitted]* (2022).

[13] David Angeli, Rishi Amrit, and James B Rawlings. "On average performance and stability of economic model predictive control". In: *IEEE transactions on automatic control* 57.7 (2011), pp. 1615–1626.

[14] Arash Bahari Kordabad and Sebastien Gros. "Q-learning of the storage function in Economic Nonlinear Model Predictive Control". In: *Engineering Applications of Artificial Intelligence* 116 (2022), p. 105343.

[15] Mario Zanon, Sébastien Gros, and Moritz Diehl. "A tracking MPC formulation that is locally equivalent to economic MPC". In: *Journal of Process Control* 45 (2016), pp. 30–42.

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

[17] Richard Cheng et al. "End-to-end safe reinforcement learning through barrier functions for safety-critical continuous control tasks". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 33. 2019, pp. 3387–3395.

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

[19] Arash Bahari Kordabad et al. "Reinforcement learning based on scenario-tree MPC for ASVs". In: *2021 American Control Conference (ACC)*. IEEE. 2021, pp. 1985–1990.

[20] Boris Houska and Mario E Villanueva. "Robust optimization for MPC". In: *Handbook of model predictive control*. Springer, 2019, pp. 413–443.

K Bias correction of discounted optimal steady state using cost modification

Preprint of [104] **Arash Bahari Kordabad** and Sebastien Gros. “Bias correction of discounted optimal steady state using cost modification”. In: *Submitted* (2022)

Bias correction of discounted optimal steady state using cost modification

Arash Bahari Kordabad[1] and Sebastien Gros[1]

[1]Department of Engineering Cybernetics, Norwegian University of Science and Technology (NTNU), Trondheim, Norway.

Abstract: In the literature of Economic Model Predictive Control (EMPC) and undiscounted Optimal Control Problem (OCP), the optimal steady-state point is an equilibrium point with the minimum stage cost. If the Economic MPC is discounted, this property does not hold, and the optimal steady-state point is not the same as the one obtained from the undiscounted EMPC. Therefore the discounted steady state point does not yield minimum stage cost and has a bias with respect to the undiscounted one. In this paper, we propose a cost modification in the discounted MPC that results in the undiscounted optimal steady-state point, i.e., the steady-state point that leads to the best stage cost. Moreover, we will show that this modification does not affect the closed-loop system behavior. We will illustrate the proposed method with a numerical example.

1 Introduction

One of the central objectives in control engineering, especially in chemical processes, power networks, etc [1, 2], is to steer the closed-loop trajectories of a given system to a steady point that has the minimum stage cost. Mathematically, the optimal steady-state problem can be formulated as a constrained optimization problem, where the cost function is the stage cost, and its constraint is the equilibrium of the point. This concept also appears in Economic Model Predictive Control (EMPC) problems, where the purpose is not tracking but to minimize a generic stage cost, such as time, energy and etc [3].

MPC schemes are generally formulated in an undiscounted setting. However, in some cases, it is reasonable to introduce a discount factor in the objective [4, 5]. Discounted OCP has drawn wide attention in, e.g., economic application [6] and social science [7]. In the discounted setting, the stage costs are weighted by a factor γ^k, where $\gamma \in (0, 1)$ is labeled discount factor, and k is the physical time index in discrete-time systems. Hence, the discount factor gives more importance to the present than the future and yields a well-posed value function [8]. A discounted infinite-horizon objective function is often the preferred setting in dynamic programming [8, 9] and reinforcement learning [10, 11].

The optimal steady state, resulting from the discounted OCP, differs from the optimal steady-state obtained from the undiscounted OCP. Although the discounted optimal steady-state is optimal in the sense of discounted OCP, the discounted optimal steady-state point does not result in the minimum one-step stage cost [12]. The bias between the discounted optimal steady-state and the undiscounted optimal steady-state depends on the discount factor, and tends to zero as the discount factor tends to one.

Recently, theories have been developed that explain the equivalency between discounted and non-discounted OCP in both deterministic [12] and stochastic systems [13]. These theories state that by modifying the stage cost in OCPs, one can establish equivalency of the discounted and undiscounted settings. I.e., they provide the same optimal policy and optimal value function. It should be noted that this modification requires knowing the exact optimal value function of the original problem [5].

Unfortunately, in some cases, the optimal value function of an undiscounted OCP is not well-posed, especially for stochastic systems. Even if it exists, the exact optimal value function of the original problem is difficult to compute for high-dimensional systems [14].

In this paper, we provide an inexpensive approximated cost modification using a second-order Taylor expansion of the optimal value function at the optimal steady state. We provide simple tools to compute the gradient and curvature needed for the approximation. Moreover, it will be shown that the approximated cost modification preserves the stability of the closed-loop system locally.

The paper is structured as follows. Section 2 provides the optimal steady-state problems in both undiscounted and discounted settings. Section 3 recalls the equivalency theorem of the discounted and undiscounted settings and provides the exact cost modification. Section 4 details the main contribution of the current paper and provides the approximated cost modification. This section states how the optimal policy based on the approximate stage cost does not invalidate the original closed-loop system stability locally. Section 5 illustrates a numerical example and Section 6 delivers a conclusion.

2 Problem Formulation

In this section, we will provide the problem formulation and detail the optimal steady-state optimization in both the discounted and undiscounted settings. Consider the following discrete-time deterministic dynamical system:

$$\boldsymbol{x}_+ = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}) \tag{1}$$

where $\boldsymbol{x}$ is the state, $\boldsymbol{u}$ is the input, $\boldsymbol{f}$ is the dynamics, and $\boldsymbol{x}_+$ is the successive state. During this transition, the system receives a scalar stage cost $L(\boldsymbol{x}, \boldsymbol{u})$. This stage cost may reflect an economic cost, often corresponding to the energy, the time or the financial cost of running a system [15]. Furthermore, the system is subject to the state and constraints as follows:

$$(\boldsymbol{x}, \boldsymbol{u}) \in \mathbb{Z} := \{(\boldsymbol{x}, \boldsymbol{u}) \,|\, \boldsymbol{h}(\boldsymbol{x}, \boldsymbol{u}) \leq 0\}\,. \tag{2}$$

The following standard assumption is essential in the OCP context, and will be used in the rest of the paper.

Assumption 1. *The stage cost $L(\cdot)$ and the dynamics $\boldsymbol{f}(\cdot)$ are continuous and at least twice differentiable functions. Moreover, set $\mathbb{Z}$ is compact.*

2.1 Undiscounted Optimal Steady-State

An infinite-horizon undiscounted OCP based on the stage cost L is defined as follows:

$$V^\star(\boldsymbol{x}) = \min_{\boldsymbol{\pi}} \quad \sum_{k=0}^{\infty} L(\boldsymbol{x}_k, \boldsymbol{\pi}(\boldsymbol{x}_k)) \tag{3a}$$

$$\text{s.t.} \quad \boldsymbol{x}_{k+1} = \boldsymbol{f}(\boldsymbol{x}_k, \boldsymbol{\pi}(\boldsymbol{x}_k))\,, \tag{3b}$$

$$(\boldsymbol{x}_k, \boldsymbol{\pi}(\boldsymbol{x}_k)) \in \mathbb{Z}, \qquad \boldsymbol{x}_0 = \boldsymbol{x}\,, \tag{3c}$$

where policy $\boldsymbol{\pi}$ is a map from the state space to the input space and $V^\star$ is the optimal value function. In some cases, the undiscounted optimal value function $V^\star$ might not be well-posed necessarily, especially for stochastic systems, but in this paper, we assume that $V^\star$ is well-posed.

We then denote the optimal policy solution of (3) by $\boldsymbol{\pi}^\star$.

An optimal steady-state pair with respect to the economic stage cost L is defined as follows:

$$(\boldsymbol{x}^\star, \boldsymbol{u}^\star) \in \arg\min_{(\boldsymbol{x},\boldsymbol{u})\in\mathbb{Z}} \quad L(\boldsymbol{x}, \boldsymbol{u}) \tag{4a}$$

$$\text{s.t.} \quad \boldsymbol{x} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}) \tag{4b}$$

Assumption 2. *Without loss of generality, we assume that $L(\boldsymbol{x}^\star, \boldsymbol{u}^\star) = 0$. If this does not hold, one can shift the stage cost to achieve $L(\boldsymbol{x}^\star, \boldsymbol{u}^\star) = 0$. Moreover we assume that $(\boldsymbol{x}^\star, \boldsymbol{u}^\star)$ is an interior point of $\mathbb{Z}$.*

Next Lemma provides a classic result that we will use in the paper.

Lemma 1. *Under Assumptions 1 and 2, the following identities hold:*

$$\boldsymbol{\pi}^\star(\boldsymbol{x}^\star) = \boldsymbol{u}^\star, \quad V^\star_{\boldsymbol{x}}(\boldsymbol{x}^\star) = \boldsymbol{\lambda}^\star \tag{5}$$

where $(\cdot)_{\boldsymbol{x}}$ is the gradient of $(\cdot)$ w.r.t. $\boldsymbol{x}$ and $\boldsymbol{\lambda}^\star$ is the optimal Lagrange multiplier of (4).

Proof. Since $(\boldsymbol{x}^\star, \boldsymbol{u}^\star)$ is an interior point of $\mathbb{Z}$, the Lagrangian of (4) reads:

$$\mathcal{L}(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{\lambda}) = L(\boldsymbol{x}, \boldsymbol{u}) + \boldsymbol{\lambda}^\top(\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}) - \boldsymbol{x}); \tag{6}$$

for $(\boldsymbol{x}, \boldsymbol{u}) \in \mathbb{Z}$. The Necessary Conditions of Optimality (NCO) of (4) reads:

$$\mathcal{L}_{\boldsymbol{x}}(\boldsymbol{x}^\star, \boldsymbol{u}^\star, \boldsymbol{\lambda}^\star) = \mathcal{L}_{\boldsymbol{u}}(\boldsymbol{x}^\star, \boldsymbol{u}^\star, \boldsymbol{\lambda}^\star) = 0, \quad \boldsymbol{x}^\star = \boldsymbol{f}(\boldsymbol{x}^\star, \boldsymbol{u}^\star) \tag{7}$$

where

$$\mathcal{L}_{\boldsymbol{x}}(\boldsymbol{x},\boldsymbol{u},\boldsymbol{\lambda}) = L_{\boldsymbol{x}}(\boldsymbol{x},\boldsymbol{u}) + \boldsymbol{f}_{\boldsymbol{x}}(\boldsymbol{x},\boldsymbol{u})\boldsymbol{\lambda} - \boldsymbol{\lambda} \tag{8a}$$

$$\mathcal{L}_{\boldsymbol{u}}(\boldsymbol{x},\boldsymbol{u},\boldsymbol{\lambda}) = L_{\boldsymbol{u}}(\boldsymbol{x},\boldsymbol{u}) + \boldsymbol{f}_{\boldsymbol{u}}(\boldsymbol{x},\boldsymbol{u})\boldsymbol{\lambda} \tag{8b}$$

and where $(\cdot)_{\boldsymbol{u}}$ is the gradient of $(\cdot)$ w.r.t. $\boldsymbol{u}$.

On the other hand, consider the Bellman equation associated with (3):

$$V^{\star}(\boldsymbol{x}) = \min_{\boldsymbol{u}} L(\boldsymbol{x},\boldsymbol{u}) + V^{\star}(\boldsymbol{f}(\boldsymbol{x},\boldsymbol{u})) \tag{9}$$

for all $\boldsymbol{x}$ with the solution $\boldsymbol{u} = \boldsymbol{\pi}^{\star}(\boldsymbol{x})$. Taking the derivation from both sides of (9), we have:

$$V^{\star}_{\boldsymbol{x}}(\boldsymbol{x}) = L_{\boldsymbol{x}}(\boldsymbol{x},\boldsymbol{u}) + \boldsymbol{f}_{\boldsymbol{x}}(\boldsymbol{x},\boldsymbol{u})V^{\star}_{\boldsymbol{x}}(\boldsymbol{f}(\boldsymbol{x},\boldsymbol{u})) \tag{10a}$$

$$\boldsymbol{0} = L_{\boldsymbol{u}}(\boldsymbol{x},\boldsymbol{u}) + \boldsymbol{f}_{\boldsymbol{u}}(\boldsymbol{x},\boldsymbol{u})V^{\star}_{\boldsymbol{x}}(\boldsymbol{f}(\boldsymbol{x},\boldsymbol{u})) \tag{10b}$$

for all $\boldsymbol{x}$ with $\boldsymbol{u} = \boldsymbol{\pi}^{\star}(\boldsymbol{x})$, specifically at $\boldsymbol{x}^{\star}$ with unknowns $\boldsymbol{\pi}^{\star}(\boldsymbol{x}^{\star})$ and $V^{\star}_{\boldsymbol{x}}(\boldsymbol{x}^{\star})$. One can verify that, using (7) the following choice:

$$\boldsymbol{\pi}^{\star}(\boldsymbol{x}^{\star}) = \boldsymbol{u}^{\star}, \quad V^{\star}_{\boldsymbol{x}}(\boldsymbol{x}^{\star}) = \boldsymbol{\lambda}^{\star} \tag{11}$$

solves (10). ■

Note that under Assumption 2 and using Lemma 1, one can verify that $V^{\star}(\boldsymbol{x}^{\star}) = 0$.

2.2 Discounted Optimal Steady-State

In the discounted setting, the stage cost at the current time takes a greater weight than the stage cost at future times. The discounted OCP can be defined as follows:

$$V^{\gamma,\star}(\boldsymbol{x}) = \min_{\boldsymbol{\pi}} \quad \sum_{k=0}^{\infty} \gamma^k L(\boldsymbol{x}_k, \boldsymbol{\pi}(\boldsymbol{x}_k)) \tag{12a}$$

$$\text{s.t.} \quad \text{(3b), (3c)}\,, \tag{12b}$$

where $\gamma \in (0,1)$ is the discount factor and $V^{\gamma,\star}$ is the optimal value function in the discounted setting. We then denote the discounted optimal policy solution of (12) by $\boldsymbol{\pi}^{\gamma,\star}$.

The optimal steady point, resulting from the discounted setting, is given by [12]:

$$(\boldsymbol{x}^{\gamma,\star}, \boldsymbol{u}^{\gamma,\star}) \in \arg \min_{(\boldsymbol{x},\boldsymbol{u})\in\mathbb{Z}} \quad L(\boldsymbol{x},\boldsymbol{u}) + (\gamma-1)V^{\star,\gamma}(\boldsymbol{x}) \tag{13a}$$

$$\text{s.t.} \quad \boldsymbol{x} = \boldsymbol{f}(\boldsymbol{x},\boldsymbol{u}) \tag{13b}$$

From the optimality of (4), one can verify that:

$$L(\boldsymbol{x}^{\star},\boldsymbol{u}^{\star}) \leq L(\boldsymbol{x}^{\gamma,\star},\boldsymbol{u}^{\gamma,\star}) \tag{14}$$

I.e. the undiscounted optimal steady-state $(\boldsymbol{x}^\star, \boldsymbol{u}^\star)$ results in better performance than the discounted optimal steady-state $(\boldsymbol{x}^{\gamma,\star}, \boldsymbol{u}^{\gamma,\star})$ at the steady state, by definition.

In this paper, we aim to find a modified stage cost $\bar{L}$ such that the associated discounted OCP delivers the undiscounted optimal steady-state point $(\boldsymbol{x}^\star, \boldsymbol{u}^\star)$ and hence corrects the bias between $(\boldsymbol{x}^{\gamma,\star}, \boldsymbol{u}^{\gamma,\star})$ and $(\boldsymbol{x}^\star, \boldsymbol{u}^\star)$. Consider the following discounted setting:

$$\bar{V}^{\gamma,\star}(\boldsymbol{x}) = \min_{\boldsymbol{\pi}} \quad \sum_{k=0}^{\infty} \gamma^k \bar{L}(\boldsymbol{x}_k, \boldsymbol{\pi}(\boldsymbol{x}_k)) \tag{15a}$$

$$\text{s.t.} \quad \text{(3b), (3c)}\,, \tag{15b}$$

where $\bar{V}^{\gamma,\star}$ is the modified optimal value function and $\bar{L}$ is the modified stage cost. The optimal policy of (15) is denoted by $\bar{\boldsymbol{\pi}}^{\gamma,\star}$. In the next section, we provide the modification $\bar{L}$ based on the theorems developed in [12, 13].

3 Exact Cost Modification

In this section, we provide the exact stage cost modification of the discounted OCP in order to correct the bias in the optimal steady-state and get the undiscounted optimal steady-state point. We use the idea, developed in [12, 13], which states the equivalency between discounted OCP and undiscounted OCP using cost modification. Before expressing this idea, we need to introduce the concept of *dissipativity* [16, 17].

The closed-loop stability of systems with economic (i.e. generic) stage costs function requires that a dissipation inequality is satisfied [18, 19]. For the undiscounted setting, system (1) with the stage cost L is (undiscounted) dissipative if there exists a continuous *storage* function μ satisfying:

$$L(\boldsymbol{x}, \boldsymbol{u}) + \mu(\boldsymbol{x}) - \mu(\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u})) \geq \alpha(\|\boldsymbol{x} - \boldsymbol{x}^\star\|) \tag{16}$$

for all $(\boldsymbol{x}, \boldsymbol{u}) \in \mathbb{Z}$ and some $\alpha \in \mathcal{K}_\infty$. Note that adding a constant in the storage function does not invalidate (16). Hence, we can assume that $\mu(\boldsymbol{x}^\star) = 0$ without loss of generality.

The stability and dissipativity condition for discounted OCP is more involved than for the undiscounted setting, and has been recently established [12]. In the discounted case, the discount factor γ has a central role in establishing the closed-loop stability of the system.

For system (1) with stage cost $\bar{L}$ and the discount factor γ, the resulting dissipativity conditions are called Strong Discounted Strict Dissipativity (SDSD). The SDSD conditions guarantee asymptotic stability of the closed-loop dynamics $\boldsymbol{f}$ with the discounted optimal policy $\bar{\boldsymbol{\pi}}^{\star,\gamma}$. More specifically, the tuple $(\boldsymbol{f}, \bar{L}, \gamma)$ is SDSD if there exists a continuous storage function μ satisfying:

$$\bar{L}(\boldsymbol{x}, \boldsymbol{u}) + \mu(\boldsymbol{x}) - \gamma\mu(\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u})) \geq \alpha(\|\boldsymbol{x} - \boldsymbol{x}^\star\|) \tag{17a}$$

$$\bar{L}(\boldsymbol{x}, \boldsymbol{u}) + \mu(\boldsymbol{x}) - \mu(\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u})) + (\gamma - 1)\bar{V}^{\star,\gamma}(\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u})) \geq \alpha(\|\boldsymbol{x} - \boldsymbol{x}^\star\|) \tag{17b}$$

for all $(\boldsymbol{x}, \boldsymbol{u}) \in \mathbb{Z}$ and some $\alpha \in \mathcal{K}_\infty$.

Next Lemma states the exact cost modification of the discounted setting $\bar{L}$ in order to get the optimal value function and the optimal policy of the undiscounted setting.

Lemma 2. *Suppose that Assumptions 1 and 2 hold, if the system $\boldsymbol{f}$ with the stage cost L is dissipative, then the following identities hold:*

$$\bar{\boldsymbol{\pi}}^{\gamma,\star}(\boldsymbol{x}) = \boldsymbol{\pi}^\star(\boldsymbol{x}), \qquad \bar{V}^{\gamma,\star}(\boldsymbol{x}) = V^\star(\boldsymbol{x})\,, \tag{18}$$

if we select the modified stage cost as follows:

$$\bar{L}(\boldsymbol{x}, \boldsymbol{u}) = L(\boldsymbol{x}, \boldsymbol{u}) + (1-\gamma)(V^\star(\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}))) \tag{19}$$

Moreover, the tuple $(\boldsymbol{f}, \bar{L}, \gamma)$ is SDSD if and only if the system $\boldsymbol{f}$ with the stage cost L is dissipative.

Proof. First, we prove the second argument. If the system $\boldsymbol{f}$ with the stage cost L is dissipative, then we define:

$$L^{\mathrm{R}}(\boldsymbol{x}, \boldsymbol{u}) := L(\boldsymbol{x}, \boldsymbol{u}) + \mu(\boldsymbol{x}) - \mu(\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u})) \geq \alpha(\|\boldsymbol{x} - \boldsymbol{x}^\star\|) \tag{20}$$

Substitution of (19) into (20) implies:

$$\bar{L}(\boldsymbol{x}, \boldsymbol{u}) + (\gamma - 1)V^\star(\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u})) + \mu(\boldsymbol{x}) - \mu(\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u})) \geq \alpha(\|\boldsymbol{x} - \boldsymbol{x}^\star\|) \tag{21}$$

We then aim to show that:

$$V^\star(\boldsymbol{x}) + \mu(\boldsymbol{x}) \geq 0 \tag{22}$$

From the definition of $V^\star$ in (3) and using a telescoping sum:

$$\begin{aligned} V^\star(\boldsymbol{x}) &= \sum_{k=0}^{\infty} L(\boldsymbol{x}_k^{\boldsymbol{\pi}^\star}, \boldsymbol{\pi}^\star(\boldsymbol{x}_k^{\boldsymbol{\pi}^\star})) = -\mu(\boldsymbol{x}) + \mu(\boldsymbol{x}_\infty^{\boldsymbol{\pi}^\star}) + \sum_{k=0}^{\infty} L^{\mathrm{R}}(\boldsymbol{x}_k^{\boldsymbol{\pi}^\star}, \boldsymbol{\pi}^\star(\boldsymbol{x}_k^{\boldsymbol{\pi}^\star})) \\ &= -\mu(\boldsymbol{x}) + \sum_{k=0}^{\infty} L^{\mathrm{R}}(\boldsymbol{x}_k^{\boldsymbol{\pi}^\star}, \boldsymbol{\pi}^\star(\boldsymbol{x}_k^{\boldsymbol{\pi}^\star})) \end{aligned} \tag{23}$$

where $\boldsymbol{x}_\infty^{\boldsymbol{\pi}^\star} := \lim_{k\to\infty} \boldsymbol{x}_k^{\boldsymbol{\pi}^\star} = \boldsymbol{x}^\star$. Then from $\mu(\boldsymbol{x}^\star) = 0$, we have:

$$V^\star(\boldsymbol{x}) + \mu(\boldsymbol{x}) = \sum_{k=0}^{\infty} L^{\mathrm{R}}(\boldsymbol{x}_k^{\boldsymbol{\pi}^\star}, \boldsymbol{\pi}^\star(\boldsymbol{x}_k^{\boldsymbol{\pi}^\star})) \geq \alpha(\|\boldsymbol{x} - \boldsymbol{x}^\star\|) \tag{24}$$

By multiplying both sides of (22) by the positive factor $1 - \gamma$:

$$(1-\gamma)V^\star(\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u})) + (1-\gamma)\mu(\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u})) \geq 0 \tag{25}$$

and summing (25) and (19):

$$\bar{L}(\boldsymbol{x}, \boldsymbol{u}) + \mu(\boldsymbol{x}) - \gamma\mu(\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u})) \geq \alpha(\|\boldsymbol{x} - \boldsymbol{x}^\star\|) \tag{26}$$

which results in SDSD for $(\boldsymbol{f}, \bar{L}, \gamma)$. If $(\boldsymbol{f}, \bar{L}, \gamma)$ is SDSD then:

$$\bar{L}(\boldsymbol{x}, \boldsymbol{u})+(\gamma - 1)V^\star(\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u})) + \mu(\boldsymbol{x}) - \mu(\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u})) \geq \alpha(\|\boldsymbol{x} - \boldsymbol{x}^\star\|) \tag{27}$$

and substitution of (19) into (27), we have:

$$L(\boldsymbol{x}, \boldsymbol{u}) + \mu(\boldsymbol{x}) - \mu(\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u})) \geq \alpha(\|\boldsymbol{x} - \boldsymbol{x}^\star\|) \tag{28}$$

which results in dissipativity for the system $\boldsymbol{f}$ and stage cost L. The first argument can be directly obtained from the second argument and Theorem 3 in [12]. ■

This Lemma shows that the cost modification in (19) produces a discounted optimal policy and optimal value functions that are the same as the undiscounted case. Moreover, the closed-loop stabilities are equivalent. One of the main consequences is then that this cost modification steers the system trajectories to the undiscounted steady state $(\boldsymbol{x}^\star, \boldsymbol{u}^\star)$ for any γ.

Unfortunately, the cost modification in (19) requires the exact optimal value function $V^\star$. In most cases, due to the curse of dimensionality, evaluating the optimal value function is extremely expensive [14]. We will then provide an inexpensive approximate cost modification that resolves these difficulties. We detail this in the next section.

4 Approximate Cost Modification

In this section, we provide an approximation of the cost modification that does not require the optimal value function $V^\star$. More specifically, we are looking for an approximated stage cost $\tilde{L} \approx \bar{L}$ without knowledge of the optimal value function $V^\star$ such that the resulting optimal policy of discounted OCP steers the system trajectory to the undiscounted optimal steady-state $(\boldsymbol{x}^\star, \boldsymbol{u}^\star)$ for any γ. Moreover, we show that this approximation preserves the (local) closed-loop stability.

In order to provide the approximate stage cost $\tilde{L}$, we will approximate the exact cost modification in (19), using a second-order Taylor expansion of the optimal value function $V^\star$ around the optimal steady-state $(\boldsymbol{x}^\star, \boldsymbol{u}^\star)$. This takes the form:

$$\begin{aligned}\bar{L}(\boldsymbol{x}, \boldsymbol{u}) \approx &\tilde{L}(\boldsymbol{x}, \boldsymbol{u}) := \\ &L(\boldsymbol{x}, \boldsymbol{u}) + (1 - \gamma)\Big(V_{\boldsymbol{x}}^\star(\boldsymbol{x}^\star)^\top \boldsymbol{g}(\boldsymbol{x}, \boldsymbol{u}) + \frac{1}{2}\boldsymbol{g}^\top(\boldsymbol{x}, \boldsymbol{u})V_{\boldsymbol{xx}}^\star(\boldsymbol{x}^\star)\boldsymbol{g}(\boldsymbol{x}, \boldsymbol{u})\Big)\,,\end{aligned} \tag{29}$$

where $\boldsymbol{g}(\boldsymbol{x}, \boldsymbol{u}) = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}) - \boldsymbol{x}^\star$, with $\boldsymbol{g}(\boldsymbol{x}^\star, \boldsymbol{u}^\star) = 0$. Using Lemma 1 and the notation definition $V_{\boldsymbol{xx}}^\star(\boldsymbol{x}^\star) := S^\star$, (29) reads as:

$$\tilde{L}(\boldsymbol{x}, \boldsymbol{u}) =L(\boldsymbol{x}, \boldsymbol{u}) + (1 - \gamma)\Big((\boldsymbol{\lambda}^\star)^\top \boldsymbol{g}(\boldsymbol{x}, \boldsymbol{u}) + \frac{1}{2}\boldsymbol{g}^\top(\boldsymbol{x}, \boldsymbol{u})S^\star\boldsymbol{g}(\boldsymbol{x}, \boldsymbol{u})\Big)\,, \tag{30}$$

Taking derivation of (10a) at $\boldsymbol{x}^\star$, $S^\star$ satisfies the following equality:

$$S^\star = L_{\boldsymbol{xx}}(\boldsymbol{x}^\star, \boldsymbol{u}^\star) + \boldsymbol{f}_{\boldsymbol{xx}}(\boldsymbol{x}^\star, \boldsymbol{u}^\star)\boldsymbol{\lambda}^\star + \boldsymbol{f}_{\boldsymbol{x}}^\top(\boldsymbol{x}^\star, \boldsymbol{u}^\star)S^\star\boldsymbol{f}_{\boldsymbol{x}}(\boldsymbol{x}^\star, \boldsymbol{u}^\star)\,, \tag{31}$$

We then define the following discounted OCP based on the approximated modified stage cost $\tilde{L}$:

$$\tilde{V}^{\gamma,\star}(\boldsymbol{x}) = \min_{\boldsymbol{\pi}} \sum_{k=0}^{\infty} \gamma^k \tilde{L}(\boldsymbol{x}_k, \boldsymbol{\pi}(\boldsymbol{x}_k)) \tag{32a}$$

$$\text{s.t. (3b), (3c)}\,, \tag{32b}$$

where $\tilde{V}^{\gamma,\star}$ is the approximated optimal value function and we denote the optimal policy solution of (32) by $\tilde{\boldsymbol{\pi}}^{\gamma,\star}$.

Next Lemma shows that the gradient and curvature of the discounted optimal value function $\tilde{V}^{\gamma,\star}$ based on the approximated stage cost $\tilde{L}$ are identical to the gradient and curvature of the undiscounted optimal value function $V^\star$, at the optimal steady-state $\boldsymbol{x}^\star$, i.e., $\boldsymbol{\lambda}^\star$ and $S^\star$, respectively. Moreover, we will show that the optimal policy $\tilde{\boldsymbol{\pi}}^{\gamma,\star}$ admits the undiscounted optimal steady-state $(\boldsymbol{x}^\star, \boldsymbol{u}^\star)$ as an equilibrium point.

Lemma 3. *The following identities hold:*

$$\tilde{\boldsymbol{\pi}}^{\gamma,\star}(\boldsymbol{x}^\star) = \boldsymbol{u}^\star, \quad \tilde{V}^{\gamma,\star}_{\boldsymbol{x}}(\boldsymbol{x}^\star) = \boldsymbol{\lambda}^\star, \quad \tilde{V}^{\gamma,\star}_{\boldsymbol{x}\boldsymbol{x}}(\boldsymbol{x}^\star) = S^\star, \tag{33}$$

Proof. The Bellman equation associated with (32) reads:

$$\tilde{V}^{\gamma,\star}(\boldsymbol{x}) = \min_{\boldsymbol{u}} \tilde{L}(\boldsymbol{x}, \boldsymbol{u}) + \gamma \tilde{V}^{\gamma,\star}(\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u})) \tag{34}$$

with the optimal solution $\tilde{\boldsymbol{\pi}}^{\gamma,\star}(\boldsymbol{x})$. Using (30) and NCO for (34):

$$\begin{aligned} \tilde{V}^{\gamma,\star}_{\boldsymbol{x}}(\boldsymbol{x}) =& L_{\boldsymbol{x}}(\boldsymbol{x}, \boldsymbol{u}) + (1-\gamma)\Big(\boldsymbol{g}_{\boldsymbol{x}}(\boldsymbol{x}, \boldsymbol{u})\boldsymbol{\lambda}^\star + \qquad (35a)\\ & \boldsymbol{g}_{\boldsymbol{x}}(\boldsymbol{x}, \boldsymbol{u}) S^\star \boldsymbol{g}(\boldsymbol{x}, \boldsymbol{u})\Big) + \gamma \boldsymbol{f}_{\boldsymbol{x}}(\boldsymbol{x}, \boldsymbol{u}) \tilde{V}^{\gamma,\star}_{\boldsymbol{x}}(\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u})) \\ \mathbf{0} =& L_{\boldsymbol{u}}(\boldsymbol{x}, \boldsymbol{u}) + (1-\gamma)\Big(\boldsymbol{g}_{\boldsymbol{u}}(\boldsymbol{x}, \boldsymbol{u})\boldsymbol{\lambda}^\star + \qquad (35b)\\ & \boldsymbol{g}_{\boldsymbol{u}}(\boldsymbol{x}, \boldsymbol{u}) S^\star \boldsymbol{g}(\boldsymbol{x}, \boldsymbol{u})\Big) + \gamma \boldsymbol{f}_{\boldsymbol{u}}(\boldsymbol{x}, \boldsymbol{u}) \tilde{V}^{\gamma,\star}_{\boldsymbol{x}}(\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u})) \end{aligned}$$

for all $\boldsymbol{x}$ with $\boldsymbol{u} = \tilde{\boldsymbol{\pi}}^{\gamma,\star}(\boldsymbol{x})$, specifically at $(\boldsymbol{x}^\star, \tilde{\boldsymbol{\pi}}^{\gamma,\star}(\boldsymbol{x}^\star))$. By selecting $\tilde{\boldsymbol{\pi}}^{\gamma,\star}(\boldsymbol{x}^\star) = \boldsymbol{u}^\star$, we have:

$$\tilde{V}^{\gamma,\star}_{\boldsymbol{x}}(\boldsymbol{x}^\star) = L_{\boldsymbol{x}}(\boldsymbol{x}^\star, \boldsymbol{u}^\star) + (1-\gamma)\boldsymbol{g}_{\boldsymbol{x}}(\boldsymbol{x}^\star, \boldsymbol{u}^\star)\boldsymbol{\lambda}^\star + \gamma \boldsymbol{f}_{\boldsymbol{x}}(\boldsymbol{x}^\star, \boldsymbol{u}^\star)\tilde{V}^{\gamma,\star}_{\boldsymbol{x}}(\boldsymbol{x}^\star)\,, \tag{36a}$$

$$\mathbf{0} = L_{\boldsymbol{u}}(\boldsymbol{x}^\star, \boldsymbol{u}^\star) + (1-\gamma)\boldsymbol{g}_{\boldsymbol{u}}(\boldsymbol{x}^\star, \boldsymbol{u}^\star)\boldsymbol{\lambda}^\star + \gamma \boldsymbol{f}_{\boldsymbol{u}}(\boldsymbol{x}^\star, \boldsymbol{u}^\star)\tilde{V}^{\gamma,\star}_{\boldsymbol{x}}(\boldsymbol{x}^\star)\,, \tag{36b}$$

using the fact that $\boldsymbol{f}_{\boldsymbol{x}} = \boldsymbol{g}_{\boldsymbol{x}}$ and $\boldsymbol{f}_{\boldsymbol{u}} = \boldsymbol{g}_{\boldsymbol{u}}$ and selecting $\tilde{V}^{\gamma,\star}_{\boldsymbol{x}}(\boldsymbol{x}^\star) = \boldsymbol{\lambda}^\star$, one can easily verify (36) using Lemma 1.

For the curvature of $\tilde{V}^{\gamma,\star}$, taking derivation of (35a) at $\boldsymbol{x}^\star$ and using the fact that $\boldsymbol{g}(\boldsymbol{x}^\star, \boldsymbol{u}^\star) = 0$, we have:

$$\tilde{V}^{\gamma,\star}_{\boldsymbol{x}\boldsymbol{x}}(\boldsymbol{x}^\star) = \tilde{L}_{\boldsymbol{x}\boldsymbol{x}}(\boldsymbol{x}^\star, \boldsymbol{u}^\star) + \gamma \boldsymbol{f}_{\boldsymbol{x}\boldsymbol{x}}(\boldsymbol{x}^\star, \boldsymbol{u}^\star)\boldsymbol{\lambda}^\star + \gamma \boldsymbol{f}^\top_{\boldsymbol{x}}(\boldsymbol{x}^\star, \boldsymbol{u}^\star)\tilde{V}^{\gamma,\star}_{\boldsymbol{x}\boldsymbol{x}}(\boldsymbol{x}^\star)\boldsymbol{f}_{\boldsymbol{x}}(\boldsymbol{x}^\star, \boldsymbol{u}^\star) \tag{37}$$

where

$$\tilde{L}_{\boldsymbol{xx}}(\boldsymbol{x}^\star, \boldsymbol{u}^\star) = L_{\boldsymbol{xx}}(\boldsymbol{x}^\star, \boldsymbol{u}^\star) + (1-\gamma)(\boldsymbol{g}_{\boldsymbol{xx}}(\boldsymbol{x}^\star, \boldsymbol{u}^\star)\boldsymbol{\lambda}^\star + \boldsymbol{g}_{\boldsymbol{x}}^\top(\boldsymbol{x}^\star, \boldsymbol{u}^\star) S^\star \boldsymbol{g}_{\boldsymbol{x}}(\boldsymbol{x}^\star, \boldsymbol{u}^\star))$$

Using (31) and $\boldsymbol{f}_{\boldsymbol{xx}} = \boldsymbol{g}_{\boldsymbol{xx}}$ and $\boldsymbol{f}_{\boldsymbol{uu}} = \boldsymbol{g}_{\boldsymbol{uu}}$, one can see that the solution $\tilde{V}^{\gamma,\star}_{\boldsymbol{xx}}(\boldsymbol{x}^\star) = S^\star$ satisfies (37). ■

This Lemma shows that the optimal policy resulting from the approximated cost modification admits the undiscounted steady state as an equilibrium point. In the following, we show that this optimal policy stabilizes the closed-system trajectories locally if the original system is dissipative.

To establish the local closed-loop stability, first, we use Lemma 2 to construct the undiscounted equivalence setting of (32). More specifically, the system $\boldsymbol{f}$ with the stage cost $\tilde{L}$ and discount factor γ is SDSD if and only if the system $\boldsymbol{f}$ is (undiscounted) dissipative with the following stage cost:

$$\tilde{L}^\gamma(\boldsymbol{x}, \boldsymbol{u}) := \tilde{L}(\boldsymbol{x}, \boldsymbol{u}) + (\gamma - 1)\tilde{V}^{\gamma,\star}(\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u})) \tag{38}$$

Note that in (38), we have used a reverse argument of (19), i.e., we built an equivalent stage cost $\tilde{L}$ which its undiscounted OCP results in the same optimal policy and optimal value function of undiscounted OCP with the stage cost $\tilde{L}$. Therefore the stability of the policy $\tilde{\boldsymbol{\pi}}^{\gamma,\star}$ turns into the (undiscounted) dissipativity of the system with the stage cost $\tilde{L}^\gamma$. Next theorem states this relationship.

Theorem 1. *Under Assumptions 1 and 2, if the system $\boldsymbol{f}$ with the stage cost L is dissipative, then the system is locally dissipative with the stage cost $\tilde{L}^\gamma$ with policy $\tilde{\boldsymbol{\pi}}^{\gamma,\star}$.*

Proof. If the system $\boldsymbol{f}$ with the stage cost L is dissipative, then there exist μ and $\alpha \in \mathcal{K}_\infty$ such that:

$$L(\boldsymbol{x}, \boldsymbol{u}) + \mu(\boldsymbol{x}) - \mu(\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u})) \geq \alpha(\|\boldsymbol{x} - \boldsymbol{x}^\star\|) \tag{39}$$

Then the aim is to show that the following inequality holds locally:

$$\tilde{L}^\gamma(\boldsymbol{x}, \boldsymbol{u}) + \mu(\boldsymbol{x}) - \mu(\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u})) \geq \tilde{\alpha}(\|\boldsymbol{x} - \boldsymbol{x}^\star\|) \tag{40}$$

for some $\tilde{\alpha} \in \mathcal{K}_\infty$ and $\boldsymbol{u} = \tilde{\boldsymbol{\pi}}^{\gamma,\star}(\boldsymbol{x})$. Using (38) and (30), (40) can be written as follows:

$$L(\boldsymbol{x}, \boldsymbol{u}) + (\gamma - 1)h(\boldsymbol{x}, \boldsymbol{u}) + \mu(\boldsymbol{x}) - \mu(\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u})) \geq \tilde{\alpha}(\|\boldsymbol{x} - \boldsymbol{x}^\star\|)$$

where

$$h(\boldsymbol{x}, \boldsymbol{u}) := \tilde{V}^{\gamma,\star}(\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u})) - (\boldsymbol{\lambda}^\star)^\top \boldsymbol{g}(\boldsymbol{x}, \boldsymbol{u}) - \frac{1}{2}\boldsymbol{g}^\top(\boldsymbol{x}, \boldsymbol{u}) S^\star \boldsymbol{g}(\boldsymbol{x}, \boldsymbol{u}) \tag{41}$$

From Lemma 3 and the Taylor theorem, we have the following:

$$h(\boldsymbol{x}, \boldsymbol{u}) \sim \mathcal{O}(\|\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}) - \boldsymbol{x}^\star\|^3) \tag{42}$$

Under assumption 1 and along the optimal policy $\tilde{\boldsymbol{\pi}}^{\gamma,\star}$, we have:

$$h(\boldsymbol{x},\boldsymbol{u}) \sim \mathcal{O}(\|\boldsymbol{x}-\boldsymbol{x}^\star\|^3) \tag{43}$$

for $\boldsymbol{u}=\tilde{\boldsymbol{\pi}}^{\gamma,\star}(\boldsymbol{x})$. It has been shown in [20] that every EMPC is locally equivalent to an LQR for dissipative problems. It yields that, locally, the stage cost L and storage function μ can be represented in the quadratic form. Then $\alpha \in \mathcal{K}_\infty$ has also quadratic form locally, i.e., $\alpha(\|\boldsymbol{x}-\boldsymbol{x}^\star\|^2)=\kappa\|\boldsymbol{x}-\boldsymbol{x}^\star\|^2$ for some positive constant κ for all $\boldsymbol{x}$ in some neighbourhood. More specifically, (39) can be written as follows:

$$L(\boldsymbol{x},\boldsymbol{u})+\mu(\boldsymbol{x})-\mu(\boldsymbol{f}(\boldsymbol{x},\boldsymbol{u})) \geq \kappa\|\boldsymbol{x}-\boldsymbol{x}^\star\|^2 \tag{44}$$

From (43) and along the optimal policy trajectory, there exists a neighbourhood around $\boldsymbol{x}^\star$ such that the following holds:

$$-\frac{\kappa}{2(1-\gamma)}\|\boldsymbol{x}-\boldsymbol{x}^\star\|^2 \leq h(\boldsymbol{x},\tilde{\boldsymbol{\pi}}^{\gamma,\star}(\boldsymbol{x})) \leq \frac{\kappa}{2(1-\gamma)}\|\boldsymbol{x}-\boldsymbol{x}^\star\|^2 \tag{45}$$

or:

$$-\frac{\kappa}{2}\|\boldsymbol{x}-\boldsymbol{x}^\star\|^2 \leq (\gamma-1)h(\boldsymbol{x},\tilde{\boldsymbol{\pi}}^{\gamma,\star}(\boldsymbol{x}))\,, \tag{46}$$

summing (44) and (46) yields:

$$L(\boldsymbol{x},\boldsymbol{u})+(\gamma-1)h(\boldsymbol{x},\boldsymbol{u})+\mu(\boldsymbol{x})-\mu(\boldsymbol{f}(\boldsymbol{x},\boldsymbol{u})) \geq \frac{\kappa}{2}\|\boldsymbol{x}-\boldsymbol{x}^\star\|^2$$

for $\boldsymbol{u}=\tilde{\boldsymbol{\pi}}^{\gamma,\star}(\boldsymbol{x})$ and some neighborhood around $\boldsymbol{x}^\star$. Selecting $\tilde{\alpha}(\cdot)=\frac{\kappa}{2}\|\cdot\|^2$ yields (41) and it completes the proof. ■

This theorem shows that under some mild assumptions on the smoothness of the dynamics $\boldsymbol{f}$ and stage cost L, the approximated cost modification $\tilde{L}$ based on the second-order Taylor approximation results in the optimal policy $\tilde{\boldsymbol{\pi}}^{\gamma,\star}$ that is stabilizing for the undiscounted optimal steady-state $(\boldsymbol{x}^\star,\boldsymbol{u}^\star)$ locally even if the OCP setting is discounted.

In the next section, we propose an example in order to illustrate the theoretical developments.

5 Numerical Example

In this section, we provide a benchmark optimal investment problem to verify the proposed method [21]. Consider the following dynamics with a non-polynomial economic stage cost:

$$x_{k+1}=u_k\,, \qquad L(x,u)=-\ln(Ax^\alpha-u), \tag{47}$$

where x is the state, u is the input, and A and $0<\alpha<1$ are the constants. The state x denotes the investment in a company and the term Ax^α is the return from this investment after one

period. Then $Ax^\alpha - u$ is the amount of money that can be used for consumption in the current time period. Then the objective is to maximize the sum of the logarithmic utility function.

The (undiscounted) optimal steady-state is:

$$x^\star = u^\star = (\alpha A)^{\frac{1}{1-\alpha}} \tag{48}$$

In the discounted setting, it is known that for the discount factor γ, the optimal value and policy functions are $V^{\gamma,\star}(x) = B + C\ln(x)$ and $\pi^{\gamma,\star}(x) = \gamma\alpha A x^\alpha$, where (see, e.g., [22, 23]):

$$B = \frac{\ln((1-\alpha\gamma)A) + \frac{\gamma\alpha}{1-\gamma\alpha}\ln(\alpha\gamma A)}{\gamma - 1}, \quad C = \frac{\alpha}{\alpha\gamma - 1} \tag{49}$$

while the discounted steady state is:

$$x^{\gamma,\star} = u^{\gamma,\star} = (\gamma\alpha A)^{\frac{1}{1-\alpha}} \tag{50}$$

The Lagrangian function is:

$$\mathcal{L}(x, u, \lambda) = -\ln(Ax^\alpha - u) + \lambda(u - x); \tag{51}$$

with

$$\mathcal{L}_x = \frac{-\alpha A x^{\alpha-1}}{Ax^\alpha - u} - \lambda, \qquad \mathcal{L}_u = \frac{1}{Ax^\alpha - u} + \lambda \tag{52}$$

Then using NCO, one can verify that:

$$\frac{-\alpha A (x^\star)^{\alpha-1}}{A(x^\star)^\alpha - x^\star} = \lambda^\star \tag{53}$$

While the gradient of the optimal value function in the undiscounted setting is:

$$V_x^\star(x^\star) = \frac{\alpha}{x^\star(\alpha - 1)} \tag{54}$$

Then one can see that (53) and (54) are same at $x^\star = (\alpha A)^{\frac{1}{1-\alpha}}$. The second-order derivation can be obtained similarly. The exact modified stage cost and approximated modified stage cost can be written as follows:

$$\bar{L}(x, u) = L(x, u) + \frac{\alpha(1-\gamma)}{\alpha - 1}\ln(u) \tag{55a}$$

$$\tilde{L}(x, u) = L(x, u) + \frac{\alpha(1-\gamma)}{\alpha - 1}\Big(\frac{1}{u^\star}(u - u^\star) - \frac{1}{2(u^\star)^2}(u - u^\star)^2\Big) \tag{55b}$$

Figure 1 shows closed-loop trajectories with the different optimal policies, including exact and approximated modified stage costs. As it can be seen, the trajectory with $\pi^{\gamma,\star}$ converges to the discounted optimal steady-state $x^{\gamma,\star}$, while the trajectories with $\bar{\pi}^{\gamma,\star}$ and $\tilde{\pi}^{\gamma,\star}$ converge to the undiscounted optimal steady-state $x^\star$.

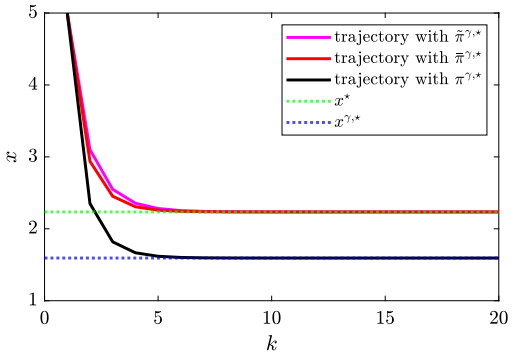


Figure 1: Closed-loop trajectories using different optimal policies and discounted and undiscounted optimal steady-state for $A = 5$, $\alpha = 0.34$ and $\gamma = 0.8$.

6 Conclusion

This paper provided an exact and approximated stage cost modification that the resulting optimal policy based on the discounted OCP steers the closed-loop system trajectories to the undiscounted optimal steady-state. The exact cost modification requires the knowledge of the optimal value function, while the approximated cost modification does not need such a requirement. We used the second-order Taylor expansion of the optimal value function to construct this approximated modified cost. We showed that this modification preserves the closed-loop stability and dissipativity property locally. We illustrated the proposed method in a benchmark example.

References

[1] RL Motard, M Shacham, and EM Rosen. “Steady state chemical process simulation”. In: *AIChE Journal* 21.3 (1975), pp. 417–436.

[2] Liam SP Lawrence, John W Simpson-Porco, and Enrique Mallada. “Linear-convex optimal steady-state control”. In: *IEEE Transactions on Automatic Control* 66.11 (2020), pp. 5377–5384.

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

[4] Martin L Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

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

[6] John Rust. “Dynamic programming”. In: *The new Palgrave dictionary of economics* 1 (2008), p. 8.

[7] Dieter Grass et al. “Optimal control of nonlinear processes”. In: *Berlino: Springer* (2008).

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

[9] Richard E Bellman and Stuart E Dreyfus. *Applied dynamic programming*. Vol. 2050. Princeton university press, 2015.

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

[11] Leslie Pack Kaelbling, Michael L Littman, and Andrew W Moore. “Reinforcement learning: A survey”. In: *Journal of artificial intelligence research* 4 (1996), pp. 237–285.

[12] Mario Zanon and Sébastien Gros. “A new dissipativity condition for asymptotic stability of discounted economic MPC”. In: *Automatica* 141 (2022), p. 110287.

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

[14] Warren B Powell. *Approximate Dynamic Programming: Solving the curses of dimensionality*. Vol. 703. John Wiley & Sons, 2007.

[15] Gabriele Pozzato et al. “Economic MPC for online least costly energy management of hybrid electric vehicles”. In: *Control Engineering Practice* 102 (2020), p. 104534.

[16] Arash Bahari Kordabad and Sebastien Gros. “Verification of dissipativity and evaluation of storage function in economic nonlinear MPC using q-learning”. In: *IFAC-PapersOnLine* 54.6 (2021), pp. 308–313.

[17] Arash Bahari Kordabad and Sebastien Gros. “Q-learning of the storage function in Economic Nonlinear Model Predictive Control”. In: *Engineering Applications of Artificial Intelligence* 116 (2022), p. 105343.

[18] James B Rawlings and Rishi Amrit. “Optimizing process economic performance using model predictive control”. In: *Nonlinear model predictive control*. Springer, 2009, pp. 119–138.

[19] Rishi Amrit, James B Rawlings, and David Angeli. “Economic optimization using model predictive control with a terminal cost”. In: *Annual Reviews in Control* 35.2 (2011), pp. 178–186.

[20] Mario Zanon, Sébastien Gros, and Moritz Diehl. “A tracking MPC formulation that is locally equivalent to economic MPC”. In: *Journal of Process Control* 45 (2016), pp. 30–42.

[21] William A Brock and Leonard J Mirman. “Optimal economic growth and uncertainty: the discounted case”. In: *Journal of Economic Theory* 4.3 (1972), pp. 479–513.

[22] Manuel S Santos and Jesus Vigo-Aguiar. “Analysis of a numerical dynamic programming algorithm applied to economic models”. In: *Econometrica* (1998), pp. 409–426.

[23] Lars Grüne, Christopher M Kellett, and Steven R Weller. “On a discounted notion of strict dissipativity”. In: *IFAC-PapersOnLine* 49.18 (2016), pp. 247–252.

References

[1] David Silver et al. “Mastering the game of Go with deep neural networks and tree search”. In: *nature* 529.7587 (2016), pp. 484–489.

[2] Qingrui Zhang, Wei Pan, and Vasso Reppa. “Model-reference reinforcement learning for collision-free tracking control of autonomous surface vehicles”. In: *IEEE Transactions on Intelligent Transportation Systems* (2021).

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

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

[5] Benjamin Karg and Sergio Lucia. “Learning-based approximation of robust nonlinear predictive control with state estimation applied to a towing kite”. In: *2019 18th European Control Conference (ECC)*. IEEE. 2019, pp. 16–22.

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

[7] Sridhar Mahadevan. “Average reward reinforcement learning: Foundations, algorithms, and empirical results”. In: *Machine learning* 22.1 (1996), pp. 159–195.

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

[9] Warren B Powell. *Approximate Dynamic Programming: Solving the curses of dimensionality*. Vol. 703. John Wiley & Sons, 2007.

[10] David Silver et al. “Deterministic policy gradient algorithms”. In: *International conference on machine learning*. PMLR. 2014, pp. 387–395.

[11] Michail G Lagoudakis and Ronald Parr. “Least-squares policy iteration”. In: *The Journal of Machine Learning Research* 4 (2003), pp. 1107–1149.

[12] 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).

[13] Hossein Nejatbakhsh Esfahani, Rafal Szlapczynski, and Hossein Ghaemi. “High performance super-twisting sliding mode control for a maritime autonomous surface ship (MASS) using ADP-Based adaptive gains and time delay estimation”. In: *Ocean Engineering* 191 (2019), p. 106526.

[14] 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.

[15] Glenn Bitar, Andreas B Martinsen, Anastasios M Lekkas, and Morten Breivik. “Two-Stage Optimized Trajectory Planning for ASVs Under Polygonal Obstacle Constraints: Theory and Experiments”. In: *IEEE Access* 8 (2020), pp. 199953–199969.

[16] E. Klintberg, J. Dahl, J. Fredriksson, and S. Gros. “An improved dual Newton strategy for scenario-tree MPC”. In: *IEEE 55th Conference on Decision and Control (CDC)*. 2016, pp. 3675–3681.

[17] Tor A Johansen, Andrea Cristofaro, and Tristan Perez. “Ship collision avoidance using scenario-based model predictive control”. In: *IFAC-PapersOnLine* 49.23 (2016), pp. 14–21.

[18] 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. DOI: 10.1109/CDC45484.2021.9683750.

[19] P. Kuhl, M. Diehl, T. Kraus, J. P. Schloder, and H. G. Bock. “A real-time algorithm for moving horizon state and parameter estimation”. In: *Computers and Chemical Engineering* 35.1 (2011), pp. 71–83.

[20] Julian Berberich, Johannes Köhler, Matthias A. Müller, and Frank Allgöwer. “Data-driven model predictive control: closed-loop guarantees and experimental results”. In: *at - Automatisierungstechnik* 69.7 (2021), pp. 608–618.

[21] Simon Muntwiler, Kim P. Wabersich, and Melanie N. Zeilinger. *Learning-based Moving Horizon Estimation through Differentiable Convex Optimization Layers*. 2021. arXiv: 2109.03962 [eess.SY].

[22] Bingheng Wang, Zhengtian Ma, Shupeng Lai, Lin Zhao, and Tong Heng Lee. *Differentiable Moving Horizon Estimation for Robust Flight Control*. 2021. arXiv: 2108.03212 [cs.RO].

[23] Benjamin Karg and Sergio Lucia. "Approximate moving horizon estimation and robust nonlinear model predictive control via deep learning". In: *Computers and Chemical Engineering* 148 (2021), p. 107266.

[24] Leslie Pack Kaelbling, Michael L. Littman, and Anthony R. Cassandra. "Planning and acting in partially observable stochastic domains". In: *Artificial Intelligence* 101.1 (1998), pp. 99–134.

[25] X. Zhong, Z. Ni, Y. Tang, and H. He. "Data-driven partially observable dynamic processes using adaptive dynamic programming". In: *in Proc. IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning*. 2014, pp. 1–8.

[26] Xuanchen Xiang and Simon Foo. "Recent Advances in Deep Reinforcement Learning Applications for Solving Partially Observable Markov Decision Processes (POMDP) Problems: Part 1—Fundamentals and Applications in Games, Robotics and Natural Language Processing". In: *Machine Learning and Knowledge Extraction* 3.3 (2021), pp. 554–581.

[27] Y. Wang, K. Velswamy, and B. Huang. "A Novel Approach to Feedback Control with Deep Reinforcement Learning". In: *IFAC-PapersOnLine* 51.18 (2018). 10th IFAC Symposium on Advanced Control of Chemical Processes ADCHEM 2018, pp. 31–36.

[28] Matthew J. Hausknecht and Peter Stone. "Deep Recurrent Q-Learning for Partially Observable MDPs". In: *CoRR* abs/1507.06527 (2015). arXiv: 1507.06527.

[29] Xiaodong Nian, Athirai A. Irissappane, and Diederik Roijers. "DCRAC: Deep Conditioned Recurrent Actor-Critic for Multi-Objective Partially Observable Environments". In: *International Foundation for Autonomous Agents and Multiagent Systems*. AAMAS '20. Auckland, New Zealand, 2020, pp. 931–938.

[30] Zhaohan Daniel Guo et al. "Neural Predictive Belief Representations". In: *CoRR* abs/1811.06407 (2018).

[31] T. Gangwani, J. Lehman, Q. Liu, and J. Peng. "Learning Belief Representations for Imitation Learning in POMDPs". In: *35th Conference on Uncertainty in Artificial Intelligence*. 2019, pp. 1–14.

[32] Hossein Nejatbakhsh Esfahani, Arash Bahari Kordabad, and Sebastien Gros. "Reinforcement learning based on MPC/MHE for unmodeled and partially observable dynamics". In: *2021 American Control Conference (ACC)* (2021), pp. 2121–2126. DOI: 10.23919/ACC50511.2021.9483399.

[33] Hossein Nejatbakhsh Esfahani, Arash Bahari Kordabad, Wenqi Cai, and Sébastien Gros. "Learning-based State Estimation and Control using MHE and MPC Schemes with Imperfect Models". In: *Submitted* (2022).

[34] Arne Groß, Antonia Lenders, Tobias Zech, Christof Wittwer, and Moritz Diehl. "Using Probabilistic Forecasts in Stochastic Optimization". In: *2020 International Conference on Probabilistic Methods Applied to Power Systems (PMAPS)*. IEEE. 2020, pp. 1–6.

[35] Arne Groß, Christof Wittwer, and Moritz Diehl. "Stochastic Model Predictive Control of Photovoltaic Battery Systems using a Probabilistic Forecast Model". In: *European Journal of Control* (2020).

[36] Warren B Powell and Stephan Meisel. "Tutorial on stochastic optimization in energy—Part I: Modeling and policies". In: *IEEE Transactions on Power Systems* 31.2 (2015), pp. 1459–1467.

[37] Pavithra Harsha and Munther Dahleh. "Optimal management and sizing of energy storage under dynamic pricing for the efficient integration of renewable energy". In: *IEEE Transactions on Power Systems* 30.3 (2014), pp. 1164–1181.

[38] Doron Lifshitz and George Weiss. "Optimal energy management for grid-connected storage systems". In: *Optimal Control Applications and Methods* 36.4 (2015), pp. 447–462.

[39] 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.

[40] 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. DOI: 10.1109/CDC45484.2021.9683333.

[41] 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.

[42] Vijay R Konda and John N Tsitsiklis. "Actor-critic algorithms". In: *Advances in neural information processing systems*. 2000, pp. 1008–1014.

[43] 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) [Submitted]*. 2021.

[44] Q. Mayne, E. C. Kerrigan, E. J. van Wyk, and P. Falugi. “Tube-based robust nonlinear model predictive control”. In: *International Journal of Robust and Nonlinear Control* 21.11 (2011), pp. 1341–1353.

[45] David Q. Mayne. “Model predictive control: Recent developments and future promise”. In: *Automatica* 50.12 (2014), pp. 2967–2986.

[46] M E. Villanueva, R Quirynen, M Diehl, B Chachuat, and B Houska. “Robust MPC via min–max differential inequalities”. In: *Automatica* 77 (2017), pp. 311–321.

[47] Z. J. Yu and L. T. Biegler. “Advanced-step multistage nonlinear model predictive control: Robustness and stability”. In: *Journal of Process Control* 84 (2019), pp. 192–206.

[48] S. Lucia, A. Ttulea-Codrean, C. Schoppmeyer, andS. Engell. “Rapid development of modular and sustainable nonlinear model predictive control solutions”. In: *Control Engineering Practice* 60 (2017), pp. 51–52.

[49] Hossein Nejatbakhsh Esfahani, Arash Bahari Kordabad, and Sebastien Gros. “Approximate Robust NMPC using Reinforcement Learning”. In: *2021 European Control Conference (ECC)* (2021), pp. 132–137. DOI: 10.23919/ECC54610.2021.9655129.

[50] James B Rawlings and Rishi Amrit. “Optimizing process economic performance using model predictive control”. In: *Nonlinear model predictive control*. Springer, 2009, pp. 119–138.

[51] David Angeli, Rishi Amrit, and James B Rawlings. “On average performance and stability of economic model predictive control”. In: *IEEE transactions on automatic control* 57.7 (2011), pp. 1615–1626.

[52] Rishi Amrit, James B Rawlings, and David Angeli. “Economic optimization using model predictive control with a terminal cost”. In: *Annual Reviews in Control* 35.2 (2011), pp. 178–186.

[53] Mario Zanon, Sébastien Gros, and Moritz Diehl. “A Lyapunov function for periodic economic optimizing model predictive control”. In: *52nd IEEE Conference on Decision and control*. IEEE. 2013, pp. 5107–5112.

[54] Timm Faulwasser, Lars Grüne, Matthias A Müller, et al. “Economic nonlinear model predictive control”. In: *Foundations and Trends® in Systems and Control* 5.1 (2018), pp. 1–98.

[55] Gabriele Pozzato, Matthias Müller, Simone Formentin, and Sergio M Savaresi. "Economic MPC for online least costly energy management of hybrid electric vehicles". In: *Control Engineering Practice* 102 (2020), p. 104534.

[56] Simon Pirkelmann, David Angeli, and Lars Grüne. "Approximate computation of storage functions for discrete-time systems using sum-of-squares techniques". In: *IFAC-PapersOnLine* 52.16 (2019), pp. 508–513.

[57] Mario Zanon and Sébastien Gros. "A new dissipativity condition for asymptotic stability of discounted economic MPC". In: *Automatica* 141 (2022), p. 110287.

[58] Dimitri P Bertsekas. *Reinforcement learning and optimal control*. Athena Scientific Belmont, MA, 2019.

[59] Jorge Nocedal and Stephen Wright. *Numerical optimization*. Springer Science & Business Media, 2006.

[60] Kay Hansel, Janosch Moos, and Cedric Derstroff. "Benchmarking the Natural Gradient in Policy Gradient Methods and Evolution Strategies". In: *Reinforcement Learning Algorithms: Analysis and Applications* (2021), pp. 69–84.

[61] Shun-ichi Amari. "Natural Gradient Works Efficiently in Learning". In: *Neural Computation* 10.2 (1998), pp. 251–276.

[62] Sham M Kakade. "A natural policy gradient". In: *Advances in neural information processing systems*. 2002, pp. 1531–1538.

[63] Thomas Furmston, Guy Lever, and David Barber. "Approximate newton methods for policy search in markov decision processes". In: *The Journal of Machine Learning Research* 17.1 (2016), pp. 8055–8105.

[64] Sean P Meyn and Richard L Tweedie. *Markov chains and stochastic stability*. Springer Science & Business Media, 2012.

[65] Romain Postoyan, Lucian Buşoniu, Dragan Nešić, and Jamal Daafouz. "Stability analysis of discrete-time infinite-horizon optimal control with discounted cost". In: *IEEE Transactions on Automatic Control* 62.6 (2016), pp. 2736–2749.

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

[67] RL Motard, M Shacham, and EM Rosen. "Steady state chemical process simulation". In: *AIChE Journal* 21.3 (1975), pp. 417–436.

[68] Liam SP Lawrence, John W Simpson-Porco, and Enrique Mallada. "Linear-convex optimal steady-state control". In: *IEEE Transactions on Automatic Control* 66.11 (2020), pp. 5377–5384.

[69] Martin L Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

[70] John Rust. "Dynamic programming". In: *The new Palgrave dictionary of economics* 1 (2008), p. 8.

[71] Dieter Grass, Jonathan P Caulkins, Gustav Feichtinger, Gernot Tragler, Doris A Behrens, et al. "Optimal control of nonlinear processes". In: *Berlino: Springer* (2008).

[72] Richard E Bellman and Stuart E Dreyfus. *Applied dynamic programming*. Vol. 2050. Princeton university press, 2015.

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

[74] Leslie Pack Kaelbling, Michael L Littman, and Andrew W Moore. "Reinforcement learning: A survey". In: *Journal of artificial intelligence research* 4 (1996), pp. 237–285.

[75] Mario Zanon and Sébastien Gros. "A new dissipativity condition for asymptotic stability of discounted economic MPC". In: *Automatica* 141 (2022), p. 110287.

[76] Nicolas Lanzetti et al. "Recurrent neural network based MPC for process industries". In: *2019 18th European Control Conference (ECC)*. IEEE. 2019, pp. 1005–1010.

[77] Felix Bünning et al. "Input convex neural networks for building MPC". In: *Learning for Dynamics and Control*. PMLR. 2021, pp. 251–262.

[78] Katrine Seel, Esten I Grøtli, Signe Moe, Jan T Gravdahl, and Kristin Y Pettersen. "Neural Network-Based Model Predictive Control with Input-to-State Stability". In: *2021 American Control Conference (ACC)*. IEEE. 2021, pp. 3556–3563.

[79] Arash Bahari Kordabad and Sebastien Gros. "Q-learning of the storage function in Economic Nonlinear Model Predictive Control". In: *Engineering Applications of Artificial Intelligence* 116 (2022), p. 105343. DOI: 0.1016/j.engappai.2022.105343.

[80] 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* 1 (2022), pp. 366–379. DOI: 10.1109/OJCSYS.2022.3221063.

[81] Kai Arulkumaran, Marc Peter Deisenroth, Miles Brundage, and Anil Anthony Bharath. “Deep reinforcement learning: A brief survey”. In: *IEEE Signal Processing Magazine* 34.6 (2017), pp. 26–38.

[82] Mathieu Granzotto, Romain Postoyan, Lucian Buşoniu, Dragan Nešić, and Jamal Daafouz. “Finite-horizon discounted optimal control: stability and performance”. In: *IEEE Transactions on Automatic Control* 66.2 (2020), pp. 550–565.

[83] Mario Zanon, Sébastien Gros, and Michele Palladino. “Stability-Constrained Markov Decision Processes Using MPC”. In: *Automatica* 143 (2022), p. 110399.

[84] Björn Lütjens, Michael Everett, and Jonathan P How. “Safe reinforcement learning with model uncertainty estimates”. In: *2019 International Conference on Robotics and Automation (ICRA)*. IEEE. 2019, pp. 8662–8668.

[85] Alexander T Schwarm and Michael Nikolaou. “Chance-constrained model predictive control”. In: *AIChE Journal* 45.8 (1999), pp. 1743–1752.

[86] R Tyrrell Rockafellar, Stanislav Uryasev, et al. “Optimization of conditional value-at-risk”. In: *Journal of risk* 2 (2000), pp. 21–42.

[87] Yinlam Chow and Mohammad Ghavamzadeh. “Algorithms for CVaR optimization in MDPs”. In: *Advances in neural information processing systems* 27 (2014).

[88] Yinlam Chow, Aviv Tamar, Shie Mannor, and Marco Pavone. “Risk-sensitive and robust decision-making: a cvar optimization approach”. In: *Advances in neural information processing systems* 28 (2015).

[89] Anton J Kleywegt, Alexander Shapiro, and Tito Homem-de-Mello. “The sample average approximation method for stochastic discrete optimization”. In: *SIAM Journal on Optimization* 12.2 (2002), pp. 479–502.

[90] Erick Delage and Yinyu Ye. “Distributionally robust optimization under moment uncertainty with application to data-driven problems”. In: *Operations research* 58.3 (2010), pp. 595–612.

[91] Emre Erdoğan and Garud Iyengar. “Ambiguous chance constrained problems and robust optimization”. In: *Mathematical Programming* 107.1 (2006), pp. 37–61.

[92] Zhaolin Hu and L Jeff Hong. “Kullback-Leibler divergence constrained distributionally robust optimization”. In: *Available at Optimization Online* (2013), pp. 1695–1724.

[93] Peyman Mohajerin Esfahani and Daniel Kuhn. “Data-driven distributionally robust optimization using the Wasserstein metric: Performance guarantees and tractable reformulations”. In: *Mathematical Programming* 171.1 (2018), pp. 115–166.

[94] Astghik Hakobyan and Insoon Yang. “Wasserstein distributionally robust motion control for collision avoidance using conditional value-at-risk”. In: *IEEE Transactions on Robotics* (2021).

[95] 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.

[96] 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. DOI: 10.23919/ECC54610.2021.9654852.

[97] 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. DOI: 10.1109/CCTA48906.2021.9659202.

[98] Arash Bahari Kordabad, Hossein Nejatbakhsh Esfahani, and Sebastien Gros. “Bias Correction in Deterministic Policy Gradient Using Robust MPC”. In: *2021 European Control Conference (ECC)* (2021), pp. 1086–1091. DOI: 10.23919/ECC54610.2021.9654962.

[99] 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. DOI: 10.23919/ACC53348.2022.9867217.

[100] Arash Bahari Kordabad and Sebastien Gros. “Functional Stability of Discounted Markov Decision Processes Using Economic MPC Dissipativity Theory”. In: *2022 European Control Conference (ECC)* (2022), pp. 1858–1863. DOI: 10.23919/ECC55457.2022.9838064.

[101] Arash Bahari Kordabad, Mario Zanon, and Sebastien Gros. “Equivalence of Optimality Criteria for Markov Decision Process and Model Predictive Control”. In: *arXiv preprint, Submitted* (2022). DOI: 10.48550/arXiv.2210.04302.

[102] Arash Bahari Kordabad, Rafal Wisniewski, and Sebastien Gros. “Safe Reinforcement Learning Using Wasserstein Distributionally Robust MPC and Chance Constraint”. In: *Submitted* (2022).

[103] Arash Bahari Kordabad, Dirk Reinhardt, Akhil S Anand, and Sebastien Gros. “Reinforcement Learning for MPC: Fundamentals and Current Challenges”. In: *Submitted* (2022).

[104] Arash Bahari Kordabad and Sebastien Gros. “Bias correction of discounted optimal steady state using cost modification”. In: *Submitted* (2022).

ISBN 978-82-326-5698-1 (printed ver.)
ISBN 978-82-326-6983-7 (electronic ver.)
ISSN 1503-8181 (printed ver.)
ISSN 2703-8084 (online ver.)

NTNU
Norwegian University of
Science and Technology