On Learning-Based Control of Dynamical Systems

Remy Hosseinkhan-Boucher

université
PARIS-SACLAY

On Learning-Based Control of Dynamical Systems

Sur les méthodes de contrôle par apprentissage des systèmes dynamiques

Thèse de doctorat de l'université Paris-Saclay

École doctorale n° 580
Sciences et Technologies de l'Information et de la Communication
Spécialité de doctorat: Informatique
Graduate School : Informatique et Sciences du Numérique
Référent : Faculté des sciences d'Orsay

Thèse préparée dans l'unité de recherche
Laboratoire Interdisciplinaire des Sciences du Numérique, (Université Paris-Saclay, CNRS),
sous la direction d'**Anne VILNAT**, professeur émérite à l'Université Paris-Saclay,
le co-encadrement de **Lionel MATHELIN**, chargé de recherche au CNRS,
et le co-encadrement d'**Onofrio SEMERARO**, chargé de recherche au CNRS.

Thèse soutenue à Paris-Saclay, le 10 avril 2025, par

Rémy HOSSEINKHAN-BOUCHER

Composition du jury

Membres du jury avec voix délibérative

Laurent CORDIER Directeur de recherche, Institut P' - CNRS - ISAE-ENSMA - Université de Poitiers	Président
Ana BUŠIĆ Chargée de recherche (HDR), Inria ARGO & ENS-PSL	Rapporteuse & Examinatrice
Tristan CAZENAVE Professeur, LAMSADE - CNRS - Université Paris Dauphine-PSL	Rapporteur & Examinateur
Emmanuel RACHELSON Professeur, ISAE-SUPAERO	Examinateur
Michèle SEBAG Directrice de recherche, LISN - CNRS - Université Paris-Saclay & Inria TAU	Examinatrice

THESE DE DOCTORAT
NNT : 2025UPASG029

université PARIS-SACLAY

ÉCOLE DOCTORALE
Sciences et technologies de l'information et de la communication (STIC)

Titre: Sur les méthodes de contrôle par apprentissage des systèmes dynamiques

Mots clés: Contrôle, Apprentissage par Renforcement, Commande Prédictive Basée sur l'Apprentissage, Apprentissage Profond, Systèmes Dynamiques, Contrôle des Écoulements

Résumé: Les impératifs environnementaux suscitent un regain d'intérêt pour la recherche sur le contrôle de l'écoulement des fluides afin de réduire la consommation d'énergie et les émissions dans diverses applications telles que l'aéronautique et l'automobile. Les stratégies de contrôle des fluides peuvent optimiser le système en temps réel, en tirant parti des mesures des capteurs et des modèles physiques. Ces stratégies visent à manipuler le comportement d'un système pour atteindre un état souhaité (stabilité, performance, consommation d'énergie).

Dans le même temps, le développement d'approches de contrôle pilotées par les données dans des domaines concurrents tels que les jeux et la robotique a ouvert de nouvelles perspectives pour le contrôle des fluides.

Cependant, l'intégration du contrôle basé sur l'apprentissage en dynamique des fluides présente de nombreux défis, notamment en ce qui concerne la robustesse de la stratégie de contrôle, l'efficacité de l'échantillon de l'algorithme d'apprentissage, et la présence de retards de toute nature dans le système.

Ainsi, cette thèse vise à étudier et à développer des stratégies de contrôle basées sur l'apprentissage en tenant compte de ces défis, dans lesquels deux classes principales de stratégies de contrôle basées sur les données sont considérées : l'apprentissage par renforcement (RL) et la commande prédictive basée sur l'apprentissage (LB-MPC). De multiples contributions sont apportées dans ce contexte.

Tout d'abord, un développement étendu sur la connexion entre les domaines du contrôle stochastique (temps continu) et du processus de décision de Markov (temps discret) est fourni pour unifier les deux approches.

Deuxièmement, des preuves empiriques sur les propriétés de régularisation de l'algorithme d'apprentissage par renforcement par maximum d'entropie sont présentées à travers des concepts d'apprentissage statistique pour mieux comprendre la caractéristique de robustesse de l'approche par maximum d'entropie.

Troisièmement, la notion d'abstraction temporelle est utilisée pour améliorer l'efficacité de l'échantillonnage d'un algorithme de commande prédictive par modèle basé sur l'apprentissage et piloté par une règle d'échantillonnage de la théorie de l'information.

Enfin, les modèles différentiels neuronaux sont introduits à travers le concept d'équations différentielles neuronales à retard pour modéliser des systèmes à temps continu avec des retards pour des applications en commande prédictive.

Les différentes études sont développées à l'aide de simulations numériques appliquées à des systèmes minimalistes issus des théories des systèmes dynamiques et du contrôle afin d'illustrer les résultats théoriques. Les expériences de la dernière partie sont également menées sur des simulations d'écoulement de fluides en 2D.

université PARIS-SACLAY

ÉCOLE DOCTORALE
Sciences et technologies de l'information et de la communication (STIC)

Title: On Learning-Based Control of Dynamical Systems

Keywords: Control, Reinforcement Learning, Model Predictve Control, Deep Learning, Dynamical Systems, Flow Control

Abstract:

Environmental needs are driving renewed research interest in fluid flow control to reduce energy consumption and emissions in various applications such as aeronautics and automotive industries. Flow control strategies can optimise the system in real time, taking advantage of sensor measurements and physical models. These strategies aim at manipulating the behaviour of a system to reach a desired state (*e.g.*, stability, performance, energy consumption).

Meanwhile, the development of data-driven control approaches in concurrent areas such as games and robotics has opened new perspectives for flow control.

However, the integration of learning-based control in fluid dynamics comes with multiple challenges, including the robustness of the control strategy, the sample efficiency of the learning algorithm, and the presence of delays of any nature in the system.

Thus, this thesis aims to study and develop learning-based control strategies with respect to these challenges where two main classes of data-driven control strategies are considered: Reinforcement Learning (RL) and Learning-based Model Predictive Control (LB-MPC). Multiple contributions are made in this context.

First, an extended development on the connection between the fields of (continuous-time) Stochastic Control and (discrete-time) Markov Decision Process is provided to bridge the gap between the two approaches.

Second, empirical evidence on the regularisation properties of the Maximum Entropy Reinforcement Learning algorithm is presented through statistical learning concepts to further understand the robustness feature of the Maximum Entropy approach.

Third, the notion of temporal abstraction is used to improve the sample efficiency of a Learning-based Model Predictive Control algorithm driven by an Information Theoretic sampling rule.

Lastly, neural differential models are introduced through the concept of Neural Delay Differential Equations to model continuous-time systems with delays for Model Predictive Control applications.

The different studies are developed with numerical simulations applied on minimalistic systems from Dynamical Systems and Control theories to illustrate the theoretical results. The training experiments of the last part are also conducted on 2D fluid flow simulations.

Contents

Symbols

Processes

X	State process
U	Control process
Y	Observation process
H	History process
v	Disturbance process
W	Brownian Motion
ξ	Initial process

Transition Probabilities

$\mathcal{P}$	State transition probability
$\mathcal{O}$	Time delay transition probability
$\mathcal{Q}$	State delay transition probability
π	Policy
$\mathcal{G}$	Observation transition probability

Operators

$\mathbb{P}$	Probability measure
Id	Identity operator
∇	Gradient operator
Δ	Laplace operator
f	Continuous Dynamics operator
g	Continuous Observation operator
F	Discrete Dynamics operator
G	Discrete Observation operator
J	Objective function
$\mathbb{E}$	Expectation operator
$\mathscr{L}$	Lagrangian
$\lambda^{\mathscr{L}}$	Adjoint state

Functions

c Cost function
V Value function
Q Q-value function
F Repartition function

Constants

T Time horizon
K Discrete time horizon
N Discretisation size
δ Time discretisation resolution

Scalars and Vectors

x_e Start equilibrium
x State point
u Control point
v Disturbance point
τ Time delay point
σ_e Start equilibrium standard deviation
γ Discount factor

Statistical Learning

θ Weights
$\mathcal{D}$ Dataset
$\mathcal{A}$ Algorithm
$\mathcal{L}$ General loss function
ℓ Learning task
m Sample complexity

Information Theory and Statistics

EIG Expected Information Gain
MI Mutual Information
$\mathcal{H}$ Entropy
D_{KL} Kullback-Leibler divergence
$\mathcal{I}$ Fisher Information
W Wasserstein Distance

Indexing

k Discrete time index
i Sampling index
n Sampling iteration/size
κ Random discrete time

η Random interdecision time
t Continuous time index

Maximum-Entropy

$\alpha^{\mathcal{H}}$ Entropy regularisation
$\mathcal{R}$ Excess Risk under noise
$\mathcal{M}$ Complexity measure

Gaussian Processes

$\mathcal{GP}$ Gaussian process (GP) distribution
μ GP mean operator
Σ GP covariance operator

Reinforcement Learning

ρ Discounted state-visitation distribution

Model Predictive Control

K^{MPC} MPC discrete time horizon
π^{MPC} MPC policy
$\widehat{\pi}^{\text{MPC}}$ MPC policy estimator

Distributional RL

Z Random objective value
F_q Quantile function
q Quantile
λ Quantile level
N_q Number of quantiles
$\mathcal{L}^{\text{QR}}$ Quantile regression loss
$\mathcal{T}$ Distributional Bellman operator

Spaces

$\mathcal{X}$ State space
$\mathcal{U}$ Control space
$\mathscr{A}_{\mathcal{U}}$ Admissible control space
$\mathscr{A}_{\Pi}$ Admissible policy space
$\mathcal{Y}$ Observation space
$\mathcal{V}$ Disturbance space
Ω Sample space
$\mathcal{F}$ Sample space σ-algebra
$\mathcal{S}$ Time delay space

Θ Weights space
$\mathscr{F}$ σ-algebra
I Time interval
d_X State dimension
d_U Control dimension
d_Y Observation dimension
d_z Spatial dimension
d_θ Weights dimension

Noise

ϵ_X State noise
ϵ_Y Observation noise

Preface

The humility, dedication, and knowledge of my teachers in the various fields of mathematics, statistics, economics, and computer science together with the power of the abstract tools emerging from those theories to solve real-world problems have always fascinated me. This respect and admiration for science and scientists is a significant motivation for my choice to pursue a PhD. Being part of the academic society and contributing to the scientific community is a great honour. Undeniably, there are numerous benefits associated with pursuing doctoral studies.

The reason for the choice of this thesis topic is not extraordinary. In 2016, the algorithm AlphaGo (Silver et al. 2016) achieved a major milestone in artificial intelligence research by defeating the world champion Go player. Prior to this, it was widely accepted that superhuman performances in Computer Go were beyond the capabilities of existing technology. The algorithm employed a decision-making process based on Deep Learning known as Reinforcement Learning (RL) which is one of the core theories used in this thesis. This approach is particularly important regarding the notion of autonomous learning and what is behind the idea of some agent "playing against itself" to improve. Thus, I got interested in the idea of contributing in the Learning-based Control field. The application to the control of fluid flows is particularly interesting since it opens the door to all the literature on Navier-Stokes equations and turbulence modelling.

Regarding the document structure, several reasons motivated the chapters' order. The first chapter introduces the challenges relative to Learning-based Control for Fluid Flows. The following two chapters introduce the mathematical framework of Stochastic Control from the continuous to discrete time setting. This allows for a unified view of the control problem that can be studied from both the discrete or continuous time angles. This set of chapters constitutes the first part of the thesis.

The second part of the thesis is dedicated to the methodological advances in Learning-based Control. The two first chapters of this part form two independent contributions to the literature that led to two separate publications in conferences (the 7^{th} International Conference in Optimization and Learning (OLA24) in Dubrovnik (Hosseinkhan Boucher, Semeraro, and Mathelin 2025) and the 6th Annual Learning for Dynamics & Control Conference (L4DC24) in Ox-

ford (Hosseinkhan Boucher, Douka, et al. 2024)). The work presented in L4DC is a collaboration with Stella Douka, during her internship within our research group. Last, the different work presented in the next two chapters are less mature. The chapter on Distributional RL aggregates the results obtained during the first months of this global research project while the chapter on neural differential models is an ongoing work on continuous control. The conclusion is the last chapter and concludes the thesis.

Finally, many people and institutions allowed me to prepare this project successfully. The next paragraph, written in French, acknowledges the set of people and institutions that contributed, directly or indirectly, to this accomplishment.

Avec l'ensemble des noms cités dans les remerciements ci-dessous, il est clairement possible de construire un graphe causal qui mène à la réalisation de ce projet de doctorat.

Tout d'abord, je tiens à remercier les membres du jury, Ana Bušić, Tristan Cazenave, Laurent Cordier, Michèle Sebag, et Emmanuel Rachelson, pour avoir accepté d'examiner ce travail de thèse et pour leurs retours constructifs. Ensuite, je remercie chaleureusement mes encadrants, Lionel Mathelin, Onofrio Semeraro, et Anne Vilnat, pour leur soutien, leur patience, et leur expertise. Par ailleurs, je remercie l'ensemble des chercheurs associés à ce projet de thèse, en particulier Luc Pastur et Sergio Chibbaro.

Je remercie aussi l'équipe d'encadrement de mon stage précédent la thèse, en particulier Michele Alessandro Bucci et Thibault Faney qui m'ont permis de m'initier au monde de la recherche académique et d'obtenir ce poste de doctorant.

Du côté des équipes de recherche, je remercie l'ensemble des membres de l'équipe Dataflot et plus largement le département mécanique-énergétique du LISN et plus particulièrement Caroline Nore, Anne Sergent, Yann Fraigneau et Didier Lucor. Je remercie également l'ensemble des membres de l'équipe Inria TAU et plus particulièrement Michèle Sebag, Guillaume Charpiat, Sylvain Chevallier, Cyril Furtlehner, et François Landes.

Au sein du laboratoire, je remercie l'équipe SAMI, notamment Laurent Pointal ainsi que l'équipe SPIL et le soutien quotidien de Romain Poirot. Je remercie aussi Christian pour ses discours quotidiens, dédiés (tous les jours à 16h) à ce que j'intègre OpenAI.

Pour la gestion des calculateurs haute performance, je remercie Rémi Lacroix et Loïc Estève pour l'IDRIS ainsi que Marco Léoni pour le mésocentre de l' Université Paris-Saclay.

Du côté des équipes pédagogiques avec qui j'ai eu l'occasion de travailler, je remercie Wassila Ouerdane (CentraleSupélec), Cécile Balkanski et Hélène Bonneau (Université Paris-Saclay) ainsi que Guillaume Charpiat (CentraleSupélec).

Sinon, je remercie Charles-Albert Lehalle pour sa considération continue et ses conseils avisés ainsi que Manfred Opper pour nos échanges constructifs lors du workshop à Cambridge. Jared Callaham, pour son accompagnement à l'usage du projet Hydrogym.

Je remercie d'ailleurs tous les chercheurs avec qui j'ai pu intéragir lors de ma participations à divers conférences et séminaires, notamment Jonathan Rivalan qui a largement égayé mon séjour à Dubrovnik, et Filipo Perotto pour sa sympathie et son esprit positif.

Pour ce qui est des institutions académiques, je remercie l'Université Paris-Saclay, l' Université Paris-Dauphine PSL (Jimmy Lamboley, Alexandre Afgoustidis, Jean-Paul. Tatiana Blondeel et bien d'autres), CentraleSupélec, l'Inria, et le CNRS.

Pour les entresprises, je remercie Luxurynsight (Antoine Auer, Jean-Louis Margoche, Jonathan Siboni), Capital Fund Management (Romain Picon, Gilles Masselot et bien d'autres), et BNP Paribas Real Estate (Samira Bouadi).

De l'autre côté, je remercie mes parents pour avoir soutenu ce long parcours.

Biensûr, je remercie tous mes co-doctorants qui ont contribué à rendre cette aventure plus agréable: Alice Lacan, Amine Saibi, Thibault Monsel, Manon Verbockhaven, Michele Quattromini, Yanis Zatout, Stéphane Février, Lucas Meyer, Emmanuel Menier, Mathieu Nastorg, Arthur Gesla, Soufiane Mrini, Nilo Schwencke, Melvin Creff, Nathan Carbonneau, Cyril, Cyriaque Rousselot, Rémi Bousquet, Romain Egelé, Sabrina Bernard, Gen.

Je remercie aussi mon équipe: Paul Amavi pour l'appui sans faille, Faaf, Samy, Doris, Yassine Guida, Bosh, L'oiseau, Pinot, Babecity (Oliv', Noé, Masco (ainsi que Massimo et Armelle), Micka (ainsi que Albert et Karina), Paulo, Beufa, Hédi, Lucas Santiago Stassart (ainsi que Fabienne), Seb, Jibé, Hakim, Amiral Nelson (maire du 9ème), Hugo, Medy, Benjamin (Richemont) Richmond, Hovo, Camel, AD, Lio, Maxence, Alli). Mais aussi, Daniel Haïk, Josh Kaji, Théo Deschamps, Hippolyte Le Roy Mayard, Orginto, Florian Bastin, Hadrien Mariaccia, Abel Nana Kouamen, Arthur Buigues, Marvin Bryant, Régis Lopez Kaufmann, Antoine Auer (maire du 3ème), Luc Baz, coach Tao, Younes, Céline, Majda, Lan, MX et Mochy, MR, Lou, Jenny, la Cheeky Family, Viviane Armand, Jules Armand, Romain Hosseinkhan-Boucher. Housni Mkouboi, Alix Mathurin.

Paris, le 22 Février 2025,

Hosseinkhan-Boucher, Rémy

1 Introduction

1.1 Motivation

1.1.1 Environmental Needs

In many areas of engineering, environmental needs are driving renewed research interest. A prime example is carbon dioxide emissions, widely considered to be one of the main causes of global warming (IPCC Core Writing Team, Lee, and Romero 2023). This urgency extends to many applications, including aeronautics, where it is recognised that optimising aerodynamic flows[1] can have a profound impact on reducing pollutant emissions and attenuating noise (Lumley and Blossey 2003). With this in mind, the role of flow control emerges as a crucial area of research, offering potential solutions to reduce energy loss and emissions.

1.1.2 Flow Control

In principle, *flow control* strategies (Ashill, Fulker, and Hackett 2005) can optimise the system in real time, taking advantage of sensor measurements and physical models. These strategies aim at influencing the behaviour of a system to reach a desired state (Trélat 2005). However, these techniques are currently only used in limited numerical and experimental cases.

1.2 Learning

1.2.1 Machine Learning

Meanwhile, the increasing computational power and storage capacity of modern computers allow for the development and scaling of data driven methods that were mostly restrained to theoretical solutions (Schmidhuber 2015). These methods belong to the larger concept of *Machine Learning (ML)* (see the book

[1] Aerodynamic drag approximately counts for 20% of the total energy loss on modern heavy duty vehicles (Vernet et al. 2014).

Mohri, Rostamizadeh, and Talwalkar 2018, for an introduction). Machine Learning can crudely be described as the science of developing algorithms that construct correspondences in between objects (mappings) based on data.

1.2.2 Learning-based Control

The combination of this field with decision or control theory gives rise to the (still broad) sub-concept of Machine Learning Control (MLC) (Sutton and Barto 2018; Duriez, Brunton, and Noack 2016; Bensoussan, Y. Li, et al. 2020; Meyn 2022).[2] In this work, this notion will also be referred to as *Learning-based Control*.[3] Two noteworthy expectations are set by this domain mixing computer science, statistics, and control.

Approximation Power

First, physical modelling comes with simplifying hypothesis allowing for the derivation of closed-form, analytical formula. However, these models are often inaccurate, especially in the presence of uncertainties or non-linearities. The approximation power of learning based models could overcome those limitations, leading to more accurate solutions.[4]

Discovering Control Strategies

Second, the discovery of some new control strategies achieving better performances can be expected. Such achievement has already been made in concurrent domain of application such as games (Silver et al. 2016), computational biology (Jumper et al. 2021), and nuclear fusion (Degrave et al. 2022).

1.3 Control of Dynamical Systems

1.3.1 Dynamical Systems

As the title of this manuscript suggests, the work presented here deals with the control of *Dynamical Systems* (Coudène 2013). This broad notion describes any

[2]The concept of *Machine Learning Control* is recent (*e.g.* the english Wikipedia page was created in April 2017 while the Reinforcement Learning page was created in 2002). As of today, no notable book chapter or review unifying the three concepts of Reinforcement Learning (RL), Learning-based Model Predictive Control (MPC) and Genetic Programming (P. Fleming and Purshouse 2002) for control has been published. The RL and MPC fields are described in the next few paragraphs of this introduction.

[3]Learning-based Control is the core topic of this manuscript. A more classical sub-topic is *Adaptive Control* (Åström and Wittenmark 1989) that considers iterative learning steps (estimation of a parametric model) to improve the control strategy.

[4]This is often at the price of interpretability. Thus, solutions combining physical inductive bias and statistical approximation are now developped (see for instance Karniadakis et al. 2021).

system endowed with an evolution law that characterises the system transition from one step to another. Thus, they encompass a large range of problems. However, the term dynamical systems should be understood in the sense of the dynamical system theory (Benoist and Paulin 2000; Viterbo 2009; Leroux 2019) which was originally inspired by the description of dynamics related to Physics and Mechanics. The controlled systems considered in this work go from simple theoretical models to Fluid Dynamics problems.

1.3.2 Two Approaches to Learning-Based Control

This work distinguishes two related approaches to Learning-Based Control: *Reinforcement Learning* and *Learning Based Model Predictive Control*.

Reinforcement Learning

Reinforcement Learning (RL) (Sutton and Barto 2018; Bertsekas and Tsitsiklis 1996; A. Agarwal, Jiang, and Kakade 2019) constructs data-driven control strategies based on a so-called *reinforcement signal* collected through the interaction with the environment (dynamical system).[5] This signal is a scalar value that quantifies the quality of the control input fed to the system. It can be seen as an instantaneous reward or a cost, depending on the problem.

Learning Based Model Predictive Control

Learning Based Model Predictive Control (MPC) (Aswani et al. 2013; Chua et al. 2018; Koller et al. 2019; Hewing et al. 2020) is a more classical approach. Basically, it combines a model learnt from dynamics data[6] with a *planning* algorithm. Planning can be defined as the process of selecting an optimal sequence of control inputs based on the model forward prediction and its associated reinforcement signal.[7] [8]

Several problems arise when considering the control of dynamical systems with Learning-Based Control. The next section presents the problems and research objectives of this PhD.

[5] Thus, approaches like reward free RL (Touati and Ollivier 2021) are here excluded from the RL definition.

[6] A major field of research in this domain is the identification of dynamical systems, termed *System Identification* (Ljung 1999).

[7] Thus, LB-MPC making use of a reinforcement signal is a form of Reinforcement Learning. However, the distinction is made here to underline the model learning and planning aspects of the algorithm.

[8] In the context of General Artificial Intelligence, the model-based approach corresponds to the concept of *World Models* (Ha and Schmidhuber 2018) and the planning aspect is referred to as *Imagination* (Z. Lin et al. 2020).

1.4 Problems and Research Objectives

1.4.1 Challenges in Learning-Based Control

Applications of Learning-Based control for Flow Control exhibit important challenges (Viquerat et al. 2022), such as:

- *Sample efficiency*: Flow control experiments are expensive and time consuming. Moreover, Learning-Based Control algorithms, such as Deep Reinforcement Learning (DRL), require a large amount of data (Plaat 2022).
- *Robustness*: Fluid dynamics are often non-stationary, chaotic, noisy or sensitive to parameters. The control strategy learnt must be robust to these perturbations.
- *Partial* observability: Sensors are noisy and limited. The control strategy should handle this partial information to achieve desired performances.
- *Delays*: As the environments are *partially observable (PO)*, the feedback signals may be delayed (post-control delay). In real-world applications, the control inference is not instantaneous (pre-control delay). The control strategy should handle these delays.

References to these issues are made throughout the document. The work presented in this manuscript aims at addressing some of these issues through a series of research projects.

1.4.2 Research Objectives

Therefore, the **research objective** of this PhD is to extend knowledge on these open-questions while contributing beyond the field of flow control. Each of the research projects presented in this manuscript addresses one of the challenges mentioned above with more or less emphasis.

As the introduction so far suggests, the work presented in this manuscript is multidisciplinary. Thus, a broad range of concepts and tools are used, borrowed from the fields of control theory, machine learning, statistics, and fluid dynamics.

1.5 Structure of the Document

The manuscript is organised as follows. This introduction is the first chapter of the document. Then, the document is divided into two main parts: the first part presents the theoretical foundations and unifying perspectives in Learning-Based Control. The second part presents the research projects conducted during the PhD. The content of each chapter is now briefly presented.

1.5.1 Theoretical Foundations and Unifying Perspectives

From Stochastic Control to Markov Decision Processes

The second chapter introduces the continuous time stochastic control concepts from which all the other notions discussed in the document (*e.g.* Markov Decision Processes (MDP), Bellman equation) can be inherited. It connects the continuous time control point of view to the discrete time framework, which is more common in the Learning-based control literature. In particular, the chapter introduces the notion of system sampling that allows for linking both continuous and discrete time worlds. The existence of this connection is useful in various applications and is a core tool used in Chapter 5. The end of this chapter deals with the numerical approximation of the continuous time control problem. The reading of this chapter is recommended before going through the part devoted to continuous time control (Chapter 7).

Learning-based Control with Discrete Decision Processes

The third chapter of this document presents the discrete time decision framework that is widespread in Learning-based control literature. Elements of learning theory are then presented in which key concepts such as learning task (loss) or generalisation error are introduced. The chapter ends with a description of Learning-based control and a presentation of the concrete dynamical systems used in the numerical experiments performed in this document. The reader only interested in the discrete time approach may solely start reading this document from Chapter 3 and ignore Chapter 7.

1.5.2 Methodological Advances in Learning Based Control

Evidence on the Regularisation Properties of Maximum Entropy Reinforcement Learning

This fourth chapter is the first of the second part. It deals with the robustness challenge of RL presenting empirical evidence on the robustness of Maximum Entropy Reinforcement Learning. It introduces the notion of complexity measure which is borrowed from statistical learning theory. Then, measures of robustness are introduced and the chapter ends with a presentation of the empirical results obtained. This work led to the publication of a paper on the proceedings of the 7^{th} International Conference in Optimization and Learning (OLA24) in Dubrovnik (Hosseinkhan Boucher, Semeraro, and Mathelin 2025).

Increasing Information for Model Predictive Control with Semi-Markov Decision Processes

Chapter 5 discusses the sample complexity approach in LB-MPC with the introduction of semi-Markov decision processes to extend a sample acquisition strategy that accelerates model learning. The chapter presents the information theoretic notion of expected information gain in the context of Gaussian process based model predictive control. Then, an extension of the approach to the semi-Markov decision process framework is presented to increase the information acquisition speed. Results on the sample efficiency of the approach are presented. This work led to a conference paper published in the proceedings of the 6th Annual Learning for Dynamics & Control Conference (L4DC24) (Hosseinkhan Boucher, Douka, et al. 2024).

Distributional Reinforcement Learning is Sample Efficient

Chapter 6 presents empirical evidence on the sample efficiency of Distributional RL. The chapter discusses the statistical approach to learn (estimate) distributions by introducing basic concepts of optimal transport theory and quantile regression. Next, the distributional perspective in RL is introduced and a state-of-the-art algorithm available in the literature is presented. The chapter ends with empirical results on the sample efficiency of the approach.

Neural Controlled Delay Differential Equations for Model Based Control

Chapter 7 presents a recent approach to model delayed dynamical systems with continuous-time neural delayed differential models. Continuous-time Reinforcement Learning is first presented, then two use cases of using delayed neural models for control are presented: delayed dynamical systems identification and partial observability handling. Finally, the neural model is presented and the chapter ends with empirical results on learning delayed or partially observable dynamical systems.

The last chapter concludes the document by summarising the contributions of the work presented and discusses future research directions.

I Theoretical Foundations and Unifying Perspectives

2 From Stochastic Control to Markov Decision Processes

The first part of this thesis begins with the present chapter, which introduces the theoretical foundations and a unifying perspective on control.

2.1 Introduction

This chapter introduces the field of stochastic control with the aim of outlining its connection with the learning-based control standard formalism. Modern frameworks for learning methods in control, such as Markov Decision Processes (MDP), have roots in the mathematical field of Control Theory. As it will be discussed in the next section, this relationship does not seem to be well-known in the Machine Learning (ML) community but tends to be increasingly considered in the literature.

2.1.1 Connecting Stochastic Control and Reinforcement Learning

Recently, the paper "A Tour of Reinforcement Learning: The View from Continuous Control" (Recht 2018) discussed the proximity between deterministic control theory and Reinforcement Learning, an interdisciplinary area of machine learning and optimal control. Two years later, a paper published in the Journal of Machine Learning Research (H. Wang, Zariphopoulou, and X. Y. Zhou 2020), entitled "Continuous Stochastic Control with Deep Reinforcement Learning", uses the connection between the *Stochastic Control* theory and MDP to propose an analysis of the maximum entropy principle in the context of Reinforcement Learning. Indeed, the central concept of exploration (closely tied to the policy entropy) is much more natural in a stochastic setting.

This chapter elaborates and extends the presentation given by H. Wang, Zariphopoulou, and X. Y. Zhou 2020 by showing how the standard Partially Observable Markov Decision Process (PO-MDP) formulation is obtained from Partially Observable Stochastic Differential Equations. This contribution could soften the gap between the two fields and provide a more solid theoretical foundation

for the learning-based control literature. Notably, carefully chosen references are provided to the reader for each step of the development. It is likely that the frontier between continuous and discrete time learning-based methods will become less marked in the future (see Croissant 2023 for a recent thesis at the intersection of the two fields and Leahy et al. 2022 for a recent work in this framework).

2.1.2 A General Framework for Diverse Applications

The choice of starting the presentation from a general, continuous-time point of view (Section 2.2) allows encompassing all the different concepts treated in this thesis. Hence, all cases presented in the subsequent chapters are particular cases of the framework introduced here.

Moreover, the generality of the presentation is broadened by the presence of a lag or *delay* affecting the system evolution. This is also motivated by the desire to unify the framework for the whole document. The question of delay will be addressed particularly in Chapter 7.

In addition, the question of sampling analogous (continuous-time) signals is also discussed at the end of the chapter (Section 2.3) since it can be related to the notion of Semi-Markov Decision Process (Sutton, Precup, and Singh 1999) that is treated in the work presented in Chapter 5.

The end of the chapter (Section 2.4) deals with the question of simulation and numerical approximation. The classical approximation scheme presented there bridges the gap between the continuous-time and discrete-time decision processes.

2.1.3 A Note on the Mathematical Development

It is important for the reader to be aware that this chapter does not attempt to provide a mathematically rigorous treatment of the highly abstract problem of Partially Observable Stochastic Control. An important list of heavy mathematical concepts proper to stochastic differentiability and infinite dimensionality are hidden from the reader but present in the mathematical references. Otherwise, the development would be very heavy and substantial work would be required to merge multiple concepts (*e.g.* the partial observability and the presence of delays). The reader is referred to W. Fleming and Rishel 1975; Øksendal 2010 for an introduction to the field of stochastic control.

The next section introduces the main concepts of stochastic control, in a general manner, before focusing on the specific cases.

2.2 Concepts of Stochastic Control

2.2.1 The notion of control

Many dynamical systems that can be observed or measured are subject to imperfectly known disturbances, possibly random. This randomness can be due to the environment, the system itself, or the measurement process. The term *nature* is often used to qualify the origin of the exogenous perturbations that affect the system. Alternatively, the system can be controlled by an endogenous[9] input called *control*.

Controlling a Dynamical System

Generally, the control U applied to the dynamics is carried out by some agent or controller. An important question concerning the design of control systems is the information available to the controller at each unit of time.
W. Fleming and Rishel 1975 mentions three main situations:

- The information available to the agent is determined *a priori*, before the beginning of the control procedure. Then, the control only depends on time, while the amount of available information is constant over time and equals the initial information. It is called "open loop" control.
- The system state X_t or history H_t is available to the controller at time t. Thus, the amount of available information then depends on time. For instance, when the history is available to the decision maker, the information increases[10]. This setting is termed as "closed-loop" or feedback control in the case of *complete observability*.
- Only a partial representation Y_t of the state is available. The quantity Y_t is often a set of system measurements that are called *observables*. Mathematically, the observables are a function of the state[11]. In this case, the

[9] The terms "exogenous" and "endogenous" are borrowed from economics (Blanchard and Johnson 1991; Acemoglu 2008).

[10] In probability theory, this idea is formalised with the concept of *filtration* (Jean-Francois Le Gall 2013) which is an increasing sequence of σ-algebras. A σ-algebra is a collection of events (subsets of the outcome space Ω). The richer the σ-algebra, the more information is available (there are more events). A σ-algebra can be generated by a random variable, in which case it represents all the possible events that can be discerned from the values described by this random variable. Notably, the σ-algebra generated by a constant random variable (poorly informative) is the certain event Ω and the impossible event $\emptyset$ (almost no information on the random experiment is conveyed by the constant random variable in the resulting σ-algebra).

[11] In practice, the function is injective since they often represent lower dimensional measurements which can have the same exact value, given two different underlying states.
The σ-algebra generated by a function of a random variable is always contained in the σ-algebra of this random variable. Consequently, the information is whether kept or lost but never created from transforming or extracting data. The two σ-algebras are the same if the function is bijective.

available information is always lower than when the full state is observable. This setting is termed as "closed-loop" or feedback control in the setting of *partial observability*.

In the next section, the problem of control in infinite dimension is described and how it can be framed as a Partially Observable Markov Decision Process.

Preliminary References on Stochastic Control

The reader is referred to Trélat 2023 for a more rigorous treatment of control of differential equations, El Karoui, Du Huu, and Jeanblanc-Picqué 1987; H. Wang, Zariphopoulou, and X. Y. Zhou 2020 to find details on continuous stochastic control and to Pan et al. 2018; Bucci et al. 2019 for recent applications with Reinforcement Learning. Note the very challenging notions of existence, uniqueness, controllability and observability of the solutions are omitted here.

The development of this chapter is inspired by multiple references in the field of Stochastic Differential Equations and Stochastic Control. In particular, this tutorial borrows concepts from Relaxed Stochastic Control for fully observable systems (W. H. Fleming and Nisio 1984; El Karoui, Du Huu, and Jeanblanc-Picqué 1987; Redjil and Choutri 2017) and Partially Observable Stochastic Control (N. Ahmed 2007; N. U. Ahmed and Xiang 1992). SDE in infinite dimension is treated in Gatarek and Goldys 1994; Gawarecki and Mandrekar 2015. An article on control of infinite-dimensional SDE is Bensoussan and Viot 1975. The topic of delayed SDE is treated in Küchler and Mensch 1992; S. Mohammed 1984; S.-E. A. Mohammed 1998; Buckwar 2000 and the control of delayed SDE in Elsanosi, Øksendal, and Sulem 2000. Other references for the notions introduced below are directly introduced in the text.

2.2.2 General Continuous-time Formulation

A general formulation of the state and observation dynamics covering most of the recent challenges in learning-based dynamical systems modelling is introduced now.

General Continuous-time Stochastic Dynamics

Let $X = (X_t)_{t\in I}$ denote the state process, subject to the control process $U = (U_t)_{t\in I}$ and $Y = (Y_t)_{t\in I}$ the observation process defined on $I \subset \overline{\mathbb{R}}_+ = \mathbb{R}_+ \cup \{+\infty\}$. Thus, $I = [t_0, T]$ for some initial time $t_0 \in \mathbb{R}_+$ and final time $T \in \mathbb{R}_+ \cup \{+\infty\}$.

Those objects are stochastic processes that are defined on a probability space $(\Omega, \mathscr{F}, \mathbb{P})$. The state and control processes evolve in some Banach spaces $\mathcal{X}$ and $\mathcal{U}$ while the observation process evolves in a finite dimensional Euclidean space $\mathcal{Y} \subset \mathbb{R}^{d_Y}$ with $d_Y \in \mathbb{N}_+$.

Now, a central object of this presentation is presented that encompasses a very broad class of dynamical systems from physics to finance. The general dynamics studied here are continuous-time stochastic process.

Definition 2.2.1 (General Dynamics - Differential)**.** *The state process X is the solution of the stochastic differential equation (SDE)*

$$\begin{cases} dX_t = f\left(X_t, X_{t-\tau_X}, U_t\right)\, dt + \epsilon_X(X_t, U_t)\, dW_t^1 \\ X_{[t_0-\tau_X,t_0]} \sim \mathbb{P}_{X_{[t_0-\tau_X,t_0]}} \end{cases} \tag{2.1}$$

where $f : \mathcal{X} \times \mathcal{X} \times \mathcal{U} \to \mathcal{X}$ is the dynamics operator*, $\epsilon_X : \mathcal{X} \times \mathcal{U} \to \mathcal{X}$ is the* state noise operator*, W^1 is a standard Brownian motion and $\tau_X \in \mathbb{R}_+$ is the state delay such that $t - \tau_X \in I$ for all $t \in I$.*

The initial condition is given by the distribution $\mathbb{P}_{X_{[t_0-\tau_X,t_0]}}$. The dynamics defined by (2.1) *being a delay-differential equation, the initial value $X_{[t_0-\tau_X,t_0]}$ is not a point in a vector space but a history process over the interval $[t_0 - \tau_X, t_0]$. The* dynamics operator *f acts between Banach spaces.*
The state noise operator*, also known as diffusion coefficient, ϵ_X is a function of the state and the control process which scales the Brownian motion W^1.*

The observation process Y is driven by the SDE[12]

$$\begin{cases} dY_t = g\left(X_t, X_{t-\tau_Y}, U_t\right)\, dt + \epsilon_Y(X_t, U_t)\, dW_t^2 \\ Y_{t_0} \sim \delta_{g_0(X_{t_0})} \end{cases} \tag{2.2}$$

where the observation operator *$g : \mathcal{X} \times \mathcal{X} \times \mathcal{U} \to \mathcal{Y}$ is a function acting from the state, delayed-state and control spaces to the observation space, the observation noise operator $\epsilon_Y : \mathcal{X} \times \mathcal{U} \to \mathcal{Y}$ is a function of the state and control processes which scales the Brownian motion W^2. The observation delay $\tau_Y \in \mathbb{R}_+$ is such that $t - \tau_Y \in I$ for all $t \in I$. The initial observation is obtained from the state through the mapping $g_0 : \mathcal{X} \to \mathcal{Y}$. The distributions of $X_{[t_0-\tau_X,t_0]}$, W^1 and W^2 are supposed to be independent.*
The set of all control processes such that (2.1) *and* (2.2) *are well-posed is denoted $\mathscr{A}_{\mathcal{U}}$ (admissible control space).*

The dynamics (2.1) and (2.2) are presented in differential form but can also be written in integral form.

Definition 2.2.2 (General Dynamics - Integral)**.** *Definition 2.2.1 can be rewritten in integral form as*

$$X_t = X_{t_0} + \int_{t_0}^{t} f\left(X_s, X_{s-\tau_X}, U_s\right)\, ds + \int_{t_0}^{t} \epsilon_X(X_s, U_s)\, dW_s^1 \tag{2.3}$$

[12]The dynamics of the observation process is inspired by W. H. Fleming and Nisio 1984. The delayed formulation is treated, for instance, in Buckwar 2000.

for the state process and

$$Y_t = g_0\left(X_{t_0}\right) + \int_{t_0}^{t} g\left(X_s, X_{s-\tau_Y}, U_s\right)\, ds + \int_{t_0}^{t} \epsilon_Y(X_s, U_s)\, dW_s^2 \tag{2.4}$$

for the observation dynamics.
The right-most integrals involving Brownian motions are stochastic integrals*. It can be interpreted as time-correlation between the diffusion coefficients ϵ_X or ϵ_Y and the Brownian motion variations dW_t^1 or dW_t^2, respectively. Remark 2.2.2 gives more details about the interpretation of the stochastic integral.*

A series of remarks are necessary to disentangle the framework presented here.

Remark 2.2.1 (Motivation)**.** *The possibility to consider the system state X_t as a (possibly random) function (e.g. $X_t(z)$, $z \in \mathbb{R}^3$) which may be a solution of a partial differential equation (PDE), is the reason for using general infinite-dimensional spaces in this presentation. This choice is not common in the learning-based control literature, see Pan et al. 2018; Bucci et al. 2019; Peitz, Stenner, et al. 2024 for a work on this topic in Reinforcement Learning.*

Now, the meaning of the stochastic integral terms in the dynamics is explained.

Remark 2.2.2 (Interpretation of the stochastic integral)**.** *For simplicity, it is supposed that $\mathcal{X} = \mathbb{R}$ such that the Brownian motion $W : I \times \Omega \to \mathbb{R}$ takes values in the real line.*

For every sequence $t_0 = t_{k_0}^n < t_{k_1}^n < \ldots < t_{k_n}^n = t$ of partitions of the interval $[t_0, t] \subset I$ such that $\max_{i=1}^n \delta_{t_{k_i}}^n \to 0$ as $n \to \infty$, where $\delta_{t_{k_i}}^n = |t_{k_i}^n - t_{k_{i-1}}^n|$,

$$\lim_{n\to\infty} \sum_{i=0}^{k_n - 1} \epsilon_X(X_{t_{k_i}^n}, U_{t_{k_i}^n})(W_{t_{k_{i+1}}^n} - W_{t_{k_i}^n}) = \int_{t_0}^{t} \epsilon_X(X_s, U_s)\, dW_s \tag{2.5}$$

in probability[13]*.*

Let the discrete-time increments of the Brownian motion *be $\delta W_{t_{k_i}^n} = W_{t_{k_{i+1}}^n} - W_{t_{k_i}^n}$. By definition of the Brownian motion,*

$$\delta W_{t_{k_i}^n} = W_{t_{k_{i+1}}^n} - W_{t_{k_i}^n} \sim \mathcal{N}(0, \delta_{t_{k_i}}^n) \tag{2.6}$$

and informally, $\delta W_{t_{k_i}^n} \to dW_t$ when $n \to \infty$. Consequently, the stochastic integration can be seen as standard integration with ***randomly distributed infinitesimal***

[13] See for instance, Jean-François Le Gall 2006 for a definition of convergence in probability. In common terms, it means that for any treshold, the probability that the absolute error, between the sequence and its limit, is above the treshold goes to zero.

time increments. *Moreover, since the partial sum of Eq.* (2.5) *approximates the stochastic integral, if the scaling parameter is constant e.g.* $\epsilon_X = 1$

$$\sum_{i=0}^{k_n-1} \delta W_{t^n_{k_i}} \simeq \int_{t_0}^{t} dW_s = W_t \tag{2.7}$$

where the last equality can be obtained by noting that the partial sum is telescopic.

The above approximation gathers some key properties.
First, the integral approximates the sum of Gaussian random variables whose scale (variance) is the time increment. Second, at time t*, the Brownian motion* W_t *is equal to the sum of those random increments. Third, by definition of the Brownian motion, the increments are independent and normally distributed: this motion can be interpreted as the limit of a discrete random walk with normally distributed increments.*

More information about this approximation can be found in Jean-Francois Le Gall 2013, Proposition 5.9 for a multidimensional Brownian motion.

The stochastic integral can be defined with respect to other stochastic processes than the Brownian motion, see Remark 2.2.6. The careful reader may notice the similitude between the stochastic integral and the Riemann-Stieltjes integral.

The following case is enlightening despite being not considered in the applications in the work presented here.

Remark 2.2.3 (Functional Brownian motion)**.** *Consider the case where* $\mathcal{X}$ *is an infinite-dimensional function space. Then a Brownian motion on* $\mathcal{X}$ *defines a trajectory in a functional space. It can be thought as a continuous sequence of random spatial functions.*

For some outcome $\omega \in \Omega$ *of the random experiment, the value of the Brownian motion* $W_t(\omega)$ *at time* t*, is a function of the space (i.e.* $W_t(\omega)(z)$ *where* z *is the spatial coordinate).*

In this general case (Guiseppe Da Prato and Zabczyk 1992), the increments of the Brownian motion define a Gaussian process.

$$\delta W_{t^n_{k_i}} = W_{t^n_{k_{i+1}}} - W_{t^n_{k_i}} \sim \mathcal{GP}(0,\, \delta^n_{t_{k_i}} 1_{z=z'}) \tag{2.8}$$

where the covariance operator $(z, z') \mapsto \delta^n_{t_{k_i}} 1_{z=z'}$ *generalises the finite dimensional scaled identity matrix* $\delta^n_{t_{k_i}} I_{d_X}$*. Hence, when the state* X_t *is a function, the dynamics are perturbed by a strong (spatial) Gaussian white noise*[14]*.*

Some important cases that are commonly encountered in the literature are now presented.

[14] Here, a strong Gaussian white noise is a stochastic process such that all coordinates are independant gaussian.

Particular Cases

A very standard setting is when the dynamics are Markovian.

Remark 2.2.4 (Markovian Dynamics)**.** *The state and the observation, dynamics* (2.1)-(2.2) *are said to be* Markovian *if their respective operators f and g, only depend on the current state X_t, and the control process U_t where the control process depends only on the instantaneous state X_t. In this case, the state process X_t, respectively the observation process Y_t, is a Markov process.*

If the dynamics are deterministic, to be Markovian means that time-derivative of the state process is a function of the instantaneous state and control only.

A Markov process that is a solution of a stochastic differential equation is called a diffusion process*.*

The initial condition can be fixed to a deterministic value.

Remark 2.2.5 (Fixed Initial Condition)**.** *Let $(x_t)_{t\in[t_0-\tau_X,t_0]} \in \mathcal{X}^{[t_0-\tau_X,t_0]}$ be an arbitrary initial history function. It is possible to fix the initial condition $X_{[t_0-\tau_X,t_0]} = x_{[t_0-\tau_X,t_0]}$ by setting $\mathbb{P}_{X_{[t_0-\tau_X,t_0]}} = \delta_{\{x_{[t_0-\tau_X,t_0]}\}}$ where $\delta_{\{x_{[t_0-\tau_X,t_0]}\}}$ is the Dirac measure at $\{x_{[t_0-\tau_X,t_0]}\}$. This way, the initial condition is deterministic (degenerated) and $X_{[t_0-\tau_X,t_0]} = x_{[t_0-\tau_X,t_0]}$ with probability one.*[15] *In the Markovian case, for $x \in \mathcal{X}$, $\mathbb{P}_{X_{t_0}} = \delta_{\{x\}}$.*

A Gaussian noise is a common choice for the state noise, but other disturbance distributions can be considered.

Remark 2.2.6 (General Noise)**.** *The brownian motion is analogous to Gaussian noise in the discrete-time setting. A larger class of continuous-time noise processes can be considered, e.g. see S.-E. A. Mohammed 1998 considers a particular type of process called semi-martingale noise (Jean-Francois Le Gall 2013; Revuz and Yor 1999), but this notion is way beyond the scope of this work.*

Remark 2.2.7. *The challenges in Learning-based Control introduced in Section 1.4.1 are addressed by the general framework presented here.*

- *The robustness aspect will be covered by the stochastic nature of the dynamics characterised by the Brownian motions W^1 and W^2 in Eq.* (2.1) *and* (2.2)*.*
- *The partial observability is addressed by the observation process Y (Eq.* (2.2)*).*
- *The delayed state $X_{t-\tau_X}$ and observation dynamics $X_{t-\tau_Y}$ should provide a wide panel of interesting configurations that feature the challenges of controlling systems with lagged information.*

[15] In probability theory, an event that occurs with probability one is said to happen *almost surely (a.s.)*

Observability

Observability is now regarded more rigorously. Throughout the document, the notion of observability will refer to the following definition

Definition 2.2.3 (Terminology on Observability)**.** *If $\mathcal{Y} = \mathcal{X}$ and $g_0 = Id$, then the system is said to be* fully observable*. Otherwise, the system is* partially observable*.*

Examples of Dynamics

A few examples are now given to illustrate the general framework presented above.

Example 2.2.1 (Deterministic Dynamics and PDE)**.** *Consider the subclass of deterministic stochastic processes $(x_t)_{t\in I}$, i.e. $(X_t(\omega))_{t\in I} = (x_t)_{t\in I}$ for any $\omega \in \Omega$. Suppose that $\mathcal{X} = L^2(\mathbb{R}^{d_z})$ with $d_z \in \mathbb{N}^*$, $\epsilon_X = 0$ and $\epsilon_Y = 0$. Then, the state dynamics given by* (2.1) *becomes*

$$\begin{cases} \partial_t x_t(z) = f\left(x_t(z), x_{t-\tau_X}(z), u_t(z)\right) \\ x_{[t_0-\tau_X,t_0]}(z) = \xi(z) \end{cases} \tag{2.9}$$

and

$$\begin{cases} \partial_t y_t(z) = g\left(x_t(z), x_{t-\tau_Y}(z), u_t(z)\right) \\ y_{t_0}(z) = g_0\left(x_{t_0}(z)\right) \end{cases} \tag{2.10}$$

where $\xi : [t_0 - \tau_X, t_0] \to L^2(\mathbb{R}^{d_z})$ is an initial history function. In particular, any delayed and controlled partially observable PDE can be represented when f is chosen as a partial derivative operator.

The well-posedness and existence of the solution of (2.9)-(2.10) *is a challenging problem in the theory of PDEs, but it will not be addressed here. To go further, the reader may be interested in the books of Cartan 1971; Evans 1998; Zuily 2002 about PDEs and Lions 1971; Bensoussan 1993; Bardi and Capuzzo-Dolcetta 2008; Trélat 2023 for controlled PDEs.*

Example 2.2.2 (Delayed Differential Equation)**.** *Consider the subclass of deterministic stochastic processes $(x_t)_{t\in I}$, i.e. $(X_t(\omega))_{t\in I} = (x_t)_{t\in I}$ for any $\omega \in \Omega$. Suppose that $\mathcal{X} = \mathbb{R}^{d_X}$ with $d_X \in \mathbb{N}^*$, $\epsilon_X = 0$ and $\epsilon_Y = 0$. Then, the state dynamics given by* (2.1) *becomes*

$$\begin{cases} \partial_t x_t = f\left(x_t, x_{t-\tau_X}, u_t\right) \\ x_{[t_0-\tau_X,t_0]} = \xi \end{cases} \tag{2.11}$$

and

$$\begin{cases} \partial_t y_t = g\left(x_t, x_{t-\tau_Y}, u_t\right) \\ y_{t_0} = g_0\left(x_{t_0}\right) \end{cases} \tag{2.12}$$

where $\xi : [t_0 - \tau_X, t_0] \to \mathbb{R}^{d_X}$ *is the initial condition.*

In particular, any controlled partially observable Delayed Differential Equation (DDE) can be represented. An ordinary differential equation (ODE) is a particular case of DDE when the delay term is ignored.

In the control-free setting, the existence and uniqueness of the solution of (2.11)-(2.12) *is guaranteed when the dynamics operator* f *is continuous and Lipschitz. The interested reader can find an important development of the DDE theory in the reference manuscripts of Kuang 1993, Smith 2010 and Hale 1971.*

Example 2.2.3 (Stochastic Navier-Stokes)**.** *Some fundamental dynamics in fluid dynamics are given by the Navier-Stokes equation. Let* $\mathcal{X} = L^2(I \times \mathbb{R}^2; \mathbb{R}^2)$*. In the stochastic setting, it reads*

$$dX_t(z_1, z_2) = \nu^{NS}(\Delta X_t(z_1, z_2) - \langle X_t(z_1, z_2), \nabla \rangle X_t(z_1, z_2) - \nabla p_t(z_1, z_2))\, dt + dW_t(z_1, z_2) \tag{2.13}$$

where ∇ *is the gradient operator* $(\partial_{z_1}, \partial_{z_2})$*,* Δ *is the Laplace operator* $\partial^2_{z_1} + \partial^2_{z_2}$*, and* $\langle X_t, \nabla \rangle$ *stands for the differential operator* $\partial_{z_1} X^1_t + \partial_{z_2} X^2_t$*, with* $X_t = (X^1_t, X^2_t)$ *the velocity field and* p_t *the pressure field at time* t*. The term* $\nu^{NS} \in \mathbb{R}^*_+$ *is the kinematic velocity.*[16] *Boundary or limit conditions can be added to Eq. 2.13, but they are omitted here for simplicity.*

For a rigorous treatment of this example, the reader may check Bensoussan and Temam 1973; E 2000; Giuseppe Da Prato and Debussche 2000; Kuksin and Shirikyan 2012; Fabbri, Gozzi, and Swiech 2017.

The question of control is now addressed by defining the associated problems the decision-maker is confronted with.

2.2.3 Control Objective

The definitions in this part extend the classical control theory presented by Trélat 2005; Trélat 2023 to the stochastic setting.

Control Problem

Suppose that $I = [t_0, T]$. Given a region of the state space $E_{\mathcal{X}} \subset \mathcal{X}$, the *control or controllability problem* is to find a control process U such that the controlled process (2.1) satisfies

$$X_{[t_0 - \tau_X, t_0]} = \xi \quad \text{and} \quad X_T \in E_{\mathcal{X}} \tag{2.14}$$

almost surely (*i.e.* with probability one) where $\xi \sim \mathbb{P}_{X_{[t_0 - \tau_X, t_0]}}$ is the initial random condition on the history (a stochastic process on $[t_0 - \tau_X, t_0]$).

[16] The superscript *NS* stands for Navier-Stokes.

Optimal Control Problem

The *optimal* control problem is a control problem as defined above, but with the constraint that the control process U *minimises a cost function*.

Let define the random total cost as the accumulated cost over the time interval $[t, T]$

$$Z\left(t, \mathbb{P}_{X_{[t-\tau_X,t]}}, U\right) = \int_t^T e^{-\gamma(s-t_0)} c\left(X_s, U_s\right) ds \tag{2.15}$$

for any $t \in I$ with $\gamma \in [0, +\infty[$ a discount factor and $c : \mathcal{X} \times \mathcal{U} \to \mathbb{R}$ an instantaneous cost function. This quantity is a random variable. Its value depends on the random initial condition, noise, and random control process. The random initial cost from initial time t_0 is denoted $Z(\mathbb{P}_{X_{[t-\tau_X,t]}}, U) \coloneqq Z(t_0, \mathbb{P}_{X_{[t_0-\tau_X,t_0]}}, U)$

The typical scalar-valued objective functional of the following form is considered

$$J\left(t, \mathbb{P}_{X_{[t-\tau_X,t]}}, U\right) = \mathbb{E}\left[\int_t^T e^{-\gamma(s-t_0)} c\left(X_s, U_s\right) ds\right] = \mathbb{E}\left[Z\left(t, \mathbb{P}_{X_{[t-\tau_X,t]}}, U\right)\right] \tag{2.16}$$

This quantity is a real number which averages the random total cost. If $t = t_0$, the objective functional $J(t_0, \mathbb{P}_{X_{[t_0-\tau_X,t_0]}}, U)$ is denoted $J(\mathbb{P}_{X_{[t_0-\tau_X,t_0]}}, U)$.

The optimal objective functional is then defined as

$$J^*\left(t, \mathbb{P}_{X_{[t-\tau_X,t]}}\right) = \inf_{U \in \mathscr{A}_{\mathcal{U}}} J\left(t, \mathbb{P}_{X_{[t-\tau_X,t]}}, U\right) \tag{2.17}$$

for any $t \in I$. The control process that minimises the cost is denoted $U^* \in \mathscr{A}_{\mathcal{U}}$. Hence, the optimal control problem solves

$$J^*\left(\mathbb{P}_{X_{[t_0-\tau_X,t_0]}}\right) = J^*\left(t_0, \mathbb{P}_{X_{[t_0-\tau_X,t_0]}}\right) = \inf_{U \in \mathscr{A}_{\mathcal{U}}} \mathbb{E}\left[\int_{t_0}^T e^{-\gamma(s-t_0)} c\left(X_s, U_s\right) ds\right] \tag{2.18}$$

In the case the initial condition is fixed (see Remark 2.2.5), the optimal objective functional is given by

$$J^*\left(x_{[t_0-\tau_X,t_0]}\right) = \inf_{U \in \mathscr{A}_{\mathcal{U}}} \mathbb{E}\left[\int_{t_0}^T e^{-\gamma(s-t_0)} c\left(X_s, U_s\right) ds \mid X_{[t_0-\tau_X,t_0]} = x_{[t_0-\tau_X,t_0]}\right] \tag{2.19}$$

Similarly, for the Markovian case, the optimal objective functional is given by

$$J^*\left(x_{t_0}\right) = \inf_{U \in \mathscr{A}_{\mathcal{U}}} \mathbb{E}\left[\int_{t_0}^T e^{-\gamma(s-t_0)} c\left(X_s, U_s\right) ds \mid X_{t_0} = x_{t_0}\right] \tag{2.20}$$

Note that the problem of the existence of an optimal partially observable control has no solution in a general way. The following remarks are intended to provide further insight into the optimal control problem.

Remark 2.2.8. *The control problem (stricto-sensu) can also be defined when $T = +\infty$. In this case, the terminal condition at time T is replaced by the condition that the state remains in $E_{\mathcal{X}}$ forever when t is sufficiently large. Moreover, the condition $\gamma < 0$ is necessary to ensure the convergence of the integral.*

Remark 2.2.9. *Note that if $E_{\mathcal{X}} = \mathcal{X}$, then the control problem is trivial, i.e. for any control process U, $X_T \in \mathcal{X}$.*

In that specific case, the optimal control problem reduces to the minimisation of the cost function (2.18)*.*

Remark 2.2.10. *Traditionally, the objective functional in finite time adds a "terminal cost" (Trélat 2023) term which is omitted here for simplicity. The terminal cost is a function of the final state X_T. This kind of cost is well-suited for task or goal-oriented problems.*

Remark 2.2.11. *If $\gamma = \log(\overline{\gamma})$ with $\overline{\gamma} \in]0, 1]$, then*

$$J^*(t, U) = \inf_{U \in \mathscr{A}_{\mathcal{U}}} \mathbb{E}\left[\int_t^T \overline{\gamma}^s c(X_s, U_s)\, ds\right] \tag{2.21}$$

which is a formulation that is often employed in the discrete case.

Remark 2.2.12 (Constrained minimisation)**.** *The optimal control problem can be seen as a constrained minimisation problem where the quantity to optimise is the objective functional* (2.18) *and the constraints are the dynamics* (2.1)-(2.2) *with initial and terminal conditions specified by* (2.14)*.*

Thus, it makes sense to consider calculus of variations (Bourguignon 2007) and constrained optimisation theory in infinite-dimensional spaces (since the control space is infinite-dimensional (Peypouquet 2015)) to solve the optimal control problem in specific cases.

An important type of cost function that is used in all the work in this thesis is now presented.

Example 2.2.4 (Quadratic Cost)**.** *The quadratic cost function is a common choice in control theory. Given two definite positive operators $A_{\mathcal{X}}$ and $A_{\mathcal{U}}$, it is defined as*

$$c(x, u) = \|x\|^2_{L^2, A_{\mathcal{X}}} + \|u\|^2_{L^2, A_{\mathcal{U}}} \tag{2.22}$$

where $\|x\|_{L^2,A} \coloneqq \langle x, Ax\rangle_{L^2}$ for any positive definite operator A and any vector x in the corresponding L^2 space. The quadratic cost is used in the Linear Quadratic Regulator (LQR) problem Trélat 2005.

Closed-Loop Control

Now the feedback control is formally defined.

Definition 2.2.4 (History-dependent control)**.** *The process $(U_t)_{t\in I}$ is said to be a* history-dependent control *if for any $t \in I$, it is a function of the past trajectory*[17] *$((Y_s)_{t_0\leq s\leq t}, (U_s)_{t_0\leq s<t})$ of the observation-control process.*
Formally[18]*, for any $t \in I$, there exists a function $u_t : \mathcal{Y}^{[t_0,t]} \times \mathcal{U}^{[t_0,t[} \to \mathcal{U}$ such that*

$$U_t = u_t\left((Y_s)_{t_0\leq s\leq t}, (U_s)_{t_0\leq s<t}\right) \tag{2.23}$$

Definition 2.2.5 (Feedback control)**.** *A feedback control process $(U_t)_{t\in I}$ is a history dependent control process where the feedback loop function u is solely a function of the instantaneous observation Y_t.*
Formally, for any $t \in I$, there exists a function $u_t : \mathcal{Y} \to \mathcal{U}$ such that

$$U_t = u_t(Y_t) \tag{2.24}$$

This kind of control is sometimes called Markovian control*.*

Remark 2.2.13. *In the field of automation, a history-dependent control is often labelled* closed-loop control *while a control that is independent of the past observations and decisions is called* open-loop control *(W. Fleming and Rishel 1975; Åström and Murray 2021).*

2.2.4 Policy

So far, a stochastic process $U = (U_t)_{t\in I}$ has been considered to control the system. Consider an outcome $\omega \in \Omega$ of a control experiment where, for instance, the random system evolution is observed or simulated.
The resulting controlled trajectory $X(\omega) = (X_t(\omega))_{t\in I}$ depends on the fixed control trajectory $U(\omega) = (U_t(\omega))_{t\in I}$. This means that once a control process U is chosen, if a controlled trajectory $(X_t(\omega))_{t\in I}$ is observed as the outcome of a random experiment, then its associated control trajectory $(U_t(\omega))_{t\in I}$ is fixed and always the same.

An example may help to grasp this concept of stochasticity[19]. Suppose the outcome space $\Omega = \{10°, 30°\}$ gives the initial temperature in a room and $U(\omega) =$ *turn on the air conditioning at 20°C* for $\omega = 30°$ and *turn on the heating at 20°C* for $\omega = 10°$.
Let $X_t(\omega)$ be the temperature in the room at time t. Then, for each scenario

[17] In probabilistic terms, this is equivalent to say the control is measurable w.r.t. the σ-algebra generated by the past observation-control process (the measurability condition is equivalent to the existence of u). Thus, the information available at time t to the controller is governed by the past trajectory.

[18] Mathematically, the question of the existence of such control is not trivial at all, especially in the stochastic setting where the history dependence, and by extension the concept of information availability, is defined in terms of measurability with respect to filtration generated by the process (El Karoui, Nguyen, and Jeanblanc-Picqué 1988).

[19] This example can be written much more formally without too much difficulty.

(initial room temperature) $\omega = 10°$ and $\omega = 30°$, the action performed is always the same (turning up or down at a prescribed level of temperature)[20].
This way, the impact of **any other decision variant** would be **unknown**. For instance, what would happen to the temperature $X_t(\omega)$ if the air conditioning was turned at 40°C instead of 20°C?
Thus, the resulting information about the environment is rather limited. Choosing a U that covers a larger, possibly randomised spectrum of decisions would help to extract more knowledge on the experiment and the system dynamics X. Ideally, for a given scenario (fixed by the outcome ω), a range of control processes is considered.

This basic example leads to the broader concept of *exploration*.

Indeed, given this fixed state trajectory $X(\omega)$, one can be interested in the behaviour of the system under different control trajectories than $U(\omega)$. This **notion is of extreme importance in the modern learning-based control field** and is called **exploration** (Ladosz et al. 2022). The exploration is a vast topic in learning-based control and notably in Reinforcement Learning. Chapter 5 will address this topic using a tool from Information Theory to increase the information extracted from the system.

From this perspective, the idea is to enlarge the control space $\mathcal{U}$ and consider that the control process U takes values in a space Π of probability measures on $\mathcal{U}$ also known as *generalised control space*. This leads to the concept of *policy*.

Definition 2.2.6 (Policy)**.** *Let Π be the space of probability measures on $\mathcal{U}$. A* policy *π, or policy process is a Π-valued stochastic process, i.e. a process $\pi = (\pi_t)_{t\in I}$ such that for any $t \in I$, $\pi_t \in \Pi$.*

Definition 2.2.7 (Stationary Policy)**.** *A policy is said to be* stationary *if there exists a probability measure $\tilde{\pi} \in \Pi$ such that for any $t \in I$, $\pi_t = \tilde{\pi} \in \Pi$.*
In this case, the policy is a fixed probability measure on $\mathcal{U}$ (by identification) and is denoted π by abuse of notation.

Remark 2.2.14 (Degenerated Policy)**.** *Let $U = (U_t)_{t\in I}$ be a control process. Suppose that $\pi = \left(\delta_{\{U_t\}}\right)_{t\in I}$ where $\delta_{\{U_t\}}$ is the Dirac measure at $\{U_t\}$. Then, the policy π is termed* degenerated *and is analogous to the control process U.*
Indeed, sampling from π deterministically returns U.

Once the policy concept is introduced, the control problem can be reformulated to incorporate the uncertainty in the control process. Originally, this has been coined as *Relaxed Stochastic Control* (El Karoui, Du Huu, and Jeanblanc-Picqué 1987).

[20]The degenerate case, where Ω contains only one outcome (scenario) implies that the dynamics are deterministic and the control is a function of the time.

2.2.5 Relaxed control

The relaxed or *exploratory* version of the dynamics (H. Wang, Zariphopoulou, and X. Y. Zhou 2020) is given by the following equation.

Definition 2.2.8 (Relaxed Dynamics - Differential)**.** *Considering the context of Definition 2.2.1, and some arbitrary policy $\pi = (\pi_t)_{t\in I}$, the relaxed dynamics are given for the state process by*

$$dX_t = f\left(X_t, X_{t-\tau_X}, \pi_t\right)\, dt + \epsilon_X(X_t, \pi_t)\, dW_t^1 \tag{2.25}$$

and for the observation process by

$$dY_t = g\left(X_t, X_{t-\tau_Y}, \pi_t\right)\, dt + \epsilon_Y(X_t, \pi_t)\, dW_t^2 \tag{2.26}$$

where

$$f\left(X_t, X_{t-\tau_X}, \pi_t\right) \coloneqq \int_{\mathcal{U}} f\left(X_t, X_{t-\tau_X}, u\right)\, \pi_t(du) \tag{2.27}$$

while the observation operator g, the noise terms ϵ_X and ϵ_Y are defined similarly. The set of all policies such that the relaxed dynamics (2.25)-(2.26) *is well-defined is denoted $\mathscr{A}_{\Pi}$.*

The notion of policy generalises the stochastic optimal control problem presented before. The next remark highlights this point.

Remark 2.2.15. *The remark 2.2.14 shows that the control process is a particular case of policy. In other words, the set of control processes is a subset of the set of policies.*

$$\mathscr{A}_{\mathcal{U}} \subset \mathscr{A}_{\Pi} \tag{2.28}$$

This is a standard observation in the stochastic control literature. It allows mathematicians to obtain optimality results more easily by reformulating (relaxing) the optimisation problem.[21]

A corresponding relaxed optimal control problem can be defined exactly as the classical optimal control problem.

Relaxed Optimal Control Problem

The *relaxed optimal control problem* is the relaxed dynamics counterpart of the optimal control problem where the policy plays the role of the control process.

Let the random total cost as the accumulated cost over the time interval $[t, I]$ be defined as

$$Z\left(t, \mathbb{P}_{X_{[t-\tau_X, t]}}, \pi\right) = \int_t^T e^{-\gamma(s-t_0)} c\left(X_s, \pi_s\right) ds \tag{2.29}$$

[21] In mathematics, especially in topology, this procedure is called *compactification* (El Karoui, Du Huu, and Jeanblanc-Picqué 1987; Shalizi 2007). Compact sets are extremely useful to find optima.

At time $t = t_0$, the random initial cost will be denoted by $Z(\mathbb{P}_{X_{[t-\tau_X,t]}}, \pi) := Z(t_0, \mathbb{P}_{X_{[t_0-\tau_X,t_0]}}, \pi)$.

The object is to find a policy π such that the following objective is minimised

$$J\left(t, \mathbb{P}_{X_{[t-\tau_X,t]}}, \pi\right) = \mathbb{E}\left[\int_t^T e^{-\gamma(s-t_0)} c\left(X_s, \pi_s\right) ds\right] = \mathbb{E}\left[Z\left(t, \mathbb{P}_{X_{[t-\tau_X,t]}}, \pi\right)\right] \tag{2.30}$$

where

$$c\left(x, \pi_s\right) := \int_{\mathcal{U}} c\left(x, u\right) \pi_s(du) \tag{2.31}$$

for any $x \in \mathcal{X}$ and π_s a probability measure on $\mathcal{U}$.

Similarly, the relaxed optimal objective functional is defined as

$$J^*\left(t, \mathbb{P}_{X_{[t-\tau_X,t]}}\right) = \inf_{\pi \in \mathscr{A}_\Pi} J\left(t, \mathbb{P}_{X_{[t-\tau_X,t]}}, \pi\right) \tag{2.32}$$

The optimal policy that minimises the cost is denoted $\pi^* \in \mathscr{A}_\Pi$. The relaxed optimal control problem is then to solve

$$J^*\left(\mathbb{P}_{X_{[t_0-\tau_X,t_0]}}\right) = J^*\left(t_0, \mathbb{P}_{X_{[t_0-\tau_X,t_0]}}\right) = \inf_{\pi \in \mathscr{A}_\Pi} \mathbb{E}\left[\int_{t_0}^T e^{-\gamma(s-t_0)} c\left(X_s, \pi_s\right) ds\right] \tag{2.33}$$

As for the control problem, when the initial distribution is degenerated, the optimal objective functional is given by

$$J^*\left(x_{[t_0-\tau_X,t_0]}\right) = \inf_{\pi \in \mathscr{A}_\Pi} \mathbb{E}\left[\int_{t_0}^T e^{-\gamma(s-t_0)} c\left(X_s, \pi_s\right) ds \mid X_{[t_0-\tau_X,t_0]} = x_{[t_0-\tau_X,t_0]}\right] \tag{2.34}$$

Similarly, for the Markovian case, the optimal objective functional is given by

$$J^*\left(x_{t_0}\right) = \inf_{\pi \in \mathscr{A}_\Pi} \mathbb{E}\left[\int_{t_0}^T e^{-\gamma(s-t_0)} c\left(X_s, \pi_s\right) ds \mid X_{t_0} = x_{t_0}\right] \tag{2.35}$$

A particular case of the relaxed optimal control problem that is at the core of Chapter 4 is now presented.

Maximum Entropy Control Problem

An important typical case is the *maximum-entropy control problem*. This approach intends to find the optimal policy that maximises the entropy of the control process. The entropy is a measure of disorder (or uncertainty) of a random variable (or its distribution). Thus, the cost function is given by

$$J_{\mathcal{H}}\left(t, \mathbb{P}_{X_{[t-\tau_X,t]}}, \pi\right) = \mathbb{E}\left[\int_t^T e^{-\gamma(s-t_0)} c\left(X_s, \pi_s\right) - \alpha^{\mathcal{H}} \mathcal{H}\left[\pi_s\right] ds\right] \tag{2.36}$$

where $\mathcal{H}$ denotes the entropy (Cover and Thomas 2006).

The regularisation introduced by the entropy term is a way to promote exploration. Chapter 4 discusses how this regularisation also impacts the robustness of the final solutions.

The following section introduces the concept of a feedback policy, also referred to as a closed-loop policy. These policies are widely regarded as the standard within the field of learning-based control.

Closed-Loop Policy

Definition 2.2.9 (History-dependent Policy)**.** *Let $t \in I$. The measure-valued process $\pi = (\pi_t)_{t\in I}$ is said to be a* history-dependent policy *if for any $t \in I$, it is a function of the past trajectory of the observation-control process $\left((Y_s)_{s\leq t}, (U_s)_{s<t}\right)$.*

Formally, for any $t \in I$, there exists a mapping (probability kernel) $\tilde{\pi} : \mathcal{Y}^{[t_0,t]} \times \mathcal{U}^{[t_0,t[} \to \Pi$ such that

$$\pi_t = \tilde{\pi}_t \left(du \mid (Y_s)_{s\leq t}, (U_s)_{s<t}\right) \tag{2.37}$$

The set of all admissible history-dependent policies is denoted $\mathscr{A}_\Pi^H$.

Definition 2.2.10 (Markovian Policy)**.** *A feedback policy $\pi = (\pi_t)_{t\in I}$ is a history-dependent policy where the associated mapping $\tilde{\pi}_t$ is solely a function of the instantaneous observation Y_t at any time t.*

Formally, for any $t \in I$, there exists a probability kernel $\tilde{\pi} : \mathcal{Y} \to \Pi$ such that

$$\pi_t = \tilde{\pi}_t \left(du \mid Y_t\right) \tag{2.38}$$

This kind of policy is sometimes called Markovian policy*. The set of all admissible Markovian policies is denoted $\mathscr{A}_\Pi^M$.*

In fact, definitions 2.2.4 and 2.2.5 from the control process section (2.2.3) are particular cases of the definitions 2.2.9 and 2.2.10 for the policy.

The next section introduces a foundational concept in control theory: the Dynamic Programming Principle. This principle allows for the solution of the optimal control problem in a recursive manner.

2.2.6 Dynamic Programming Principle

Background

The 12-page preface to Richard Bellman's seminal book *Dynamic Programming* (R. E. Bellman 1957) is a great way to get started with the concept Dynamic Programming (DP). Some elements of the author's introduction are given here.

Factually, the term "Dynamic Programming" refers to a specific framework furnishing a "versatile tool" to deal with the mathematical theory of multi-stage decision processes. A multi-stage decision process is a process that is controlled at several stages: decisions or controls are made to modify the underlying

dynamics. In continuous time, a controlled process such as (2.1) is viewed as a process on which an infinite number of decisions is made over the allotted time.

On the first hand, the notion of "programming" traditionally reflects the act or process of making plans or scheduling. In fact, it is intrinsically linked to the concept of planning. This naturally contains the decision-making part of the topic.

On the other hand, the term "dynamics" translates the temporal property of the method to solve control problems. In Bellman's own words, "time plays a significant role" and "the order of operations may be crucial" in Dynamic Programming.

A cornerstone result of this theory is given by the so-called *principle of optimality* also known as Dynamic Programming Principle or functional equations (Puterman 2014). Ronald Howard calls it the Recurrence Relation in his founding book Howard 1960[22].
The following section is devoted to the statement of this principle

Mathematical Formulation

First of all, here the dynamics are supposed to be Markovian. The following remark gives further details on this assumption.

Remark 2.2.16 (Markovian Assumption)**.** *To formulate the Dynamic Programming Principle, the dynamics are assumed to be Markovian as well as the policy which belongs to $\mathscr{A}_{\Pi}^{M}$. As Kolmogorov himself stated in his foundational paper Kolmogoroff 1931, a non-Markovian process can always be transformed into a Markovian one by considering a higher-dimensional state space. Section 7.1.3 also provides a development on the Markovian assumption. See also Hale 1971; S. Mohammed 1984 for a treatment of delay differential equations as functional differential equations.*

In mathematical statistics, the *filter* of a partially observable (also called hidden) stochastic process is the conditional distribution of the instantaneous state given all the past observations. It is sometimes referred to as a *belief state* (X. Chen et al. 2022), Section 7.1.2 provides more details on this concept.

Formally, it is defined as follows

Definition 2.2.11 (Filter)**.** *For any $s, t \in I$, the filter $\mathbb{P}_{X_t}^{Y_{[t,s]}}$ of the partially observable dynamics* (2.1)-(2.2) *is defined as*

$$\mathbb{P}_{X_s}^{Y_{[t,s]}}(dx) \coloneqq \mathbb{P}\left(X_s \in dx \mid Y_{[t,s]}\right) \tag{2.39}$$

[22]Note that both Bellman's and Howard's books are considered as edifying works of the sequential decision-making theory.

Hence, $\mathbb{P}_{X_s}^{Y_{[t,s]}}$ is the conditional distribution of the state process at time s given all the observations from t to s.

Remark 2.2.17 (Kalman Filter)**.** *The Kalman Filter is an essential instance of this concept in the field of control theory and signal processing (Kalman and Bucy 1961). It is obtained in the specific case of linear Gaussian dynamics and observations.*

In this part, the optimal objective functional at time $t \in [t_0, T]$ is $J^*(t, \rho)$ where ρ is any state distribution supported on $\mathcal{X}$ (*e.g.* $\rho = \mathbb{P}_{X_t}$, the initial distribution of the Markovian version of Eq. (2.1)-(2.2)). Thus, when $\rho = \delta_{\{x\}}$, for some $x \in \mathcal{X}$, the dynamics at time t start from the state x and the optimal objective functional is usually denoted $J^*(t, x)$.

In a very general way, the Dynamic Programming Principle (DPP) (R. Bellman 1957; R. E. Bellman 1957) can be stated as follows for a finite horizon $T < +\infty$.

Theorem 2.2.1 (Dynamic Programming Principle - Filtered Version)**.** *Let $T < +\infty$ and the dynamics of Eq. (2.1)-(2.2) be Markovian. The optimal objective functional J^* satisfies the Dynamic Programming Principle (DPP).*

$$J^*(t, \mathbb{P}_{X_t}) = \inf_{\pi \in \mathscr{A}_\Pi^M} \mathbb{E}\left[\int_t^{\tilde{t}} e^{-\gamma(s-t_0)} c(\mathbb{P}_{X_s}^{Y_{[t,s]}}, \pi_s) ds + J^*(\tilde{t}, \mathbb{P}_{X_{\tilde{t}}}^{Y_{[t,\tilde{t}]}})\right] \tag{2.40}$$

for any $t \in I$ and $\tilde{t} \in [t, T]$ where

$$c(\mathbb{P}_{X_s}^{Y_{[t,s]}}, \pi_s) \coloneqq \int_{\mathcal{X}} \int_{\mathcal{U}} c(x, u) \pi_s(du) \mathbb{P}_{X_s}^{Y_{[t,s]}}(dx) \tag{2.41}$$

This general version of the DPP with filters is due to El Karoui 1987.

When the dynamics are fully observed, the filter has a simple form

$$\mathbb{P}_{X_s}^{Y_{[t,s]}}(dx) = \mathbb{P}\left(X_s \in dx \mid Y_{[t,s]}\right) = \mathbb{P}\left(X_s \in dx \mid X_{[t,s]}\right) = \delta_{X_s} \tag{2.42}$$

where δ_{X_s} is the Dirac measure at X_s. Naturally, the cost function simplifies and the classical DPP is recovered.

Theorem 2.2.2 (Dynamic Programming Principle - Fully Observed Version)**.** *Let $T < +\infty$ and the dynamics of Eq. (2.1)-(2.2) be Markovian. The optimal objective functional J^* satisfies the Dynamic Programming Principle (DPP).*

$$J^*(t, \mathbb{P}_{X_t}) = \inf_{\pi \in \mathscr{A}_\Pi^M} \mathbb{E}\left[\int_t^{\tilde{t}} e^{-\gamma(s-t_0)} c(X_s, \pi_s) ds + J^*(\tilde{t}, \mathbb{P}_{X_{\tilde{t}}})\right] \tag{2.43}$$

for any $t \in I$ and $\tilde{t} \in [t, T]$ where

$$c(X_s, \pi_s) \coloneqq \int_{\mathcal{U}} c(X_s, u) \pi_s(du) \tag{2.44}$$

When the time horizon is infinite, a version of the DPP is given by the following theorem. Recall that $J^*(\rho) := J^*(0, \rho)$.

Theorem 2.2.3 (Dynamic Programming Principle - Infinite Horizon)**.** *Let $T = +\infty$ and the dynamics of Eq.* (2.1)-(2.2) *be Markovian. The optimal objective functional J^* satisfies the Dynamic Programming Principle (DPP).*

$$J^*(t, \mathbb{P}_{X_t}) = \inf_{\pi \in \mathscr{A}_\Pi^M} \mathbb{E}\left[\int_t^{\tilde{t}} e^{-\gamma(s-t_0)} c(X_s, \pi_s) ds + e^{-\gamma \tilde{t}} J^*(t, \mathbb{P}_{X_{\tilde{t}}})\right] \tag{2.45}$$

for any $t \in I$ and $\tilde{t} \in [t, +\infty]$.

Formulas (2.40),(2.43) and (2.45) are the continuous-time versions of the Dynamic Programming Principle.

In simple words and considering the case of Theorem 2.2.2 where $\mathbb{P}_{X_t} = \delta_{\{x\}}$ for some state $x \in \mathcal{X}$ (the initial state is known and fixed), the DPP is a consistency condition which holds between the value of the optimal objective functional for a given state and its possible successors states (Sutton and Barto 2018).
This recursive relation shows that the expected objective value of the state $x \in \mathcal{X}$, or some distribution over the states $\mathbb{P}_{X_t}$ at time t, can be split into two components: the immediate cost between t and $\tilde{t}$ which is the integral term in (2.40)-(2.43) and the discounted expected value of the objective starting from successive states $X_{\tilde{t}}$.

Hamilton-Jacobi-Bellman

By letting $\tilde{t} \to t$, a differential form is obtained from the DPP equations. It yields a second order, nonlinear partial differential equation called Hamilton-Jacobi-Bellman equations (HJB).[23]
A smooth solution of the HJB equation is a candidate solution of the DPP equation. The so-called *verification theorem* shows that this solution coincides with the optimal objective function of the DPP. The main drawback of this approach is that optimal objective functions are not smooth in general (Pham 2009) thus not all regular HJB solutions can represent optimal objective functions.
Consequently, a type of weak solutions of partial differential equations called *viscosity solutions* has been developed in the 1980s by Michael Crandall and Pierre-Louis Lions as a suitable framework to study stochastic control problems (Bardi and Capuzzo-Dolcetta 2008; Trélat 2005).

The link from the continuous time to the discrete time point of view start with the following section.

[23]The equation is not presented here for the sake of brevity.

2.3 Sampling

Even though many real-life control systems are continuous in time, it is sometimes practical to transport the framework in the discrete-time realm. Notably, this widens the range of possible methods to solve the control problem, at the price of approximation errors. Moreover, it allows also representing the problem from an analogue signal to a numeric signal point of view (Salomon 2010).

2.3.1 From Analogue to Digital

An *analogue* signal represents a continuously variable physical quantity. In practice, this signal is first sampled and a sequence of points is obtained. The set of those chronologically ordered measurements is called the *digital* signal.

The idea of sampling consists of defining a discrete time system such that the trajectories of this discrete time system and its corresponding continuous time system coincide at the sampling times (Grüne and Pannek 2011, Chapter 2). This control setup is also known as *sampled-data systems*.

Time Partition

With this in mind, the interval $I = [t_0, T] \subset \overline{\mathbb{R}}_+$ is discretised with a sequence of $K \in \overline{\mathbb{N}}^*$ time points representing sampling instants. Let $[\![t_0, K]\!]$ denote the set of integers from t_0 to K, denoted by $[\![t_0, K]\!] = \{t_0, \dots, K\}$. If $K = +\infty$, then $[\![t_0, K]\!] = \overline{\mathbb{N}}$.

Definition 2.3.1 (Deterministic Time Partition)**.** *Let $K \in \overline{\mathbb{N}}^*$. A deterministic time partition is a collection $(s_k)_{k=0}^K$ of time points in I such that*

$$t_0 \leq s_0 < s_1 < \ldots < s_K \leq T \tag{2.46}$$

Note that when $K = \{+\infty\}$, the collection is an ascending sequence of elements in the interval I.

Remark 2.3.1 (Discrete-time Indexing)**.** *Henceforth, a discrete time stochastic process $(\tilde{X}_{s_k})_{k=0}^K$ that is indexed by a deterministic time partition $(s_k)_{k=0}^K$ shall be directly indexed by the index k of the time points s_k such that the stochastic process is denoted $(\tilde{X}_k)_{k=0}^K$. However, the time partition indexing is kept for the rest of this chapter to stress the link between the continuous-time and discrete-time processes.*

Parenthetically, it is important to identify conditions under which the sampled sequence faithfully represents the original signal. An important theorem, called Shannon-Nyquist, gives conditions under which the original signal can be reconstructed. This sampling theorem is the bridge between the analogue (physical) world and the discrete-time (computational) world of digital signal processing.

There is a whole theory called Signal Processing treating the question of sampling and other important related problems. The book Brémaud 2001 is a great introduction, and the above-mentioned theorem is presented there.

Nonetheless, a more general type of discretisation can be considered.

Random Time Partition

Some sampling procedures may be subject to irregularity or perturbations. In addition, the sampling times can themselves be controlled by an external agent. Thus, it can be beneficial to work with *random sampling times*[24] which are specified by a random time partition.

Definition 2.3.2 (Random Time Partition)**.** *Let $K \in \overline{\mathbb{N}}^*$. A random time partition is a collection $(\kappa_k)_{k=0}^K$ of I-valued random variables such that*

$$t_0 \leq \kappa_0 < \kappa_1 < \ldots < \kappa_K \leq T \tag{2.47}$$

Note that when $K = \{+\infty\}$, the collection is an increasing sequence of random variables in I.

The reader is invited to think of a random time partition as a noisy version of a deterministic time partition. For instance, a Gaussian noise can be added to the deterministic time partition to get a perturbed but still ordered partition. This would represent the potential jitter in the sampling times.

Example 2.3.1. *Let $(s_k)_{k=0}^K$ be a deterministic time partition. A noisy time partition $(\kappa_k)_{k=0}^K$ is obtained by adding a Gaussian noise to the deterministic time partition*

$$\kappa_k \sim \mathcal{N}\left(s_k, \sigma_{s_k}^2\right) \tag{2.48}$$

where $\sigma_{s_k}^2 \in \mathbb{R}_+$ is chosen small enough to keep the order of the time points (this can be probabilistically quantified, but there exists always a positive probability of having a permutation since the support of the normal distribution is unbounded).

A better choice may be to consider finite support distributions or to define this random partition recursively.

Indeed, as suggested in Example 2.3.1, the choice of the noise distribution is not trivial. Moreover, the time partition can be generated dynamically, based on previous data.

As a matter of fact, the time partition can represent the moments where the system is interrogated[25]. Possibly, it could define the times when the control process is updated. Thus, sampling times are also called *decision times* or *decision epochs*. This gives rise to the concept of *inter-decision times*.

[24] In probability theory, those are called stopping times.

[25] Measured or probed.

Definition 2.3.3 (Inter-decision Time)**.** *The inter-decision time is the time elapsed between two consecutive decision times (or sampling times).*
For a random time partition $(\kappa_k)_{k=0}^K$*, the inter-decision times are the collection* $(\eta_k)_{k=1}^K$ *of random variables such that*

$$\eta_k \coloneqq \kappa_k - \kappa_{k-1} \tag{2.49}$$

Remark 2.3.2. *In the deterministic case,*

$$\eta_k \coloneqq s_k - s_{k-1} \tag{2.50}$$

Now, core components of the sampled version of the continuous time controlled stochastic process are defined: the discrete time distributions and transition probabilities. They provide a mechanism to describe random motion (Revuz and Yor 1999).

2.3.2 Discrete time Distributions and Transition Kernels

Initial Probability Distributions

For some time partition $s_{-r_X} = t_0 - \tau_X < \ldots s_{-i} < \ldots < s_{-1} < s_0 = t_0$, with $r_X \in \mathbb{N}^*$, of the history function domain $[t_0 - \tau_X, t_0]$, the initial probability distribution of the state process is denoted $\mathbb{P}_{X_{s_{-r_X}},\ldots,X_{s_0}}(dx_{-r_X}, \ldots, dx_0)$.
The initial observation Y_{t_0} is determined by the initial state as stated in (2.2)

$$\mathbb{P}_{Y_{t_0}}(dy_0) \coloneqq \delta_{g_0(X_0)} \tag{2.51}$$

The initial inter-decision time η_0 is t_0.

$$\mathbb{P}_{\eta_0}(d\sigma_0) \coloneqq \delta_{\{t_0\}}(d\sigma_0) \tag{2.52}$$

Discrete-time State Transition Probability

The system (2.1) considered in this manuscript is stationary (the operator f does not depend on time). Thus, a time independent (a.k.a. homogeneous) transition probability can be defined. However, this transition kernel depends on the random inter-decision time η_k whose realisation $\sigma \in \mathcal{S}$ determines the duration before the next decision time.
For any $k \in [\![0, K]\!]$, the discrete-time state transition probability is defined as

$$\begin{aligned}\mathcal{P}(dx'' \mid x, x', u, \sigma) &\coloneqq \mathbb{P}_{X_{s_k+\eta_k}}(dx'' \mid X_{s_k} = x, X_{s_k-\tau_X} = x', U_{s_k} = u, \eta_k = \sigma) \\ &= \mathbb{P}_{X_{s_k+\sigma}}(dx'' \mid X_{s_k} = x, X_{s_k-\tau_X} = x', U_{s_k} = u) \\ &= \mathbb{P}_{X_\sigma}(dx'' \mid X_0 = x, X_{-\tau_X} = x', U_0 = u) \quad \text{(by stationarity)}\end{aligned} \tag{2.53}$$

for any inter-decision time σ and $s_k \in I$. Thus, $\mathcal{P}$ is independent of k.
This collection of probability measures that are (measurably) indexed (here by

the elements in $\mathcal{X}\times\mathcal{X}\times\mathcal{U}\times\mathcal{S}$) are called *transition kernels* or transition functions and plays a fundamental role in the stochastic process theory.

Now, detailed explanations of this unusual definition of $\mathcal{P}$ are given. The following description is inspired by Puterman 2014, Chapter 11 and the references cited therein.

In the scope of this thesis, the processes $(X_{\kappa_k})_{k=0}^K$ and $(U_{\kappa_k})_{k=0}^K$ are considered as sampled versions of the continuous-time processes $(X_s)_{s\in I}$ and $(U_s)_{s\in I}$. Those discrete-time processes are called *sampled processes*, *decision processes*, or *embedded processes*, while their continuous-time counterparts are called *natural processes* or *underlying processes*.

This distinction underlines the classical control settings where it is not possible nor necessary to control the system at any time. Sometimes, what transpires between decision epochs provides no relevant information to the decision maker. In general, the system state (natural process) may vary between decision epochs. However, the control is only allowed at sampling times (decision process). An important instance of such a setting is when the control process is not continuous but **piecewise constant** (*a.k.a.* jump process) and the decisions performed are the modifications of the control signal. This is a usual choice, and this will be the case for all the work presented in this thesis. Additionally, the approach examined here enables the continuous-time nature of the system dynamics to be maintained.

The process $(X_s)_{s\in I}$ equipped with the random time sequence $(\kappa_k)_{k=0}^K$ such that $(X_{\kappa_k})_{k=0}^K$ is a Markov chain[26] is called a *semi-Markov decision process* (See Lévy 1954; Harlamov 2004, for the uncontrolled case).

Here, the random variable X_{κ_k} is measurable w.r.t. the product $\mathscr{F}_{\mathcal{X}}\otimes\mathscr{F}_I$ of the σ-algebra $\mathscr{F}_{\mathcal{X}}$ endowing the state space $\mathcal{X}$ and the σ-algebra $\mathscr{F}_{\mathcal{X}}$ generated by the intervals on $I=[0,T]$.[27] Then, for any $\omega\in\Omega$, the point $X_{\kappa_{k(\omega)}}(\omega)\in\mathcal{X}$ represents the value of the stochastic process $X(\omega)=(X_s(\omega))_{s\in I}$ at the random time $\kappa_k(\omega)$.

Discrete-time Inter-decision Time Transition Probability

The transition kernel $\mathcal{O}$ specifies the inter-decision time conditional distribution for any $k\in[\![0,K]\!]$.

$$\mathcal{O}(d\sigma \mid x,x',u) \coloneqq \mathbb{P}_{\eta_k}\left(d\sigma \mid X_{s_k}=x, X_{s_k-\tau_X}=x', U_{s_k}=u\right) \tag{2.54}$$

There are multiple interesting cases to consider. For instance, the inter-decision times conditional distribution is independent of the state and control

[26] In this case the policy and the inter-decision times are necessarily Markovian.

[27] Basically, this means the random variable X_{κ_k} is indexed by a random time κ_k. Consequently, this variable carries information relative to events in both the state space and the time interval.

processes ($\mathcal{O}(d\sigma \mid x, x', u) = \mathcal{O}(d\sigma)$) for all $(x, x', u) \in \mathcal{X} \times \mathcal{X} \times \mathcal{U}$. Also, the dependence on $u \in \mathcal{U}$ highlights the impact of an exogenous agent (the sampler) on sampling times. Another typical case is when the inter-decision times distribution is degenerated ($\mathcal{O}(d\sigma) = \delta_{\{s'\}}(d\sigma)$) for some $s' \in I$. In this case, the sampling corresponds to a deterministic, equispaced time partition.

Another important historical and seminal case comes up when $\mathcal{P}$ is Markovian where $\mathcal{P}(dx'' \mid x, x', u, \sigma) = \mathcal{P}(dx'' \mid x, u, \sigma)$ for all $(\sigma, x, x', u) \in \mathcal{S} \times \mathcal{X} \times \mathcal{X} \times \mathcal{U}$.
Originally, the introduction of the inter-decision times η was motivated by the need to model problems where the system state changes at random, irregular times. The duration for which a stochastic process stays in the same state is called *sojourn time* or *exit time*.

Notable examples are queueing control and equipment maintenance that are more naturally modelled by allowing system interaction at random times. Thus, for those problems there is an intrinsic notion of sojourn time, *i.e.* the time spent in a state before a transition occurs.
In addition, when the sojourn time in a specific state, for a finite state space, (respectively region, for a continuous state space) follows a geometric distribution (respectively exponential distribution), the embedded state process $(X_s)_{s\in I}$ is necessarily a Markov chain.

Incidentally, the idea of studying processes with more general inter-decision times came independently and almost simultaneously from Paul Lévy and Walter Laws Smith in the mid-1950s (Grabski 2016). At that time, certain practical problems compelled researchers to seek an adequate mathematical description. Attempts to apply Markov models to these problems were sometimes unsatisfactory because the exponential distribution of the sojourn times was not always appropriate (Harlamov 2004, Preface).
Consequently, the community was looking for processes that are Markovia at decision times but with inter-decision times that are not necessarily exponential. The semi-Markov process was born.

Again, the use of semi-markov models for modelling dynamics where the system state is piecewise constant is not really the aim of the work presented in this thesis. The state is supposed to vary continuously between decision times. Rather, the inter-decision times will be used in Chapter 5 to construct a randomised dataset of type $\mathcal{D} = (X_{\kappa_k}, U_{\kappa_k}, X_{\kappa_{k+1}})_{k=0}^{K}$ that maximises the information in a sense which will be precisely defined.

In traditional presentations, the transition law $\mathcal{Q}$ of $X_{\kappa_{k+1}}$ on $\mathcal{X} \times I$ is the central object of interest. It is given here for the sake of completeness, but it will not be used in the rest of the document.

$$\mathcal{Q}(dx''d\sigma \mid x, x', u) \coloneqq \mathbb{P}_{X_{\kappa_k}+\eta_k}(dx''d\sigma \mid X_{s_k} = x, X_{s_k-\tau_X} = x', U_{s_k} = u, \kappa_k = s_k) \tag{2.55}$$

Again, this equation can be simplified by the stationarity of the dynamics.

Remark that this distribution is linked to $\mathcal{P}$ and $\mathcal{O}$ by

$$\mathcal{Q}(dx''d\sigma \mid x, x', u) = \int_{d\sigma} \mathcal{P}(dx'' \mid x, x', u, \sigma')\mathcal{O}(d\sigma' \mid x, x', u) \tag{2.56}$$

If $\mathcal{P}$ is independent of σ, then

$$\mathcal{Q}(dx''d\sigma \mid x, x', u) = \mathcal{P}(dx'' \mid x, x', u)\mathcal{O}(d\sigma \mid x, x', u) \tag{2.57}$$

This is an example of conditional independence[28] of X_{s_k} and η_k given the random variables X_{s_k}, $X_{s_k-\tau_X}$ and U_{s_k}.

Discrete-time Observation Transition Probability

The observation kernel $\mathcal{G}$ reads

$$\mathcal{G}(dy \mid x, x', u) \coloneqq \mathbb{P}_{Y_{s_k}}(dy \mid X_{s_k} = x, X_{s_k-\tau_Y} = x', U_{s_k} = u) \tag{2.58}$$

Discrete-time Policy

A discretised version of the history process is required to define history-based policies. First of all, the *discrete-time history space* is defined for $k \in [\![0, K]\!]$ as

$$\mathscr{H}_k^{\eta,X,Y,U} = (\mathcal{S} \times \mathcal{X} \times \mathcal{Y} \times \mathcal{U})^{k-1} \times \mathcal{S} \times \mathcal{X} \times \mathcal{Y} \tag{2.59}$$

Note that the control is excluded from the k-th product.
In a similar manner, the *discrete-time observable history space* is defined as

$$\mathscr{H}_k^{\eta,Y,U} = (\mathcal{S} \times \mathcal{Y} \times \mathcal{U})^{k-1} \times \mathcal{S} \times \mathcal{Y} \tag{2.60}$$

Let $(s_k)_{k=0}^K$ be a deterministic time partition. A *history process* is a $\mathscr{H}_k^{\eta,X,Y,U}$ valued random variable representing the sampling history up to time s_k defined as

$$H_{s_k}^{\eta,X,Y,U} = \left(\eta_{s_0}, X_{s_0}, Y_{s_0}, U_{s_0}, \dots, U_{s_{k-1}}, \eta_{s_k}, X_{s_k}, Y_{s_k}\right) \tag{2.61}$$

Similarly, an *observable history process* is defined as

$$H_{s_k}^{\eta,Y,U} = \left(\eta_{s_0}, Y_{s_0}, U_{s_0}, \dots, U_{s_{k-1}}, \eta_{s_k}, Y_{s_k}\right) \tag{2.62}$$

A point in the history space $\mathscr{H}_k^{\eta,X,Y,U}$ is denoted by

$$h_{s_k}^{\eta,X,Y,U} = \left(\sigma_{s_0}, x_{s_0}, y_{s_0}, u_{s_0}, \dots, u_{s_{k-1}}, \sigma_{s_k}, x_{s_k}, y_{s_k}\right) \in \mathscr{H}_k^{\eta,X,Y,U} \tag{2.63}$$

[28] In probability, two objects (events, random variables, σ-algebras) are independent if their joint distribution is the product of their marginal distributions. Inspired by common sense, two events are independent if the probability of their joint occurrence is the product of their individual probabilities.

Again,

$$h_{s_k}^{\eta,Y,U} = \left(\sigma_{s_0}, y_{s_0}, u_{s_0}, \ldots, u_{s_{k-1}}, \sigma_{s_k}, y_{s_k}\right) \in \mathscr{H}_k^{\eta,Y,U} \tag{2.64}$$

Other spaces such as $\mathscr{H}_k^{Y,U}$ and their corresponding points and random variables can be defined in a similar manner. In the case $K = \{+\infty\}$, the limit space $\mathscr{H}_\infty^{\eta,X,Y,U}$ is defined as

$$\mathscr{H}_\infty^{\eta,X,Y,U} = (\mathcal{S} \times \mathcal{X} \times \mathcal{Y} \times \mathcal{U})^{\mathbb{N}} \tag{2.65}$$

and the associated objects similarly (*e.g.* $H_\infty^{\eta,X,Y,U}$, $h_\infty^{\eta,X,Y,U}$ are sequences indexed by $\mathbb{N}$).

Echoing the continuous-time policy definition in Definition 2.2.6, a discrete-time policy is defined as follows

Definition 2.3.4 (Discrete-time Policy)**.** *A discrete-time policy denoted $(\pi_k)_{k=0}^K$ is a measure-valued discrete-time stochastic process such that $\pi_k \in \Pi$ for any $k \in [\![0, K]\!]$.*

Definition 2.3.5 (Discrete-time History-dependent Policy)**.** *A discrete-time history-dependent policy $(\pi_k)_{k=0}^K$ is a discrete-time policy such that for any $k \in [\![0, K]\!]$, it is a function of the past trajectory of the observation-control process $H_k^{\eta,Y,U}$.*

Formally, there exists a mapping $\tilde{\pi} : \mathscr{H}_k^{\eta,Y,U} \to \Pi$ such that

$$\pi_k = \tilde{\pi}_k\left(du \mid H_k^{\eta,Y,U}\right) \tag{2.66}$$

Thus, for $\omega \in \Omega$,

$$\pi_k(\omega) = \tilde{\pi}_k\left(du_k \mid H_k^{\eta,Y,U}(\omega)\right) = \tilde{\pi}_k\left(du_k \mid h_k^{\eta,Y,U}\right) \tag{2.67}$$

which writes

$$\pi_k(\omega) = \tilde{\pi}_k\left(du_k \mid \sigma_{s_0}, y_{s_0}, u_{s_0}, \ldots, u_{s_{k-1}}, \sigma_{s_k}, y_{s_k}\right) \tag{2.68}$$

The definition for Markovian policies is straightforward.

Definition 2.3.6 (Discrete-time Markovian Policy)**.** *A discrete-time Markovian policy $(\pi_k)_{k=0}^K$ is a discrete-time policy such that for any $k \in [\![0, K]\!]$, it is a function of the current state X_k. Formally, there exists a mapping $\tilde{\pi} : \mathcal{X} \to \Pi$ such that*

$$\pi_k = \tilde{\pi}_k(du \mid X_k) \tag{2.69}$$

Thus for $\omega \in \Omega$,

$$\pi_k(\omega) = \tilde{\pi}_k(du_k \mid X_k(\omega)) = \tilde{\pi}_k(du_k \mid x_k) \tag{2.70}$$

2.3.3 Discrete-time Process distribution

In this part, the Markovian case is considered

$$\mathcal{P}(dx'' \mid x, x', u, \sigma) = \mathcal{P}(dx'' \mid x, u, \sigma) \tag{2.71}$$

Markovian case

In the Markovian case, $\tau_X = 0$ and $\tau_Y = 0$, thus

$$\mathcal{P}(dx'' \mid x, x', u, \sigma) = \mathbb{P}_{X_{s_k+\sigma}}(dx'' \mid X_{s_k} = x, U_{s_k} = u, \eta_{s_k} = \sigma) = \mathcal{P}(dx'' \mid x, u, \sigma) \tag{2.72}$$

and

$$\mathcal{O}(d\sigma \mid x, x', u) = \mathbb{P}_{\eta_k}(d\sigma \mid X_{s_k} = x, U_{s_k} = u) = \mathcal{O}(d\sigma \mid x, u) \tag{2.73}$$

and

$$\mathcal{G}(dy \mid x, x', u) = \mathbb{P}_{Y_{s_k}}(dy \mid X_{s_k} = x, U_{s_k} = u) = \mathcal{G}(dy \mid x, u) \tag{2.74}$$

and

$$\mathcal{Q}(dx''d\sigma \,|\, x, x', u) = \mathbb{P}_{X_{\kappa_k+\eta_k}}(dx''d\sigma \,|\, X_{s_k} = x, U_{s_k} = u, \kappa_k = s_k) = \mathcal{Q}(dx''d\sigma \,|\, x, u) \tag{2.75}$$

Notably, the initial probability $\mathbb{P}_{X_{s_{-r_X}},\dots,X_{s_0}}(dx_{-r_X}, \dots, dx_0)$ does not generate a history function any more and boils down to the probability $\mathbb{P}_{X_0}(dx_0)$.

Now, the probability distribution $\mathbb{P}_{H_{s_k}^{\eta,X,Y,U}}$ of a random history $H_{s_k}^{\eta,X,Y,U} = \left(\eta_{s_0}, X_{s_0}, Y_{s_0}, U_{s_0}, \dots, U_{s_{k-1}}, \eta_{s_k}, X_{s_k}, Y_{s_k}\right)$ for any finite $k \in [\![0, K]\!] \setminus \{+\infty\}$ can be written in a useful recursive form involving only initial probabilities and transition kernels.

Proposition 2.3.1 (PO-SMDP Distribution)**.** *Let $k \in [\![0, K]\!] \setminus \{+\infty\}$, the distribution of the history process $H_{s_k}^{\eta,X,Y,U}$ is given by*

$$\begin{aligned}\mathbb{P}_{H_{s_k}^{\eta,Y,U}}(d\sigma_{s_0}, dx_{s_0}, dy_{s_0}, du_{s_0}, \dots, d\sigma_{s_k}, dx_{s_k}, dy_{s_k}) = {} & \mathbb{P}_{\eta_{s_0}}(d\sigma_{s_0})\mathbb{P}_{X_{s_0}}(dx_{s_0})\mathbb{P}_{Y_{s_0}}(dy_{s_0}) \\ & \pi_{s_0}(du_{s_0} \mid y_{s_0})\mathcal{O}(d\sigma_{s_1} \mid x_{s_0}, u_{s_0})\mathcal{P}(dx_{s_1} \mid x_{s_0}, u_{s_0})\mathcal{G}(dy_{s_1}, \mid u_{s_0}x_{s_1}) \\ & \pi_{s_1}(du_{s_1} \mid h_{s_0}^{\eta,Y,U}, y_{s_1}) \dots\dots \pi_{s_{k-1}}(du_{s_{k-1}} \mid h_{s_{k-2}}^{\eta,Y,U}, y_{s_k})\mathcal{O}(d\sigma_{s_k} \mid x_{s_{k-1}}, u_{s_{k-1}}) \\ & \mathcal{P}(dx_{s_k} \mid x_{s_{k-1}}, u_{s_{k-1}})\mathcal{G}(dy_{s_k}, \mid u_{s_k}x_{s_k})\end{aligned} \tag{2.76}$$

In the case $K = \{+\infty\}$, the distribution $\mathbb{P}_{H_\infty^{\eta,X,Y,U}}$ of the history process $H_\infty^{\eta,X,Y,U}$ is an extension[29] *of the finite dimensional distributions given by* (2.76)*. of the finite dimensional distributions given by* (2.76)*.*

Proof. The first part of the proposition is obtained by applying the chain rules for conditional distributions. Then, the probability distribution can be extended to the infinite horizon case by the Ionescu-Tulcea Theorem (Neveu 1970; Klenke 2007). □

[29] Here, an extension means that the infinite dimensional probability measure $\mathbb{P}_{H_\infty^{\eta,X,Y,U}}$ is equal to the finite dimensional probability measure $\mathbb{P}_{H_{s_k}^{\eta,X,Y,U}}$ on the subspaces of possible trajectories of size $k \in \mathbb{N}$ (called k-dimensional marginal distributions or projections).

The recursive formula (2.76) of Proposition 2.3.1 is useful to sample the history process and implement Monte Carlo methods (Stoehr 2017). Moreover, it is a core component of the Markov Decision Process (MDP) and their generalisation (PO-MDP, SMDP, etc.).

The extension to the non-Markovian case is challenging.

2.4 Simulation and Numerical Approximation

In the previous section, the shift from the continuous-time world to the discrete-time has been motivated by the necessity to measure and process the signal through a simplified and more tractable representation.

On the other hand, it is not always possible nor desirable to interact with the real system directly. A variety of reasons can be invoked: the high cost of experiment, the danger of the environment, the difficulty of accessing the system, the need for reproducibility, and the necessity of trying a large number of independent scenarios[30], (see Bélanger, Venne, and Paquin 2010, for arguments in the domain of Real-Time Simulation).

In this context, numerical simulations are essential to overcome these technical limitations. Despite being subject to inaccuracies, their proficiency has been proven in numerous fields such as plasma control (Degrave et al. 2022), Robotics (Choi et al. 2021), and more broadly in Physics and Engineering (Steinhauser 2013). These simulations can also be computationally costly while requiring calibration[31] (parameter estimation). Thus, understanding and analysing the different simulation schemes is important to anticipate outcomes and to design efficient algorithms.

The field of Numerical Analysis (Legendre 2021) is the theoretical backbone of simulation engineering. Its purpose is to design and analyse numerical calculation methods or algorithms. This section presents the basic methodology to approximate the continuous-time controlled stochastic process (2.1)-(2.2) by its discrete-time counterpart. A complete presentation of SDE numerical approximation is given in Kloeden and Platen 1992. The paper Buckwar 2000 develops the theory for the delayed case (SDDE).

2.4.1 Approximation with Delayed Dynamics and the Euler-Maruyama Scheme

The continuous-time controlled stochastic process (2.1)-(2.2) is approximated by a discrete-time controlled stochastic process through the standard Euler-

[30] In statistics, this is linked to the notion of *statistical significance* (M. Hoffman 2015; Colas, Sigaud, and Oudeyer 2018).

[31] In robotics, this is often called *Sim2Real* transfer or *reality gap* (Koos, Mouret, and Doncieux 2010; Höfer et al. 2021).

Maruyama scheme.[32]This scheme is the standard approach for SDE numerical approximation.

Delay differential equations lead to various complications in their solution from both the theoretical and numerical points of view (Bellen and Zennaro 2013). The main difficulty is due to the delay term τ_X which is in general unknown or unpredictable (sometimes τ_X depends on the time or the state). Similar difficulties arise in the observation process with the delay term τ_Y.

The first approaches to the numerical solution of deterministic DDE in the sense of (2.11) go back to the 1950s. Indeed, the seminal approach of Elsgolts 1964 imposed serious constraints on the time partition a.k.a. *time mesh*. Given a sequence of N points forming a deterministic time partition[33] $(t_k)_{k=0}^N$, it can be imposed that for all $k \in [\![0, N]\!]$, either $t_k - \tau_X < t_0$ or $t_k - \tau_X \in (t_k)_{k=0}^N$ (a time partition of this type is called τ_X-valid). In this way, the following Euler approximation of the state process is well-defined. The constraint can be reinforced by requiring that the time partition is also τ_Y-valid.

Definition 2.4.1 (Euler-Maruyama Approximation of the General Dynamics)**.** *For a τ_X-valid time partition $(t_k)_{k=0}^N$ with time step size $\delta_{t_k} = t_{k+1} - t_k$, define $\tau_k = t_k - \tau_X$.*

The general discrete-time approximation of the state process is given by:

$$X_{t_{k+1}} = X_{t_k} + f\left(X_{t_k}, X_{t_k-\tau_X}, U_{t_k}\right)\delta_{t_k} + \epsilon_X(X_{t_k}, U_{t_k})\,\delta W_{t_k}, \tag{2.77}$$

where $\delta W_{t_k} = W_{t_{k+1}} - W_{t_k}$ are the Brownian increments.

Similarly, for the observation process and a time partition that is τ_Y-valid, the discrete-time approximation is:

$$Y_{t_{k+1}} = Y_{t_k} + g\left(X_{t_k}, X_{t_k-\tau_Y}, U_{t_k}\right)\delta_{t_k} + \epsilon_Y(X_{t_k}, U_{t_k})\,\delta W^2_{t_k}, \tag{2.78}$$

where $\delta W^2_{t_k}$ are the Brownian increments associated with the observation noise.

Since the increments are independent and normally distributed, the recursive formula (2.77) and (2.78) can be simplified with the following remark.

Remark 2.4.1 (Euler-Maruyama Scheme)**.** *The discrete-time approximation of the state process* (2.77) *and observation process* (2.78) *can be rewritten as:*

$$\begin{cases} X_{t_{k+1}} = X_{t_k} + f\left(X_{t_k}, X_{t_k-\tau_X}, U_{t_k}\right)\delta_{t_k} + \mathcal{N}\left(0, \epsilon_X(X_{t_k}, U_{t_k})\right) \\ Y_{t_{k+1}} = Y_{t_k} + g\left(X_{t_k}, X_{t_k-\tau_Y}, U_{t_k}\right)\delta_{t_k} + \mathcal{N}\left(0, \epsilon_Y(X_{t_k}, U_{t_k})\right) \end{cases} \tag{2.79}$$

where $\mathcal{N}(0, \epsilon_X(x, u))$ is a Gaussian process with covariance operator $\epsilon_X(x, u)$.[34]*The noise for the observation process is defined similarly.*

[32]Simply the extension of the Euler method to the stochastic differential equation case.

[33]In the remaining, the *time mesh* will always be a deterministic time partition. For a modern example of adaptive stochastic mesh construction, see Kelly and O'Donovan 2024.

[34]Remember, ϵ_X and ϵ_Y are operator-valued mappings. In the finite-dimensional case, they are positive definite matrices. They define quadratic forms.

By defining discrete-time dynamics operators, the recursive formulations can be simplified further.

Definition 2.4.2 (General Discrete-time Dynamics)**.** *Let the discrete-time dynamics operator F defined as*

$$F \coloneqq Id + f\delta_{t_k} \tag{2.80}$$

and define the observation operator G similarly as

$$G \coloneqq Id + g\delta_{t_k} \tag{2.81}$$

Suppose that the time delay τ_X allows for a τ_X-valid time partition, and the same holds for the observation process with τ_Y. Then, the state delay index is defined as

$$r_X \coloneqq \tau_X \quad \textit{with} \quad t_k - \tau_X = t_{k-r_X} \tag{2.82}$$

for $r_X \in \mathbb{N}^*$. Hence, the value $k - \tau_X$ is viewed as an index in the time partition. The observation delay index r_Y is defined similarly.

The discrete-time dynamics of the state and observation processes are given by:

$$\begin{cases} X_{k+1} = F\left(X_k, X_{k-r_X}, U_k\right) + \delta_{t_k}\mathcal{N}\left(0, \epsilon_X(X_k, U_k)\right), \\ Y_{k+1} = G\left(X_k, X_{k-r_Y}, U_k\right) + \delta_{t_k}\mathcal{N}\left(0, \epsilon_Y(X_k, U_k)\right). \end{cases} \tag{2.83}$$

The discrete-time dynamics provide an approximation to the continuous-time dynamics over discrete time steps δ_{t_k}, with noise terms accounting for the randomness introduced by the stochastic processes.

In this manner, a discrete-time formulation is obtained from a continuous-time stochastic system.

Moreover, since τ_X-valid time partitions are too restrictive, the following remark introduces a more general but less accurate approach.

Remark 2.4.2 (Approximation of the Delayed State and Observation Processes)**.** *Approximations of the state X_{k-r_X} and observation Y_{k-r_Y} can be performed when the time partition is not τ_X-valid or τ_Y-valid. For instance, Feldstein 1964 introduced piecewise constant or linear interpolation for the state process while a similar approach can be applied to the observation process. This construction is based on the values of the state and observation processes $(X_k)_{k=0}^N$ and $(Y_k)_{k=0}^N$ at the discrete times $(t_k)_{k=0}^N$. This results in approximated delayed state and observation processes $\hat{X}_{k-r_X}$ and $\hat{Y}_{k-r_X}$ that can be plugged into the discrete-time dynamics of equations* (2.83) *and* (2.79)*.*

Moreover, the linear equispaced time partition is a common choice for numerical simulations.

Example 2.4.1 (Equispaced Time Partition)**.** *Let $\delta_{t_k} = \delta$ for any $k \in [\![0, N]\!]$. Then, the time partition is equispaced: $t_k = k\delta$.*

Finally, the quantities involving time integrals can also be discretised with Riemann-sum.

Remark 2.4.3 (Riemann Sum Discretization)**.** *In the discrete-time approximation of a continuous-time controlled stochastic system, quantities involving time integrals can be approximated using the Riemann sum. For a general integral over the time interval $[t, T]$ of any integrable function $\tilde{c}$, the integral*

$$\int_t^T \tilde{c}(s)\, ds \tag{2.84}$$

can be discretised as a Riemann sum:

$$\sum_{k=0}^{N-1} \tilde{c}(t_k)\delta_{t_k}, \tag{2.85}$$

where $(t_k)_{k=0}^N$ is a time partition on $[t, T]$ and $\delta_{t_k} = t_{k+1} - t_k$ is the time step size (see Lamboley 2022, for a reminder on Riemann sums).

Similarly, for stochastic integrals the approximation of Remark 2.2.2 given by Eq. (2.5) *is performed.*

2.5 Delay, Sampling Times and Discretisation Compatibility

Whether for sampling or numerical approximation purposes, time partitions have been a central element in the previous constructions. Up to know, the specification of time partitions has been left open except for the τ_X-validity condition in Section 2.4.1. From a practical point of view, some choices of partitions are rather natural and convenient.

Nonetheless, a possible kind of incompatibility between sampling times and discretisation times may arise when the sampling times do not coincide with the discretisation times. Therefore, a notion of compatibility is defined and discussed in the following.

In this section, it is supposed that the random sampling times are given by the random partition $(\kappa_k)_{k=0}^K$ and the discretisation times are given by the deterministic partition $(t_k)_{k=0}^N$. The compatibility condition is now defined.

Definition 2.5.1 (Discretisation and Sampling Times Compatibility)**.** *The sampling times $(\kappa_k)_{k=0}^K$ are said to be compatible with the discretisation times $(t_k)_{k=0}^N$ when they are a subsequence of the discretisation times (a.s.).*

Hence, the compatibility condition ensures sampling is well-defined (with probability one) for a given discretisation.

A common case of compatibility is when the sampling times are the same as the discretisation times.

Example 2.5.1 (Sampling-times Equal to Discretisation-times)**.** *Let the sampling times be equal to the discretisation times, i.e.* $\kappa_k = t_k$ *for all* $k \in [\![0, K]\!]$.[35] *Then, the sampling times are compatible with the discretisation times and* $K = N$.

Consequently, a ubiquitous setting encountered across fields is when these two assumptions hold:

- the time interval is uniformly partitioned as in Example 2.4.1
- the sampling times are the same as the discretisation times as in Example 2.5.1.

In other words, $\kappa = (t_k)_{k=0}^N$ and $t_k = k\delta$.

The following remark discusses the case where not all points used during simulation are sampled.

Remark 2.5.1 (Sampling Frequency)**.** *By Definition 2.5.1, being compatible implies* $K \leq N$. *In that case, the sampling frequency is always lower than the discretisation frequency.*

On the other hand, when the sampling times are not compatible with the discretisation times, an approximation approach that is similar to the one of Remark 2.4.2 can be pursued. This way, the sampled quantities are estimated using the trajectory described by the discretisation times used for numerical approximation.

2.6 Conclusion

In this chapter, the continuous-time stochastic control problem was introduced (Section 2.2). The goal was to set a framework that is general enough to encompass the range of problems encountered in this thesis.

The continuous-time stochastic control problem was formulated as the solution of a stochastic differential equation with delay. The notion of policy (Sections 2.2.4-2.2.5) that generalises the concept of control and that is widely used in the Reinforcement Learning literature was introduced.

In addition, the Dynamic Programming Principle (Section 2.2.6) was outlined as a central way to solve the continuous time stochastic control problem. This principle is a key concept in the field of Reinforcement Learning.

Then, the question of sampling (Section 2.3) was addressed to link the continuous time problem to the discrete time setting. This gives rise to the concept of sampled-data systems.

[35] The notation $[\![0, K]\!]$ stands for the set of integers from 0 to K included. This will be used throughout the document.

Finally, the numerical approximation of the continuous time stochastic control problem was discussed (Section 2.4). The compatibility between the sampling times and the discretisation times was defined (Section 2.5). This compatibility is crucial for the well posedness of the problem.

3 Learning-based Control with Discrete Decision Processes

This chapter introduces the discrete time point of view of the decision process which is widely used in the Learning-Based Control literature.

First, the general discrete-time decision process is defined (Section 3.1) together with the discrete time version of Dynamic Programming (Section 3.1.4). The transition probabilities characterisation is presented which connects to the notions of sampling and data for learning applications (Section 3.1.2).

The Learning Theory (Section 3.2) is then introduced in a sufficient generality to encompass the range of problems encountered in this thesis.

Subsequently, the application of the Learning Theory to the optimal control problem, namely Learning-based Control, is introduced (Section 3.3). Multiple important concepts and paradigms that are used throughout the thesis are presented, such as the policy iteration procedure or Model Predictive Control.

Finally, the chapter concludes with a presentation of the several controlled dynamics that are of interest in the field of control of Dynamical Systems (Section 3.4). All the systems presented in this section are used in the various numerical experiments of this thesis.

3.1 Discrete-Time Decision Processes

In this section, discrete-time decision processes are introduced.

3.1.1 General Discrete Decision Process

A general discrete-time formulation of the state and observation dynamics for learning-based control is given as follows.

Definition 3.1.1 (General Discrete Decision Process - Recurrence)**.** *The state process $X = (X_k)_{k\in\mathbb{N}}$ is governed by the following discrete-time stochastic recurrence equations:*

$$\begin{cases} X_{k+1} = F\left(X_k, X_{k-r}, U_k\right) + \mathcal{N}\left(0, \epsilon_X(X_k, U_k)\right) \\ X_{[\![-r,0]\!]} \sim \mathbb{P}_{X_{[\![-r,0]\!]}} \end{cases} \tag{3.1}$$

where:

- $F : \mathcal{X} \times \mathcal{X} \times \mathcal{U} \to \mathcal{X}$ *is the discrete dynamics operator*
- $r \in \mathbb{N}$ *is the time delay index*
- ϵ_X *is a kernel valued function of* $\mathcal{X} \times \mathcal{U}$*. Thus for any* $x \in \mathcal{X}$ *and* $u \in \mathcal{U}$*,* $\epsilon_X(x, u)$ *is the kernel of a Gaussian process (a quadratic form). If* $\mathcal{X}$ *is a function space, the kernel is a covariance operator, otherwise if* $\mathcal{X}$ *is a finite-dimensional space, the kernel is a covariance matrix.*
- $\mathbb{P}_{X_{[\![-r,0]\!]}}$ *is the distribution of the history process* $X_{[\![-r,0]\!]} = (X_0, \dots, X_{-r})$

The observation process $Y = (Y_k)_{k\in\mathbb{N}}$ *follows a similar discrete-time stochastic process:*

$$\begin{cases} Y_{k+1} = G\left(X_k, X_{k-r}, U_k\right) + \mathcal{N}\left(0, \epsilon_Y(X_k, U_k)\right) \\ \quad Y_0 \sim \delta_{G_0(X_0)} \end{cases} \tag{3.2}$$

where:

- $G : \mathcal{X} \times \mathcal{X} \times \mathcal{U} \to \mathcal{Y}$ *is the observation operator*
- ϵ_Y *is defined similarly to* ϵ_X
- $G_0 : \mathcal{X} \to \mathcal{Y}$ *is the initial observation operator*

This kind of system will now be referred to as *decision process* or discrete *controlled process*. The procedure to get a discrete-time controlled process from a continuous-time random dynamical system modelled as a partially observable stochastic differential equation has been described in the previous chapter. Notably, Remark 2.3.1 and the exposition in Section 2.5 ensure the expression is well-defined and can be derived from a stochastic delayed differential equation.

Notations are kept consistent with the continuous-time case whenever possible, the context should make the distinction clear. For instance, the admissible policy space is still denoted by $\mathscr{A}_\Pi$ and the subset of Markovian admissible policies by $\mathscr{A}_\Pi^M$.

3.1.2 Transition Probabilities Characterisation

In the discrete case, the transition probabilities are given by

$$\begin{aligned} \mathcal{P}(dx'' \mid x, x', u, \sigma) &\coloneqq \mathbb{P}_{X_{k+\eta_k}}(dx'' \mid X_k = x, X_{k-r} = x', U_k = u, \eta_k = \sigma) \\ \mathcal{O}(ds \mid x, x', u) &\coloneqq \mathbb{P}_{\eta_k}(ds \mid X_k = x, X_{k-r} = x', U_k = u) \\ \mathcal{G}(dy \mid x, x', u) &\coloneqq \mathbb{P}_{Y_k}(dy \mid X_k = x, X_{k-r} = x', U_k = u) \\ \pi_k(du \mid H_k^{\eta,Y,U}) &\coloneqq \mathbb{P}_{U_k}(du \mid H_k^{\eta,Y,U} = h_k^{\eta,Y,U}) \end{aligned} \tag{3.3}$$

for any $k \in \mathbb{N}$, $x, x', x'' \in \mathcal{X}$, $u \in \mathcal{U}$, $\sigma \in \mathbb{N}^*$. Note the support of the interdecision time η_k is now in $\mathbb{N}^*$.

There is an alternative formulation of this discrete-time controlled process that is commonly used in modern literature. In a particular case, this formulation is equivalent to the one given in Definition 3.1.1.

Definition 3.1.2 (General Discrete Decision Process - Transition)**.** *The tuple given by* $(\mathcal{X}, \mathcal{U}, \mathcal{Y}, \mathcal{P}, \mathcal{G}, \mathcal{O}, \pi)$ *is called* discrete decision process*.*

Several important cases of discrete decision processes in the field of control are now presented.

Partially Observable Semi-Markov Decision Process (PO-SMDP)

When the state transition probability and the observation transition probability are Markovian, in the sense

$$\mathcal{P}(dx'' \mid x, x', u, \sigma) = \mathcal{P}(dx'' \mid x, u, \sigma) \tag{3.4}$$

$$\mathcal{G}(dy \mid x, x', u) = \mathcal{G}(dy \mid x, u) \tag{3.5}$$

for all $x, x', x'' \in \mathcal{X}$, $u \in \mathcal{U}$ and $\sigma \in \mathbb{N}^*$, the process is called a *Partially Observable Semi-Markov Decision Process* (PO-SMDP).

Partially Observable Markov Decision Process (PO-MDP)

Again, suppose the state and observation transition probabilities are Markovian, if the interdecision time transition probability is degenerated to a constant value, *i.e.*

$$\mathcal{O}(ds \mid x, x', u) = \delta_{\{1\}}(ds) \tag{3.6}$$

Then the resulting process is called a *Partially Observable Markov Decision Process* (PO-MDP).

Semi-Markov Decision Process

Now suppose that the state is Markovian and the system is fully observable, *i.e.* the observation operator is the identity operator, $G = Id$. Equivalently, the observation transition probability is degenerated on the identity operator, *i.e.*

$$\mathcal{G}(dy \mid x, x', u) = \delta_{\{x\}}(dy) \tag{3.7}$$

Markov Decision Process (MDP)

Finally, the most important case coined *Markov Decision Process* (MDP) by the applied mathematician Richard Bellman in the 1950s is when the system is fully observable (3.7) and the interdecision time is degenerated to a constant value

(3.6). This is also called *Controlled Markov Process* (CMP) by Dynkin and Yushkevich 1979. The origin of those processes can be traced back to the work of Richard Bellman in the 1950's (R. Bellman 1957; R. E. Bellman 1957) and Ronald Howard in the 1960's (Howard 1960).

3.1.3 On the Equivalence of Formulations

Consequently, given a system of discrete-time stochastic recurrence equations given by Definition 3.1.1, one can extract transition probabilities as in (3.3) and obtain a decision process in the sense of Definition 3.1.2. Reciprocally, it can be questioned whether, given a specification of transition kernels as in Definition 3.1.2, a probability distribution $\tilde{\mathbb{P}}$ exists such that the transition probabilities are determined by the relations in (3.3).

The answer is affirmative and guaranteed under weak conditions by the Ionescu-Tulcea theorem (Neveu 1970; Loève 1977; Klenke 2007): given the transition probabilities aforementioned, there exists a probability distribution $\tilde{\mathbb{P}}$ such that the transition probabilities satisfy the relations in (3.3).

To go further, the question can be extended to the existence of a system of stochastic recurrence equations characterised by a state evolution operator $\tilde{F}$ and an observation operator $\tilde{G}$ such that $X_{k+1} = \tilde{F}(X_k, X_{k-r}, U_k, \epsilon_X^k)$ and $Y_{k+1} = \tilde{G}(X_k, X_{k-r}, U_k, \epsilon_Y^k)$ where ϵ_X^k and ϵ_Y^k are *i.i.d.* random variables[36] . As a result, it would be equivalent to specify either a recurrence equation or a set of transition probabilities. It happens that in the fully observable, Markovian case (MDP) the existence of the state evolution operator is guaranteed (Gihman and Skorohod 1979). Both ways have their advantages and drawbacks, in terms of interpretability and practicality (Onésimo Hernández-Lerma and Lasserre 1996).

3.1.4 Discrete Control Problem

Here we present the essential elements of the optimal control problem in discrete time. A detailed treatment of the general continuous-time case is given in Chapter 2.

Optimal Control Problem (Discrete Time)

The optimal control problem in discrete time follows a similar structure to the continuous-time case, and can be obtained by approximating the time integral by a Riemann sum, (see Remark 2.4.3). The policy $\pi = (\pi_k)_{k\in\mathbb{N}}$ must *minimise a cost function* defined over a finite ($K < +\infty$) or infinite horizon ($K = +\infty$).

[36]The existence of such a system is sufficient to ensure the existence of a probability distribution $\tilde{\mathbb{P}}$ verifying the desired properties.

The random total cost from step k in discrete time is defined as

$$Z\left(k, \mathbb{P}_{X_{[\![-r,0]\!]}}, \pi\right) = \sum_{i=k}^{K} \gamma^i c\left(X_i, \pi_i\right) \tag{3.8}$$

for any discrete-time step $k \in [\![0, K]\!]$, where $\gamma \in [0, 1]$ is a discount factor, and $c : \mathcal{X} \times \mathcal{U} \to \mathbb{R}$ is the instantaneous cost function. From the initial step, the total cost is denoted by $Z(\mathbb{P}_{X_{[\![-r,0]\!]}}, \pi) = Z(0, \mathbb{P}_{X_{[\![-r,0]\!]}}, \pi)$.

This quantity is a random variable, and the expectation of the random total cost is the objective functional.

The typical objective functional in discrete time is then given by

$$J\left(k, \mathbb{P}_{X_{[\![-r,0]\!]}}, \pi\right) = \mathbb{E}\left[\sum_{i=k}^{K} \gamma^i c\left(X_i, \pi_i\right)\right] = \mathbb{E}\left[Z\left(k, \mathbb{P}_{X_{[\![-r,0]\!]}}, \pi\right)\right] \tag{3.9}$$

Remark 3.1.1 (Series Convergence)**.** *The control problem is also defined for an infinite horizon, i.e., when $K = +\infty$. In this case, the condition $\gamma < 1$ is sufficient to ensure convergence of the sum when the cost function is bounded.*

The optimal objective functional in discrete time is then defined as

$$J^*\left(k, \mathbb{P}_{X_{[\![-r,0]\!]}}\right) = \inf_{\pi \in \mathscr{A}_\Pi} J\left(k, \mathbb{P}_{X_{[\![-r,0]\!]}}, \pi\right) \tag{3.10}$$

for any $k \in [\![0, K]\!]$.

Thus, the optimal control problem becomes

$$J^*\left(\mathbb{P}_{X_{[\![-r,0]\!]}}\right) = J^*\left(0, \mathbb{P}_{X_{[\![-r,0]\!]}}\right) = \inf_{\pi \in \mathscr{A}_\Pi} \mathbb{E}\left[\sum_{i=0}^{K} \gamma^i c\left(X_i, \pi_i\right)\right] \tag{3.11}$$

Exactly as in the continuous-time case, when the initial condition is fixed, the optimal objective functional is given by

$$J^*\left(k, x_{[\![-r,0]\!]}\right) = J^*\left(k, \delta_{x_{[\![-r,0]\!]}}\right) = \inf_{\pi \in \mathscr{A}_\Pi} \mathbb{E}\left[\sum_{i=k}^{K} \gamma^i c\left(X_i, \pi_i\right) \mid X_{[\![-r,0]\!]} = x_{[\![-r,0]\!]}\right] \tag{3.12}$$

The optimal objective from the initial step $k = 0$ is denoted $J^*\left(x_{[\![-r,0]\!]}\right) = J^*\left(0, x_{[\![-r,0]\!]}\right)$.

Similarly, for the Markovian case, the optimal objective functional is given by

$$J^*\left(k, x\right) = J^*\left(k, \delta_x\right) = \inf_{\pi \in \mathscr{A}_\Pi} \mathbb{E}\left[\sum_{i=k}^{K} \gamma^i c\left(X_i, \pi_i\right) \mid X_0 = x\right] \tag{3.13}$$

Again, $J^*\left(x\right) = J^*\left(0, x\right)$ and this quantity is commonly called *optimal value function*.

Another fundamental concept which remains to be introduced is the optimal expected total cost when both the initial condition and the control are fixed. This mapping is here called *optimal Q-function* and defined in the Markovian case as

$$Q^{*}\left(x,u\right)=\inf_{\pi\in\mathscr{A}_{\Pi}}\mathbb{E}\left[\sum_{i=0}^{K}\gamma^{i}c\left(X_{i},\pi_{i}\right)\mid X_{0}=x,U_{0}=u\right] \tag{3.14}$$

Given any policy $\pi\in\mathscr{A}_{\Pi}$ and for $k=0$, the *value function* $x\mapsto J(x,\pi)$ and the *Q-function* $(x,u)\mapsto Q(x,u,\pi)$ are defined as the expectation on which the infimum is taken in the optimal objective functional (Eq. (3.13)) and the optimal Q-function (Eq. (3.14)), respectively.

Consequently,

$$J^{*}(x)=\inf_{\pi\in\mathscr{A}_{\Pi}}J(x,\pi) \tag{3.15}$$

and

$$Q^{*}(x,u)=\inf_{\pi\in\mathscr{A}_{\Pi}}Q(x,u,\pi) \tag{3.16}$$

for any $x\in\mathcal{X}$ and $u\in\mathcal{U}$.

Maximum Entropy Control Problem

The Maximum Entropy Control Problem in the discrete-time case is characterised by the objective functional

$$J_{\mathcal{H}}\left(k,\mathbb{P}_{X_{[\![-r,0]\!]}},\pi\right)=\mathbb{E}\left[\sum_{i=k}^{K}\gamma^{i}c\left(X_{i},\pi_{i}\right)-\alpha^{\mathcal{H}}\mathcal{H}\left[\pi_{i}\right]\right] \tag{3.17}$$

This objective is referred to as the *soft* objective functional in the RL literature Haarnoja, Tang, et al. 2017. The optimal soft objective functional follows the same notation rules as in the standard case (a.k.a. hard objective functional). An important particular case is the optimal soft value function, defined as

$$J_{\mathcal{H}}^{*}\left(x\right)=\inf_{\pi\in\mathscr{A}_{\Pi}}\mathbb{E}\left[\sum_{i=0}^{K}\gamma^{i}c\left(X_{i},\pi_{i}\right)-\alpha^{\mathcal{H}}\mathcal{H}\left[\pi_{i}\right]\mid X_{0}=x\right] \tag{3.18}$$

Thus, the optimal soft Q-function is defined as

$$Q_{\mathcal{H}}^{*}\left(x,u\right)=\inf_{\pi\in\mathscr{A}_{\Pi}}\mathbb{E}\left[\sum_{i=0}^{K}\gamma^{i}c\left(X_{i},\pi_{i}\right)-\alpha^{\mathcal{H}}\mathcal{H}\left[\pi_{i}\right]\mid X_{0}=x,U_{0}=u\right] \tag{3.19}$$

In discrete time, the dynamic programming principle is referred to as the *Bellman equation*. Again, in this case the dynamics are supposed to be Markovian thus $\mathbb{P}_{[\![k-r,k]\!]}=\mathbb{P}_{X_{k}}$ for any $k\in\mathbb{N}$, where $\mathbb{P}_{[\![k-r,k]\!]}$ is the distribution of the history process $X_{[\![k-r,k]\!]}=(X_{k-r},\ldots,X_{k})$.

Theorem 3.1.1 (Bellman Equation)**.** *The optimal objective functional satisfies the Dynamic Programming Principle*

$$J^*\left(k, \mathbb{P}_{X_k}\right) = \inf_{\pi \in \mathscr{A}_\Pi^M} \mathbb{E}\left[\sum_{i=k}^{k+j} \gamma^i c\left(X_k, \pi_k\right) + J^*\left(k+j+1, \mathbb{P}_{X_{k+j+1}}\right)\right] \tag{3.20}$$

for any $k \in [\![0, K]\!]$*, where* $j \geq 0$*.*

Moreover, a crucial functional equation can be obtained for the optimal objective functional when $K = +\infty$*, known as the* Bellman equation

$$J^*\left(k, \mathbb{P}_{X_k}\right) = \inf_{\pi \in \mathscr{A}_\Pi^M} \mathbb{E}\left[\sum_{i=k}^{k+j} \gamma^i c\left(X_k, \pi_k\right) + \gamma^{j+1} J^*\left(k, \mathbb{P}_{X_{k+j+1}}\right)\right] \tag{3.21}$$

Thus for $j = 0$*, the Bellman equation is*

$$J^*\left(k, \mathbb{P}_{X_k}\right) = \inf_{\pi \in \mathscr{A}_\Pi^M} \mathbb{E}\left[\gamma^k c\left(X_k, \pi_k\right) + \gamma J^*\left(k, \mathbb{P}_{X_{k+1}}\right)\right] \tag{3.22}$$

and if $k = 0$*, the Bellman equation is*

$$J^*\left(\mathbb{P}_{X_0}\right) = \inf_{\pi \in \mathscr{A}_\Pi^M} \mathbb{E}\left[c\left(X_0, \pi_0\right) + \gamma J^*\left(\mathbb{P}_{X_1}\right)\right] \tag{3.23}$$

These latter equations are fundamental in the dynamic programming theory and define an infinite dimensional (function space) fixed-point problem (this fixed-point property will be used in the next chapters to build learning-based control algorithms). Hence, the DPP is sometimes called the *functional equation* in R. E. Bellman 1957.

Many learning algorithms are based on the Bellman equation. The equations (3.21) and (3.23) are the basis of learning based control theory and algorithms (Sutton and Barto 2018; A. Agarwal, Jiang, and Kakade 2019; Bensoussan, Y. Li, et al. 2020; Meyn 2022). Chapter 6 discusses an extension of the Bellman operator to the random total cost defined in Eq. (3.8).

The next section will focus on the learning theory before presenting the learning based control approaches.

3.2 Learning Theory, Generalisation and Complexity Measures

3.2.1 Statistical Learning

Notations and Definition

Basically, three elements are essential in learning theory[37] (Shalev-Shwartz and Ben-David 2014): a dataset $\bar{\mathcal{D}}$, a model $\bar{f}$, and a learning task $\bar{\ell}$.[38]First, the data are modelled by some random variable $\bar{Z}$ with distribution $\mathbb{P}_{\bar{Z}}$ with values in a domain $\bar{\mathcal{Z}}$. The dataset $\bar{\mathcal{D}}$ usually contains $m_{\bar{\mathcal{D}}}$ identically and independently distributed (i.i.d.) samples[39] $\bar{Z}$, *i.e.* $\bar{\mathcal{D}} = \left(\bar{Z}_1, \dots, \bar{Z}_{m_{\bar{\mathcal{D}}}}\right)$ where $\bar{Z}_i \sim \mathbb{P}_{\bar{Z}}$. Thus, $\mathbb{P}_{\mathcal{D}}$ is the joint distribution of the elements of the dataset $\bar{\mathcal{D}}$. Second, the model is an element $\bar{f}$ of the hypothesis class $\mathcal{F}$ which is often infinite dimensional. Third, the learning task is defined by a loss function $\bar{\ell}$ which quantifies the task error the model $\bar{f}$ makes for a given observation $\bar{Z}$.

Definition 3.2.1 (Loss function)**.** *A loss function is a mapping $\bar{\ell} : \mathcal{F} \times \bar{\mathcal{Z}} \to \mathbb{R}_+$ that maps a hypothesis and an observation to a positive real number. Sometimes, the loss function is referred to as a* learning task*.*

Example 3.2.1 (Quadratic loss in Supervised Learning)**.** *Regarding* supervised learning *one has $\bar{\mathcal{Z}} = \bar{\mathcal{X}} \times \bar{\mathcal{Y}}$, the observation and label spaces and possibly $\mathcal{F} = \left(\bar{f}_\theta\right)_{\theta \in \Theta}$ with $\Theta \subset \mathbb{R}^d_\theta$ some parameters space of dimension d_θ associated to the quadratic loss $\bar{\ell}(\bar{f}_\theta, (x, y)) = (\bar{f}_\theta(x) - y)^2$ for any $(x, y) \in \bar{\mathcal{X}} \times \bar{\mathcal{Y}}$.*

Generalisation

As stated in Mohri, Rostamizadeh, and Talwalkar 2018, "Machine Learning is fundamentally about generalization". Roughly, this can be understood as the ability of a model (or hypothesis) $\bar{f}$ to perform well on unseen data or data not used to estimate the model. In the standard supervised learning setting, the generalisation error is defined on the set $\mathcal{F}$ of all possible models, the so-called *hypothesis set* (this set is traditionally denoted by $\mathcal{H}$ but this symbol is kept for the entropy).

Definition 3.2.2 (Generalisation Error)**.** *The generalisation error $\bar{J}$, also called the* risk*, of a hypothesis $\bar{f} \in \mathcal{F}$ is defined as*

$$\bar{J}(\mathbb{P}_{\bar{Z}}, \bar{f}) = \mathbb{E}_{\mathbb{P}_{\bar{Z}}}[\bar{\ell}(\bar{f}, \bar{Z})] \tag{3.24}$$

where $\mathbb{E}^{\mathbb{P}_{\bar{z}}}$ denotes the expectation w.r.t. the distribution $\mathbb{P}_{\bar{Z}}$.

The optimal generalisation error is defined as

$$\inf_{\bar{f}' \in \mathcal{F}} \bar{J}(\mathbb{P}_{\bar{Z}}, \bar{f}') \tag{3.25}$$

[37]Here the classical, also known as *frequentist*, approach is presented. Details and comparisons between statistical approaches are given in Section 3.2.3.

[38]In this section, notations from statistical learning theory are introduced with a bar over the symbols to distinguish them from the RL ones.

[39]This hypothesis is sometimes relaxed in the literature (Steinwart, Hush, and Scovel 2009).

for a fixed data distribution $\mathbb{P}_{\bar{Z}}$.

If the optimal generalisation error (3.25) *is attained by a model* $\bar{f}^*$*, it is called the* oracle model*.*

In other words, *learning* is the process of finding a hypothesis $\bar{f} \in \mathcal{F}$ from a dataset $\bar{\mathcal{D}}$ that minimises the generalisation error $\bar{J}$, hence being by definition a generalisation problem.

Example 3.2.2 (Mean square error (MSE))**.** *Regarding the supervised learning setting with quadratic loss, one obtains the commonly called* mean square error $\bar{J}(\mathbb{P}_{\bar{X},\bar{Y}}, \bar{f}) = \mathbb{E}_{\mathbb{P}_{\bar{X},\bar{Y}}}[(\bar{f}(\bar{X}) - \bar{Y})^2]$ *where* $\bar{Z} = (\bar{X}, \bar{Y})$ *has been chosen.*

Learning-based Control case

In the case of reinforcement learning, one can be interested in the generalisation error in terms of regret (Y. Duan, Jin, and Z. Li 2021) for a given policy $\bar{f} = \bar{\pi} \in \mathcal{F} = \Pi$. There are several definitions of this concept in the literature, in a general form it can be defined as

Definition 3.2.3 (Regret)**.** *The regret for a given algorithm* $\mathcal{A}$ *is defined as*

$$\textit{Regret}\,(\bar{\pi}) = \bar{J}^{\bar{\pi}} - \min_{\bar{\pi} \in \mathcal{F}} \bar{J}\,(\bar{\pi}) = \bar{J}\,(\bar{\pi}) - \bar{J}\,(\bar{\pi}^*) \tag{3.26}$$

Hence the regret can be generally understood as the spread between the performance when taking optimal decisions and the target policy performance.

In the next section, a rigorous answer to the fundamental question of learning theory, pioneered by Leslie Valiant (Valiant 1984), is presented.

3.2.2 Probably Approximately Correct Learning

The principal objective of *statistical learning* is to provide bounds on the generalisation error, so-called *generalisation bounds*. In what follows, it is assumed that an algorithm $\mathcal{A}$ returns a hypothesis $\bar{f} \in \mathcal{F}$ from a dataset $\bar{\mathcal{D}}$. Note the dataset $\bar{\mathcal{D}}$ is random and the algorithm $\mathcal{A}$ is a randomised algorithm.

As the hypothesis set $\mathcal{F}$ typically used in machine learning is infinite, a practical way to quantify the generalisation ability of such a set must be found. This is done by introducing *complexity measures*, which enable the derivation of generalisation bounds.

Definition 3.2.4 (Complexity measure)**.** *A complexity measure is a mapping* $\mathcal{M} : \mathcal{F} \to \mathbb{R}_+$ *that maps a hypothesis to a positive real number.*

According to Neyshabur, Bhojanapalli, et al. 2017 from which this formalism is inspired, an appropriate complexity measure satisfies several properties. In the case of parametric models $\bar{f}_\theta \in \mathcal{F}(\Theta)$ with $\theta \in \Theta \subset \mathbb{R}^{d_\theta}$, it should

increase with the dimension d_θ of the parameter space Θ as well as being able to identify when the dataset $\bar{\mathcal{D}}$ contains totally random, spurious, or adversarial data. Moreover, it can distinguish between models learnt with zero training errors and the same dataset $\bar{\mathcal{D}}$ but different final local optima θ^* obtained, for instance, through a randomised optimisation.

A simple example of complexity measure for a parametric hypothesis π_θ is the feature dimension of the parameter $\mathcal{M}(\bar{f}_\theta, \bar{\mathcal{D}}) = d_\theta$ or its ℓ_2-norm $\mathcal{M}(\bar{f}_\theta, \bar{\mathcal{D}}) = \|\theta\|_2$. More generally, complexity measures can also be defined for a whole hypothesis set $\mathcal{F}$ as $\mathcal{M}(\mathcal{F}, \bar{\mathcal{D}})$. Various examples of such complexity measures exist, such as the fundamental *Vapnik-Chervonenkis (VC) dimension* in binary classification and the *Rademacher complexity* $\mathcal{M}(\bar{f}, \bar{\mathcal{D}}) = \mathcal{R}ad(\mathcal{F}) = \mathcal{R}ad(\bar{f})$ for any $\bar{f} \in \mathcal{F}$ that measures the degree to which a hypothesis set $\mathcal{F}$ correlates with random noise, in a larger scope than classification (Mohri, Rostamizadeh, and Talwalkar 2018).

Given an algorithm $\mathcal{A}$, one can wonder if it is able to return a hypothesis $\bar{f} \in \mathcal{F}$ from a dataset $\bar{\mathcal{D}}$ of size $m_{\bar{\mathcal{D}}}$ such that the generalisation error $J(\bar{f})$ is close to the optimal generalisation error $\min_{\bar{f}' \in \mathcal{F}} \bar{J}(\bar{f}')$. This is the goal of the *Probably Approximately Correct (PAC) learning* framework (Valiant 1984) which is stated below for the sake of exhaustiveness.

Definition 3.2.5 (PAC Learning)**.** *A hypothesis set $\mathcal{F}$ is PAC learnable if there exists a learning algorithm $\mathcal{A}$ such that for any $\bar{\eta}, \bar{\delta} > 0$ and any distribution $\mathbb{P}_{\bar{Z}}$ over the data, there exists a sample size $m_{\mathcal{F},\bar{\eta},\bar{\delta}}$ such that running $\mathcal{A}$ with a given random sample $\bar{\mathcal{D}}$ of size $m_{\bar{\mathcal{D}}} \geq m_{\mathcal{F},\bar{\eta},\bar{\delta}}$, the algorithm $\mathcal{A}$ returns a hypothesis $\bar{f} \in \mathcal{F}$ such that*

$$\mathbb{P}_{\bar{Z}}\left[\bar{J}\left(\bar{f}\right) - \min_{\bar{f}' \in \mathcal{F}} \bar{J}\left(\bar{f}'\right) \leq \bar{\eta}\right] \geq 1 - \bar{\delta} \tag{3.27}$$

One can note how PAC learning relies on *sample-complexity* through the number of samples $m_{\mathcal{F},\bar{\eta},\bar{\delta}}$.

3.2.3 Estimation

The field of statistics is vast and several important approaches have been developed creating a whole range of subfields. In this part, the problem of learning, which is an instance of statistical *estimation* theory, is thus categorised. Throughout the different chapters of this thesis, different kinds of estimation approaches belonging to different statistical paradigms are used. Hence, it is appropriate to give concise explanations of these approaches here.

Let $\mathcal{F}$ be the space mentioned in Section 3.2.1 that is assumed to be a submanifold of an infinite dimensional manifold (such that a notion of dimension on this space ($\dim(\mathcal{F})$) can be defined). In practice, the hypothesis space $\mathcal{F}$ represents the class of candidate models that can be learnt from the data, and the goal is to find the best procedure $\mathcal{A}$ to perform the learning task, *i.e.* to

find the minimiser of the generalisation error $\bar{J}$ in $\mathcal{F}$ or at least finding a model close to this minimiser.

Multiple categories of estimation, depending on the nature of the hypothesis space $\mathcal{F}$, are commonly defined.

Parametric Estimation

In the case where the hypothesis space $\mathcal{F}$ is finite dimensional ($\dim(\mathcal{F}) = d_\theta < \infty$), the estimation problem is called *parametric estimation*. The usual case is when $\mathcal{F} = \Theta \subset \mathbb{R}^{d_\theta}$. Note that any family indexed by a finite dimensional set of parameters (e.g. $\left(\bar{f}_\theta\right)_{\theta\in\Theta}$) can be seen as a parametric family. The tradition of considering the problem of statistical estimation as the estimation of a finite number of parameters goes back to Sir Ronald Aylmer Fisher.

Non-parametric Estimation

On the other hand, parametric models sometimes provide inaccurate representations of the underlying statistical structure (Tsybakov 2008). Thus, it can be more appropriate to consider the estimation on a functional space directly ($\dim(\mathcal{F}) = \infty$). In that case, the estimation problem is called *non-parametric estimation*.

Frequentist Statistics

In the above presentation, the loss function considered in Definition 3.2.1 is a function of two elements: a model (called hypothesis) $\bar{f}$ and a data point $\bar{z}$. Hence, the loss $\bar{\ell}(\bar{f}, \bar{z})$ is parameterised by the input data $\bar{z}$. Consequently, the comparison between two models $\bar{f}$ and $\bar{f}' \in \mathcal{F}$ is made difficult since no ordering, even partial, is defined. By averaging the loss over the data distribution $\mathbb{P}_{\bar{Z}}$, a partial ordering is defined: this is the frequentist approach.

From Definition 3.2.2, the generalisation error is the expectation of the loss function w.r.t. the data distribution $\mathbb{P}_{\bar{Z}}$. The intuition behind this error measure is the following. Suppose that some algorithm $\mathcal{A}$ returns a hypothesis $\bar{f}$ from a dataset $\bar{\mathcal{D}} = \left(\bar{Z}_1, \ldots, \bar{Z}_{m_{\bar{\mathcal{D}}}}\right)$. The generalisation error averages (integrates) over all possible points that do not necessarily belong to the dataset $\bar{\mathcal{D}}$, such that the learning task (the error metric) depends only on the candidate model $\bar{f}$ and the data distribution $\mathbb{P}_{\bar{Z}}$. Thus, this approach is called *classical, or frequentist statistical* inference. The term "frequentist" appropriately stands for the fact the unknown data distribution $\mathbb{P}_{\bar{Z}}$ can be approximated with its empirical counterpart (called empirical distribution) denoted $\widehat{\mathbb{P}}_{\bar{Z}} = \frac{1}{m_{\bar{\mathcal{D}}}} \sum_{i=1}^{m_{\bar{\mathcal{D}}}} \delta_{\bar{Z}_i}$ representing the distribution obtained from the frequencies of the data in $\bar{\mathcal{D}}$.
Hence, the empirical risk $\widehat{\bar{J}}_{\bar{\mathcal{D}}}(\bar{f}) = \frac{1}{m_{\bar{\mathcal{D}}}} \sum_{i=1}^{m_{\bar{\mathcal{D}}}} \bar{\ell}(\bar{f}, \bar{Z}_i)$ approximates the generalisation error.

Definition 3.2.6 (Empirical Generalisation Error)**.** *The empirical generalisation error is defined as*

$$\widehat{J}_{\bar{\mathcal{D}}}(\bar{f}) = \frac{1}{m_{\bar{\mathcal{D}}}} \sum_{i=1}^{m_{\bar{\mathcal{D}}}} \bar{\ell}(\bar{f}, \bar{Z}_i) \tag{3.28}$$

In this way, the frequentist school defines its elementary notion of optimality: minimising the empirical risk. PAC learning described in Section 3.2.2 is a frequentist approach. This approach has several drawbacks, two of which are particularly important.

First, the classical school supposes the distribution $\mathbb{P}_{\bar{Z}}$ is somehow fixed while the associated statistical experiments generating the data are repeatable under the same conditions. This setting is difficult to verify in practice.

Second, the empirical distribution requires a number of samples that grows with the dimension of the input data.[40] Consequently, the frequentist approach requires a large number of samples to be efficient.

Other arguments against the frequentist approach are given in the landmark book of Robert 2007 on the Bayesian view of statistics. Methodologies of both schools are used in the work presented in this thesis. The reader interested in the frequentist approach is referred to Barra 1971; M. Hoffman 2015.

Bayesian Statistics

The central idea of Bayesian statistics is to consider the unknown quantity of interest (θ^* in the parametric case or $\bar{f}^*$ in non-parametric case) as a random variable[41]: the hypothesis space $\mathcal{F}$ is endowed with a prior distribution ($\mathbb{P}_{\bar{f}^*}$ in the non-parametric case or $\mathbb{P}_{\theta^*}$ in the parametric case). This prior distribution represents what is known about the hypothesis before observing the data.

This randomness shall be understood as the decision maker or agent belief in the true value of the optimal model. Thus, a distribution $\mathbb{P}_{\bar{f}^*}$ is defined over the hypothesis space $\mathcal{F}$. If the hypothesis space is finite dimensional $\mathcal{F} = \Theta \subset \mathbb{R}^{d_\theta}$, the prior distribution $\mathbb{P}_{\theta^*}$ is defined over the parameter space Θ. The choice of a prior distribution is not trivial, and a considerable part of the Bayesian literature is dedicated to this topic.

Note that the above definition refers to the PAC-Bayesian theory, while the classic Bayesian theory assigns a prior distribution to the data distribution $\mathbb{P}_{\bar{Z}}$ itself. PAC-Bayesian algorithms are motivated by a desire to provide an informative prior encoding information about the expected experimental setting but still having PAC performance guarantees over all *i.i.d.* settings.

[40] This is known as the *curse of dimensionality* (Bach 2024).

[41] In Bayesian Statistics, the unknown target $\bar{f}^*$ does not necessarily vary, thus the term "random" may not be very appropriate. In probability theory, a random variable is defined as a measurable function from a probability space to a measurable space. Hence, the possibility to assign a value $\mathbb{P}_{\bar{f}^*}(B_{\mathcal{F}})$ to a region $B_{\mathcal{F}} \subset \mathcal{F}$ of the hypothesis space which describes how likely the unknown model $\bar{f}^*$ belongs to this region is the essential concept in Bayesian statistics.

The reader interested in the Bayesian approach is referred to Robert 2007 and Rousseau 2009.

Estimators

The notion of learning algorithm $\mathcal{A}$ is historically associated with the statistical learning literature, while the statisticians prefer the closely related notion of *estimator*. A bit more formally, a learning algorithm $\mathcal{A}$ is a mapping that assigns a hypothesis $\bar{f} \in \mathcal{F}$ to a dataset $\bar{\mathcal{D}} \in \bar{\mathcal{Z}}^{m_{\bar{\mathcal{D}}}}$. In fact, the term *estimator* generalises this concept being a mapping that assigns any object from a dataset.

Definition 3.2.7. *An estimator, or a statistic, is a (measurable[42]) function of a dataset $\bar{\mathcal{D}}$.*

When the estimation target is an unknown function $\bar{f}^* \in \mathcal{F}$, the estimator is denoted $\hat{\bar{f}}$, and when the target is a parameter $\theta^* \in \mathcal{F} = \Theta$, the estimator is denoted $\widehat{\theta}$. In those cases, where the estimator returns an element of the hypothesis space $\mathcal{F}$, the algorithm $\mathcal{A}$ and the estimator are equivalent. Note that the estimators are functions of the data, thus $\hat{\bar{f}} = \hat{\bar{f}}(\bar{\mathcal{D}})$ and $\widehat{\theta} = \widehat{\theta}(\bar{\mathcal{D}})$.

Given a class of estimators, how to choose the optimal one, and what is a notion of optimality? A basic tool is the notion of risk or *loss function* as defined in Section 3.2.1 that allows comparing the performances of different estimators.

Notably, an important procedure in the Machine Learning literature is the (stochastic) gradient descent algorithm Mandt, M. D. Hoffman, and Blei 2017 to find the optimal estimator if the risk function is differentiable such that the gradient of this empirical generalisation error can be computed. In the work presented in this thesis, the gradient descent algorithm used on parametric models is the Adam algorithm Kingma and Ba 2015.

3.2.4 Decision Theory

Statistical decision theory is concerned with the problem of making decisions in the presence of statistical knowledge. Classical statistics are directed towards the use of sample information (from the data gathered) in making inference about the unknown state of nature or system, represented by the parameter θ^*. In decision theory, this information is coupled with the system features in order to make the best decision. In addition to the extracted information, a measure of the decision consequences is performed through the use of a *loss* function. The choice of function among a class of hypothesis class $\mathcal{F}$ is the decision in the statistical learning theory. The incorporation of the loss function is due to Abraham Wald. In economics, the loss function is called the *utility function*.

[42] This ensures a probability measure can assign a probability to the estimator potential values. Estimators are then random variables, they may have a mean and a variance.

The other kind of information that is not extracted from the statistical experiment is called *prior information*. This information arises from other sources such as knowledge built on past or similar experiments. The approach of statistics which seeks to use prior information and is termed as Bayesian analysis. Bayesian analysis and Decision Theory are naturally linked, partly because of their common use of information that does not directly result from experimental trials and partly because of theoretical ties. Note also, there is also non-Bayesian Decision Theory and a statistical Bayesian point of view that is not necessarily linked to Decision Theory. The book Berger 1985 is a classic reference on the subject.

The next part is devoted to the application of learning theory to control problems.

3.3 Learning-based Control

Historically, the first application of learning theory to control problems can be traced back at least to the *adaptive control* theory (Åström and Wittenmark 1989) which concerns the design of controllers for controlled systems that depend on unknown quantities such as the dynamics f, the disturbances ϵ_X and ϵ_Y or any other object in the dynamics presented in (2.1)-(2.2). This definition can be extended to the case of unknown cost functions c or anything that is unknown to the decision maker.

Rudolf Kalman (Kálmán 1958), was one of the first to propose a kind of learning-based control setting called "Self-optimising Control System". Indeed, he was already interested in building a "machine" that "adjusts itself automatically to control an arbitrary dynamics process", paving the way to the learning-based or machine-learning control field. From his own words, "this machine represents a new concept in the development of control systems". Once again Bellman pioneered the adaptive control theory in the 1960s with other great researchers, an extended bibliography of Adaptive Control early days can be found in Åström and Wittenmark 1989, p. 38 and subsequent.

3.3.1 Adapting Learning Theory to Control

Basically, Learning-based Control brings together the fields of Control Theory presented in Chapters 2 and 3 and Learning theory presented in Section 3.2.

Several categories of learning-based control can be distinguished depending on the unknown target object to be learnt. Taking back the notation of Section 3.2.1, the target object $\bar{f} \in \mathcal{F}$ can now represent any central object of the control problem that have been presented.

Strong Realisability Assumption

To simplify the exposition, all the target objects such as the policy space or the operator spaces are supposed to be contained in the hypothesis spaces that are considered. This means that quantities such as the optimal policy π^*, the optimal value function J^* or the true dynamics f are in the hypothesis spaces $\mathcal{F}$. Hence, those objects are learnable.

Data distribution

In learning-based control, a random data point $\bar{Z}$ may be a complete trajectory defined in Chapter 2 such as $\bar{Z} = H_k$ or $\bar{Z} = H_\infty$, an observed state $\bar{Z} = X_k$, control $\bar{Z} = U_k$ at a given time k or any pair or combination of them with other observed quantities (observations, inter-decision times, etc.).

Thus, the data distribution $\mathbb{P}_{\bar{Z}}$ is often the distribution of the finite or infinite observed trajectory. For instance, in the discrete case the data distribution may be $\mathbb{P}_{\bar{Z}} = \mathbb{P}_{H_k}$ or $\mathbb{P}_{\bar{Z}} = \mathbb{P}_{H_\infty}$, and a similar distribution is defined in the continuous case with the continuous time history.

Loss Function

A common natural choice for the loss function is given by $\bar{\ell}(\bar{f}, \bar{Z}) = \bar{\ell}(\pi, H_\infty) = \sum_{i=k}^{K} \gamma^i c\left(X_k, U_k\right)$ where the distribution of H_∞ depends, of course, on the policy π, time, and initial state distribution.

Dynamics Learning

In the case of dynamics learning, the target object is often the true dynamics, *i.e.* $\bar{f} = f \in \mathcal{F}$. or the transition kernel $\bar{f} = \mathcal{P} \in \mathcal{F}$. Non-parametric estimators of models are denoted with a hat, *e.g.* $\widehat{f}$ or $\widehat{\mathcal{P}}$ and belong also to the hypothesis space $\mathcal{F}$.

When the problem is parametric $(\mathcal{F} = \Theta)$, the associated estimators are denoted f_{θ^*} or $\mathcal{P}_{\theta^*}$ for $\theta^* \in \Theta$ the true parameter of the dynamics, and the estimator of the weights is denoted $\widehat{\theta}$.

The methods using dynamics estimation are called *model-based* methods because they use a "model" of the dynamics to make decisions.

Policy Learning

Regarding Policy Learning, the learning target can be the optimal policy, *i.e.* $\bar{f} = \pi^*$. If the problem is nonparametric $(\pi^* \in \mathcal{F} = \Pi)$, the corresponding estimator is denoted $\widehat{\pi}$. Similarly, in the parametric case, the optimal policy π_{θ^*} is associated with the true parameter $\theta^* \in \Theta$ and the estimator of the weights is denoted $\widehat{\theta}$.

Value Learning

Value Learning aims at approximating the objective (a.k.a. value) function $(k, x, \pi) \mapsto J(k, x, \pi)$. As above, the non-parametric estimator of the value function is denoted $\widehat{J}$ and the parametric estimator is denoted J_θ for $\theta \in \Theta$.

3.3.2 Reinforcement Learning

Reinforcement Learning is a very large field that has been developed in the last decades. The reader is referred to the comprehensive book of Sutton and Barto 2018 for a detailed introduction to the field.

Definition

Throughout this thesis, Reinforcement Learning is defined as the process of learning an optimal policy $\pi^* \in \mathscr{A}_\Pi$ from a decision performance feedback termed *reinforcement signal*. In the present context, the reinforcement signal transmitted to the decision maker (controller) in a state $x \in \mathcal{X}$, when a decision $u \in \mathcal{U}$ is taken, is given by the cost $c(x, u)$.

Policy Iteration

A fundamental two-steps procedure called *policy iteration* is performed to learn the optimal policy. Iteratively, the following two stages are performed in order to obtain a new policy $\pi' \in \mathscr{A}_\Pi$ that performs better than the previous policy $\pi \in \mathscr{A}_\Pi$.

- *Policy Evaluation*: The value function $J(\,\cdot\,, \pi)$ of the policy is approximated by an estimator $\widehat{J}(\,\cdot\,, \pi)$.
- *Policy Improvement*: The policy is updated such that the new policy is better than the previous one in terms of the objective function *i.e.* the new policy $\pi \in \mathscr{A}_\Pi$ is such that $\widehat{J}(\,\cdot\,, \pi') \leq \widehat{J}(\,\cdot\,, \pi)$

In some simple cases such as when the state and control spaces are finite (tabular), the policy iteration algorithm is guaranteed to converge to the optimal policy. Those cases rather belong to the field of Dynamic Programming. Thus, they are not considered as proper RL settings.

In many other cases, neither the evaluation nor the improvement steps are performed exactly. The methods thus belong to the case of Dynamic Programming with Function Approximation. The policy evaluation is approximated by the value function estimation, and the policy improvement is rarely possible in a closed form, notably in the case of continuous control problems. Hence, Reinforcement Learning is defined as the application of the policy iteration procedure, based on reinforcement signals, with learning algorithms to approximate the value function and the policy.

Value-based Methods

Value-based methods are a class of Reinforcement Learning methods that aim at learning the value function or the Q-function of the (possibly optimal) policy.

Several ways exist to learn the value function in a supervised learning fashion. The most common approaches are the Temporal Difference (TD) methods (Tsitsiklis and Van Roy 1997) and the Q-learning algorithm (Watkins and Dayan 1992; Tsitsiklis 1994; Melo 2001). Those methods supervise the learning of the (Q-)value function by using a target computed from some type of Bellman equation (Theorem 3.1.1).

Alternatively, the learning label can be an empirical estimate of the value function obtained from a Monte Carlo simulation (here an empirical distribution of the controlled trajectory is derived). However, this approach is not always feasible in practice due to the high variance of the Monte Carlo estimator and the high computational cost of the simulation (curse of the trajectory size dimensionality).

Actor-based Methods

In this class of methods, the principal idea is to learn the policy directly from the reinforcement signal. They are called actor-based because the policy is sometimes called the actor in the Reinforcement Learning literature.

A common approach is to use the policy gradient theorem to update the policy in the direction of the gradient of the objective function (R. J. Williams, Peng, and H. Li 1991; Sutton, McAllester, et al. 1999).

Another example of actor-based class of methods is the gradient-free *Policy Search* approach (Sigaud and Stulp 2019) where the reinforcement signal is collected to evaluate the policy performance.

Actor-Critic Methods

Actor-critic approaches (Konda and Tsitsiklis 1999) combine the two previous classes of methods. This kind of procedure reduces the variance and is appreciated for its computational congeniality, even though it introduces bias in the estimation (due to bootstrapping).

A critic is a (Q-)value function estimator that is used to construct a bootstrapped target of the cumulative cost by means of the Bellman equation (Theorem 3.1.1). For instance, the critic $\widehat{J}$ is used to construct estimator $\widehat{J}'$ of the (Q-)value of some state X_k at iteration $k \in \mathbb{N}$ which reads $\widehat{J}'(X_k) = c(X_k, U_k) + \gamma \widehat{J}(X_{k+1})$. In this case the estimator $\widehat{J}'$ is called a bootstrap estimator since it is a function of an estimator. Moreover, the critic is called as such because it evaluates the policy performance starting from the next state (at iteration $k+1$). This way, the estimator critics the decision taken at the current state (at iteration k).

Model-free vs. Model-based

One essential difference between the fields of Dynamic Programming and Reinforcement Learning is the access to the model of the dynamics f or equivalently, the transition kernel $\mathcal{P}$. Thus, the initial works in Reinforcement Learning were model-free, *i.e.* the dynamics are unknown and algorithms are based on the estimation of the value function or the policy from the reinforcement signal.

In the case of model-based Reinforcement Learning (Moerland et al. 2022), the dynamics are approximated by an estimator $\widehat{f}$ or $\widehat{\mathcal{P}}$. Then, a wide range of methods can be used to solve the control problem. Chapter 7 is essentially based on the ideas of a model-based Reinforcement Learning article.

Off-policy vs. On-policy

As mentioned for instance in Section 3.3.1, learning is based on some data distribution derived from the interaction of the agent with the environment (dynamic system). Suppose that the actual control policy $\pi \in \mathscr{A}_\Pi$ is fixed. If the data distribution used for learning is independent of the policy π, then the learning algorithm is said to be *off-policy*. Otherwise, the learning algorithm is said to be *on-policy*.

Several empirical advantages and drawbacks are associated with each type of learning. Off-policy learning is often more efficient in terms of sample complexity, but it may suffer from high variance and instability. On-policy learning is more stable but may require more samples to converge.

3.3.3 Learning-based Model Predictive Control

Model Predictive Control

As stated in the introduction of the thesis (see Section 1.3.2), Model Predictive Control (MPC) (Grüne and Pannek 2011) is a control strategy that combines two main ingredients: a model of the system state dynamics $\widehat{\mathcal{P}}$ and an optimisation problem.

For each decision time $k \in \mathbb{N}$, the MPC approach consists of solving a finite-horizon optimal control problem. Formally it defines the following policy

$$\pi^{\mathsf{MPC}}(x) = u_0^* \tag{3.29}$$

$$s.t. \quad (u_0^*, \ldots, u_{K^{\mathsf{MPC}}}^*) = \underset{(u_0,\ldots,u_{K^{\mathsf{MPC}}}) \in \mathcal{U}^{K^{\mathsf{MPC}}+1}}{\arg\min} \mathbb{E}\left[\sum_{k=0}^{K^{\mathsf{MPC}}} c\left(\widehat{X}_k, u_k\right) \mid \widehat{X}_0 = x\right] \tag{3.30}$$

where $K^{\mathsf{MPC}} \leq K$ is the MPC planning horizon, $x \in \mathcal{X}$ is the current state and $(\widehat{X}_k)_{k\in[0,K^{\mathsf{MPC}}]}$ is the state trajectory when the state transition probability is given by the model $\widehat{\mathcal{P}}$.

The policy obtained with MPC on $\widehat{\mathcal{P}}$ is denoted by π^{MPC}. The history process under π^{MPC} is denoted by $H^{\text{MPC}} = (H_k^{\text{MPC}})_{k\in\mathbb{N}}$, it is an approximation of the optimal history process $(H_k^*)_{k\in\mathbb{N}}$ and the random variable H_K^{MPC} is an approximation of the optimal trajectory H_K^*. The objective function under π^{MPC} is denoted by J^{MPC}.

When the system is partially observed, the MPC policy is computed using a filter (see Definition 2.2.11 and Remark 2.2.17).

Fewer works have been done on the MPC with partially observed systems, the article Copp and Hespanha 2017 and the review Findeisen et al. 2003 are good references on this topic.

Model Learning

In Learning-based Model Predictive Control, the model of the system dynamics is learnt from data. This means that the model $\widehat{\mathcal{P}}$ is an estimator of the true dynamics $\mathcal{P}$. Any approach from Section 3.2.3 can be used to learn the model.

This approach opposes the physics-based model predictive control where the model is derived from the physics of the system.

Cross-Entropy Method

In this work, the MPC procedure is performed with the *iCEM* algorithm, an improved version of the *Cross Entropy Method (CEM)* (Rubinstein and Kroese 2004; Pinneri et al. 2021), a zeroth order optimisation algorithm based on Monte Carlo estimation.

Now concrete examples of dynamical systems are presented. They range from standard models used in control and dynamical system theory to more complex models used in the Flow Control literature.

3.4 Example of Dynamical Systems as Discrete Decision Processes

3.4.1 On the Spatial Discretisation

The potentially infinite dimensional function spaces are discretised such that $\mathcal{X} \simeq \mathbb{R}^{d_X}$, $\mathcal{Y} \simeq \mathbb{R}^{d_Y}$ and $\mathcal{U} \simeq \mathbb{R}^{d_U}$. Any function of some space is represented by a finite-dimensional vector in the corresponding space (the finite approximation vector should represent a function by containing a rich enough collection of its images). The simulation and numerical approximation is done according to the standard scheme presented in Section 2.4. Such discretisation is discussed and applied in several Reinforcement Learning works (e.g. Pan et al. 2018; Bucci et al. 2019; Tallec, Blier, and Ollivier 2019).

All environments start from a neighbourhood of some reference state $x_e \in \mathcal{X}$ which can be an equilibrium of the system.[43] More precisely, the initial state is drawn from a Gaussian distribution centred at the reference state, *i.e.* $\mathbb{P}_{X_0} \sim \mathcal{N}(x_e, \sigma_e^2 I_{d_X})$ where $\sigma_e > 0$ is the standard deviation of the distribution. Regarding the Navier-Stokes flows, the initial state is an arbitrary element of the state space that belongs to the attractor of the system (the set of states that the system tends to reach after a long time, *i.e.* the ergodic system behaviour).

3.4.2 Lorenz 63' System

In the study of deterministic chaos, one of the most prominent systems is given by the *Lorenz 63'* differential equations. Those equations model the unpredictable behaviour usually associated with the weather. Over the years, this system inspired several works from the control community (Vincent and Yu 1991) and is given as follows for some positive β_i, $i = 1, 2, 3$ and an additive control input:

$$\begin{aligned} \partial_t x_t^1 &= \beta_1(x_t^2 - x_t^1) + u_t^1 \\ \partial_t x_t^2 &= x_t^1(\beta_2 - x_t^3) - x_t^2 + u_t^2 \\ \partial_t x_t^3 &= x_t^1 x_t^2 - \beta_3 x_t^3 + u_t^3 \end{aligned} \tag{3.31}$$

In particular, when $\beta_1 = 10$, $\beta_2 = 28$ and $\beta_3 = \frac{8}{3}$ it has chaotic solutions and three unstable equilibria $x_{e_i^*}$ for $i = 1, 2, 3$, which are considered as a reference state x_e for the resulting MDP.

Note there is no spatial dimension in this system, thus the state space is finite-dimensional with $\mathcal{X} = \mathbb{R}^{d_X}$, $d_X = 3$ and $\mathcal{U} = \mathbb{R}^{d_U}$, $d_U = 3$. The observable operator g chosen in this work is the identity, $g = Id$ thus $\mathcal{Y} = \mathcal{X}$ and the discrete operator is implicitly obtained during Runge-Kutta 4 integration. In the experiments, the initial reference state is set to $x_e = x_{e_1^*}$. Illustrations of the Lorenz system used in this work are given in Figure 3.1.

3.4.3 Kuramoto-Sivashinsky

The second dynamical system in question is the *Kuramoto-Sivashinsky* (KS) equation. It is a well-known unidimensional partial differential equation which exhibits spatio-temporally chaotic behaviour and describes many physical settings such as stability of flame fronts or reaction-diffusion systems (Cvitanović, Davidchack, and Siminos 2010). In this work, the KS equation is given for any $z \in \mathcal{Z}$ by

$$\partial_t x_t(z) = -x_t(z)\partial_z x_t(z) - \partial_z^2 x_t(z) - \partial_z^4 x_t(z) + A_{\mathsf{KS}}(u_t)(z) \tag{3.32}$$

[43] An equilibrium is a state where the system remains if no external forces are applied. In the ODE case, it is a state $x_e \in \mathcal{X}$ such that $F(x_e) = 0$ for any $t \in I$, where F is the system dynamics. In other words, the velocity of the system is null at the equilibrium, hence the state remains constant and does not depend on time.

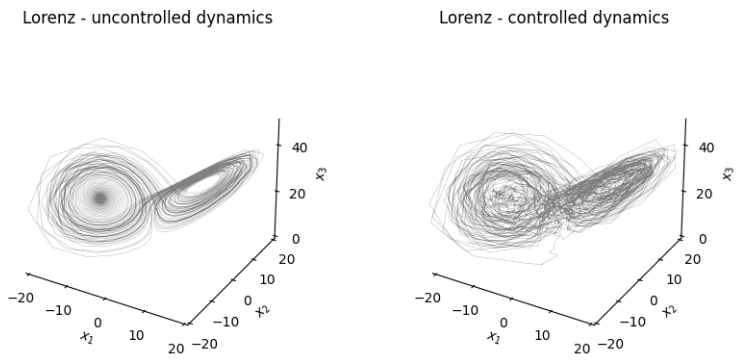


Figure 3.1: Trajectory of the Lorenz system ((3.31)) in this work without control (left) and with random actuation (right). The system represented here is the one used in this work, and the trajectory horizon is ten times greater than the one used in the experiments. Note the initial state $X_0 \sim \mathcal{N}(x_{e_1^*}, \sigma_e^2 I_d)$ is randomly picked in the vicinity of the equilibrium $x_{e_1^*}$ which is at the centre of the left butterfly wing.

where the spatial domain is given by $\mathcal{Z} = [0, L_{\mathcal{X}}]$ with periodic boundary conditions ($x_t(z + L_{\mathcal{X}}) = x_t(z)$ for any $z \in \mathcal{Z}$ and $t \in I$), A_{KS} is an actuation operator that models actuator interactions with the system.

The observation mapping $g : \mathcal{X} \to \mathcal{Y} = \mathbb{R}^{d_Y}$ models the d_Y spatially equidistant sensors and is given, for any $1 \leq i \leq d$ and any $t \in [0, T]$, by

$$g_i(x_t) = \langle x_t,\, g_{\tilde{\mu}_i, s} \rangle_{L^2} \tag{3.33}$$

where $g_{\tilde{\mu}_i,s}$ is a Gaussian density with mean $\tilde{\mu}_i \in [0, L_{\mathcal{X}}]$ and standard deviation s. Thus, $(\tilde{\mu}_i)_{1 \leq i \leq d}$ and $(s_i)_{1 \leq i \leq d_Y}$ represent respectively the barycenter and scale of the sensors. Similarly, the actuation mapping $A_{\text{KS}} : \mathcal{U} = \mathbb{R}^{d_U} \to \mathcal{X}$ is given, for any $1 \leq i \leq d_U$ and any $t \in [0, T]$, by

$$A_{\text{KS}}(u)(z) = \sum_{j=1}^{d_U} u^j h_{\bar{\mu}_j, \bar{s}}(z) \tag{3.34}$$

for any $z \in \mathcal{Z}$, $u \in \mathcal{U}$ where $h_{\bar{\mu}_i,s}$ is a Gaussian function with mean $\bar{\mu}_i \in [0, L_{\mathcal{X}}]$ and standard deviation $\bar{s}$.
Here, the control $u = (u^j)_{1 \leq j \leq d_U}$ is vector valued and each coordinate represents the intensity of the actuation at a given location $\bar{\mu}_j$. The role of A_{KS} is to map those intensities to the system state space $\mathcal{X}$.

This construction is inspired by Bucci et al. 2019 and in the same fashion, the spatial domain space is chosen with $L_{\mathcal{X}} = 22$. In this setting, the dynamics have 4 unstable equilibria $x_{e_i^*}(z)$ for $i = 0, 1, 2, 3$ (spatially dependent on $\mathcal{Z} = [0, L_{\mathcal{X}}]$ and time independent functions) which are considered as reference state x_e for the resulting PO-MDP. Especially, $x_{e_0^*}(z) = 0$ is the constant, null function on the spatial domain.

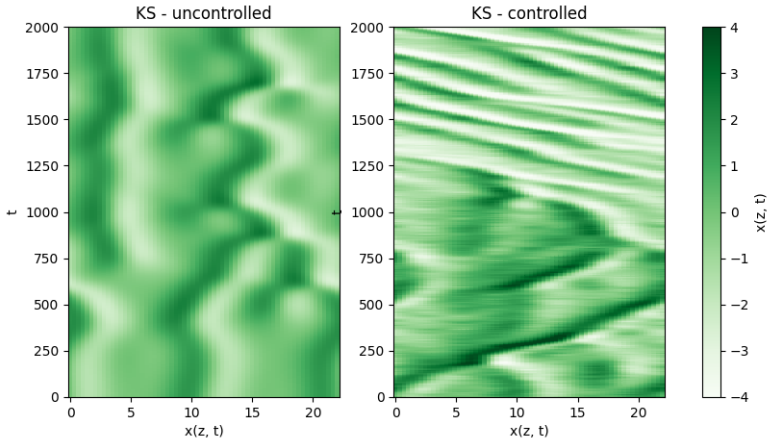


Figure 3.2: Trajectory of the Kuramoto-Sivashinsky system from (3.32) without control (left) and with random actuation (right). The system represented here is the one used in this work, and the trajectory horizon is ten times greater than the one used in the experiments. Note the initial state $X_0 \sim \mathcal{N}(x_{e_2^*}, \sigma_e^2 I_d)$ is randomly picked in the vicinity of the equilibrium $x_{e_2^*}(z)$

Indeed, in this setting the system shares some interesting properties found in the Navier-Stokes equations, which is a more realistic and practical but computationally expensive model of fluid dynamics (Viquerat et al. 2022).

Finally, all Banach spaces $\mathcal{X}$, $\mathcal{Y}$ and $\mathcal{U}$ are discretised, with $d_X = 64$, $d_Y = 8$ and $d_U = 8$ where the sensors and actuators are equidistantly distributed in the spatial domain. In the experiments, the initial reference state is set to $x_e = x_{e_2^*}$. The equation being stiff, a particular method (Cox and Matthews 2002) is used to proceed to the integration of the PDE. Notably, the periodic boundary conditions allow using the Fourier transform to compute the spatial derivatives. The resulting dynamical system is illustrated in Figure 3.2.

3.4.4 Pendulum

A simple benchmark problem in control theory is the pendulum (Khalil 2002). The dynamics of the pendulum are described by the following ordinary differential equation:

$$\begin{aligned} \partial_t x_t^1 &= x_t^2 \\ \partial_t x_t^2 &= -\frac{g_\mathrm{P}}{l_\mathrm{P}} \sin(x_t^1) + \frac{1}{m_\mathrm{P}} u_t \end{aligned} \tag{3.35}$$

where x_t^1 is the angle of the pendulum with respect to the vertical axis, x_t^2 is the angular velocity of the pendulum, u_t is the control input, g_P is the acceleration due to gravity, l_P is the length of the pendulum, and m_P is the mass of the pendulum. In the experiments, the parameters are set to the Gym default values $g_\mathrm{P} = 30.0$, $l_\mathrm{P} = 2.0$ and $m_\mathrm{P} = \frac{l_\mathrm{P}^2}{3}$.

In general, the control objective is to stabilise the pendulum in the upright position. The control input is bounded with $u_t \in [-a_{\mathsf{P}}, a_{\mathsf{P}}]$ for some $a_{\mathsf{P}} > 0$ and any $t \in I$. The pendulum has two equilibria $x_{e_0^*} = (0, 0)$ and $x_{e_1^*} = (\pi, 0)$ which is unstable. The initial reference state is set to $x_e = x_{e_0^*}$. The state space is $\mathcal{X} = \mathbb{R}^{d_X}$, $d_X = 2$ and the control space is $\mathcal{U} = \mathbb{R}^{d_U}$, $d_U = 1$. The observable operator g chosen is the identity, $g = Id$ thus $\mathcal{Y} = \mathcal{X}$ and the discrete operator is implicitly obtained with Euler integration.

The reader is referred to Towers et al. 2024 for a more detailed description of the inverted pendulum problem and its practical implementation.

3.4.5 Van der Pol Oscillator

A classical equation of nonlinear dynamics is the Van der Pol oscillator (Khalil 2002). This system was originally introduced by the Dutch physicist Van der Pol to study oscillations in vacuum tube circuits. Then, it became a fundamental example in nonlinear oscillation theory (Atay 1998), hosting a large quantity of interesting dynamical behaviours.

The equation is given by

$$\begin{aligned} \partial_t x_t^1 &= x_t^2 + u_t^1 \\ \partial_t x_t^2 &= \epsilon_{\mathsf{VDP}}(1 - (x_t^1)^2)x_t^2 - x_t^1 + u_t^2 \end{aligned} \tag{3.36}$$

where x_t^1 is the position of the oscillator, x_t^2 is the velocity of the oscillator, $\epsilon_{\mathsf{VDP}} > 0$ is a parameter that controls the nonlinearity of the system.

The only equilibrium of the system is the origin $x_{e_0^*} = (0, 0)$. Otherwise, all solutions are periodic, and the system exhibits limit cycle behaviour. The control input is bounded by $u_t \in [-a_{\mathsf{VDP}}, a_{\mathsf{VDP}}]$ for some $a_{\mathsf{VDP}} > 0$. In the experiments, the parameter is set to $\epsilon_{\mathsf{VDP}} = 1.5$, $a_{\mathsf{VDP}} = 1.0$.

The state space is finite-dimensional with $\mathcal{X} = \mathbb{R}^{d_X}$, $d_X = 2$ and $\mathcal{U} = \mathbb{R}^{d_U}$, $d_U = 2$. The observation operator is the identity, $g = Id$ thus $\mathcal{Y} = \mathcal{X}$ and the discrete-time operator is implicitly obtained with the Dormand-Prince 5 integration method (Hairer, Nørsett, and Wanner 2008) from the *torchdde* library (Monsel et al. 2024).

3.4.6 Mackey-Glass

The *Mackey-Glass* equation is a representative instance of delay-induced chaos. Originally, it was introduced in M. C. Mackey and Leon Glass 1977 to model the dynamics of circulating blood cells in the human body. The equation is a nonlinear ordinary differential equation that describes the evolution of a monitored variable. The value of a state variable is sensed, and appropriate changes are made in the production (or decay) rates of blood cell concentration. A delayed state term models the time lag between sensing and response. The delayed

differential dynamics read

$$\partial_t x_t = \beta_{\text{MG}} \frac{x_{t-\tau_X}}{1 + x_{t-\tau_X}^{n_{\text{MG}}}} - \gamma_{\text{MG}} x_t + u_t \tag{3.37}$$

where β_{MG}, γ_{MG}, and n_{MG} are positive parameters. The time delay τ_X is a positive constant such that $t - \tau_X \in I = [t_0, T]$ for any $t \in I$. The control input u is a bounded function of time and is absent in the original formulation.

In the dynamical system literature, this kind of equation is said to belong to the class of feedback (delay) systems as the delayed state is fed back into the dynamics. Given the parameter values, the system exhibits multiple attractors such as fixed points, periodic orbit, and chaotic attractors (Kiss and Röst 2017). Because of the practical difficulties induced by the stiffness of the equation when choosing a chaotic parameter regime, a configuration exhibiting a periodic orbit is chosen. More precisely, the parameters are set to $\beta_{\text{MG}} = 2.0$, $\gamma_{\text{MG}} = 1.0$, $n_{\text{MG}} = 8.0$. This dynamics possesses two equilibria $x_{e_0^*} = 0$ and $x_{e_1^*} > 0$. The time delay is set to $\tau_X = 1.0$. Even though the complexity of the system does not reach the chaotic regime, the essential property of interest is the delayed feedback.

Note there is no spatial dimension in this system, thus the state space is finite-dimensional with $\mathcal{X} = \mathbb{R}^{d_X}$, $d_X = 1$ and $\mathcal{U} = [-a_{\text{MG}}, a_{\text{MG}}]$, for some $a_{\text{MG}} > 0$. Then, $d_U = 1$. The observable operator g chosen is the identity, $g = Id$ thus $\mathcal{Y} = \mathcal{X}$ and the discrete operator is implicitly obtained with the Runge-Kutta 4 DDE solver from the *torchdde* library (Monsel et al. 2024). In the experiments, the initial reference state is set to $x_e = x_{e_1^*}$.

For a thorough description and historical notes on the reasoning behind the construction of the Mackey-Glass equation, the reader is referred to L. Glass and M. Mackey 2010.

3.4.7 Navier-Stokes 2-Dimensional Flow

In the following example, a particular case of the Navier-Stokes equations introduced earlier in Example 2.13 for numerical simulations is presented.

Example 3.4.1 (Adimensional Navier-Stokes)**.** *Some fundamental system in fluid dynamics is governed by the Navier-Stokes equation. Let the velocity field be denoted as* $(x_t(z_1, z_2))_{t \in \mathbb{R}_+}$ *and the pressure field as* $(p_t(z_1, z_2))_{t \in \mathbb{R}_+}$ *for any* $(z_1, z_2) \in \mathcal{Z}_{NS} \subset \mathbb{R}^2$ *where* $\mathcal{Z}_{NS}$ *is the spatial domain. The adimensionalised Navier-Stokes equation reads:*

$$\frac{\partial x_t}{\partial t} + \langle x_t, \nabla \rangle x_t = -\nabla p_t + \frac{1}{\text{Re}} \Delta x_t, \tag{3.38}$$

for any $t \in I$*, where the same notation as in Example 2.13 is used.*

The velocity field $x = (x^1, x^2)$ *is adimensionalised with respect to a characteristic velocity* U_∞*, while the spatial coordinates* $(z_1, z_2) \in \mathcal{Z}_{NS}$ *are scaled with a characteristic length* L_{NS} *that generally depends on the spatial domain (e.g. physical object*

length). The Reynolds number Re *is defined as:*

$$\mathrm{Re} = \frac{U_\infty L_{NS}}{\nu_{NS}},$$

where ν_{NS} *is the kinematic viscosity. This number is a parameter that controls the "complexity" of the flow and helps to predict fluid flow patterns. At low Reynolds numbers, flows tend to be dominated by laminar (constant streamlines) flow, while at high Reynolds numbers, flows tend to be turbulent.*

The flow incompressibility hypothesis is made to simplify the problem. This hypothesis is expressed by the divergence-free condition:

$$\mathrm{div}(x_t) = 0 \tag{3.39}$$

for any $t \in I$*, where* div *is the divergence operator (Chorin and Marsden 2013).*

For this form of the equation, boundary conditions also play a crucial role since they notably define the geometry of the problem. Importantly, all control inputs for the next examples are embedded in the boundary conditions. For instance, a blowing or suction strategy at a specific location in the spatial domain can be modelled by some specific boundary conditions. See Holmes et al. 2012 for a general description and the references attached to the particular flows below.

The flow control interface is managed by *Hydrogym* (Paehler et al. 2023) which is built on top of *Firedrake* (Ham et al. 2023), an automated system for the solution of partial differential equations using the finite element method (FEM) (Allaire 2005) and the *Unified Form Language (UFL)* (Alnaes et al. 2013) from the FEniCS project (Baratta et al. 2023). The flow is integrated in time with the semi-implicit backward differentiation formula (Semi-implicit BDF) method (Forti and Dedè 2015). Thus, the exact implementation of the flows is available at the following address: `https://github.com/dynamicslab/hydrogym/tree/main`.

In the case of Navier-Stokes flows, any system state $x \in \mathcal{X}$ is a vector field. Then, the state space $\mathcal{X}$ is the space of vector fields on the corresponding spatial domain $\mathcal{Z}_{\mathrm{NS}}$. Those large state spaces (infinite dimensional) are numerically discretised with a finite element approach. Moreover, measurements are taken at discrete locations in the spatial domain, leading to a finite-dimensional observation space $\mathcal{Y} = \mathbb{R}^{d_Y}$ with $d_Y \in \mathbb{N}^*$.

Now the three Navier-Stokes benchmark problems are presented in the following paragraphs.

Cylinder Flow

The cylinder flow is a classical benchmark problem in fluid dynamics. It consists of a two-dimensional flow around a circular cylinder in a uniform stream. The characteristic length L_{NS} is the diameter of the cylinder here.

Above a critical Reynolds $\mathrm{Re} \approx 50$, the uncontrolled flow is linearly unstable and eventually reaches a post-transient state of periodic vortex shedding, the

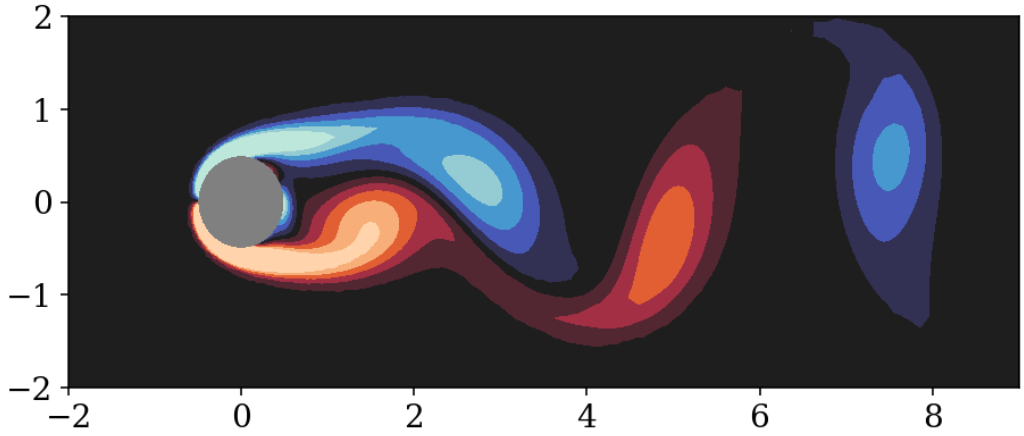


Figure 3.3: Illustration of the cylinder flow problem. Shown here is the velocity (vector) field of the flow around a circular cylinder at Reynolds $\mathrm{Re} = 100$.

well-known von Kármán vortex street. From a flow control perspective, this is a benchmark problem in stabilisation and drag reduction. Generally, the objective is to reduce the drag force (less commonly, the lift force) acting on the cylinder. This setup incorporate sufficient challenges for control strategies, such as the nonlinearity of the flow and the actuation through partially observable measurements.

Two measurements ($d_Y = 2$) are extracted from the flow: the lift and drag coefficients acting on the cylinder. The flow actuation is performed by two jets normal to the cylinder wall relative to the flow direction. Mass flow rates representing blowing or suction on the cylinder wall are injected, following Rabault et al. 2019. Hence, $d_U = 1$ since the actuation intensity of one jet is equal to the opposite of the other. The control space $\mathcal{U}$ is symmetrical and bounded by the maximum actuation intensity which is specified by the user.

Complementary to the Hydrogym interface, the reader can refer to Sipp and Lebedev 2007 for a detailed description of the uncontrolled flow and Rabault et al. 2019 for the associated control problem.

Fluidic Pinball

The pinball flow extends the cylinder flow by adding two additional cylinders in the wake of the main cylinder. Originally, this flow was introduced for testing flow control laws with low computational cost, while being physically complex enough to host a range of interacting frequencies Deng et al. 2018. This is a relatively new benchmark for multiple inputs-multiple outputs nonlinear flow control. This configuration exhibits a large range of flow behaviours, from steady state to chaotic dynamics.

Similarly to the cylinder flow, the pinball flow is characterised by the lift and drag coefficients acting on the three cylinders. The characteristic length L_{NS}

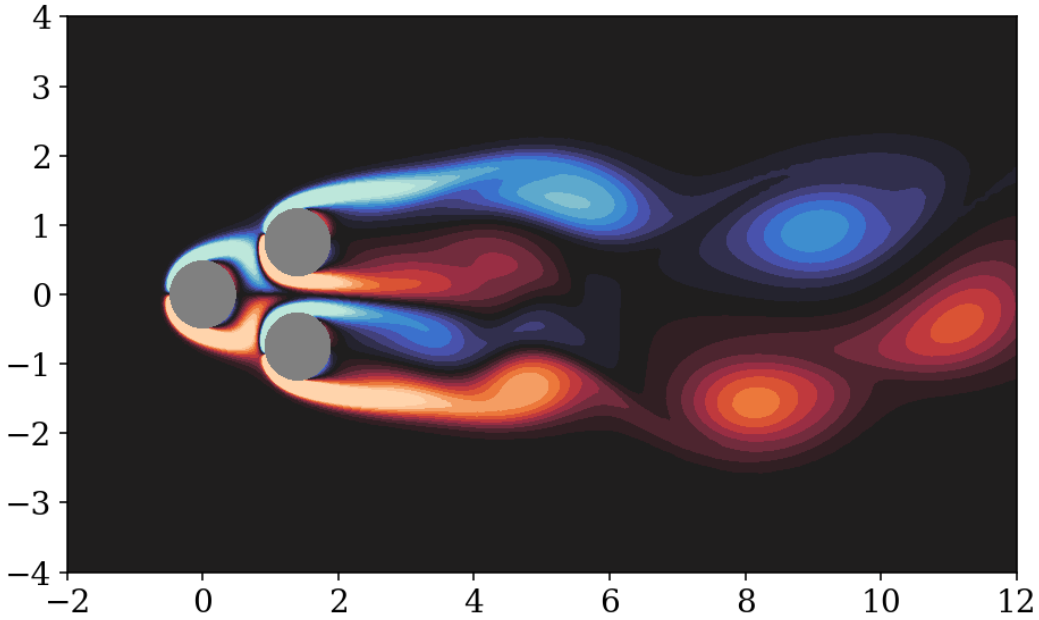


Figure 3.4: Illustration of the fluidic pinball problem. Shown here is the velocity (vector) field of the flow around a pinball at Reynolds $\mathrm{Re} = 130$.

is the diameter of any cylinder. However, the actuation is performed by rotating the cylinders around their axis. Consequently, the control space $\mathcal{U}$ is the bounded space of rotations of the cylinders, and the control dimension $d_U = 3$ is the number of cylinders. The measurements are taken at the same locations as the cylinder flow, leading to $d_Y = 2 \times 3 = 6$.

Complementary to the Hydrogym interface, the reader can refer to Deng et al. 2018 for a detailed description of the uncontrolled flow and Cornejo Maceda et al. 2021 for the associated control problem. See Peitz, Otto, and Rowley 2020 for a recent application of the Koopman operator theory to control the fluidic pinball.

Cavity Flow

Another particular flow which exhibits a rich variety of behaviours is the open Cavity Flow. This is a benchmark of commonly called *separated fluid flow*.

From Barbagallo, Schmid, and Huerre 2009: "This type of flow exhibits a recirculating component (confined geometrically to the cavity) as well as a strong shear layer that forms at the top of the cavity and, for sufficiently high Reynolds number, becomes unstable and settles into a characteristic periodic motion."

A blowing and suction strategy is applied to the cavity flow control problem. The characteristic length L_{NS} is the depth of the cavity here. The sensor measurement is located at the top-right corner of the cavity. The actuation is performed by a jet at the top-left corner (upstream edge) of the cavity. Con-

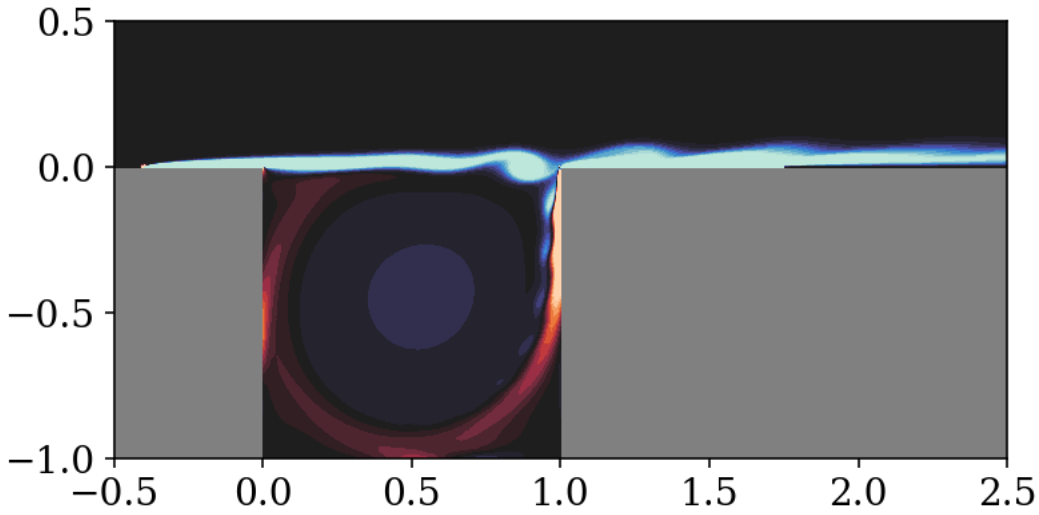


Figure 3.5: Illustration of the cavity flow problem. Shown here is the velocity (vector) field of the flow around a square cavity at Reynolds $\mathrm{Re} = 7500$.

cretely, the control input is the intensity of the jet, and the control space $\mathcal{U}$ is the bounded space of the jet intensity. The control dimension $d_U = 1$ is the number of actuators. The observation space $\mathcal{Y}$ is the set of wall-normal shear stress measurements evaluated on a neighbourhood of at the top-right corner of the cavity (sensor location). Shear stress is critical in understanding drag forces on surfaces.

Complementary to the Hydrogym interface, the exact configuration with geometric and numerical details is fully described in Sipp and Lebedev 2007 and the control setup is described in Barbagallo, Schmid, and Huerre 2009.

3.5 Conclusion

This chapter introduced the discrete-time version of the controlled process and its associated optimality problem, which is widely used in the Learning-Based Control literature.

First of all, the general discrete time decision process was defined (Section 3.1) together with the discrete-time version of Dynamic Programming (Section 3.1.4). The Learning Theory (Section 3.2) was then introduced in a sufficient generality to encompass the range of problems encountered in this thesis. Then, Learning-based Control was introduced (Section 3.3) as the application of estimation and learning techniques to the optimal control problem.

Multiple important concepts and paradigms that are used throughout the thesis were presented, such as the policy iteration procedure or Model Predictive Control.

Finally, the chapter concluded with a presentation of the several controlled

dynamics that are of interest in the field of control of Dynamical Systems (Section 3.4).

II Methodological Advances in Learning Based Control

4 Evidence on the Regularisation Properties of Maximum Entropy Reinforcement Learning

This chapter is an attempt to address the robustness challenge discussed in the introduction of the thesis (Section 1.4.1). Here, robustness to white noise is considered in the context of Reinforcement Learning and some empirical evidence is provided to support the hypothesis that Maximum Entropy Reinforcement Learning policies are more robust than their non-regularised counterparts.

This work led to the publication of a paper on the proceedings of the 7th International Conference in Optimization and Learning (OLA24) in Dubrovnik (Hosseinkhan Boucher, Semeraro, and Mathelin 2025).

4.1 Introduction

Maximum Entropy Reinforcement Learning (R. J. Williams, Peng, and H. Li 1991) aims to solve the problem of learning a policy which optimises a chosen utility criterion while promoting the entropy of the policy. The standard way to account for the constraint is to add a Lagrangian term to the objective function. This entropy-augmented objective is commonly referred to as the soft objective.

There are multiple advantages in solving the soft objective over the standard objective. For instance, favouring stochastic policies over deterministic ones allows learning multi-modal distributions (Haarnoja, Tang, et al. 2017). In addition, agent stochasticity is a suitable way to deal with uncertainty induced by Partially Observable Markov Decision Processes (PO-MDP). Indeed, there are PO-MDP such that the best stochastic adapted policy can be arbitrarily better than the best deterministic adapted[44] policy (Sigaud and Buffet 2010).
Furthermore, several important works highlight both the theoretical and exper-

[44] In this context, the term "stochastic adapted policy" is a conditional distribution on the control space $\mathcal{U}$ given the observation space $\mathcal{Y}$ since this type of policy is "adapted" from Markovian policies in fully observable MDPs.

imental *robustness* of those policies under noisy dynamics and rewards (Eysenbach and Levine 2022).

Related to the latter notion of robustness, the maximum-entropy principle exhibits non-trivial generalisation capabilities, which are desired in real-world applications (Haarnoja, A. Zhou, Abbeel, et al. 2018).

However, the reasons for such robustness properties are not yet well understood. Thus, further investigations are needed to grasp the potential of the approach and to design endowed algorithms. A clear connection between Maximum-Entropy RL and their robustness properties is important and intriguing.

Meanwhile, recent work in the deep learning community discusses how some complexity measures on the neural network model are related to generalisation and explains typically observed phenomena (Neyshabur, Bhojanapalli, et al. 2017). In fact, these complexity measures are derived from the learnt model, they bound the PAC-Bayes generalisation error, and are meant to identify which of the local minima generalise well.

As a matter of fact, a relatively recent trend in statistical learning suggests that generalisation is not only favoured by the regularisation techniques (*e.g.* dropout) but mainly because of the flatness of the local minima (Hochreiter and Schmidhuber 1997; Dinh et al. 2017; Keskar et al. 2017). The reasons for such regularity properties remain an open problem. This work aims to address these points in the context of Reinforcement Learning, and addresses the following questions:

What is the bias introduced by entropy regularisation? Are the aforementioned complexity measures also related to the robustness of the learnt solutions in the context of Reinforcement Learning?

In that respect, by defining a notion of robustness against noisy contamination of the observable, a study on the impact of the entropy regularisation on the robustness of the learnt policies is first conducted. After explaining the rationale behind the choice of the complexity measures, a numerical study is performed to validate the hypothesis that some measures of complexity are good robustness predictors. Finally, a link between the entropy regularisation and the flatness of the local minima is treated through the information geometry notion of Fisher Information.

The chapter is organised as follows. Section 4.2 introduces the background and related work, Section 4.3 presents the problem setting. Section 4.4 is the core contribution of this chapter. This section introduces the rationale behind the studied complexity measures from a learning theory perspective, as well as their expected relation to robustness. Lastly, Section 4.5 presents the experiments related to the policy robustness as well as their complexity, while Section 4.6 examines the results obtained. Finally, Section 4.8 concludes the chapter.

4.2 Related work

Maximum Entropy Policy Optimisation In Haarnoja, A. Zhou, Abbeel, et al. 2018, the generalisation capabilities of entropy-based policies are observed where multimodal policies lead to optimal solutions. It is suggested that maximum entropy solutions aim to learn all the possible ways to solve a task. Hence, transfer learning towards more challenging objectives is made easier, as it is demonstrated in their experiment. This study investigates the impact of adopting policies with greater randomness on their robustness. The impact of the entropy regularisation on the loss landscape has been recently studied in (Z. Ahmed et al. 2019). They provide experimental evidence about the smoothing effect of entropy on the optimisation landscape. The present study aims specifically to answer the question in Section 3.2.4 of their paper: *Why do high entropy policies learn better final solutions?* This work extends their results from a complexity measure point of view. Recently, (Neu, Jonsson, and Gómez 2017; Derman, Geist, and Mannor 2021) studied the equivalence between robustness and entropy regularisation on regularised MDP.

Flat minima and Regularity The notion of local minima flatness was first introduced in the context of supervised learning by Hochreiter and Schmidhuber 1997 through the Gibbs formalism (Haussler and Opper 1997). Progressively, different authors stated the concept with geometric tools such as first order (gradient) or second order (Hessian) regularity measures (Zhao, Zhang, and Hu 2022; Keskar et al. 2017; Sagun, Bottou, and LeCun 2017; Yoshida and Miyato 2017; Dinh et al. 2017). In a similar fashion, Chaudhari et al. 2019 uses the concept of local entropy to smooth the objective function.
In the scope of Reinforcement Learning, Z. Ahmed et al. 2019 observed that flat minima characterise maximum entropy solutions, and entropy regularisation has a smoothing effect on the loss landscape, reducing the number of local optima. A central objective of this present study is to investigate this latter property further and relate it to the field of research on robust optimisation. Lastly, among the few recent studies on the learning and optimisation aspects of RL, Gogianu et al. 2021 shows how a well-chosen regularisation can be very effective for deep RL. Indeed, they explain that constraining the Lipschitz constant of only one neural network layer is enough to compete with state-of-the-art performances on a standard benchmark.

Robust Reinforcement Learning A branch of research related to this work is the study of robustness with respect to the uncertainty of the dynamics, namely *Robust Reinforcement Learning* (Robust RL), which dates back to the 1970's (Satia and Lave 1973). Correspondingly, in the field of control theory, echoes the notion of robust control and especially H_∞ control (K. Zhou, J.C. Doyle, and Glover 1996), which also appeared in the mid-1970s after observing Linear Quadratic

Regulator (LQR) solutions are very sensitive to perturbations while not giving consistent enough guarantees (J. Doyle 1996).
More specifically, the Robust RL paradigm aims to control the dynamics in the worst-case scenario, *i.e.* to optimise the minimal performance for a given objective function over a set of possible dynamics through a min-max problem formulation. This set is often called *ambiguity set* in the literature. It is defined as a region in the space of dynamics close enough w.r.t. to some divergence measure, such as the relative entropy (Nilim and Ghaoui 2003). Closer to this work, the recent paper from Eysenbach and Levine 2022 shows theoretically how Maximum-Entropy RL policies are inherently robust to a certain class of dynamics of fully observed MDP. The finding of their article might still hold in the partially observable setting as any PO-MDP can be cast as fully observed MDP with a larger state-space of probability measures (Onésimo Hernández-Lerma and Lasserre 1996), provided the ambiguity set is adapted to a more complicated space.

4.3 Problem Setup and Background

4.3.1 Partially Observable Markov Decision Process with Gaussian noise

First, the stochastic control problem when noisy observations are available to the agent is formulated. The study focuses on *Partially Observable Markov Decision Processes (PO-MDP)* with Gaussian noise of the form (M. P. Deisenroth and Peters 2012):

$$\begin{aligned} X_{k+1} &= F\left(X_k,\, U_k\right) \\ Y_k &= G\left(X_k\right) + \epsilon_Y, \quad \epsilon_Y \sim \mathcal{N}(0, \sigma_Y^2 I_d) \end{aligned} \tag{4.1}$$

with $X_k \in \mathcal{X}$, $U_k \in \mathcal{U}$ and $Y_k \in \mathcal{Y}$ for any $k \in \mathbb{N}$, where $\mathcal{X}$, $\mathcal{U}$ and $\mathcal{Y}$ are respectively the corresponding state, action, and observation spaces. The initial state starts from a reference state x_e^* on which centred Gaussian noise with diagonal covariance $\sigma_e^2 I_d$ is additively applied, $X_0 \sim \mathcal{N}(x_e^*,\, \sigma_e^2 I_d)$. Associated with the dynamics, an instantaneous cost function $c : \mathcal{X} \times \mathcal{U} \to \mathbb{R}_+$ is also given to define the control model.

In the context of this chapter, a *policy* π is a transition kernel on $\mathcal{U}$ given $\mathcal{Y}$, *i.e.* a distribution on actions conditioned on observations. This kind of policy is commonly used in literature but can be very poor in the partially observable setting where information is missing. Together, a control model, a policy π, and an initial distribution $\mathbb{P}_{X_0}$ on $\mathcal{X}$ define a stochastic process with distribution $\mathbb{P}^{\pi,\epsilon_Y}$ (Proposition 2.3.1) where the superscript ϵ_Y highlights the dependency on the observation noise ϵ_Y. Similarly, one denotes by $\mathbb{P}^{\pi}$ the distribution of the process when the noise is zero almost-surely, *i.e.* $\mathbb{P}^{\pi} = \mathbb{P}^{\pi,0}$. More details about the PO-MDP control problem can be found in Onésimo Hernández-Lerma and

Lasserre 1996; Cassandra 1998.

Here, the maximum-entropy control problem is to find a policy π^* which minimises the following performance criterion

$$J_m^{\pi,\epsilon_Y} = \mathbb{E}^{\pi,\epsilon_Y}\left[\sum_{k=0}^{K}\gamma^k c\left(X_k, U_k\right)\right] - \alpha^{\mathcal{H}}{}_m \mathbb{E}^{\pi,\epsilon_Y}\left[\sum_{k=0}^{K}\gamma^k \mathcal{H}(\pi(\cdot \mid X_k))\right], \tag{4.2}$$

where $K \in \mathbb{N}$ is a given time horizon, $\mathbb{E}^{\pi,\epsilon_Y}$ denotes the expectation under the probability measure $\mathbb{P}^{\pi,\epsilon_Y}$, $\mathcal{H}$ denotes the differential entropy (Cover and Thomas 2006) and $\alpha^{\mathcal{H}}{}_{\mathcal{M}}$ is a time-dependent weighting parameter that evolves over training time $m \leq m_{\mathcal{D}} = |\mathcal{D}|$ with $|\mathcal{D}|$ being the total number of times the agent interacts with the system such that all observations used by the learning algorithm form the dataset $\mathcal{D}$ at the end of the training procedure (when $m_{\mathcal{D}}$ environment interactions are done).
In the $\alpha^{\mathcal{H}}_m = 0$ case, J_m^{π,ϵ_Y} is denoted J^{π,ϵ_Y}. Here, the quantity J^{π,ϵ_Y} is called the value function or, more generally, *loss* (see also Section 3.1.4).[45]
Moreover, the performance gap for dynamics with noisy and noiseless observables will be considered in the sequel. In this context, the *(rate of) excess risk under noise* is defined as the difference between the loss under noisy dynamics and the loss under noiseless dynamics:

Definition 4.3.1 (Excess Risk Under Noise)**.** *The excess risk under noise of a policy π for a PO-MDP with dynamics given by Eq.* (4.1) *is defined as:*

$$\mathcal{R}^{\pi} = \mathbb{E}^{\pi,\epsilon_Y}\left[\sum_{k=0}^{K}\gamma^k c\left(X_k, U_k\right)\right] - \mathbb{E}^{\pi}\left[\sum_{k=0}^{K}\gamma^k c\left(X_k, U_k\right)\right] = J^{\pi,\epsilon_Y} - J^{\pi} \tag{4.3}$$

Similarly, the rate of excess risk under noise is defined as:

$$\mathring{\mathcal{R}}^{\pi} = \frac{J^{\pi,\epsilon_Y} - J^{\pi}}{J^{\pi}} = \frac{\mathcal{R}^{\pi}}{J^{\pi}} \tag{4.4}$$

Note that in the above definition, expectations are taken with respect to the probability measure $\mathbb{P}^{\pi,\epsilon_Y}$ and $\mathbb{P}^{\pi}$ respectively. The *rate of excess risk under noise* represents the performance degradation after noise introduction in value function units. In the sequel, arguments to heuristically derive complexity measures will be developed, allowing to predict the excess risk under noise and provide numerical evidence showing maximum-entropy policies are more robust regarding this metric. Hence, maximum-entropy policies implicitly learn a robust control policy in the sense of Definition 4.3.1.

In the next section, some concepts of statistical learning theory are introduced. Then, complexity measures will be defined to quantify the regularisation power of the maximum-entropy objective of Eq. (4.2).

[45] This notation is more convenient than the one used in Chapter 3 when the context is clear.

4.4 Complexity Measures and Robustness

4.4.1 Complexity Measures

The principal objective of *statistical learning* is to provide bounds on the generalisation error, so-called *generalisation bounds*. In the following, it is assumed that an algorithm $\mathcal{A}$ returns a hypothesis $\pi \in \mathcal{F}$ from a dataset $\mathcal{D}$. Note that the dataset $\mathcal{D}$ is random and the algorithm $\mathcal{A}$ is a randomised algorithm.

As the hypothesis set $\mathcal{F}$ typically used in machine learning is infinite, a practical way to quantify the generalisation ability of such a set must be found. This quantification is done by introducing *complexity measures*, enabling the derivation of generalisation bounds.

Definition 4.4.1 (Complexity measure)**.** *A complexity measure is a mapping $\mathcal{M} : \mathcal{F} \to \mathbb{R}_+$ that maps a hypothesis to a positive real number.*

According to Neyshabur, Bhojanapalli, et al. 2017 from which this formalism is inspired, an appropriate complexity measure satisfies several properties. In the case of parametric models $\pi_\theta \in \mathcal{F}(\Theta)$ with $\theta \in \Theta \subset \mathbb{R}^b$, it should increase with the dimension b of the parameter space Θ as well as being able to identify when the dataset $\mathcal{D}$ contains totally random, spurious or adversarial data. As a result, finding good complexity measures $\mathcal{M}$ allows the quantification of the generalisation ability of a hypothesis set $\mathcal{F}$ or a model π and an algorithm $\mathcal{A}$.

4.4.2 Complexity measures for PO-MDP with Gaussian Noise

This work studies heuristics about generalisation bounds on the optimal *excess risk under noise* from Definition 4.3.1when the optimal policy π_{θ^*} is learnt with an algorithm $\mathcal{A}$ on the non-noisy objective J^π, where $\alpha^{\mathcal{H}}_m = 0$ for any m.

Definition 4.4.2 ((Rate of) Excess Risk Under Noise Bound)**.** *Given an optimal policy π^* learnt with an algorithm $\mathcal{A}$ on the non-noisy objective J^π, the optimal excess risk under noise bound is a real-valued mapping φ such that*

$$\mathcal{R}^{\pi^*} \leq \varphi(\mathcal{M}(\pi^*, \mathcal{D}),\, m_{\mathcal{D}},\, \eta,\, \delta) \tag{4.5}$$

and φ is increasing with the complexity measure $\mathcal{M}$ and the sample complexity $m_{\mathcal{D}}$. The definition is similar to the rate of excess risk under noise bound where $\mathring{\mathcal{R}}^{\pi^*}$ is used instead of $\mathcal{R}^{\pi^*}$.

Hence, by considering a learning algorithm $\mathcal{A}$ with a parameterised family given by $\mathcal{F}(\Theta) = (\pi_\theta)_{\theta\in\Theta}$, $\Theta \subset \mathbb{R}^b$, such that $\theta = (\theta_\mu, \theta_{\sigma_\pi})$ with a Gaussian policy $\pi_\theta(\cdot \mid x) \sim \mathcal{N}(\mu_{\theta_\mu}(x), \text{diag}(\theta_{\sigma_\pi}))$, $x \in \mathcal{X}$, - where μ_{θ_μ} is a shallow multi-layer feed-forward neural network (with depth-size $l = 2$, width $w = 64$ neurons, weights matrix $(\theta^i_\mu)_{1\leq i\leq l}$) and $\text{diag}(\theta_{\sigma_\pi})$ is a diagonal matrix of dimension $d_U =$

$\dim(\mathcal{U})$ parameterising the variance[46] — to learn the optimal policy π_{θ^*}, multiple complexity measures $\mathcal{M}$ are defined and details on their underlying rationale are given below.

Norm based complexity measures

First, the so-called norm-based complexity measures are functions of the norm of some subset of the parameters of the model. For instance, a common norm-based measure calculates the product of the operator norms of the neural network linear layers. The measures are commonly used in the statistical learning theory literature to derive bounds on the generalisation gap, especially in the context of neural networks (Neyshabur, Tomioka, and Srebro 2015; Golowich, Rakhlin, and Shamir 2018; Miyato et al. 2018).
In fact, the product of the linear layers norm of a standard class of multi-layer neural networks (including Convolutional Neural Networks) serves as an upper bound on the often intractable Lipschitz constant of the network (Miyato et al. 2018). Thus, controlling the linear layers weights magnitude increases the regularity of the model.

Consequently, the following complexity measures are defined:

- $\mathcal{M}(\pi_\theta, \mathcal{D}) = \|\theta_\mu\|_p$
- $\mathcal{M}(\pi_\theta, \mathcal{D}) = \Pi_{i=1}^{l} \|\theta_\mu^i\|_p$ where θ_μ^i is the i^{th} layer of the network μ_{θ_μ}

In this context $\|\cdot\|_p$ with $p = 1,\ 2,\ \infty$ denotes the p-operator norm while $p = F$ denotes the Frobenius norm, which is discarded for the first case of the full parameters vector θ_μ (since Frobenius norm is defined for matrix).

Flatness based complexity measures

On the other hand, another measure of complexity is given by the flatness of the optimisation local minimum (see Section 4.2 for a brief overview). As McAllester 2003; Neyshabur, Bhojanapalli, et al. 2017 have pointed out, the generalisation ability of a parametric solution is controlled by two key components in the context of supervised learning: the norm of the parameter vector and its flatness w.r.t. to the objective function.

One might wonder if a similar robustness property still holds in the setting of Reinforcement Learning. In this manner, complexity measures quantifying the flatness of the solution are needed. Concretely, the interest lies in the flatness of the local minima of the objective function J^π. As stated earlier, there are several ways to quantify the flatness of a solution with metrics derived from the gradient or curvature of the loss function at the local optimum, such as the

[46] Note this choice of state-independent policy variance is inspired by Z. Ahmed et al. 2019 and simplifies the problem.

Hessian's largest eigenvalue—otherwise spectral norm (Keskar et al. 2017) or the trace of Hessian (Dinh et al. 2017).

Moreover, as discussed in Section 4.2, Z. Ahmed et al. 2019 observed that *maximum entropy solutions are characterised by flat minima* while entropy regularisation has a smoothing effect on the loss landscape. *Hence, a central objective of this present study is to investigate this latter property further and relate it to the robustness aspect of the resulting policies*. However, instead of dealing directly with the Hessian of the objective J^π this work proposes a measure based on the conditional Fisher Information $\mathcal{I}$ of the policy due to its link with a notion of model regularity in the parameter space.

Definition 4.4.3 (Conditional Fisher Information Matrix)**.** *Let $x \in \mathcal{X}$ and π_θ a policy identified by its conditional density for a parameter $\theta \in \Theta \subset \mathbb{R}^b$ and suppose ρ is a distribution over $\mathcal{X}$. The conditional Fisher Information Matrix of the vector θ is defined under some regularity conditions as*

$$\mathcal{I}(\theta) = -\mathbb{E}^{X\sim\rho, U\sim\pi_\theta(\cdot|X)}\left[\nabla^2_\theta \log \pi_\theta(U \mid X)\right], \tag{4.6}$$

where ∇^2_θ denotes the Hessian matrix evaluated at θ.

Note that the distribution over states ρ is arbitrary and can be chosen as the discounted state visitation measure ρ^π induced by the policy π (A. Agarwal, Jiang, and Kakade 2019) or the stationary distribution of the induced Markov process if the policy is Markovian and the MDP ergodic[47] as it is done in Kakade 2001.

As a matter of fact, it has already been mentioned in the early works of policy optimisation (Kakade 2001) that this quantity $\mathcal{I}$ might be related to the Hessian of the objective function. Indeed, the Hessian matrix of the standard objective function reads (see Shen et al. 2019 for a proof):

$$\nabla^2_\theta J^{\pi_\theta} = \mathbb{E}^{\pi_\theta}\left[\sum_{k,i,j=0}^{K} c\left(X_k, U_k\right)\left(\Pi_1^{i,j} + \Pi_2^i\right)\right]. \tag{4.7}$$

where the second order quantities (matrix valued) are given by

$$\Pi_1^{i,j} \coloneqq \nabla_\theta \log \pi_\theta\left(U_i \mid X_i\right) \nabla_\theta \log \pi_\theta\left(U_j \mid X_j\right)^T \tag{4.8}$$

$$\Pi_2^i \coloneqq \nabla^2_\theta\left[\log \pi_\theta\left(U_i \mid X_i\right)\right] \tag{4.9}$$

As suggested by the author mentioned above (S. Kakade), Eq. (4.7) might be related to $\mathcal{I}$ although being weighted by the cost c. Indeed, the Hessian of the state-conditional log-likelihoods ($\nabla^2_\theta \log \pi_\theta$ on the rightmost part of the expectation of Eq. (4.7)) belongs to the objective-function Hessian $\nabla^2_\theta J^{\pi_\theta}$ while the Fisher Information $\mathcal{I}(\theta)$ is an average of the Hessian of the policy log-likelihood.

[47] With these choices, the following holds: $\mathbb{E}^{\rho^\pi(dx)\pi(du|x)} = \mathbb{E}^\pi$ up to taking the expectation *w.r.t.* the state-control space (no subscript under X and U) or the trajectory space (with subscripts such as X_k and U_k as trajectory coordinate) (A. Agarwal, Jiang, and Kakade 2019).

In any case, the conditional FIM measures the regularity of a critical component of the objective to be minimised. Thus, the trace of the conditional FIM of the mean actor network parameter θ_μ is suggested as a complexity measure

- $\mathcal{M}(\pi_\theta, \mathcal{D}) = Tr(\mathcal{I}(\theta_\mu)) = Tr(-\mathbb{E}^{X\sim\rho^\pi, U\sim\pi_\theta(\cdot|X)}\left[\nabla^2_{\theta_\mu} \log \pi_\theta(U \mid X)\right])$.

Moreover, in the context of classification, a link between the degree of stochasticity of optimisation gradients (leading to flatter minima (Mulayoff and Michaeli 2020; Xie, Sato, and Sugiyama 2021)) and the FIM trace during training has recently been revealed in Jastrzebski et al. 2021. Magnitudes of the FIM eigenvalues may be related to loss flatness and norm-based capacity measures to generalisation ability (Karakida, Akaho, and Amari 2019) in deep learning.

4.5 Experiments

4.5.1 Robustness under noise of Maximum Entropy Policies

The first hypothesis is that maximum entropy policies are more robust to noise than those trained without entropy regularisation (which plays the role of control experiments). Consequently, the robustness of the controlled policy π_{θ^*} is compared with the robustness of the maximum entropy policy $\pi^{\alpha^\mathcal{H}}_{\theta^*}$ for different temperature evolutions $\alpha^\mathcal{H} = (\alpha^\mathcal{H}_m)_{0\leq m\leq m_\mathcal{D}}$.

In this view, and since inter-algorithm comparisons are characterised by high uncertainty (Henderson et al. 2018; Colas, Sigaud, and Oudeyer 2018; R. Agarwal et al. 2021), only one algorithm $\mathcal{A}$ (*Proximal Policy Optimisation* (PPO) (Schulman, Wolski, et al. 2017)) is retained while results on multiple entropy constraint levels $\alpha^\mathcal{H} = (\alpha^\mathcal{H}_m)_{0\leq m\leq m_\mathcal{D}}$ are examined.

In this regard, ten independent PPO models are trained for each of the five arbitrarily chosen entropy temperatures $\alpha^{\mathcal{H}i} = (\alpha^{\mathcal{H}i}_m)_{0\leq m\leq m_\mathcal{D}}$ where $i \in \{1, \ldots, 5\}$, on dynamics without observation noise, *i.e.* where $\sigma^2_Y = 0$. The entropy coefficients linearly decay during training, and all vanish ($\alpha^\mathcal{H}_m = 0$) when m reaches one-fourth of the training time $m_{1/4} = \lfloor \frac{m_\mathcal{D}}{4} \rfloor$ in order to replicate a sort of exploration-exploitation procedure, ensuring that all objectives J^π_m are the same whenever $m \geq m_{1/4}$, *i.e.* $J^\pi_m = J^\pi$. This choice is different but inspired by Z. Ahmed et al. 2019 as they optimise using only the *policy gradient* and manipulate the standard deviation of Gaussian policies directly, whereas, in the present approach, it is done implicitly with an adaptive entropy coefficient. An algorithm that learns a model with a given entropy coefficient $\alpha^\mathcal{H} = (\alpha^\mathcal{H}_m)_{0\leq m\leq m_\mathcal{D}}$ is denoted as $\mathcal{A}_{\alpha^\mathcal{H}}$.

The chosen chaotic systems are the *Lorenz* (Vincent and Yu 1991) (with $m_\mathcal{D} = 10^6$) and *Kuramoto-Sivashinsky (KS)* (Bucci et al. 2019) (with $m_\mathcal{D} = 2{\cdot}10^6$) controlled differential equations. The default training hyperparameters from the library *Stable-Baselines3* (Raffin et al. 2021) are used.

4.5.2 Robustness against Complexity Measures

So far, three separate analyses on the 5×10 models obtained have been performed on the *Lorenz* and *Kuramoto-Sivashinsky (KS)* controlled differential equations.
First, as mentioned before, the robustness of the models for each of the chosen entropy temperatures $\alpha^{\mathcal{H}i}$ is tested against the same dynamics but now with a noisy observable, *i.e.* $\sigma_Y > 0$. Second, norm-based complexity measures introduced in Section 4.4.2 are evaluated and compared to the generalisation performances of the distinct algorithms $\mathcal{A}_{\alpha^{\mathcal{H}}}$. Third, numerical computation of the conditional distribution of the trace of the Fisher Information Matrix given by Eq. (4.6) is performed to test the hypothesis that this regularity measure is an indicator of robust solutions. The state distribution ρ^{π_θ} is naturally chosen as the state visitation distribution induced by the policy π_θ. The following section discusses the results of those experiments.

4.6 Results

This section provides numerical evidence of maximum entropy's effect on the robustness, as defined by the Excess Risk Under Noise defined by Eq. (4.3). Then, after quantifying robustness, the relation between the complexity measures defined in Section 4.4.2 and robustness is studied.

4.6.1 Entropy Regularisation induces noise robustness

In the first place, a distributional representation[48] of the rate of excess risk under noise defined in Eq. (4.3) is computed for each of the 5×10 models obtained with the PPO algorithm $\mathcal{A}_{\alpha^{\mathcal{H}i}}$, $i \in \{1, \dots, 5\}$ and different levels of observation noise $\sigma_Y > 0$.

First and foremost, the results shown in Figure 4.1 indicate that the noise introduction to the system observable Y of KS and Lorenz leads to a global decrease in performance, as expected.

The robustness to noise contamination of the two systems is improved by initialising the policy optimisation procedure up to a certain intermediate entropy coefficient threshold $\alpha^{\mathcal{H}i} > 0$. Once this value is reached, two respective behaviours are observed depending on the system. In the case of the Lorenz dynamics, the robustness continues to improve after this entropy threshold,

[48] By replacing the expectation operator $\mathbb{E}$ with the conditional expectation $\mathbb{E}[\,\cdot \mid X_0]$ in the definition of $\mathcal{R}^\pi$ in (4.3), the quantity becomes a random variable for which the distribution can be estimated by sampling the initial state distribution $X_0 \sim \mathcal{N}(x_e^*, \sigma_e^2 I_d)$. In fact, taking the conditional expectation gives the difference of the standard *value functions* under $\mathbb{P}^\pi$ and $\mathbb{P}^{\pi, \epsilon_Y}$.

whereas the opposite trend is observed for KS (particularly with the maximal entropy coefficient chosen).

Hence, the sole introduction of entropy-regularisation in the objective function impacts the robustness. This behaviour difference between Lorenz and KS might be explained by the variability of the optimisation landscapes that can be observed with respect to the chosen underlying dynamics as underlined in Z. Ahmed et al. 2019.

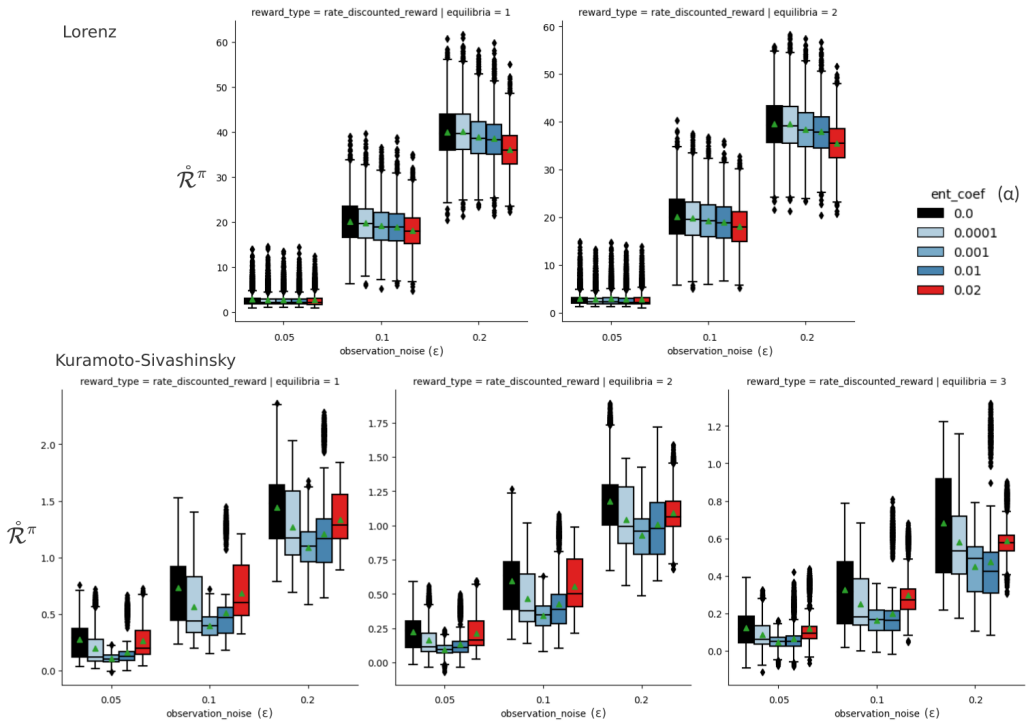


Figure 4.1: Distributional representation of the rate of excess risk under noise $\mathring{\mathcal{R}}^{\pi}$ conditioned on the $\alpha^{\mathcal{H}^i}$ used during optimisation for different initial state distribution $X_0 \sim \mathcal{N}(x_e^*, \sigma_e^2 Id)$. Each of the rows corresponds to one of the dynamical systems of interest. Each of the columns corresponds to one of the initial state distributions of interest. There are two non-zero fixed points (equilibria) x_e^* for Lorenz and three for KS. From top to bottom: KS; Lorenz.
For each box plot, three intensities σ_Y for the observation noise ϵ_Y are evaluated. As expected, when the uncertainty regarding the observable Y increases through the variance σ_Y of the observation signal noise ϵ_Y, the policy performance decreases globally ($\mathring{\mathcal{R}}^{\pi}$ increases). Moreover, the rate of excess risk under noise tends to decrease when $\alpha^{\mathcal{H}^i}$ increases in the Lorenz case, whereas it decreases up to a certain entropy threshold for KS before increasing again.

4.6.2 Maximum entropy as a norm-based regularisation on the policy

Norm-based complexity measures introduced in Section 4.4.2 are now evaluated. For a complexity measure $\mathcal{M}$ to be considered significant, it should be correlated with the robustness of the model.

Accordingly, the different norm-based measures presented in Section 4.4.2 are estimated. Figure 4.2 shows the layer-wise product norm of the policy actor network parameters ($\mathcal{M}(\pi_\theta, \mathcal{D}) = \Pi_{i=1}^{l} \|\theta_\mu^i\|_p$) w.r.t. to their associated entropy coefficient $\alpha^{\mathcal{H}^i}$ for all the 50 independently trained models.

Again, policies obtained with initial $\alpha^{\mathcal{H}^i} > 0$ exhibit a trend toward decreasing complexity measure values as $\alpha^{\mathcal{H}}$ increases up to a certain threshold of the entropy coefficient. Similarly to Section 4.6.1, the complexity measure continues to decrease after surpassing this threshold for the Lorenz system. On the other hand, in the KS case, $\mathcal{M}(\pi_\theta, \mathcal{D})$ increases again once its entropy threshold is reached, notably for the larger entropy coefficient.

Moreover, the measures tend to be much more concentrated when $\alpha^{\mathcal{H}^i} > 0$, especially in the case of KS (except for the higher $\alpha^{\mathcal{H}^i}$).

This may indicate that the entropy regularisation acts on the uncertainty of the policy parameters. Likewise, similar observations can be made for the total

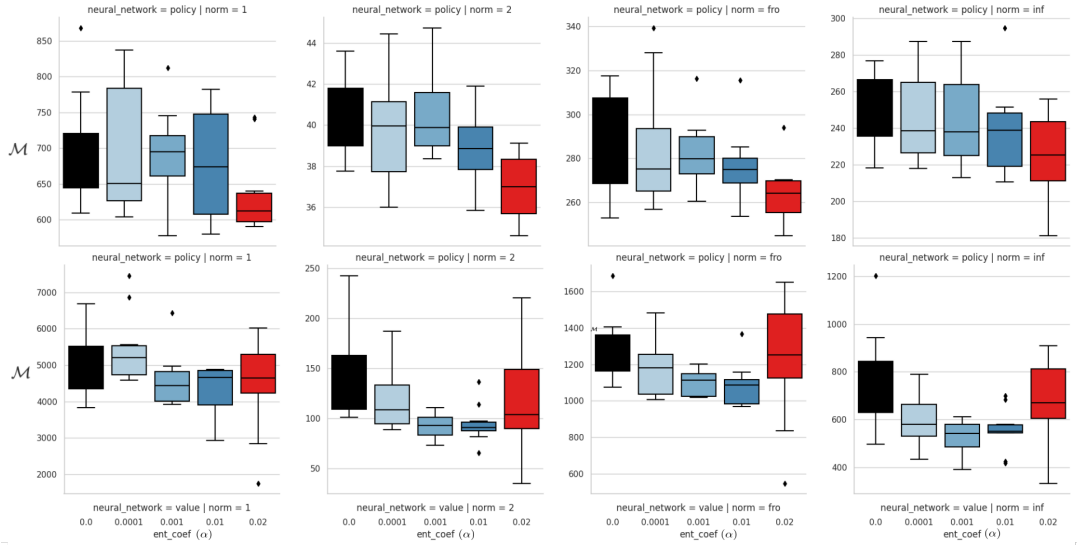


Figure 4.2: Measures of complexity $\mathcal{M}(\pi_\theta, \mathcal{D}) = \Pi_{i=1}^{l} \|\theta_\mu^i\|_p$ with $p = 1, 2, \infty, F$ conditioned on the $\alpha^{\mathcal{H}^i}$ used during optimisation. Each row corresponds to one of the dynamical systems of interest while columns represent a different norm order p. From top to bottom: Lorenz and KS.
For the Lorenz case, the barycenters of the measures tend to decrease when $\alpha^{\mathcal{H}^i}$ increases. Regarding KS, passing a threshold, the complexity increases with the entropy again. In addition, the measures are much more concentrated when $\alpha^{\mathcal{H}^i} > 0$. For $p = 2, F$, the separation of the measures w.r.t. the different $\alpha^{\mathcal{H}^i}$ is more pronounced.

norm of the parameters but are not introduced here for the sake of brevity.

Consequently, this experiment highlights an existing correlation between maximum entropy regularisation and norm-based complexity measures. As this complexity measure is linked to the Lipschitz continuity of the policy, one might wonder if the regularity of the policy is more directly impacted. This is the purpose of the next section.

4.6.3 Maximum entropy reduces the average Fisher-Information

Another regularity measure is considered: the average trace of the Fisher information ($\mathcal{M}(\pi_\theta, \mathcal{D}) = Tr(\mathcal{I}(\theta_\mu)) = Tr(-\mathbb{E}^{X\sim\rho, U\sim\pi_\theta(\cdot|X)}\left[\nabla^2_{\theta_\mu} \log \pi_\theta(U \mid X)\right])$). As discussed in 4.4.2, this quantity reflects the regularity of the policy and might be related to the flatness of the local minima of the objective function.

Figure 4.3 shows the distribution under π_θ of the trace of the state conditional Fisher Information of the numerical optimal solution $\theta^*_{\mu,\alpha^{\mathcal{H}^i}}$ for the policy w.r.t. the $\alpha^{\mathcal{H}^i}$ used during optimisation. In other words, a kernel density estimator of the distribution of $Tr(\mathcal{I}(\pi_{\theta^*_{\mu,\alpha^{\mathcal{H}^i}}}(\cdot \mid X)))$ when $X \sim \rho^{\pi_{\theta^*}}$ is represented. The results of this experiment suggest first, this distribution is skewed negatively and has a fat right tail. This means some regions of the support of $\rho^{\pi_{\theta^*}}$ provide FIM trace with extreme positive values, meaning the regularity of the policy may be poor in these regions of the state space.
A comparison of the distribution w.r.t. the different $\alpha^{\mathcal{H}^i}$ sheds further light on the relation between robustness and regularity. In fact, there appears to be a correspondence between the robustness, as indicated by the rate of excess risk under noise $\mathring{\mathcal{R}}^\pi$ shown in Figure 4.1 and the concentration of the trace distribution toward larger values (*i.e.* more irregular policies) when the model is less robust.

Meanwhile, under the considerations of 4.4.2 and since it is known that entropy regularisation favours flat minima in RL (Z. Ahmed et al. 2019),these experimental results support the hypothesis of an existing relationship between robustness, objective function flatness around the solution θ^* and conditional Fisher information of θ^*.

For a complementary point of view, a supplementary experiment regarding the sensitivity of the policy updates during training w.r.t. to different level of entropy is also presented in Section 4.7.

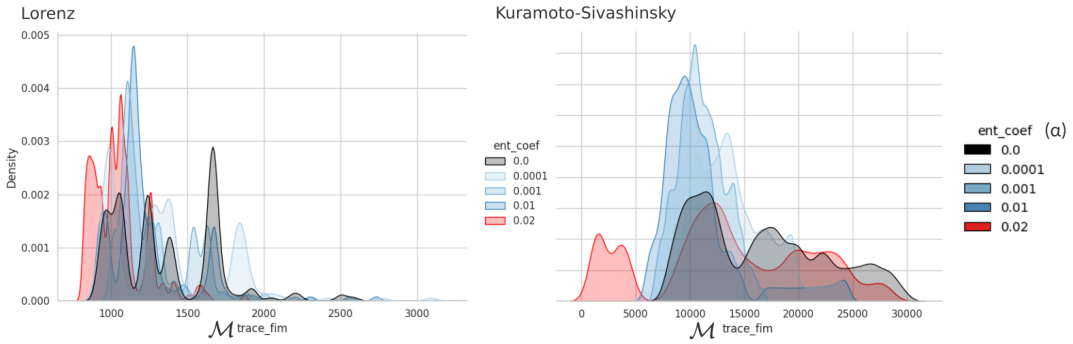


Figure 4.3: Distribution of the trace of the (conditional) Fisher information of the numerical optimal solution $\theta^*_{\mu,\alpha^{\mathcal{H}i}}$ for the policy w.r.t. the $\alpha^{\mathcal{H}i}$ used during optimisation. From left to right: Lorenz and KS environments. Colours: control experiment $\alpha^{\mathcal{H}i} = 0$ (black); intermediate entropy level $\alpha^{\mathcal{H}i}$ (blue); largest $\alpha^{\mathcal{H}i}$ (red).
A skewed distribution towards (relatively) larger values is observed for all controlled dynamical systems. Moreover, those right tails exhibit high kurtosis, especially for the control experiment (black) and the model with the larger entropy coefficient (red) for the KS system. Finally, solutions with intermediate entropy levels (blue) are much more concentrated—have lower variance than the others. About Lorenz, the barycenter of the more robust model (red) is shifted towards lower values than the others.

4.7 Complement: Weights sensitivity during training

This section is intended to provide complementary insights on the optimisation landscape induced by the entropy coefficient $\alpha^{\mathcal{H}}$ during training from the *conservative* or *trust region* policy optimisation point of view (Kakade and Langford 2002; Schulman, Levine, et al. 2015).

Let $\left(\theta_m^{\alpha^{\mathcal{H}}}\right)_{m=1}^{m_{\mathcal{D}}}$ be the sequence of weights of the policy during the training of the model for some initial entropy coefficient $\alpha^{\mathcal{H}}$. The conditional Kullback-Leibler divergence between the policy identified by the parameters $\theta_m^{\alpha^{\mathcal{H}}}$ and the subsequent policy defined by the parameters $\theta_{m+1}^{\alpha^{\mathcal{H}}}$ is given by

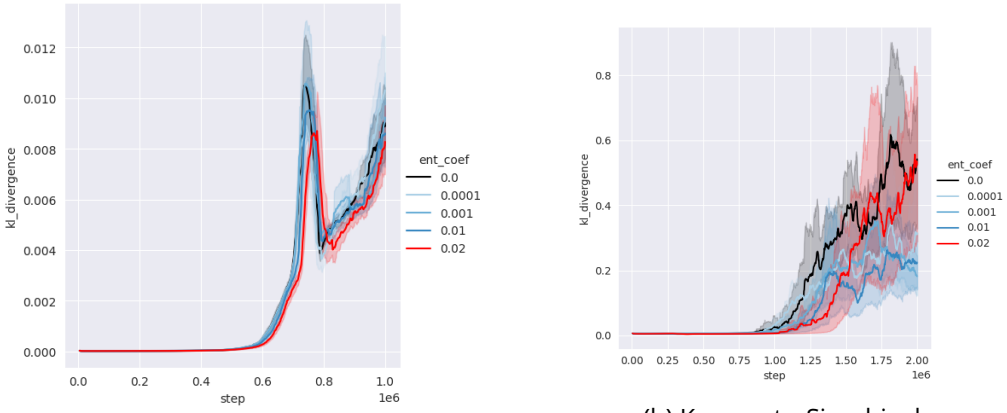
$$D_{\mathrm{KL}}\left(\theta_m^{\alpha^{\mathcal{H}}},\, \theta_{m+1}^{\alpha^{\mathcal{H}}}\right) = \mathbb{E}^{X\sim\rho}\left[\int_{\mathcal{U}} \log\left(\frac{\pi_{\theta_m^{\alpha^{\mathcal{H}}}}(du|X)}{\pi_{\theta_{m+1}^{\alpha^{\mathcal{H}}}}(du|X)}\right) \pi_{\theta_{m+1}^{\alpha^{\mathcal{H}}}}(du \mid X)\right].$$

The above quantity is a measure of the divergence from the policy at time m to the policy at time $m+1$. Thus it may provide information on the local stiffness of the optimisation landscape during training.

Figure 4.4 shows the evolution of the Kullback-Leibler divergence between two subsequent policies during training for the Lorenz and KS controlled differential equations. Regarding the Lorenz system, the maximal divergence is reached for the optimisation performed with the two lowest $\alpha^{\mathcal{H}i}$ while increasing entropy seems to slightly reduce the divergence. On the other hand, the

highest divergence values observed for the KS system are reached for $\alpha^{\mathcal{H}^i} = 0$ and the maximal entropy coefficient. This observation is coherent with the results of the previous sections and suggests that the entropy coefficient $\alpha^{\mathcal{H}}$ impacts the optimisation landscape during training.

Interesting questions regarding the optimisation landscape and its link with the Fisher Information (through the point of view of Information Geometry (Amari 1998)) are raised by the results of this section but are left for future work.



(a) Lorenz

(b) Kuramoto-Sivashinsky

Figure 4.4: Evolution of $D_{\text{KL}}\left(\theta_m^{\alpha^{\mathcal{H}}}, \theta_{m+1}^{\alpha^{\mathcal{H}}}\right)$ during training for the Lorenz and KS controlled differential equations. For Lorenz, the maximal divergence is reached for the optimisation performed with $\alpha^{\mathcal{H}^i} = 0$ and the second lowest $\alpha^{\mathcal{H}^i}$. Regarding KS, the highest divergence values are observed for $\alpha^{\mathcal{H}^i} = 0$ and the maximal entropy coefficient.

4.8 Discussion

In this study, the question of the robustness of maximum entropy policies under noise is studied. After introducing the notion of complexity measures from the statistical learning theory literature, numerical evidence supports the hypothesis that maximum entropy regularisation induces robustness under noise. Moreover, norm-based complexity measures are shown to be correlated with the robustness of the model. Then, the average trace of the Fisher Information is shown to be a relevant indicator of the regularity of the policy. This suggests the existence of a link between robustness, regularity and entropy regularisation.

5 Increasing Information for Model Predictive Control with Semi Markov Decision Processes

This chapter aims at addressing the problem of the sample complexity (see Section 1.4.1) of the learning process in the context of Learning-based Model Predictive Control (LB-MPC). This work led to a conference paper published in the proceedings of the 6th Annual Learning for Dynamics & Control Conference (L4DC24) (Hosseinkhan Boucher, Douka, et al. 2024).

5.1 Introduction

As discussed in the introduction of this thesis (Sections 1.2 and 1.3), *Machine Learning Control (MLC)* is an interdisciplinary area of statistical learning and control theory that solves model-free optimal control problems (Duriez, Brunton, and Noack 2016). Among the multiple approaches of the vast field of data-driven control, two classes have received notable attention by the ML community: *Learning-Based Model Predictive Control (LB-MPC)* (Hewing et al. 2020) and *Model-Based Reinforcement Learning (MB-RL)* (Abbeel, Quigley, and Ng 2006; Recht 2018; Moerland et al. 2022). The former refers to the combination of *Model Predictive Control (MPC)*, an optimisation method based on a sufficiently descriptive model of the system dynamics (Grüne and Pannek 2011), and learning methods which enable the improvement of the prediction model from recorded data while possibly modelling uncertainty (Aswani et al. 2013; Koller et al. 2019). The latter combines general function approximators such as linear models (Tsitsiklis and Van Roy 1997), or more generally neural networks (Sutton, McAllester, et al. 1999), with *Dynamic Programming (DP)* (R. E. Bellman 1957) principles to solve the underlying optimisation problem.

Despite the recent impressive results in learning complex dynamical models (Ha and Schmidhuber 2018), the *sample complexity* of the learning process remains a major issue in the field of data-driven control (Kakade 2003; G. Li et al. 2021, and see the references therein), in which the sample complexity is defined as the sample size required to learn a good approximation of the tar-

get concept (Mohri, Rostamizadeh, and Talwalkar 2018). For this reason, recent works (Mehta, Char, et al. 2022; Mehta, Paria, et al. 2022) in LB-MPC have focused on the design of exploration strategies based on the Information Theory concept of *Expected Information Gain (EIG)* or negative *Conditional Mutual Information (CMI)* (Lindley 1956). The resulting criterion allows for quantifying the gain of information given by a new state-control observation on the estimated optimal system trajectory. Hence, this tool can be used as an acquisition function to guide the exploration of the state-control space. The concept of acquisition function is borrowed from the field of *Bayesian Optimisation (BO)*. In particular, the work of Mehta, Char, et al. 2022 relies on the broader black-box BO framework of Neiswanger, K. A. Wang, and Ermon 2021.

In a setting where the data is collected along the trajectory of the dynamical system of interest, the diversity of the resulting dataset (which may be characterised by the quantity of information) is conditioned on the subsequent states of the system. Informally, the setting in which the sampling procedure is constrained by the current system state may introduce information redundancy if the system exhibits high auto-correlation or if the current state is in a slowly evolving region of the state space. Indeed, as shown in Figure 5.1 (auto-correlation from a perturbated fixed point of a controlled Lorenz 63' system), the auto-correlation from an initial state can be high on average for a long period of time while the control intensity allows reducing the correlation of the sequence of states.

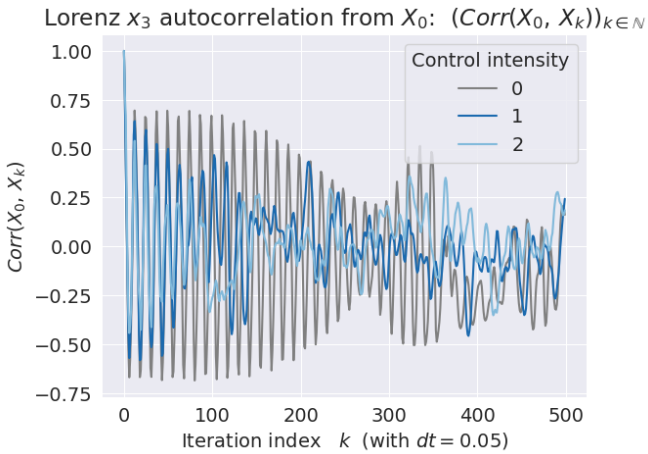


Figure 5.1: $(Corr(X_0, X_k))_{k \in \mathbb{N}}$ for the controlled Lorenz system x_3 component under multiple control intensities.

However, for dynamical systems characterised by a wide range of time scales, the notion of *temporal abstraction*, described in the following paragraphs, (Precup 2000; Machado et al. 2023) may play a key role in overcoming the issue mentioned here.

Abstraction in Artificial Intelligence refers to a broad range of techniques

in order to provide a more compact representation of the problem at hand (Boutilier and Dearden 1994; Banse et al. 2023). In the framework of *Markov Decision Process (MDP)*,the work of Sutton, Precup, and Singh 1999 sheds light on the limitation induced by standard MDP modeling: "There is no notion of a course of action persisting over a variable period of time. [...] As a consequence, conventional MDP methods are unable to take advantage of the simplicities and efficiencies sometimes available at higher levels of temporal abstraction."

Temporal abstraction can refer to the concept of selecting the right level of time granularity to facilitate the description of the world model to achieve a given task. In simpler words, in the present case, temporal abstraction is the idea of representing and reasoning about actions and states at different timescales and duration.

In the present work, temporal abstraction through *Semi-Markov Decision Processes (SMDP)* modeling is introduced to improve the informativeness of the sequential exploration of the state-control space. SMDP modeling is shown to obtain a better sample complexity of the dynamics model estimator. This article thus extends the previous work of Mehta, Paria, et al. 2022 by introducing temporal abstraction to the acquisition function. The chapter is organised as follows. Section 5.2 reviews the related works. Section 5.3 introduces the problem setting. Section 5.4 presents the hypothesis and the experimental setup while Section 4.6 presents the results and Section 5.6 concludes the chapter.

5.2 Related Works

Information Driven Model-Based Control The foundations of the *Bayesian Experimental Design* have been laid by the seminal work of Lindley 1956 where the author presents a measure of the information provided by an experiment. More recently, MacKay 1992 termed *Expected Information Gain (EIG)* a measure of the information provided by an observation allowing, in his own terms, to *actively select particularly salient data points*. In the field of LB-MPC, such a criterion has been used to cherry-pick the most informative state-control pair to learn the dynamics of the system (Mehta, Char, et al. 2022; Mehta, Paria, et al. 2022). Their work is based on the broader Bayesian Optimisation method of Neiswanger, K. A. Wang, and Ermon 2021 designed to optimise "blackbox" functions. An extensive review of Bayesian Optimisation and its applications is available in this latter paper.

Learning-Based Model Predictive Control The history of learning based modelling may be traced back to the seminal work by Stratonovich 1960 in probability theory which stimulated several contributions, notably the work of Kalman and Bucy 1961, that were to compose a body of work generally referred to as

filtering theory. More recently, Kamthe and M. Deisenroth 2018 model the system dynamics with Gaussian Processes (GP) and use MPC for data efficiency. GPs are also used in the PILCO model (M. Deisenroth and C. Rasmussen 2011) which has a high influence in MB-RL. Koller et al. 2019 model the uncertainty of the system dynamics for safe-RL. The work of Bonzanini and Mesbah 2020 presents a stochastic LB-MPC strategy to handle this uncertainty.

Semi-Markov Decision Processes Temporal abstraction in Reinforcement Learning was pioneered in Sutton 1995 and Precup and Sutton 1997; Precup 2000. Specifically, Sutton 1995 proposed learning a model and value function at different levels of temporal abstraction. The actions in SMDPs take variable amounts of time and are intended to model temporally-extended courses of action. Recent works for continuous-time control use variants of Neural Ordinary Differential Equations to model dynamics delays (Du, Futoma, and Doshi-Velez 2020; S. Holt et al. 2023). A classical use of SMDP is for queueing control and equipment maintenance (Puterman 2014) where time-delays are prominent.

5.3 Problem Setting

5.3.1 Control Model

This work considers a control model given by the following d-dimensional discrete time dynamical system X (Duflo 1997) on a probability space $(\Omega, \mathcal{F}, \mathbb{P})$ defined by

$$\begin{aligned} X_{k+1} &= F\left(X_k,\, U_k,\, v_k\right) \\ X_0 &\sim \mathcal{N}(x_e,\, \sigma_e^2 Id_{d_X}) \end{aligned} \tag{5.1}$$

with $X_k \in \mathcal{X}$, $U_k \in \mathcal{U}$ and $v_k \in \mathcal{V}$ for any $k \in \mathbb{N}$, where $\mathcal{X}$, $\mathcal{U}$ and $\mathcal{V}$ are respectively the corresponding state, control and disturbance spaces. The initial state starts from a reference state $x_e \in \mathcal{X}$ (a system equilibrium or *fixed point*[49]) on which centered Gaussian noise with diagonal covariance is additively applied, $X_0 \sim \mathcal{N}(x_e,\, \sigma_e^2 Id_{d_X})$. The *i.i.d.* random process $(v_k)_{k\in\mathbb{N}}$ is such that v_k is independent of all previous states and controls for any $k \in \mathbb{N}$. The distribution of v_k for any $k \in \mathbb{N}$ is denoted by $\mathbb{P}_v$. Coupled with the dynamics, an instantaneous cost function $c : X \times U \to \mathbb{R}_+$ is also given to define the control model.

In the sequel, it will be convenient to define the control model as a *Markov Control Model (MCM)* (O. Hernández-Lerma 1989) defined by the following transition probability $\mathcal{P}$ on $\mathcal{X} \times \mathcal{U}$:

$$\mathcal{P}(B_{\mathcal{X}},\, (x,u)) = \int_{\mathcal{V}} 1_{B_{\mathcal{X}}}\left(F(x,u,v)\right) \mathbb{P}_v(dv) = \mathbb{P}_v(\{v \in \mathcal{V} \mid F(x,u,v) \in B_{\mathcal{X}}\}) \tag{5.2}$$

[49] In this work a fixed point is considered as a point of the state space $x_e \in \mathcal{X}$ such that $F(x_e, 0, 0) = x_e$.

for any $B_{\mathcal{X}} \in \mathcal{B}(\mathcal{X})$ (Borel σ-algebra) and $(x, u) \in \mathcal{X} \times \mathcal{U}$. The function 1 is the indicator function.

Hence, the conditional distribution of X_{k+1} given X_k and U_k is given by

$$\mathcal{P}(B_{\mathcal{X}}, (x, u)) = \mathbb{P}(X_{k+1} \in B_{\mathcal{X}} \mid X_k = x, U_k = u) \tag{5.3}$$

for any $B_{\mathcal{X}} \in \mathcal{B}(\mathcal{X})$.

Additionally, in this context, a *policy* π is a transition probability on $\mathcal{U}$ given $\mathcal{X}$, *i.e.* a distribution on controls conditioned on states. In the rest of this chapter, $\pi(du \mid x) = \delta_{\{u\}}$ is the Dirac measure at u. Hence the notation is simplified to $\pi(x) = u$.

Together, a control model, a policy π and an initial distribution on $\mathcal{X}$ define a stochastic process with distribution $\mathbb{P}^{\pi}$ on the space of trajectories $(\mathcal{X} \times \mathcal{U})^K$. The distribution of the process is given by $\mathbb{P}(dx_0 du_0 dx_1 \dots) = \mathbb{P}_{X_0}(dx_0)\pi(du_0 \mid dx_0)\mathcal{P}(dx_1 \mid dx_0, du_0) \dots$ More details on the stochastic process are given in Section 2.3.3 and Onésimo Hernández-Lerma and Lasserre 1996; Puterman 2014. Lastly, the history process $(H_k)_{k\in\mathbb{N}}$ is defined as $H_k = (X_0, U_0, \dots, X_k)$ for any $k \in \mathbb{N}$. When $k = K$, H_K is called the *trajectory* of the process. The process $(X_k, U_k, X_{k+1})_{k\in\mathbb{N}}$ is called the *transition process* and the marginal process $(X_k)_{k\in\mathbb{N}}$ is called a *Markov Decision Process (MDP)*.

5.3.2 Control Problem

The studied control problem is to find a policy π^* which minimises the following performance criterion

$$J^{\pi} = \mathbb{E}^{\pi}\left[\sum_{k=0}^{K} c(X_k, U_k)\right] \tag{5.4}$$

where $K \in \mathbb{N}$ is a given time-horizon and $\mathbb{E}^{\pi}$ denotes the expectation under the probability measure $\mathbb{P}^{\pi}$. Here, the quantity J^{π} is called the value function or objective function (see Section 3.1.4). The history process under π^* is called the optimal history process and is denoted by $(H_k^*)_{k\in\mathbb{N}}$ and the random variable H_K^* is called the *optimal trajectory*.

In this work, the optimal policy π^* is estimated with *Model Predictive Control (MPC)* applied on a model of the dynamics.

The MPC procedure (see Section 3.3.3) is performed with the *iCEM* algorithm, an improved version of the *Cross Entropy Method (CEM)* (Rubinstein and Kroese 2004; Pinneri et al. 2021), a zeroth order optimisation algorithm based on Monte-Carlo estimation.

5.3.3 Gaussian Process Modeling

The use of *Gaussian Process (GP)* regression to model relevant quantities of controlled dynamical systems has long been proposed (Kuss and C. Rasmussen

2003; M. Deisenroth and C. Rasmussen 2011; Kamthe and M. Deisenroth 2018) notably for its distributional nature thus its ability to model uncertainty. By definition, a GP is a stochastic process (here indexed by $\mathcal{X} \times \mathcal{U}$) such that any finite collection of random variables has a joint Gaussian distribution.

Continuing from the aforementioned papers, Gaussian Process regression is used to model the transition probability $\mathcal{P}$ with a model estimator $\hat{\mathcal{P}}_{\mathcal{D}}$ such that

$$\hat{\mathcal{P}}_{\mathcal{D}}(\,\cdot\,,\,(x,u)) \sim \mathcal{N}\left(\mu(x,u),\,\Sigma\left((x,u),\,(x,u)\right) \mid \mathcal{D}\right) \tag{5.5}$$

where μ and Σ are respectively the mean and covariance functions of the GP and $\mathcal{D}$ is a dataset of observations from the transition process $(X_k, U_k, X_{k+1})_{k\in\mathbb{N}}$. The distribution $\hat{\mathcal{P}}_{\mathcal{D}}$ of Equation (5.5) is the predictive posterior distribution of the GP conditioned on the dataset $\mathcal{D}$ (the reader is referred to C. E. Rasmussen and C. K. I. Williams 2006 for more details on GP regression). The processes $\hat{X}$, $\hat{U}$ and $\hat{H}$ are respectively the state, control and history processes under the approximate model and the same rules of notation apply as for the original processes. The MPC policy obtained with the approximate model $\hat{\mathcal{P}}_{\mathcal{D}}$ is denoted by $\widehat{\pi}^{\mathsf{MPC}}$. The history process under $\widehat{\pi}^{\mathsf{MPC}}$ is denoted by $\widehat{H}^{\mathsf{MPC}} = (\widehat{H}_k^{\mathsf{MPC}})_{k\in\mathbb{N}}$ and the objective function under $\widehat{\pi}^{\mathsf{MPC}}$ is denoted by $\widehat{J}^{\mathsf{MPC}}$.

Notably, *this work focuses on the sample complexity required to estimate a model $\hat{\mathcal{P}}_{\mathcal{D}}$ of the true dynamics $\mathcal{P}$ accurate enough to obtain a MPC policy $\widehat{\pi}^{\mathsf{MPC}}$ that is close to the optimal policy π^*.*

Hence, two time units are considered: the sampling iteration n which represents the number of observations gathered from the system so far, and the time index k of the current state X_k of the underlying dynamical system X. It is supposed in the following that $n \leq k$: it is not possible to gather more observations than the number of time steps of the system.

5.3.4 Expected Information Gain

For a fixed sampling budget n and a fixed configuration (*e.g.* the horizon K^{MPC}, the number of samples for the Monte-Carlo estimation of the cost or the other hyper-parameters of the iCEM algorithm) to perform the MPC procedure π^{MPC}, the control performance mainly lies in the quality of the model estimator $\hat{\mathcal{P}}_{\mathcal{D}_n}$. It depends on two main elements: the choice a priori of the mean and kernel functions μ and Σ and the collection $\mathcal{D}_n$ of n observations. From the work of Mehta, Char, et al. 2022; Mehta, Paria, et al. 2022, the selection of the observations can be guided by the maximisation of the *Expected Information Gain (EIG)* on the optimal trajectory.

Let suppose the time iteration k of the underlying observed process X is equal to the number of samples gathered, *i.e.* $k = n$ and the dataset is already collected[50] at the sampling iteration n such that $\mathcal{D}_n = ((x_i, u_i, x'_i))_{i=0}^{n-1}$ and denote by (X_n, U_n) a new random state-control pair to draw from the system. The

[50] In this specific case of $k = n$, the dataset $\mathcal{D}_n$ simply contains the whole past trajectory of

goal is to select the state-control pair (x, u) that maximises the *Expected Information Gain* EIG on the optimal trajectory which is defined by

$$\mathsf{EIG}_n(x,u) \coloneqq \mathcal{H}_1 - \mathbb{E}_{\mathbb{P}_{X_{n+1}|\mathcal{D}_n,\, X_n=x,\, U_n=u}}\left[\mathcal{H}_2\left(x,u,X_{n+1}\right)\right] \tag{5.6}$$

with

$$\mathcal{H}_1 \coloneqq \mathcal{H}\left[\hat{H}_K^* \mid \mathcal{D}_n\right] \tag{5.7}$$

$$\mathcal{H}_2\left(x,u,X_{n+1}\right) \coloneqq \mathcal{H}\left[\hat{H}_K^* \mid \mathcal{D}_n, X_n = x, U_n = u, X_{n+1}\right] \tag{5.8}$$

where $\mathcal{H}$ denotes the differential entropy of a random variable. In other words, given a level of uncertainty $\mathcal{H}\left[\hat{H}_K^* \mid \mathcal{D}_n\right]$ on the optimal trajectory $\hat{H}_K^*$, the EIG measures the reduction of this uncertainty when the dataset of the model estimator is augmented with the transition tuple (x, u, X_{n+1}).

An intriguing interpretation can be made by noticing that (5.6) is also equal to the negative *Conditional Mutual Information (CMI)* (Pinsker 1964; Cover and Thomas 2006) of the optimal trajectory $\hat{H}_K^*$ and the new state X_{n+1} given the dataset $\mathcal{D}_n$ and the state-control pair (X_n, U_n).[51] Thus, maximising the EIG is equivalent to minimising the CMI between the optimal trajectory and the new transition tuple hence tending to draw new states sharing less information with the optimal trajectory conditioned on the dataset $\mathcal{D}_n$ and the event $(X_n = x, U_n = u)$. Indeed, by definition, the CMI quantifies the independence between the distribution of the optimal trajectory and the distribution of the new state given both the dataset and the current state-control pair.

By symmetry of the EIG, a more tractable formulation is given by

$$\mathsf{EIG}_n(x,u) \coloneqq \mathcal{H}_1'\left(x,u\right) - \mathop{\mathbb{E}}_{\mathbb{P}_{\hat{H}_T^*|\mathcal{D}_n}}\left[\mathcal{H}_2'(x,u,\hat{H}_T^*)\right] \tag{5.9}$$

with

$$\mathcal{H}_1'(x,u) \coloneqq \mathcal{H}\Big[X_{n+1} \mid \mathcal{D}_n,\, X_n = x,\, U_n = u\Big] \tag{5.10}$$

$$\mathcal{H}_2'(x,u,\hat{H}_T^*) \coloneqq \mathcal{H}\left[X_{n+1} \mid \mathcal{D}_n, X_n = x,\, U_n = u,\, \hat{H}_T^*\right] \tag{5.11}$$

It is in practice estimated by Monte-Carlo sampling as detailed in Section 5.4.

In the original work of Mehta, Char, et al. 2022, the EIG is maximised with a greedy Monte-Carlo algorithm (uniform sampling) that selects the next state-control pair (x, u) to interact with the true system and subsequently update the dataset $\mathcal{D}_n$ with the new transition tuple $(x,\, u,\, x')$ where x' is sampled from the

X, it is a realisation of H_n, in other words $\mathcal{D}_n = H_n(\omega)$ for some random outcome $\omega \in \Omega$.

[51] Here and after, a slight abuse of notation is made as the dataset $\mathcal{D}_n$ should be written $\mathcal{D}_n = ((x_i, u_i, x'_i))_{i=0}^{n-1}$ since the sole random quantities are X_n and $\hat{H}_K^*$ but it is omitted for the sake of readability.

true transition probability $\mathcal{P}(\,\cdot\,,(x,u))$. It assumes any state-control pair (x,u) can be evaluated and queried at any time step. The authors' algorithm is called *Bayesian Active Reinforcement Learning (BARL)*; the dataset and EIG obtained with this algorithm are denoted by $\mathcal{D}_n^{\text{BARL}}$ and EIG^{BARL} respectively. In this setting, the dataset support is the whole state-control space, $\text{Supp}(\mathcal{D}_n^{\text{BARL}}) = (\mathcal{X}\times\mathcal{U}\times\mathcal{X})^n$.

However, in many real-world applications, the system is not always controllable and the state-control pairs that can be queried are limited to a subset induced by the system trajectory. This constraint has been considered in the work following the original paper (Mehta, Paria, et al. 2022) where the authors proposed to restrict the dataset support to the trajectory of the system. This second algorithm is called[52] *Trajectory Information Planning (TIP)* and similarly the dataset and EIG obtained with this algorithm are denoted by $\mathcal{D}_n^{\text{TIP}}$ and EIG^{TIP} respectively.

In this case, the dataset support is limited to the trajectory of the system, $\text{Supp}(\mathcal{D}_n^{\text{TIP}}) \subseteq \{((x_k,\,u_k,\,x_{k+1}))_{k=1}^n \in (\mathcal{X}\times\mathcal{U}\times\mathcal{X})^n \mid \exists (v_k)_{k=1}^n \in \mathcal{V}^n,\, x_{k+1} = F(x_k,\,u_k,\,v_k),\, 0\leq i\leq n\} \subseteq (\mathcal{X}\times\mathcal{U}\times\mathcal{X})^n = \text{Supp}(\mathcal{D}_n^{\text{BARL}})$. This set inclusion implies that the optimal EIG obtained with TIP is lower than the one obtained with BARL provided the transition probability estimator $\hat{\mathcal{P}}_{\mathcal{D}_n}$ are the same for both algorithms for a fixed current state $x\in\mathcal{X}$, $\max_{\{(x,u'),\,u'\in\mathcal{U}\}}\text{EIG}(x,u') \leq \max_{\{(x',u')\in\mathcal{X}\times\mathcal{U}\}}\text{EIG}(x',u')$.

Besides, the latter algorithm (TIP) does not take into account the potential benefits of including dynamics time scales in the sampling process. In the next section, an extension of the TIP algorithm is proposed to increase the EIG for each of the sampling iterations through the *introduction of delayed state-control pairs in the setting of Semi-Markov Decision Processes (SMDP)*. The new algorithm builds upon TIP by considering the inclusion of temporally-extended actions in the data-collection procedure to reach more distant system states that are not reachable with the original TIP algorithm, hence increasing the amount of information gathered from the system. A similar use of action repetition improves learning in Deep-RL (Sharma, A. S. Lakshminarayanan, and Ravindran 2017; A. Lakshminarayanan, Sharma, and Ravindran 2017).

5.3.5 Semi-Markov Decision Processes Extension

A formal definition of *temporal abstraction* is given through the concept of *options* defined by Sutton, Precup, and Singh 1999 where it refers to *temporally extended courses of action*. This concept has been shown by Parr 1998 to be equivalent to the construction of *Semi-Markov Decision Processes (SMDP)* which are defined below.

Let call *decision epoch* the time index k of the underlying dynamics $(X_k)_{k\in\mathbb{N}}$

[52] It is important to mention that the main asset of TIP is to provide a whole trajectory as input to the EIG, which is not used in this work. Thus, only the property of querying observation by following the trajectory of the system is used here.

defined by equation (5.1). *Semi-Markov Control Models (SMCM)* generalise the concept of MCM by letting the decisions be random variables. Indeed, consider a strictly increasing random sequence $(\kappa_j)_{j\in\mathbb{N}}$ of integers. The random quantities $\eta_j = \kappa_j - \kappa_{j-1}$ with support in some finite space $\mathcal{S} \subsetneq \mathbb{N}\setminus\{0\}$ are called *inter-decision times* and the random index κ_j are called *random decision epochs*. The resulting stochastic process $(X_{\kappa_j})_{j\in\mathbb{N}}$ is called a *semi-Markov Decision Process*. For a more detailed probabilistic construction, see Puterman 2014, p. 534 and O. Hernández-Lerma 1989, p. 15.

In the scope of this work, *SMDP are used to model the temporal extension of the control process*. The corresponding SMCM is introduced by first extending the control space from $\mathcal{U}$ to $\mathcal{U}\times\mathcal{S}$ such that the temporal extension of the control is encoded in the last coordinate of the control tuple, and the new dynamics is given by $\mathcal{P}^{\text{SMDP}}(dx' \mid (x,(u,\sigma))) = \mathbb{P}(X_{k+\sigma} \mid X_k = x, U_{k:k+\sigma-1} = u)$ where $U_{k:k+\sigma-1} = u$ means that *the control process is constant* between k and $k+\sigma-1$. The latter definition illustrates the fact that during the inter-decision time $\eta = \sigma$, the control process is constant and equal to u.

From now on, this construction allows to enlarge the support of the dataset $\mathcal{D}_n$, for a fixed number of observations n while maintaining a rollout, trajectory-based sampling procedure. Indeed, the dataset support is now $\text{Supp}(\mathcal{D}_n^{\text{SM-TIP}}) \subseteq \{((x_{k_j}, u_{k_j}, x_{k_j+1}))_{j=1}^n \in (\mathcal{X}\times\mathcal{U}\times\mathcal{X})^n \mid \exists (v_k)_{k=1}^{n\sup(\mathcal{S})} \in \mathcal{V}^{n\sup(\mathcal{S})}, x_{k+1} = F(x_k,u_k,v_k),\ 0\le k\le n\sup(\mathcal{S}),\ (k_j)_{j=1}^n \in \mathcal{S}^n,\ k_j < k_{j+1}\}$, the transitions tuples extracted from the set of all possible subsequences of the trajectory up to the maximal reachable time value.

Therefore, $\text{Supp}(\mathcal{D}_n^{\text{TIP}}) \subseteq \text{Supp}(\mathcal{D}_n^{\text{SM-TIP}})$. Consequently, this suggests an extension of the EIG to the SMDP setting. Let $\sigma \in \mathcal{S}$ be an inter-decision time and $\mathcal{D}_n^{\text{SM-TIP}}$ be the dataset under the SMDP setting at the sampling iteration n, the resulting $\text{EIG}_n^{\text{SM-TIP}}(x,(u,\sigma))$ is defined as

$$\text{EIG}_n^{\text{SM-TIP}}(x,(u,\sigma)) \coloneqq \mathcal{H}_1''(x,u,\sigma) - \mathbb{E}_{\mathbb{P}_{\hat{H}_T^*|\mathcal{D}_n}}\Big[\mathcal{H}_2''(x,u,\sigma,\hat{H}_T^*)\Big] \tag{5.12}$$

$$\mathcal{H}_1''(x,u,\sigma) \coloneqq \mathcal{H}\Big[X_{\kappa_n+\sigma+1}|\mathcal{D}_n,\ X_{\kappa_n}=x, U_{\kappa_n:\kappa_n+\sigma}=u, \kappa_n\Big] \tag{5.13}$$

$$\mathcal{H}_2''(x,u,\sigma,\hat{H}_T^*) \coloneqq \mathcal{H}\Big[X_{\kappa_n+\sigma+1}\,|\,\mathcal{D}_n, X_{\kappa_n}=x, U_{\kappa_n:\kappa_n+\sigma}=u, \hat{H}_T^*, \kappa_n\Big] \tag{5.14}$$

Hence, this measure allows the introduction of temporal abstraction in the sampling procedure by considering the inter-decision delay to increase the potential information gain. However, despite being tractable in trajectory rollout settings, the metric defined by (5.12) needs to look ahead in the future to be computed (non-causal). Last, note that $\text{EIG}^{\text{SM-TIP}}(x,u,1) = \text{EIG}^{\text{TIP}}(x,u)$.

5.4 Method and Experiments

The main objective of this work is to demonstrate the increase in the total information gathered from a system with the introduction of temporal abstraction

via the $\text{EIG}^{\text{SM-TIP}}$ measure. To this end, a comparison between the original TIP algorithm and the proposed SMDP extension is performed on two controlled dynamical systems, the Inverted Pendulum (Trélat 2005) and the Lorenz Attractor (Vincent and Yu 1991).

The algorithm controls the path of the dynamical system $(X_k)_{k\in\mathbb{N}}$ and collects observations $(X_i, U_i, X_{i+1})_{i=0}^{n-1}$ to populate the dataset $\mathcal{D}_n$ and improve the GP transition probability estimator $\hat{\mathcal{P}}_{\mathcal{D}_n}$. The indices n and k are respectively the sampling iteration and the time index of the underlying dynamical system $(X_k)_{k\in\mathbb{N}}$. The TIP algorithm supposes $n = k$ (data collected at each time step) while $n \leq k$ (there are time steps where no data is collected) for the SMDP extension. In the SMDP case, the inter-decision time η_n rules the optional sampling procedure which defines the random decision epochs $\kappa_n = \kappa_{n-1} + \eta_n$. The random decision epochs κ_n define when the algorithm can query the system $(X_k)_{k\in\mathbb{N}}$.

To estimate $\text{EIG}_n^{\text{SM-TIP}}$, a collection of bootstrapped future states, candidate control points and inter-decision times are sampled. The bootstrapped future states $X_{\kappa_n+\sigma} = x_\sigma$ are estimated with the GP model. This may lead to a bias in the estimation of the EIG due to the bootstrapping error. The candidate control points and inter-decision times (u, σ) are sampled from a uniform distribution Unif $(\mathcal{U} \times \mathcal{S})$ at time κ_n to solve $\arg\max_{(u,\sigma)\in\mathcal{U}\times\mathcal{S}} \text{EIG}_n^{\text{SM-TIP}}(x_\sigma, (u, \sigma))$. In this work, $\mathcal{S} = \{1, \ldots, \sigma_{\max}\}$ for some $\sigma_{\max} \in \mathbb{N}$. The $\text{EIG}_n^{\text{SM-TIP}}$ is estimated by the Monte-Carlo estimator $\widehat{\text{EIG}}_n^{\text{SM-TIP}}(x, (u, \sigma))$ given by

$$\mathcal{H}\left[X_{\kappa_n+\sigma+1} \mid \mathcal{D}_n, X_{\kappa_n} = x, U_{\kappa_n+\sigma} = u, \kappa_n\right] - \frac{1}{m}\sum_{i=1}^{m} \widehat{\mathcal{H}}_2''(x, u, \sigma, \hat{H}_{k_i}^{\text{MPC}}) \tag{5.15}$$

with

$$\widehat{\mathcal{H}}_2''(x, u, \sigma, \hat{H}_{k_i}^{\text{MPC}}) \coloneqq \mathcal{H}\left[X_{\kappa_n+\sigma+1} \mid \mathcal{D}_n, X_{\kappa_n} = x, U_{\kappa_n+\sigma} = u, \hat{H}_{k_i}^{\text{MPC}}, \kappa_n\right] \tag{5.16}$$

where m is the number of Monte-Carlo samples of the optimal trajectory $\hat{H}_{k_i}^{\text{MPC}}$ under $\hat{\mathcal{P}}_{\mathcal{D}_n}$. The entropy values are easily computed since the conditional distribution of the new state given the dataset, and the current state-control pair is a Gaussian distribution with mean and covariance given by the GP posterior. More details on this procedure and the settings used are available in the paper of Mehta, Paria, et al. 2022.

Every two sampling iterations n, the MPC policy $\widehat{\pi}^{\text{MPC}}$ is evaluated on the true system and the objective function is computed. Four independent experiments with different maximal inter-decision time $\sigma_{\max} \in \{1, 2, 4, 8\}$ are performed. For each of the experiments, the algorithm is run for 10 independent trials (seeds) to alleviate the variability proper to data-driven control methods (Henderson et al. 2018). The cost function is defined as $(x, u) \mapsto c(x, u) \coloneqq \|x\|^2$ in the case of the Lorenz attractor while the classic Gym (Brockman et al. 2016) cost function (also norm-based) is used for the Inverted Pendulum. The sampling

budget is set to $n_{\max} = 100$ for the Lorenz system and $n_{\max} = 200$ for the Inverted Pendulum. To implement the SMDP, the system is stepped forward in time with the action kept constant during inter-decision times. Details on the implementation and experimental settings are available on https://github.com/ReHoss/lbmpc_semimarkov.

5.5 Results

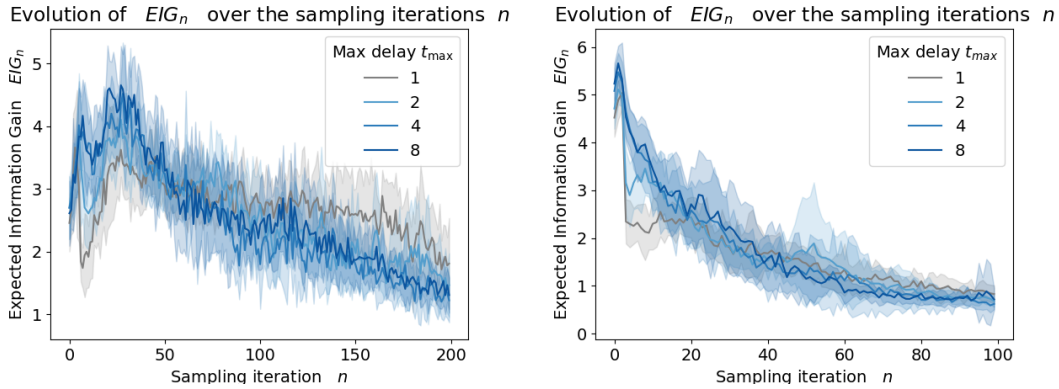
Among the relevant quantities to be reported, the evolution of the EIG, the interdecision times and the evaluation of the objective function are of interest to question the hypothesis raised in Section 5.4.

First, the evolution of the amount of information gathered during sampling through a comparison of $(\mathrm{EIG}_n^{\mathsf{TIP}})_{n=1}^{n_{\max}}$, and $(\mathrm{EIG}_n^{\mathsf{SM\text{-}TIP}})_{n=1}^{n_{\max}}$ presented in Figure 5.2 to assess the impact of the SMDP extension. Second, the corresponding inter-decision times $(\eta_n)_{n=1}^{n_{\max}}$ are shown in Figure 5.3 to evaluate the necessity of temporal abstraction. Lastly, the evolution of the objective function $J^{\widehat{\pi}^{\mathsf{MPC}}}$ from 5 fixed initial conditions X_0 is shown as a function of the sampling iteration n in Figure 5.4 to analyse the effective results of the proposed method. For all the figures, the shaded area represents the standard error over the 10 independent trials.

About the first point, one can observe that in all cases, the EIG is larger for SM-TIP than for TIP ($\sigma_{\max} = 1$) until one-fourth of the sampling budget is reached. This suggests that the SMDP extension is beneficial to the information gathering process at the beginning of the sampling procedure. This may be explained by the fact that the inter-decision times allow to de-correlate the collected states via the same mechanism illustrated in Figure 5.1. Note also that, in the case of Lorenz (Figure 5.2a), the EIG after approximately half of the sampling procedure is superior for TIP than SM-TIP since more information (state-actions pairs minimising the mutual information) remain to be gathered.

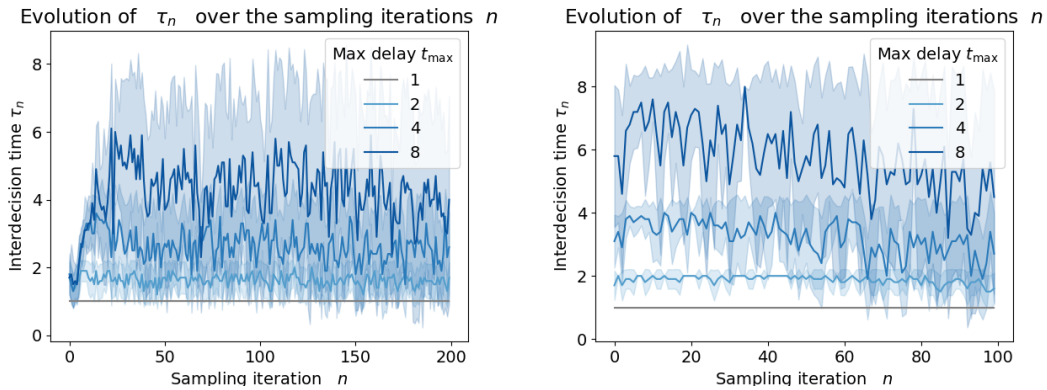
Examining the chosen inter-decision times $(\eta_n)_{n=1}^{n_{\max}}$, it can first be observed that globally $\eta_n > 1$ for the SMDP algorithms (where $\sigma_{\max} > 1$). This shows that the sequential maximal EIG is approximately reached for inter-decision times that are larger than the original MDP decision times. This confirms the relevance of temporal abstraction to increase the information gathering process. However, the inter-decision times are not necessarily always equal to $\sigma_{\max}$, suggesting the more informative observations are not always the temporally most distant ones.

Moving on to the objective function, in the case of the Lorenz system (Figure 5.4a), the evaluation performances show the learning speed is greater for the SM-TIP settings ($\sigma_{\max} > 1$) than for the TIP setting ($\sigma_{\max} = 1$). For the Pendulum case (Figure 5.4b), except for the SM-TIP setting where $\sigma_{\max} = 8$, the proposed approach shows better sample complexity since very few iterations



(a) Lorenz (b) Pendulum

Figure 5.2: Evolution of the Expected Information Gain $\text{EIG}^{\text{SM-TIP}}$ over the number of sampling iterations.



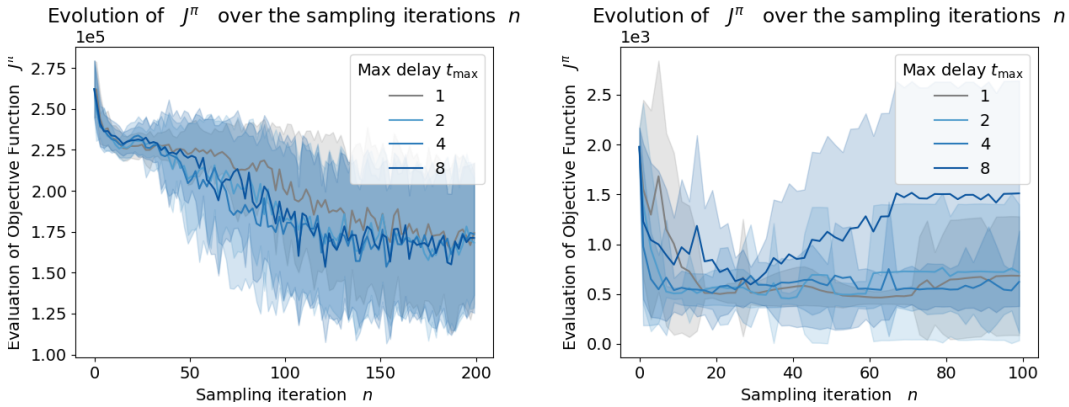
(a) Lorenz (b) Pendulum

Figure 5.3: Inter-decision time η chosen by the SMDP during training.

are required to reach optimality (light blue curves ($\sigma_{\max} \in \{2, 4\}$) are below the grey curve ($\sigma_{\max} = 1$) for the first (up to $n = 20$) sampling iterations. Furthermore, one of the reasons the $\sigma_{\max} = 8$ fails to achieve optimal performances is likely the *bootstrapping prediction error* (not shown in this document) which increases with $\sigma_{\max}$. Indeed, as mentioned in Section 5.4 due to the non-causal property of $\text{EIG}^{\text{SM-TIP}}$, there exists a trade-off between the temporal extension of the dynamics to reach the new region of the state space and the bootstrapping error which increases with the temporal extension.

5.6 Conclusion

This study demonstrates that, when restricted to the trajectory of the system, the total information gathered for a given sampling budget can be increased by



(a) Lorenz

(b) Pendulum

Figure 5.4: Evolution of the objective function $J^{\widehat{\pi}^{\text{MPC}}}$ to evaluate the system during training.

introducing temporal abstraction through the usage of SMDPs. Results show that learning the dynamics of the Inverted Pendulum and the Lorenz system is more data-efficient with the use of temporally-extended actions.

Future work may extend this methodology to more complex systems, leveraging the flexibility of SMDPs. These systems may have the potential to reach highly informative regions and efficiently capture rapid changes in system dynamics, as the information content can be increased when considering the time resolution as a decision variable.

In summary, this work offers a concise yet comprehensive glimpse into the potential of SMDPs in Model Predictive Control. The results on known systems establish a robust foundation for broader applications and unveil potential future advancements in control strategies.

6 Distributional Reinforcement Learning is Sample Efficient

This chapter presents the main results obtained during the first months of the PhD project, on the application of Distributional Reinforcement Learning to chaotic dynamical systems.

6.1 Introduction

A modern approach called *Distributional Reinforcement Learning (D-RL)* defined in Bellemare, Dabney, and Munos 2017 shows impressive capabilities, both in terms of policy performance and data efficiency. The *distributional* aspect of learning describes the approximation of probability distributions in opposition to classical regression. In the case of Reinforcement Learning, the distribution of the random total cost given an initial state and control pair is considered.

6.1.1 Learning Distributions

The inference of unknown probability distribution has a long and complex history in the pattern recognition and statistics literature.

Machine Learning and Statistics

The reader may consult Kearns et al. 1994 and the references therein to learn more about the origins of the question of probability distribution approximation. Learning distributions is a core concept of generative modelling (Hinton, Osindero, and Teh 2006) and unsupervised learning (Hastie, Tibshirani, and J. Friedman 2013). It is the central goal in nonparametric density estimation and Bayesian statistics (see Section 3.2.3).

Reinforcement Learning

In RL, a generative model for the random total cost given an initial state and control pair is learned. This approach has been initially considered in a Bayesian setting (see Section 3.2.3) to quantify the information acquired by exploration

(Dearden, N. Friedman, and Russell 1998), and later for risk management applications (Morimura et al. 2010). More recently, this idea became the core concept of the distributional RL paradigm. In the foundational paper Bellemare, Dabney, and Munos 2017, the fundamental importance of the total cost as a distribution, in contrast to the total cost in expectation, is shown. The theoretical properties of the distributional approach are exposed, and its difference with classical RL is presented. Notably, an operator acting on conditional distributions is defined, echoing the Bellman operator acting on conditional expectations in classical RL.

6.1.2 Advantages of the Distributional Approach

Multiple benefits of the distributional approach have been identified in the literature.

Stability and Sample Efficiency

Among them, the distributional Bellman operator preserves multi-modality in value distributions, which may improve the stability of the learning process. Therefore, D-RL algorithms show improved empirical sample efficiency. Moreover, Bellemare, Dabney, and Munos 2017 argue that D-RL algorithms are more robust against non-stationary (time-dependent) policy than standard RL, and more globally, this paradigm makes the reinforcement learning process significantly better behaved.

A Novelty in the Flow Control Literature

Being recently introduced, the distributional approach has not been tested by the research community interested in dynamical systems connected with flow control.[53] Indeed, while a large part of the D-RL publications apply to robotics environment, the field of flow control could benefit from the potential advantages of this method.

6.1.3 Research Objectives and Experimental Setup

Miniaturised Chaotic Systems

Important properties of fluid flows such as chaos or symmetry are well incorporated in simple, miniaturised, chaotic systems such as the Lorenz or Kuramoto-Sivashinsky dynamics. They are an appropriate testbed for evaluating Deep Reinforcement Learning before scaling up to more complex systems such as Navier-Stokes (Cvitanović, Davidchack, and Siminos 2010).

[53] This work has been conducted during winter 2022–2023 when no paper on the application of D-RL to flow control has been issued.

Hypotheses and Objectives

This work tests two main hypotheses on the distributional approach to RL applied to chaotic dynamical systems. First, the distributional approach is more sample efficient than the classical approach for the control of representative chaotic systems. Second, the distributional generalises better for controlling from other parts of the state space than the classical approach. The two questions try to address the sample efficiency and the robustness challenges of RL in the context of flow control (those challenges are introduced in Section 1.4.1).

6.2 Distributional Reinforcement Learning

6.2.1 The Distributional Perspective

Consider the random total cost defined in Eq. (3.8) in a Markovian setting (the initial history distribution only depends on X_0).

$$Z\left(\mathbb{P}_{X_0}, \pi\right) = \sum_{i=0}^{K} \gamma^i c\left(X_k, \pi_k\right) \tag{6.1}$$

To alleviate the notation, this random variable is simply denoted Z in the following but the reader should keep in mind that it depends on the initial distribution $\mathbb{P}_{X_0}$.

Total Cost Conditional Distribution

The principal feature of the paradigm defined by Distributional Reinforcement Learning is to consider value *distributions* instead of value functions. The value function for a state or a state-control pair under some policy π gives the expected value of the random total cost given the initial state or state-control pair. In contrast, D-RL considers the distribution under some policy of the random total cost conditioned on an initial state or state-control pair.

Definition 6.2.1 (Conditional Random Objective Function)**.** *The conditional random objective function is defined as the conditional distribution of the random total cost Z given (X_0, U_0). For all $(x, u) \in \mathcal{X} \times \mathcal{U}$, the conditional random objective function is denoted by $Z(x, u)$ for $(X_0, U_0) = (x, u)$.*

Consequently, a closed-form expression of the conditional random objective function is given by

$$Z(x, u) \coloneqq c(x, u) + \gamma \sum_{i=1}^{K} \gamma^{i-1} c\left(X_i, \pi_i\right) \tag{6.2}$$

By definition of conditional probability, the quantity $Z(x, u)$ is still a random variable for all $(x, u) \in \mathcal{X} \times \mathcal{U}$. Note that this definition is equivalent as choosing $\mathbb{P}_{X_0} = \delta_{\{x\}}$ and $\pi = (\pi_k)_{k \in [\![0, K]\!]}$ where $\pi_0 = \delta_{\{u\}}$ in Eq. (6.1).

Link with the Q-function

Moreover, as the random total cost $Z(x, u)$ is a random variable for any $(x, u) \in \mathcal{X} \times \mathcal{U}$, the expectation of this random variable is well-defined if $Z(x, u)$ is integrable. This expectation is the Q-value function (see Section 3.1.4).

Proposition 6.2.1 (Link between Z-function and Q-function)**.** *The expectation of the conditional random objective function Z is the Q-function Q.*

$$Q(x, u) = \mathbb{E}\left[Z(x, u)\right] \tag{6.3}$$

for any $(x, u) \in \mathcal{X} \times \mathcal{U}$.

Proof. The proof is straightforward by definition of the soft Q-function and the conditional random objective function. Use Definition 11.5.2 in Jean-François Le Gall 2006. □

The Distributional Bellman Operator

The Distributional Bellman operator is the analogue to the Bellman operator (Theorem 3.1.1) in the distributional setting. Again, such fixed point operator is defined for criteria with infinite horizon.

Definition 6.2.2 (Distributional Bellman Operator)**.** *Suppose that $K = +\infty$. The distributional Bellman operator $\mathcal{T}$ is*

$$\mathcal{T}^{\pi} Z(x, u) \coloneqq c(x, u) + \gamma \mathbb{P}\left[Z\left(X_1, U_1\right) \mid X_0 = x, U_0 = u\right] \tag{6.4}$$

for any $(x, u) \in \mathcal{X} \times \mathcal{U}$. The notation $\mathbb{P}\left[Z\left(X_1, U_1\right) \mid X_0 = x, U_0 = u\right]$ denotes the conditional distribution of the random total cost Z given $(X_0, U_0) = (x, u)$.

An alternative form of the operator is

$$\mathcal{T}^{\pi} Z(x, u) \coloneqq c(x, u) + \gamma Z\left(X', U'\right) \tag{6.5}$$

where $X' \sim \mathcal{P}(\,\cdot \mid x, u)$ and $U' \sim \pi(\,\cdot \mid X')$.

Remark 6.2.1. *The distribution $\mathcal{T}^{\pi} Z(x, u)$ has multiple sources of randomness:*

- *The randomness of the next state-control pair (X', U')*
- *The randomness of the total cost $Z\left(X', U'\right)$*

It can be shown that the distributional Bellman operator is a contraction mapping for a specific metric based on the Wasserstein distance (Bellemare, Dabney, and Munos 2017). This allows for generalising temporal difference learning algorithms to the distributional setting.

6.2.2 Distributional RL with Quantile Regression

Being scalar-valued, the random total cost of Equation (6.2) can be regarded as an element of a metric space where the metric is totally characterised by its cumulative distribution function (c.d.f.).

Cumulative Distribution Functions and Quantiles Functions

Let $F_{\bar{Z}}$ be the cumulative distribution function of any real-valued random variable $\bar{Z}$. The c.d.f. is defined by $F_{\bar{Z}}(q) = \mathbb{P}(\bar{Z} \leq q)$ for any $q \in \mathbb{R}_+$. Let $F_{\bar{Z}}^{-1}$ be the general inverse c.d.f. (also called *quantile function*) defined by $F_{\bar{Z}}^{-1}(p) := \inf\{q \in \mathbb{R}_+ \mid F_{\bar{Z}}(q) \geq p\}$ for any $p \in [0, 1]$. The value $F_{\bar{Z}}^{-1}(p)$ is the p-quantile of the random variable $\bar{Z}$. When $F_{\bar{Z}}$ is continuous and strictly increasing, $F_{\bar{Z}}^{-1}$ coincide with the inverse function of $F_{\bar{Z}}$ (otherwise the mapping is not bijective).

A well-known metric space on probability distributions can be based on the concept of quantile functions.

Wasserstein Distance

Because D-RL focuses on distributions, a notion of distance between random variables (more generally over distributions) is relevant. The Wasserstein distance (Villani 2008; Santambrogio 2015) is a suitable choice for this purpose. This metric measures the cost of transporting the probability mass of one distribution to the other, for some arbitrary chosen cost (e.g. the L^p norm). A practical formulation of this distance in the unidimensional case (real random variables) is given in terms of quantile functions.

Definition 6.2.3 (Wasserstein Distance - Quantile Version)**.** *The Wasserstein distance of order $p \in [1, +\infty[$ between two real random variables $\bar{Z}_1$ and $\bar{Z}_2$ is defined by*

$$W_p\left(\mathbb{P}_{\bar{Z}_1}, \mathbb{P}_{\bar{Z}_2)}\right) = \left(\int_0^1 \left|F_{\bar{Z}_1}^{-1}(\omega) - F_{\bar{Z}_2}^{-1}(\omega)\right|^p d\omega\right)^{\frac{1}{p}} \tag{6.6}$$

where $\mathbb{P}_{\bar{Z}_i}$ and $F_{\bar{Z}_i}$ are the distribution and the c.d.f. of $\bar{Z}_i$ for $i \in \{1, 2\}$, respectively.

The objective of the method presented in Dabney et al. 2018 is to construct an optimal estimator $\widehat{Z}$ of the target conditional distribution Z that minimises the Wasserstein distance between the true distribution Z and the estimator $\widehat{Z}$. However, the Wasserstein distance is not directly minimised in practice.

Wasserstein Gradients are Biased

Viewed as a risk function, the Wasserstein distance exhibits an important limitation for practical applications based on gradient descent optimisation. Indeed, its estimation using empirical distribution leads to biased gradients. For instance, say that the target distribution $\mathbb{P}_{\bar{Z}_1}$ is approximated by the empirical

distribution $\widehat{\mathbb{P}}_{\bar{Z}_1}$ and that the distribution $\mathbb{P}_{\bar{Z}_2}$ is parametrised by an element $\bar{f} \in \mathcal{F}$ or $\theta \in \Theta$ such that $\mathbb{P}_{\bar{Z}_2} = \mathbb{P}_{\bar{f}}$ or $\mathbb{P}_{\bar{Z}_2} = \mathbb{P}_\theta$. Then, the following result due to Bellemare, Danihelka, et al. 2017 holds

Proposition 6.2.2 (Biased Gradient of the Wasserstein Distance)**.** *There exists a (target) distribution $\mathbb{P}_{\bar{Z}_1}$ approximated by its corresponding empirical distribution $\widehat{\mathbb{P}}_{\bar{Z}_1} = \frac{1}{N}\sum_{i=1}^{N} \delta_{\{\bar{Z}_1{}^{(i)}\}}$ for some $N \in \mathbb{N}^*$ and there exists a parametrised distribution $\mathbb{P}_\theta$ with parameter $\theta \in \Theta$ such that*

$$\arg\min_{\theta\in\Theta} \nabla_\theta W_p^p\left(\mathbb{P}_{\bar{Z}_1}, \mathbb{P}_\theta\right) \neq \arg\min_{\theta\in\Theta} \mathbb{E}\left[\nabla_\theta W_p^p\left(\widehat{\mathbb{P}}_{\bar{Z}_1}, \mathbb{P}_\theta\right)\right] \tag{6.7}$$

where W_p is the Wasserstein distance of order $p \in [1, +\infty[$ defined in Definition 6.2.3.

The gradient of the Wasserstein distance to the power p is biased when estimated using empirical distributions.

Proof. See Bellemare, Danihelka, et al. 2017. □

This result is relatively weak but contraindicates a general use of gradient-based optimisation over a Wasserstein risk for estimating an approximation $\widehat{Z}$ of the random total cost distribution Z. Within this context, an approach based on the use of a loss function that allows unbiased gradient estimation is being sought.

Quantile Distribution

To this end, a specific space of distribution is constructed such that a Wasserstein metric minimisation can be achieved through a learning task that exhibits unbiased gradients. Hence, for each $(x, u) \in \mathcal{X} \times \mathcal{U}$, the authors of the reference paper define what they call a *quantile (conditional) distribution* that is a mapping from $\mathcal{X} \times \mathcal{U}$ to the set of discrete uniform probability distributions on a finite set of quantiles $(q_i)_{i\in[\![1,N_q]\!]}$ which will be learnt by the algorithm.

Definition 6.2.4 (Quantile Distribution)**.** *A N_q-quantiles distribution $\widehat{Z}$ is a mapping*

$$\begin{aligned} \widehat{Z} : \mathcal{X} \times \mathcal{U} &\to \mathbb{P}_{(\mathsf{Unif}, N_q)} \\ (x, u) &\mapsto \mathsf{Unif}\left(\left(\widehat{q}_i(x, u)\right)_{i\in[\![1,N_q]\!]}\right) \end{aligned} \tag{6.8}$$

where $\mathbb{P}_{(\mathsf{Unif},N_q)}$ is the set of discrete uniform probability distributions on N_q atoms and $\mathsf{Unif}((\widehat{q}_i)_{i\in[\![1,N_q]\!]})$ is the uniform probability distribution on the set $(\widehat{q}_i)_{i\in[\![1,N_q]\!]}$. Hence, for any $(x, u) \in \mathcal{X} \times \mathcal{U}$, $\widehat{Z}(x, u) \sim \mathsf{Unif}((\widehat{q}_i(x, u))_{i\in[\![1,N_q]\!]})$. The set of quantile distributions is denoted by $\mathcal{F}_{QR}$.

Remark 6.2.2. *Since quantile regression involves a family of risk functions indexed by quantile levels $\lambda \in [0, 1]$, the computational complexity of algorithms based on this approach is N_q times higher than the classical regression methods.*

Remark 6.2.3. *For a fixed quantile order $\lambda \in [0,1]$, the task ℓ_λ is an asymmetric convex functional penalising overestimation errors with weight λ and underestimation errors with weight $1-\lambda$.*

This way, conditional distributions in $\mathcal{F}_{\text{QR}}$ are characterised by a finite set of quantiles functions. Thus, this approach amounts to estimating optimal quantiles $(\widehat{q}_i(x,u))_{i\in[\![1,N_q]\!]}$ for each $(x,u) \in \mathcal{X}\times\mathcal{U}$ such that the Wasserstein distance between the true distribution $Z(x,u)$ and the estimator $\widehat{Z}(x,u)$ is minimised. Now, the hypothesis space considered is the set of all conditional quantile distributions which is denoted by $\mathcal{F}_{\text{QR}}$. Thus, $\widehat{Z}(x,u) \in \mathcal{F}_{\text{QR}}$.

It turns out that it can be shown (Dabney et al. 2018) that for a uniform discretisation of $[0,1]$ into N_q probability values $(\lambda_i)_{i\in[\![1,N_q]\!]} \coloneqq (\frac{i}{N_q})_{i\in[\![1,N_q]\!]}$, the optimal quantiles such that the estimator $\widehat{Z}(x,u)$ minimises the Wasserstein distance $W_1(Z(x,u),\widehat{Z}(x,u))$ in the sense of Definition 6.2.3, between the true distribution $Z(x,u)$ and the estimator $\widehat{Z}(x,u)$, are the quantiles $(\widehat{q}_i^*(x,u))_{i\in[\![1,N_q]\!]} = (F^{-1}_{Z(x,u)}(\tilde{\lambda}_i))_{i\in[\![1,N_q]\!]}$ where[54] $\widetilde{\lambda}_i \coloneqq (\tilde{\lambda}_i)_{i\in[\![1,N_q]\!]} = (\frac{\lambda_{i-1}+\lambda_i}{2})_{i\in[\![1,N_q-1]\!]}$. Consequently, an optimal estimator for the Wasserstein distance considered here is given by $\widehat{Z}^*(x,u) = \text{Unif}((q_i^*(x,u))_{i\in[\![1,N_q]\!]})$.

Now, the question of the quantile estimation arises. Indeed, minimising the Wasserstein distance W_1 amounts to estimating optimal quantiles.

Quantile Regression

Naturally, *quantile regression* (Koenker 1994; Koenker 2005) is a suitable technique for this purpose: A family of learning tasks is specified as a collection of functions $(\ell_\lambda)_{\lambda\in[0,1]}$ where each task is indexed by a probability value $\lambda \in [0,1]$.

Definition 6.2.5 (Quantile Regression)**.** *Quantile regression is defined as the minimisation of the following collection of learning tasks*

$$\ell_\lambda(\bar{Z},q) \coloneqq (\bar{Z}-q)\left(\lambda - 1_{(\bar{Z}<q)}\right) = \begin{cases} \lambda|\bar{Z}-q| & \text{if } \bar{Z}\geq q \\ (1-\lambda)|\bar{Z}-q| & \text{if } \bar{Z}<q \end{cases} \tag{6.9}$$

for any quantile level $\lambda \in [0,1]$ and $q \in supp(\bar{Z})$ where $supp(\bar{Z})$ is the support of the random variable $\bar{Z}$ (the smallest closed set such that the probability of the random variable $\bar{Z}$ being outside this set is zero). $F^{-1}_{q\bar{Z}}(\lambda)$ is a critical point of ℓ_λ for any $\lambda \in [0,1]$, i.e. a global minimum (by convexity).

In practice, for a set of N_q quantile levels $(\lambda_i)_{i\in[\![1,N_q]\!]}$, the average task risk for quantile regression is defined as

$$\mathcal{L}^{\text{QR}}_{(\lambda_i)_{i\in[\![1,N_q]\!]}}\left(\bar{Z},(q_i)_{i\in[\![1,N_q]\!]}\right) \coloneqq \frac{1}{N_q}\sum_{i=1}^{N_q}\mathbb{E}^{\mathbb{P}_{\bar{Z}}}\left[\ell_{\lambda_i}\left(\bar{Z},q_i\right)\right] \tag{6.10}$$

[54] Those mid-quantile values recall how the median ($\frac{1}{2}$-quantile) is optimal in L^1 regression, which is used to define the W_1 distance.

It is important to note that quantile regression exhibits *no bias* in the gradient estimation discussed above.

No Bias in Gradient Estimation for Quantile Distributions

Importantly, the Wasserstein distance W_1 restricted to the space $\mathcal{F}_{\text{QR}}$ of quantile distributions (uniform probability distribution on N_q points) can be minimised without bias in the gradient estimation. In other words, when the distributions considered are quantile distributions, the Wasserstein distance W_1 can be minimised *indirectly* through the unbiased minimisation of the average quantile regression risk, provided the quantile are well-chosen (mid-points). However, this approach restricts the hypothesis space significantly and nothing guarantees that the target distribution belongs to this space.

Though, in Reinforcement Learning, the target distribution is unknown and the temporal difference target is used as a proxy[55]. Moreover, the crucial point is to find a fixed point of the Bellman equation. It happens that the distributional Bellman operator from Definition 6.2.1, restricted on $\mathcal{F}_{\text{QR}}$ is a contraction mapping for a particular metric based on W_1, which permits the application of basic dynamic programming algorithms.

Consequently, for a fixed collection $\widetilde{\lambda} = (\widetilde{\lambda}_i)_{i\in[\![1,N_q]\!]}$ of quantile levels, the estimates $\widehat{q}_Z \coloneqq (\widehat{q}_{Z,i})_{i\in[\![1,N_q]\!]}$ of the optimal $\widetilde{\lambda}$-quantiles $q^* \coloneqq (q_i^*)_{i\in[\![1,N_q]\!]}$ are obtained by minimising the average quantile regression risk $\mathcal{L}_{\widetilde{\lambda}}^{\text{QR}}$ where $\widehat{Z} \sim \text{Unif}((\widehat{q}_{Z,i})_{i\in[\![1,N_q]\!]})$.

The next section sets the foundations of the Distributional-RL algorithm used in this work: the Distributional *Truncated Quantile Critics* algorithm introduced in Kuznetsov et al. 2020 for which the core concept is an extension of the Soft Actor-Critic algorithm Dabney et al. 2018 to its distributional version (see also J. Duan et al. 2022).

6.3 Distributional Soft Actor-Critic

The maximum-entropy principle is a central concept in this thesis and appears again in this part of the document. The distributional method used for this work is based on a prominent algorithm in the field of Maximum-Entropy Reinforcement Learning: the Soft Actor-Critic algorithm, which was introduced in Haarnoja, A. Zhou, Abbeel, et al. 2018; Haarnoja, A. Zhou, Hartikainen, et al. 2019.

[55] As mentioned in Bellemare, Dabney, and Munos 2017 and Sutton and Barto 2018: "learn a guess from a guess".

6.3.1 Soft Actor-Critic

As its name suggests, the Soft Actor-Critic algorithm belongs to the family of *actor-critic* algorithms. As discussed in Section 2.2.5, the term "Soft" refers to the entropy regularisation term added to the standard ("hard") Bellman equation.

The optimality equations for the soft-objective have been derived in Ziebart 2010. Later a schema of policy iteration (see Section 3.3.2) that fits well with function approximation has been proposed by Haarnoja, Tang, et al. 2017 through the use of Deep Energy-Based Models (EBMs) for the policy.[56]

Soft Bellman Equation

Consider the soft objective problem defined in Eq. (3.18)-(3.19) for an infinite horizon problem and a Markov policy. Then the objectives read

$$J^*_{\mathcal{H}}(x) = \inf_{\pi \in \mathscr{A}_\Pi} \mathbb{E}\left[\sum_{i=0}^{\infty} \gamma^i c(X_i, \pi_i) - \alpha^{\mathcal{H}} \mathcal{H}\left[\pi_i(\cdot \mid X_i)\right] \mid X_0 = x\right] \tag{6.11}$$

for the soft value function and

$$Q^*_{\mathcal{H}}(x, u) = \inf_{\pi \in \mathscr{A}_\Pi} \mathbb{E}\left[\sum_{i=0}^{\infty} \gamma^i c(X_i, \pi_i) - \alpha^{\mathcal{H}} \mathcal{H}\left[\pi_i(\cdot \mid X_i)\right] \mid X_0 = x, U_0 = u\right] \tag{6.12}$$

for the soft Q-function.

It can be shown (Haarnoja, Tang, et al. 2017) that an optimal Markov policy π^* exists and its conditional probability density is given by

$$\pi^*_{\mathcal{H}}(u \mid x) = \exp\left(\frac{1}{\alpha^{\mathcal{H}}}\left(Q^*_{\mathcal{H}}(x, u) - V^*_{\mathcal{H}}(x)\right)\right) = \frac{1}{C^\pi_{\mathcal{H}}(x)} \exp\left(\frac{1}{\alpha^{\mathcal{H}}} Q^*_{\mathcal{H}}(x, u)\right) \tag{6.13}$$

for all $(x, u) \in \mathcal{X} \times \mathcal{U}$ with $C^\pi_{\mathcal{H}}(x)$ a normalisation constant incorporating the value function $V^*_{\mathcal{H}}$. Consequently, it is crucial to find the optimal soft Q-function $Q^*_{\mathcal{H}}$ to derive the optimal policy π^*. When the set of policy Π is Markovian, a Dynamic Programming equation can be derived from (6.12) and (6.13). The resulting equation is called *soft Bellman equation*.

Soft Policy Iteration

Then, a policy iteration scheme (see Section 3.3.2) can be employed to iteratively solve the problem of finding the optimal policy:

[56] There is an important link between the maximum-entropy principle and the Gibbs distribution that is prominent in statistical mechanics (Mézard and Montanari 2009; Mohri, Rostamizadeh, and Talwalkar 2018). The only thing that matters here is that the policy is a Gibbs distribution, i.e. probability measure with density $x \mapsto \frac{1}{C_Z} \exp(-\beta E(x))$ where C_Z is a normalisation constant.

- The policy evaluation consists in approximating the soft Q-function $Q^{\pi}_{\mathcal{H}}$ for a given policy π. This step is achieved by solving the soft Bellman equation for the soft Q-function by fixed-point iteration.
- The policy improvement step updates the policy toward its known closed form given in Eq. (6.13) using the soft Q-function obtained in the previous step by solving

$$\pi'_{\mathcal{H}}(\cdot \mid x) = \arg\min_{\pi'' \in \Pi} D_{\mathsf{KL}}\left(\pi''(\cdot \mid x) \,\middle\|\, \frac{1}{C^{\pi}_{\mathcal{H}}(x)} \exp\left(\frac{1}{\alpha^{\mathcal{H}}} Q^{\pi}_{\mathcal{H}}(x, \cdot)\right)\right) \tag{6.14}$$

  where $C^{\pi}_{\mathcal{H}}(x)$ is the normalisation constant of the distribution $\pi(\cdot \mid x)$ which is absorbed by the learning rate of any gradient-based optimisation algorithm. Hence, the resulting policy $\pi'_{\mathcal{H}}$ gets closer to the family of policy characterised by such exponential form. This way, this policy is necessarily closer to the optimal policy $\pi^{*}_{\mathcal{H}}$. Thus, this step improves the policy at each iteration.

Soft Actor-Critic

By estimating both a policy and a value function, the *Soft Actor-Critic* algorithm enters the category of *actor-critic* methods (see Section 3.3.2). In the sequel, the policy estimator is denoted by $\widehat{\pi}_{\mathcal{H}}$ and the soft Q-function estimator by $\widehat{Q}_{\mathcal{H}}$. Moreover, it is supposed that two $\widehat{\pi}_{\mathcal{H}}$-dependent data distribution are given, possibly by sampling from the environment or collecting historical data from a simulation. The first distribution $\mathbb{P}_{X}^{\widehat{\pi}_{\mathcal{H}}}$ is a measure on the state space $\mathcal{X}$ and the second distribution $\mathbb{P}_{X,U,X',U'}^{\widehat{\pi}_{\mathcal{H}}}$ is a measure on $\mathcal{X} \times \mathcal{U} \times \mathcal{X} \times \mathcal{U}$. This probability measure can be thought as a state occupancy measure and observed transitions distribution, respectively under the policy $\widehat{\pi}_{\mathcal{H}}$.

- The policy evaluation step solves the soft Bellman equation by minimising the quadratic risk between the soft Q-function and its one-step ahead, forward expression In the maximum-entropy setting, if the policy belongs to the class of energy-based policy given by Eq. (6.13), then the soft Bellman equation can be rewritten as (the proof is given in Haarnoja, A. Zhou, Abbeel, et al. 2018)

$$Q^{*}_{\mathcal{H}}(x,u) = c(x,u) + \gamma \int_{\mathcal{X}\times\mathcal{U}} Q^{*}_{\mathcal{H}}(x',u') - \alpha^{\mathcal{H}} \log \pi(u' \mid x')\, \mathcal{P}(dx' \mid x,u) \pi(u' \mid x)\, du' \tag{6.15}$$

  Consequently, the policy evaluation step consists in the optimisation of the quadratic risk defined by

$$\widehat{Q}'_{\mathcal{H}} = \arg\min_{\widehat{Q}_{\mathcal{H}} \in \mathcal{F}} \int_{\mathcal{X}\times\mathcal{U}\times\mathcal{X}\times\mathcal{U}} (\widehat{Q}_{\mathcal{H}}(x,u) - \widetilde{Q}_{\mathcal{H}}(x,u,x',u'))^2 \mathbb{P}_{X,U,X',U'}^{\widehat{\pi}_{\mathcal{H}}}(dx,du,dx',du') \tag{6.16}$$

where $\mathcal{F}$ is the hypothesis space of the soft Q-function and the temporal difference target $\widetilde{Q}_{\mathcal{H}}$ is given by

$$\widetilde{Q}_{\mathcal{H}}(x,u,x',u') = c(x,u) + \gamma \widehat{Q}_{\mathcal{H}}\left(x',u'\right) - \alpha^{\mathcal{H}} \log \pi\left(u' \,|\, x'\right) \tag{6.17}$$

- Given an estimate of the soft Q-function $\widehat{Q}_{\mathcal{H}}$, the policy improvement part updates the policy by using an estimation of the target policy in Eq. (6.13) and optimising the averaged Kulback-Leibler (KL) divergence between the current policy and the target policy from Eq. (6.14)

$$\widehat{\pi}'_{\mathcal{H}}\left(\cdot \mid x\right) = \arg\min_{\pi'' \in \Pi} \int_{\mathcal{X}} D_{\mathsf{KL}}\left(\pi''(\cdot \mid x) \,\middle\|\, \frac{1}{C^{\pi}_{\mathcal{H}}(x)} \exp\left(\frac{1}{\alpha^{\mathcal{H}}} \widehat{Q}_{\mathcal{H}}(x,\cdot)\right)\right) \mathbb{P}_X^{\widehat{\pi}_{\mathcal{H}}}\left(dx\right) \tag{6.18}$$

Since the target function $\widetilde{Q}_{\mathcal{H}}$ depends itself on $\widehat{Q}_{\mathcal{H}}$, the minimisation of the risk in Eq. (6.16) can be challenging in practice.[57] In statistics, the use of an estimator to build another estimator such as in Eq. (6.16) is known as *bootstrapping*.

Parametrisation and Learning

These algorithms have been designed amidst the recent advances in Deep Reinforcement Learning applied to Robotics where problems are high dimensional and state and control domains are continuous. Suitable function approximators are thus needed to learn in such environments. In this vein, neural networks are considered. However, instead of running policy evaluation and policy improvement to convergence, the algorithm alternates between optimising both networks with Stochastic Gradient Descent.

It should be noted that the need of neural networks is not necessarily clear in the context of flow control where only a few observations are available (that are considered as states when applying such algorithms).

Under those circumstances, the hypothesis spaces are parametrised $\mathcal{F}_\pi = \Theta_\pi$ and $\mathcal{F}_Q = \Theta_Q$ where Θ_π and Θ_Q are the neural network weights spaces of the policy and the soft Q-function, respectively. Consequently, the policy and the soft Q-function become parametric estimators $\widehat{\pi}_{\mathcal{H}} = \pi^{\theta_\pi}_{\mathcal{H}}$ and $\widehat{Q}_{\mathcal{H}} = Q^{\theta_Q}_{\mathcal{H}}$.

6.3.2 Combining Distributional Reinforcement Learning and Soft Actor-Critic

By adapting the Distributional Bellman operator defined in Eq. (6.4)-(6.5), in its maximum-entropy form described by the soft Bellman equation defined in

[57] Actually, another approximator of $x \mapsto \int_{\mathcal{U}} Q^{\pi}_{\mathcal{H}}(x,u)\pi(u \mid x)du$ is used in the original SAC paper to stabilise the learning process (see Haarnoja, A. Zhou, Abbeel, et al. 2018, p. 5). However, the SAC formulation given here is identical as the one considered in the TQC paper to build their distributional counterpart of SAC. Polyak averaging (Polyak and Juditsky 1992) can be used to stabilise the learning process.

Eq. (6.15), which also defines an operator), a maximum entropy version of the Distributional Bellman operator can be defined.

Quantile Temporal-Difference Learning

The distributional approach considered here is based on the quantile regression method presented in Section 6.2.2. The first goal is to estimate the distribution $Z(x,u)$ for any $(x,u) \in \mathcal{X} \times \mathcal{U}$ with a quantile distribution $\widehat{Z}(x,u) \in \mathcal{F}_{\mathsf{QR}}$. This quantile distribution is characterised by a set of quantiles $(\widehat{q}_{Z,i}(x,u))_{i \in [\![1,N_q]\!]}$.

Thus, for a fixed state-control pair $(x,u) \in \mathcal{X} \times \mathcal{U}$, an approximation of the minimisation objective (empirical risk) is now given by

$$\widehat{\mathcal{L}^{\mathsf{QR}}}_{(x,u),(\widetilde{\lambda}_i)_{i \in [\![1,N_q]\!]}} \left((\widetilde{Z}_i)_{i \in [\![1,N]\!]}, (\widehat{q}_{Z,i})_{i \in [\![1,N_q]\!]} \right) \coloneqq \frac{1}{N_q N} \sum_{i=1}^{N_q} \sum_{j=1}^{N} \ell_{\widetilde{\lambda}_i} \left(\widetilde{Z}_i(x,u), \widehat{q}_{Z,i}(x,u) \right) \tag{6.19}$$

where $\widetilde{Z}_i(x,u) \coloneqq c(x,u) + \gamma \widehat{Z}_i(X',U') - \alpha^{\mathcal{H}} \log \pi\left(U' \,|\, X'\right)$ for any $i \in [\![1,N_q]\!]$ with $X' \sim \mathcal{P}(\cdot \,|\, x,u)$ and $U' \sim \pi(\cdot \,|\, x)$. The target $(\widetilde{Z}_i(x,u))_{i \in [\![1,N]\!]}$ is a collection of i.i.d. samples generated from an i.i.d. realisation $(\widehat{Z}(x,u))_{i \in [\![1,N]\!]}$ of the random total cost $\widehat{Z}(x,u) \sim \mathsf{Unif}((q_{\widehat{Z},i})_{i \in [\![1,N_q]\!]})$. Note that the empirical risk defined in Eq. (6.19) is a function of $(x,u) \in \mathcal{X} \times \mathcal{U}$.

In other terms, the conditional distribution $\widehat{Z}$ is first used to generate samples of the random total cost. Then, a temporal difference target distribution $\widetilde{Z}$ is constructed from these samples. The goal is to find new quantiles that are closer to the temporal difference target distribution. This way, a fixed point of the distributional Bellman operator is approximated. This method is known as *Quantile Temporal-Difference Learning* and has been analysed thoroughly in Rowland et al. 2024. This fixed point is a conditional (quantile) distribution.

In practice, the risk $\widehat{\mathcal{L}^{\mathsf{QR}}}_{(x,u),(\widetilde{\lambda}_i)_{i \in [\![1,N_q]\!]}}$ is not minimised for any state-control pair $(x,u) \in \mathcal{X} \times \mathcal{U}$, but the average risk over a dataset of state-control pairs is optimised. The following averaged risk is considered

$$\widehat{\mathcal{L}^{\mathsf{QR}}}_{(\widetilde{\lambda}_i)_{i \in [\![1,N_q]\!]}} \coloneqq \int_{\mathcal{X} \times \mathcal{U}} \widehat{\mathcal{L}^{\mathsf{QR}}}_{(x,u),(\widetilde{\lambda}_i)_{i \in [\![1,N_q]\!]}} \left((\widetilde{Z}_i)_{i \in [\![1,N]\!]}, (\widehat{q}_{Z,i})_{i \in [\![1,N_q]\!]} \right) \mathbb{P}^{\widehat{\pi}}_{X,U}(dx,du) \tag{6.20}$$

Hence, the random total cost approximator $\widehat{Z}$, being characterised by the quantile estimators $(\widehat{q}_{Z,i})_{i \in [\![1,N_q]\!]}$, is optimised by minimising the average quantile regression risk $\widehat{\mathcal{L}^{\mathsf{QR}}}$. In other words, the policy evaluation step is replaced by the quantile regression step and solves

$$\widehat{Z}' \sim \mathsf{Unif}\left((\widehat{q}_{Z',i})_{i \in [\![1,N_q]\!]} \right) = \operatorname*{arg\,min}_{Z'' \in \mathcal{F}_{\mathsf{QR}}} \widehat{\mathcal{L}^{\mathsf{QR}}}_{(\widetilde{\lambda}_i)_{i \in [\![1,N_q]\!]}} \left((\widetilde{Z}_i)_{i \in [\![1,N]\!]}, (\widehat{q}_{Z'',i})_{i \in [\![1,N_q]\!]} \right) \tag{6.21}$$

which is equivalent to a minimisation over the quantiles $(\widehat{q}_{Z',i})_{i\in[\![1,N_q]\!]}$ since the distribution belongs to the family $\mathcal{F}_{\mathsf{QR}}$ of quantile distributions (uniform probability distribution on N_q points).

$$(\widehat{q}_{Z',i})_{i\in[\![1,N_q]\!]} = \underset{(\widehat{q}_{Z'',i})_{i\in[\![1,N_q]\!]}}{\arg\min}\ \widehat{\mathcal{L}^{\mathsf{QR}}}_{(\widetilde{\lambda}_i)_{i\in[\![1,N_q]\!]}}\left((\widetilde{Z}_i)_{i\in[\![1,N]\!]}, (\widehat{q}_{Z'',i})_{i\in[\![1,N_q]\!]}\right) \tag{6.22}$$

To sum up, $\widehat{Z}_{\mathcal{H}}$ replaces $\widehat{Q}_{\mathcal{H}}$ and is characterised by the quantile estimators $\widehat{q}_Z = (\widehat{q}_{Z,i})_{i\in[\![1,N_q]\!]}$, $\widetilde{Z}_{\mathcal{H}}$ replaces $\widetilde{Q}_{\mathcal{H}}$ and is characterised by the temporal difference target quantile estimators $\widetilde{q} \coloneqq (\widetilde{q}_i)_{i\in[\![1,N_q]\!]} = (c(x,u) + \gamma\widehat{Z}_{\mathcal{H}}(x',u') - \alpha^{\mathcal{H}}\log\pi\left(u'\,|\,x'\right))_{i\in[\![1,N_q]\!]}$, and the policy evaluation step is replaced by the quantile regression step where the value is now atomised into quantiles.

Remark 6.3.1. *In practice, the quantile regression loss leads to unstable optimisation, and an adaptation of the Huber loss for quantile regression is preferred.*

Policy Improvement

The policy improvement step is straightforward. It is performed by policy gradient on the estimated Q-function $\widehat{Q}_{\mathcal{H}}$ obtained from the random total cost approximator $\widehat{Z}_{\mathcal{H}}$. Indeed, by Proposition 6.2.1, the Q-function is the expectation of the random total cost. But since $\widehat{Z}_{\mathcal{H}} \sim \mathrm{Unif}((\widehat{q}_{Z,i})_{i\in[\![1,N_q]\!]})$, its expectation is well known and is given by $\mathbb{E}[\widehat{Z}_{\mathcal{H}}] = \widehat{Q}_{\mathcal{H}} = \frac{1}{N_q}\sum_{i=1}^{N_q}\widehat{q}_{Z,i}$.

Thus, the policy improvement step of Eq. (6.18) simplifies to

$$\begin{aligned}\widehat{\pi}'_{\mathcal{H}}\left(\cdot\mid x\right) &= \underset{\pi''\in\Pi}{\arg\min}\int_{\mathcal{X}\times\mathcal{U}}\left(\frac{1}{N_q}\sum_{i=1}^{N_q}\widehat{q}_{Z,i}(x,u) - \alpha^{\mathcal{H}}\log\pi''\left(u\,|\,x\right)\right)\mathbb{P}^{\widehat{\pi}_{\mathcal{H}}}\left(dx,du\right)\\ &= \underset{\pi''\in\Pi}{\arg\min}\int_{\mathcal{X}\times\mathcal{U}}\frac{1}{N_q}\sum_{i=1}^{N_q}\widehat{q}_{Z,i}(x,u)\mathbb{P}^{\widehat{\pi}_{\mathcal{H}}}\left(dx,du\right) - \alpha^{\mathcal{H}}\int_{\mathcal{X}}\mathcal{H}\left[\pi''\left(\cdot\,|\,x\right)\right]\mathbb{P}^{\widehat{\pi}_{\mathcal{H}}}\left(dx\right)\end{aligned} \tag{6.23}$$

The method aims to minimise the Wasserstein distance between the random total cost $\widehat{Z}$ and its corresponding temporal difference target distribution $\widetilde{Z} \coloneqq c(x,u) + \gamma\widehat{Z}(x',u') - \alpha^{\mathcal{H}}\log\pi\left(u'\,|\,x'\right)$ for all $(x,u)\in\mathcal{X}\times\mathcal{U}$, $x'\sim\mathcal{P}(\cdot\,|\,x,u)$ and $u'\sim\pi(\cdot\,|\,x)$. This way, a fixed point of the distributional Bellman operator is approximated.

Parametrisation and Learning

As well as in the non-distributional case, the hypothesis spaces are parameterised $\mathcal{F}_{\widehat{\pi}} = \Theta_{\widehat{\pi}}$ and $\mathcal{F}_{\widehat{Z}} = \mathcal{F}_{\mathsf{QR}} = \Theta_{\widehat{Z}}^{N_q}$ such that the family of quantile distributions is the collection of N_q-tuples spaces since $(\widehat{q}_{Z,i})_{i\in[\![1,N_q]\!]} = (q_{Z,i}^{\theta_Z})_{i\in[\![1,N_q]\!]}$.

6.3.3 Complementary Features

The core of the method has been presented in the previous sections. However, being in practice parametrised by neural networks, the learning process can be extremely brittle in practice (Hasselt et al. 2018).

Stabilisation and Variance Reduction

Common features can be added to the algorithm to improve its stability and performance. The following features are prominent:

- **Ensembling**: The Q-function or Z-function can be approximated by a set of estimators to reduce the variance.
- **Polyak Averaging**: The temporal difference target $\widetilde{Q}$ or $\widetilde{Z}$ can be updated using Polyak averaging (Polyak and Juditsky 1992) to improve the behaviour of the stochastic approximation.
- **Huber Loss**: The quantile regression loss can be replaced by a smoother Huber loss type to stabilise the learning process Dabney et al. 2018.

Overestimation Correction by Quantile Truncation

A well-known issue for algorithms combining Q-learning and function approximation is the overestimation of the Q-function (Thrun and Schwartz 1999; Hasselt 2010; Fujimoto, Hoof, and Meger 2018).

In fact, the distributional RL method used in this work called *Truncated Quantile Critics* (TQC) has been designed to address this issue Kuznetsov et al. 2020. By truncating the top n_{trunc} quantiles of the estimated distribution $\widehat{Z}_{\mathcal{H}} \sim \text{Unif}\left((\widehat{q}_{Z,i})_{i\in[\![1,N_q]\!]}\right)$, to get the truncated distribution $\widehat{Z}^{\text{trunc}}_{\mathcal{H}} \sim \text{Unif}((\widehat{q}_{Z,i})_{i\in[\![1,N_q-n_{\text{trunc}}]\!]})$, the overestimation of the Q-function is reduced. Indeed, truncation allows for arbitrary granular overestimation control by biasing the distribution towards the lower quantiles.

Finally, TQC implements the three features mentioned above: ensembling[58], Polyak averaging, and Huber loss in addition to quantile truncation to improve the stability and performance of the algorithm. In this work, the implementation of TQC from *Stable Baselines3* is used Raffin et al. 2021.

The next section presents an application of the TQC algorithm to the control of a chaotic system. A comparison with the state-of-the-art method is provided to gain insights on the particular features of the TQC algorithm and its potential for flow control applications.

[58]The ensemble is composed of n_{ens} of Z-functions, thus n_{ens} collections of N_q quantiles.

6.4 On the sample efficiency of Distributional Reinforcement Learning

This work extends the results obtained in Bucci et al. 2019 on the control of the *Kuramoto-Sivashinksy (KS)* partial differential equation with *Deep Deterministic Policy Gradient (DDPG)* (Lillicrap et al. 2019). The KS system exhibits chaotic properties and possesses similarities with the Navier-Stokes equations (see Section 3.4.3).

The performance of other baseline deep RL algorithms for the control of chaotic systems is the first question addressed in this work. Indeed, Bucci et al. 2019 solely used DDPG to control the KS system. Being an off-policy algorithm, it is prone to instability (Matheron, Sigaud, and Perrin 2020). Moreover, on-policy algorithms and maximum-entropy reinforcement learning (Chapter 4) may also be suitable for the control of chaotic systems.

The performance criterion considered here is the sample efficiency of the algorithms. Since Distributional Reinforcement Learning (DRL) shows promising results in terms of learning speed and stability (Bellemare, Dabney, and Munos 2017), it is of interest to compare it with the other baseline algorithms. Consequently, the first question addressed in this work is

- How does the sample efficiency of Distributional RL compare to other baseline deep RL algorithms for the control of chaotic systems?

The second question is related to the generalisation capability of the learned policies to initial distribution $\mathbb{P}_{X_0}$. Thus, the second question addressed in this work is

- How do policies learnt with Distributional RL generalise to out of training initial conditions compared to other baseline deep RL algorithms?

The next section presents the experiments performed to gather insights regarding the previous questions.

6.5 Experiments

In order to answer the questions raised in the previous section, a series of experiments are conducted. Basically, they build on the training of deep RL algorithms on the Kuramoto-Sivashinsky PDE (see Section 3.4.3).

Four standard deep RL algorithms and one instance of deep Distributional RL are considered in the following experiments. Each of them represents a particular aspect of the RL literature (On-policy, Off-policy, Maximum Entropy). Concretely, the following algorithms are considered:

- Deep Deterministic Policy Gradient (DDPG), off-policy gradient based algorithm (Lillicrap et al. 2019).

- Trust Region Policy Optimisation (TRPO), on-policy algorithm with a trust region mechanism to ensure smooth policy updates (Schulman, Levine, et al. 2015).
- Proximal Policy Optimisation (PPO), computationally efficient extension of TRPO (Schulman, Wolski, et al. 2017).
- Soft Actor-Critic (SAC), a deep learning approach to the maximum entropy version of the Bellman equation (see Section 6.3).
- Truncated Quantile Critics (TQC), a Distributional Reinforcement Learning extension of SAC, with critic ensembling and value function correction by large quantile (extreme values) truncation (see Section 6.3.3).

All the optimisation procedures (*a.k.a.* training or learning processes) are carried out with the default hyperparameters from *Stable Baselines3* that often corresponds to the original paper configuration or benchmark configurations. Finally, each of the algorithms is trained over five *i.i.d.* runs (random seeds).

The first question asked in Section 6.4 is now addressed.

6.5.1 On the sample efficiency of Distributional Reinforcement Learning

In the perspective of answering the first question, training on the KS environment is analysed. Results of the training process are presented in Figure 6.1. The training time unit is given by the number $m \in \mathbb{N}^*$ of interactions with the environment (which determines the sample complexity). This number also corresponds to the number of decisions (control input). In the case of this experiment, a limited budget $m = 2 \times 10^5$ is set. The initial distribution $\mathbb{P}_{X_0}$ is defined such that $X_0 \sim \mathcal{N}(x_e, \sigma_e^2 Id_{d_X})$, *i.e.* the initial state is randomly picked in the vicinity of the equilibrium $x_e = x_{e_2^*}$ with perturbation noise $\sigma_e = 10^{-1}$. The controlled trajectory length of the KS environment is set to $K = 200$.

Figure 6.1 shows that the training dynamics of TQC minimises the objective criterion throughout the training process compared with the other algorithms. The performance spread is significant for TQC against the other algorithms. Otherwise, the best algorithms are instances of on-policy algorithms (TRPO and PPO).

6.5.2 Ablation Study

An ablation study is performed to evaluate the influence some features of particular importance implemented by TQC. The features considered are the critic ensemble size, the number of quantiles and the maximum entropy regularisation coefficient. The number of critics is varied from 1 to 2, the number of quantiles is equal to 1 or 25 and the entropy regularisation coefficient is set to zero

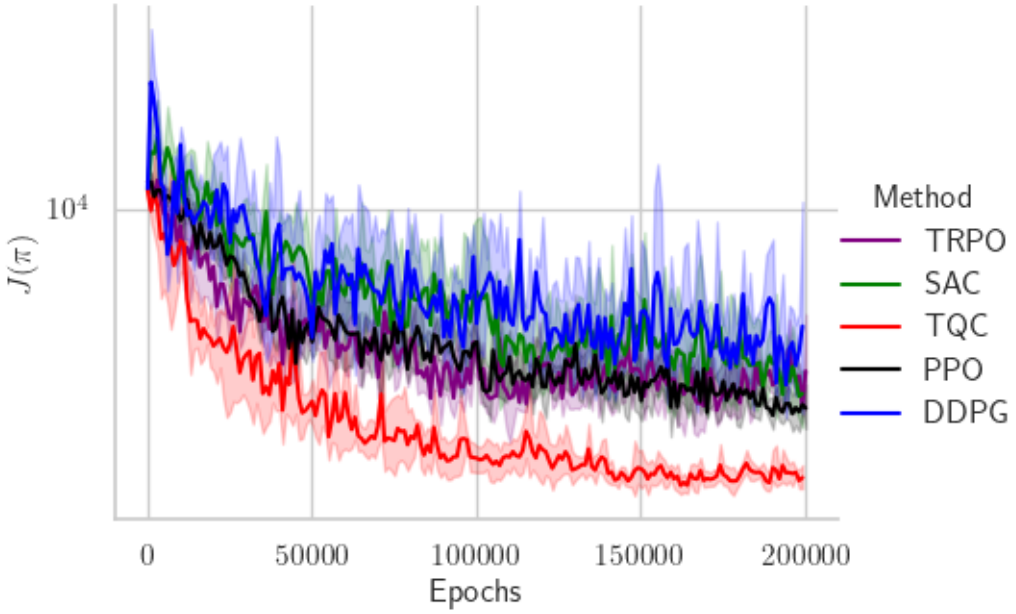


Figure 6.1: Training dynamics of the deep RL algorithms on the KS system over 2×10^5 environment steps. The x-axis is the number of environment steps, and the y-axis is the objective value.

or a learnable value (which is inherited from the follow-up SAC paper Haarnoja, A. Zhou, Hartikainen, et al. 2019 by the same authors). This ablation study is performed over 5 independent runs. The results are presented in Figure 6.2.

First, the limit case when the number of quantiles is set to $N_q = 1$ fails to converge. This case amounts to learning the median of the distribution. Second, the critic ensemble size has a slight impact on the performance. Ensembling seems to have a minor positive impact. Third, when removing the entropy regularisation, the performance is degraded and the training dynamics are less regular. This suggests that the entropy regularisation adds smoothness to the learning landscape (Chapter 4 discusses this phenomenon). Thus, the combination of quantile regression and maximum entropy reinforcement learning seems to have the largest impact in this setup of the KS system.

6.5.3 Generalisation to other initial conditions

The second question mentioned in Section 6.4 is now addressed. The generalisation capability of the learned policies to out-of-training initial conditions is evaluated. To this end, the algorithm achieving the best performance after TQC in the previous experiment is retained (namely PPO) for a comparison analysis.

Recall that the Kuramoto-Sivashinsky system is initialised with a Gaussian perturbation around the equilibrium $x_{e_2^*}$. The initial distribution $\mathbb{P}_{X_0}$ is defined

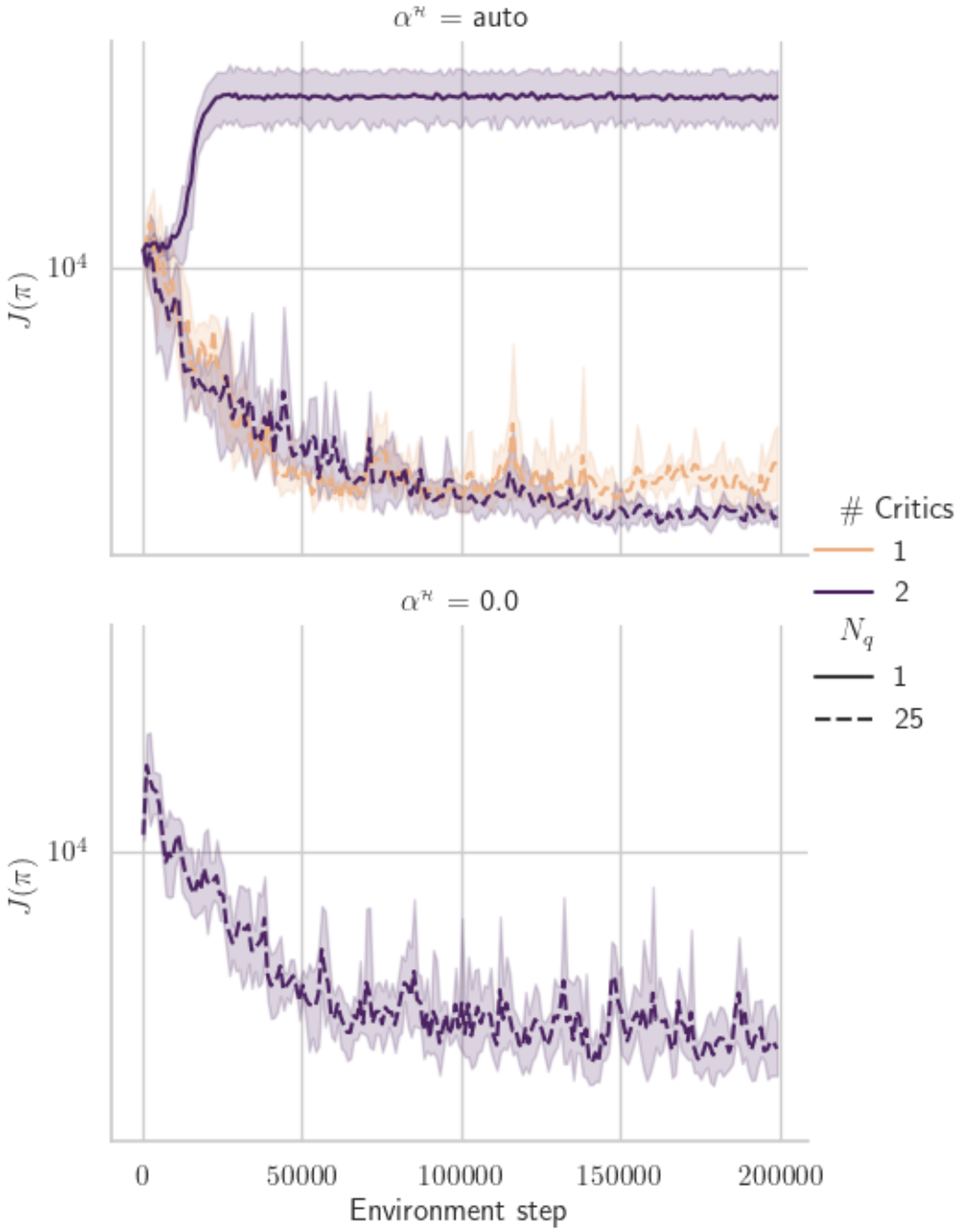


Figure 6.2: Ablation study. Comparison of TQC performance for varying sizes of the critic ensemble and varying number of quantiles N_q with a default learnable entropy coefficient $\alpha_\theta^{\mathcal{H}}$ (top). Reference performance without entropy regularisation (bottom). The x-axis represents the number of environment steps, and the y-axis is the objective value.

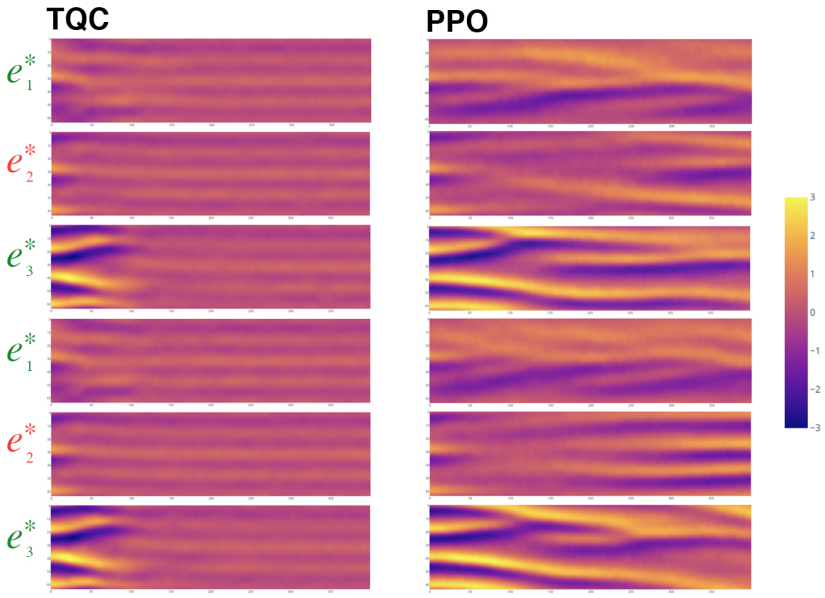


Figure 6.3: Qualitative comparison of TQC and PPO on other initial condition distributions. Learnt policies are evaluated on twice the environment time horizon (K). Each heatmap represents the controlled state (y-axis) of the KS system w.r.t to the time (x-axis). Rows: different initial points $x_e = x_{e_i^*}$ such that $X_0 \sim \mathcal{N}(x_e, \sigma_e^2 I_d)$. Columns: TQC, PPO.

such that $X_0 \sim \mathcal{N}(x_e, \sigma_e^2 I_d)$. As stated in Section 3.4.3, the KS partial differential equation exhibits four equilibria $(x_{e_i^*})_{i \in [\![1,4]\!]}$.[59] The training is performed with $x_e = x_{e_2^*}$, $\sigma_e = 10^{-1}$, and the evaluation presented in Figure 6.3 is performed with $x_e \in \left\{x_{e_1^*}, x_{e_2^*}, x_{e_3^*}\right\}$ with twice the rollout length used for training $2K = 400$. Figure 6.4 shows the result for the same setting, but the intensity of the noise (standard deviation) is 10 times higher. In this case, $X_0 \sim \mathcal{N}(x_e, 10\sigma_e)$.

Some observations can be extracted from the figures. On one hand, TQC stabilises the dynamics relatively well from any starting $x_e \in \left\{x_{e_1^*}, x_{e_2^*}, x_{e_3^*}\right\}$ for a time horizon of $2K$, while PPO stabilises only dynamics starting from $x_e = x_{e_2^*}$ for a rollout length equal to K. On the other hand, TQC is robust to the increase of noise intensity.

Note that quantitative results evaluating the objective function on controlled trajectories from other initial conditions confirm the qualitative observation. However, those results are kept from the reader, for the sake of brevity.

[59] Since the state space $\mathcal{X}$ for KS is a function space. The equilibria are function of the space $z \in [0, L_{\mathcal{X}}] \to x_{e_i^*}(z)$. In practice, those functions are discretised with a finite dimension $d_X \in \mathbb{N}_+$

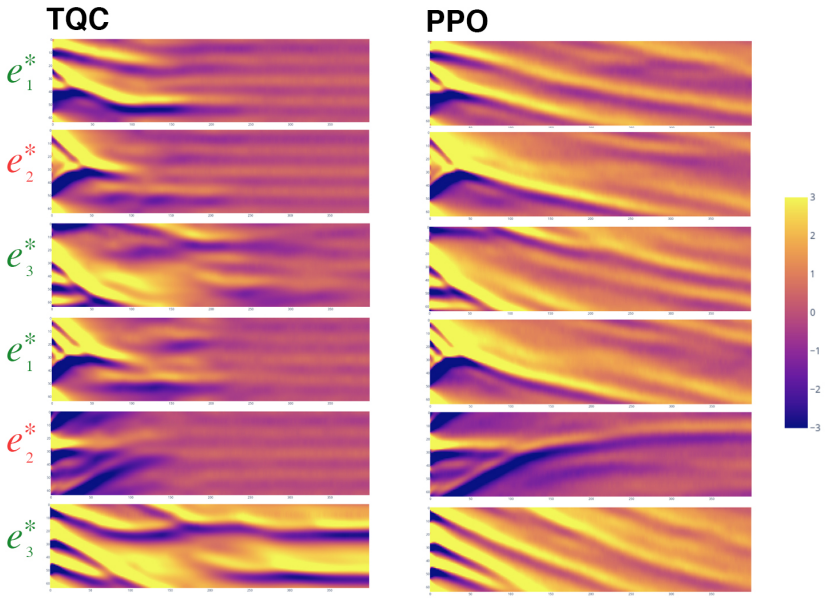


Figure 6.4: Qualitative comparison of TQC and PPO on other initial condition distributions. Learnt policies are evaluated on twice the environment time horizon ($2K = 400$). In this case the noise intensity is set to $10\sigma_e > 0$, ten times the value used for training. Each heatmap represents the controlled state (y-axis) of the KS system w.r.t to the time (x-axis). Rows: different initial points $x_e = x_{e^*_i}$ such that $X_0 \sim \mathcal{N}(x_e, \sigma_e^2 I_d)$. Columns: TQC, PPO.

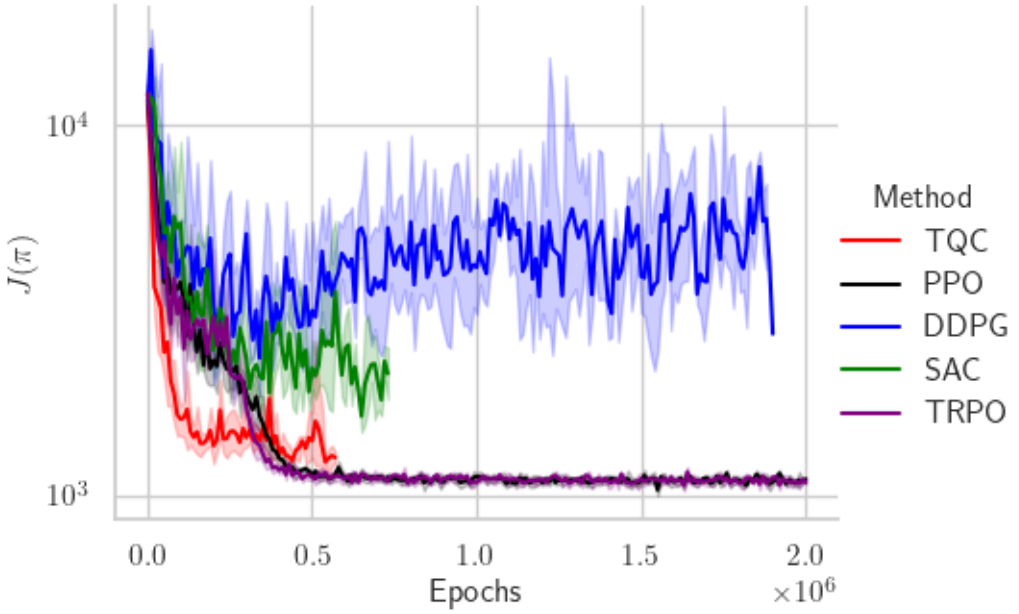


Figure 6.5: Training dynamics of the deep RL algorithms on the KS system over 2×10^6 environment steps. The x-axis is the number of environment steps, and the y-axis is the objective value. Some algorithms do not have training dynamics data after some point since all algorithms are allowed the same training clock-time but do not have the same sample complexity.

6.5.4 Asymptotic performance

This last part investigates the asymptotic performance of the algorithms. Consequently, the sample size is increased by a factor of 10 such that $m = 2 \times 10^6$. Figure 6.5 shows the training dynamics of the deep RL algorithms on the KS system over $m = 2 \times 10^6$ environment steps. TQC presents impressive learning speed, but the on-policy algorithms show better performances after roughly $m = 4 \times 10^5$ environment steps. Moreover, TQC is roughly four times slower than the on-policy algorithms.

6.6 Conclusion

This chapter introduces the Distributional Reinforcement Learning framework and its application to the control of chaotic systems. The Kuramoto Sivashinsky PDE is used as a benchmark system. The DRL algorithm used is the Truncated Quantile Critics algorithm, which is an extension of the SAC algorithm. The performance of TQC is compared to other baseline deep RL algorithms and shows promising results in terms of sample efficiency. However, the algorithm is computationally slower than all other baseline algorithms.

Further work includes understanding the reasons behind the performance of TQC and the application to more complex fluid flows.

7 Towards Neural Controlled Delay Differential Equations for Model Based Control

This chapter presents a project that aims to address particular challenges of data driven control with continuous time dynamics modelling. The neural differential equation framework for continuous time approximation is presented with its underlying motivations, then preliminary results are presented and discussed.

7.1 Introduction

This chapter introduces a unified way of addressing, to some extent, at least three main challenges in data-driven control: sampling time robustness, partial observability, and dynamics delays. These questions will be discussed in the following sections.

The underlying idea is originally motivated by the need to model the lag between actuator and sensor signals observed in the Cavity Flow Control problem. It appears that a neural model of the continuous-time control-free dynamics called *Complementary Deep Reduced Order Model* (CD-ROM) (Menier et al. 2023), that aims at reconstructing the full system state dynamics with a history of incomplete observations, was developed within the research group associated with this work.

In this context, bibliographic research led to three central articles on Model-Based Reinforcement Learning in continuous time that were used to frame and build the present study. The first paper Du, Futoma, and Doshi-Velez 2020 models semi-Markov Decision Processes (with random decision times, see Sections3.1.2) with Neural Ordinary Differential Equations (NODE), an actor-critic algorithm and optionally Model Predictive Control (MPC). The second one (Yildiz, Heinonen, and Lähdesmäki 2021) addresses the question of robustness to time discretisation schemes in Reinforcement Learning using a Continuous-Time Reinforcement Learning (CTRL) approach with NODE. The last paper (S. Holt et al. 2023) introduces a learning-based control method for continuous-time delayed

dynamics, where the authors combine a continuous-time model of the dynamical system, which is built over the Laplace transform S. I. Holt, Qian, and Schaar 2022, with a gradient-based MPC algorithm.

Those three articles encompass the challenges aforementioned and provide a solid reference for the development of the present work. The next section addresses the question of the robustness to time discretisation and irregular sampling times of learning-based control algorithms.

Nagabandi et al. 2018; Meng, Gorbet, and Kulić 2021; Bradtke and Duff 1994

7.1.1 Continuous-Time Reinforcement Learning: Temporal Abstraction

As emphasised in the introduction of this manuscript (Section 1.4.1), the question of robustness is a significant concern for data-driven flow control. Knowing already that reinforcement learning methods are extremely sensitive to the choice of training hyperparameters, a growing interest regarding the robustness of RL methods to time or spatial discretisation schemes has emerged recently in the literature (Tallec, Blier, and Ollivier 2019). For instance, Kidger et al. 2020 show that their particular neural ODE model[60] is the continuous-time limit of a Recurrent Neural Network (RNN) that handles irregular time series, in a non-controlled setting. Those developments are motivated by the recent advances and challenges in the application of learning-based control to real-world systems (Dulac-Arnold, Mankowitz, and Hester 2019). This brings again the concept of *temporal abstraction* that was addressed extensively in Chapter 5 where a method governing the data sampling times is used to reduce the sample complexity of a model learning procedure.

In this chapter, the abstraction is obtained by design through modelling the continuous-time dynamics of the system to approximate an optimal solution of the continuous-time control problem. This data-driven approach is now referred to as *Continuous-Time Reinforcement Learning* (CTRL) in the literature (Munos and Bourgine 1997; Doya 2000), but it amounts to be a learning-based approach to Stochastic Optimal Control defined in Chapter 2.

7.1.2 Partial Observability: Information States

There has been a long history of research work on both the theoretical and practical aspects of Partially Observed Markov Decision Processes (POMDPs) since the publication of the paper Åström 1965 that laid the theoretical foundation of the problem. As in the standard RL literature, at least two main branches of research can be identified. First, a theory referred to as “exact”

[60] This model is also termed as Neural Controlled Differential Equation (NCDE) in that paper but for another reason (the control in Rough path theory is the signal against which the dynamics is integrated).

relying on Dynamic Programming methods where optimality guarantees are obtained (Sigaud and Buffet 2010, Chapter 7). In general, the exact algorithms obtained have a large computational complexity with respect to the size of the state and action spaces due to the complexity of the constructed representations of the system. To counter this, the second branch of research focuses on "approximate" solutions (Cassandra 1998). A concise but complete historical overview of the different approaches for exact and approximate planning with POMDPs is given in Subramanian et al. 2022. The PhD thesis Cassandra 1998 and the seventh chapter of the book Sigaud and Buffet 2010 provide a strong presentation of the question, mainly in the finite state and action space case (exact, tabular case). For a presentation in general state and action spaces, the reader is referred to the book O. Hernández-Lerma 1989. The article Alt, Schultheis, and Koeppl 2020 considers the problem of partial observability in continuous time.

Several approaches encountered in the literature construct an augmented state based on the history of observations such that the augmented dynamics become Markovian (Bertsekas 2000). A first important approach is made with a random representation of the state, called the *belief state*.

Belief State

The belief state approach shares the same idea as the Bayesian approach presented in Section 3.2.3, where a guess of the state is represented by a probability distribution. This also echoes to the relaxed control theory presented in Section 2.2.5 where the control is a probability distribution over the control space (O. Hernández-Lerma 1989). In the PO-MDP case, the belief state is a probability distribution over the state space which represents the agent's belief about the current state of the system. In fact, the belief state is in general a filter (see Definition 2.2.11) that estimates the state of the system given the history of observations (Andrieu and Doucet 2002). The belief state is updated at each time step by the observation and the control. Viewing the problem in the space of belief states, the PO-MDP is transformed into a Markov Decision Process (MDP) where the state space is the space of belief states. A drawback of this approach is the curse of dimensionality, as the belief state is a probability distribution over the state space for which the complexity grows exponentially with the dimension of the state space (Sigaud and Buffet 2010).

Information States and Sufficient Statistics

Since the history process constructed from any stochastic process is a Markov process, methods based on Markovian representations such as Dynamic Programming (see Section 2.2.6) can be applied using the history process as augmented state for a augmented Markov Decision Process (MDP). However,

this default approach is not efficient because algorithms are carried out over a space of expanding dimension (Bertsekas 2000).

An alternative to the filter (belief state) approach is to consider a deterministic transformation of the history process that preserves the information of the system. A function $\varphi(H_t)$ of the random observation history H_t at time $t \in I$ (or h_t in the deterministic case, see Section 2.3.2) can be considered. This transformation of the history is called the *information state* at time $t \in I$. The information map φ should be viewed as a mapping from the space of maximal-length history to a space of finite dimension.

On the contrary, if the information map was chosen as $\varphi = Id$, the information state would be the history itself ($\varphi(H_t) = H_t$). In this case, the dimension of the information state would increase with time, which is not desirable for algorithmic applications.

Thus, the information state has the property to produce a compressed representation of the information for which the dimension does not increase w.r.t. time. As mentioned in the previous paragraph, the information conservation property defined by the information state is formally defined as $\varphi(H_t)$ being a sufficient statistic Barra 1971; M. Hoffman 2015 for the histrory process H_t . In probabilistic terms, the two random variables convey the same information about the system.[61] Precisely, conditional probabilities given the information state are the same as conditional probabilities given the history process.

Takens' Theorem

Another point of view coming from the theory of dynamical systems is given by the Takens' theorem (Takens 1981; Noakes 1991). This theorem states that a deterministic dynamical system can be reconstructed from scalar-valued partial measurements of the system. In other words, a well-chosen information state (see Section 7.1.2) can be used to reconstruct, up to a diffeomorphism (Fejoz 2017), the manifold on which the system evolves (thus the geometry is not preserved while the ergodic statistics are) (Coudène 2013). This result is also known as the delay embedding theorem because the information state is defined as a finite size rolling window of the history process. The size of this window is called the embedding dimension and is strictly greater than twice the dimension of the system attractor. A stochastic version of the Takens' theorem is given in Barański, Gutman, and Śpiewak 2020.

[61] Echoing the notion of information available to the agent or controller (see the footnotes of Section 2.2), a sufficient statistic for a random process H_t is a random variable $\varphi(H_t)$ that generates the same σ-algebra as the history process. Consequently, conditioning on the information state $\varphi(H_t)$ is equivalent to conditioning on the history process H_t. It is sufficient to know the information state to reconstruct the possible events that can be identified from the history process. The advantage of the information state is that it is of fixed dimension and does not increase with time.

7.1.3 Delay in Dynamical Systems

Delayed Dynamical systems can be seen as partially observable systems where the observation may be the state of the system at a previous time. In this particular case of imperfect state information Bertsekas 2000, it is natural to construct information states based on the history of observations as discussed in one of the seminal papers Kim and Jeong 1987 (see also Bertsekas 2000, p. 35). Later on, White III 1988 proves analytically that the observation history improves performances. Bander and White 1999 shows a sufficient statistic in the presence of observation delay is a specific belief state. M. Agarwal and Aggarwal 2021; W. Wang et al. 2024 use the delay information to construct augment the state with a delay-aware window of the past history.

Some work explicitly constructs MDP from delayed MDP (Altman and Nain 1992; Katsikopoulos and Engelbrecht 2003; B. Chen et al. 2021).

Walsh et al. 2008 uses a model-based RL approach to handle observation and rewards delays. Lancewicki, Rosenberg, and Mansour 2022 is the first study that considers regret minimisation in the important setting of MDP with delayed feedbacks. This paper provides bounds and also considers adversarially changing costs. To go further, Ramstedt et al. 2020 considers random delays in the observation process.

7.2 Neural Controlled Delay Differential Equations

7.2.1 Vector Field Parameterisation

Considering a Partially Observed Differential equation as defined in Example 2.2.2.[62] For such a system, the state dynamics are characterised by the operator f and the observation dynamics by the operator g. Also, a parametric hypothesis space $\mathcal{F}_f = \Theta_f$ for the state dynamics and $\mathcal{F}_g = \Theta_g$ for the observation dynamics are considered. The term *Neural Controlled Delay Differential Equations* refers here to the deterministic differential equation obtained by parameterising the state and observation dynamics operators by $\theta_f \in \Theta_f$ and $\theta_g \in \Theta_g$, respectively. In general, those parametrised models are neural architectures such as feedforward or convolutional neural networks. This leads to the following system of equations:

$$\partial_t x_t = f_{\theta_f}\left(x_t, x_{t-\tau_X}, u_t\right) \tag{7.1}$$

and

$$\partial_t y_t = g_{\theta_g}\left(x_t, x_{t-\tau_Y}, u_t\right) \tag{7.2}$$

where $\theta_f \in \Theta_f$ and $\theta_g \in \Theta_g$ and the time delays $\tau_X \in \mathbb{R}^+$ and $\tau_Y \in \mathbb{R}^+$ are fixed and known. The time interval is finite and given by $I = [t_0, T]$ with $T < \infty$. A

[62] The system need not to be necessarily defined on the whole space $\mathbb{R}^{d_X} \times \mathbb{R}^{d_X} \times \mathbb{R}^{d_U}$ but on any open set of $\mathcal{X} \times \mathcal{X} \times \mathcal{U}$, where the equation is well-defined.

solution[63] of the neural differential equation above is denoted by $(x_t^\theta)_{t\in I}$ where $\theta \in \Theta_f$ for the state and a similar notation is used for the observation.

Henceforth, in this work and more generally in the field of Physics Informed Machine Learning (Karniadakis et al. 2021), neural differential equations are positioned within the domain of system identification (Ljung 1999). Actually, recent efforts (Ayed et al. 2019; Rackauckas et al. 2021; Menier et al. 2023; Monsel et al. 2024) have been made to apply neural differential models for modelling physical systems, particularly those of computational fluid dynamics. In the control context, they pertain to the category of model-based methods (see Section 3.3.1).

As a matter of fact, it is worth mentioning that the neural differential equation approach was originally designed to approximate and abstract very deep neural networks. The following remark gives details about the origins of the method.

Remark 7.2.1. *During the early days of modern deep learning for computer vision, neural architectures became increasingly deep (Simonyan and Zisserman 2015). However, a degradation problem has been exposed: the model accuracy saturates and degrades as the network depth increases. To overcome this issue, the ResNet architecture was introduced (He et al. 2015). The heart of the ResNet architecture is the residual block defined as*

$$x_{t+1} = x_t + \bar{f}_{j,\theta_f}(x_t) \tag{7.3}$$

where $\bar{f}_{j,\theta_f}$ is the j-th residual layer (or block of layers) with parameters θ_f. Given an input data point x_0, the output x_T of the ResNet model is obtained by propagating the input through the residual layers together with the remaining layers of the network.

On the other hand, when discretising a Markovian and non-controlled version of the state dynamics given by Eq. (7.1) *with an Euler scheme, the following equation is obtained:*

$$x_{t+1} = x_t + \delta f_{\theta_f}(x_t) \tag{7.4}$$

where δ is the time step of the Euler scheme. If the time step is absorbed by the model $\bar{f}_{j,\theta_f}$ in Eq. (7.4)*, the residual block in Eq.* (7.3) *is recovered. Thus, for a neural differential equation with a fixed time horizon T, the time step of the Euler scheme controls the depth of the neural network. This way, neural network architectures and vector fields maintain a close relationship. A data point is propagated through the neural network layers in a similar way as a point following the flow defined by a vector field. The seminal Neural Ordinary Differential Equation (NODE) model (R. T. Q. Chen et al. 2018) paved the way for this connection.*

[63] The existence of the solution should be guaranteed when the control and the operators are continuous w.r.t. time. For a non-continuous control or the uniqueness question, a combination of the general version of the Cauchy-Lipschitz theorem (Trélat 2005) with its delay differential equation version (Hale 1971) is required.

The neural differential equation approach was initiated by R. T. Q. Chen et al. 2018. The PhD thesis Kidger 2021 provides a comprehensive manual on the subject.

The neural differential equation approach combines several advantages. First, it benefits from the approximation power of neural networks. Second, it allows for a memory-efficient gradient computation. Third, it benefits from the strong theoretical understanding of differential equations.

7.2.2 About Optimisation

Learning Task

The optimisation is described for state dynamics learning but is equivalent for observation dynamics. The learning task is defined by the L^2 distance between a piece of trajectory $(x_t)_{t\in I'}$ issued from the neural vector field f_{θ_f} and its corresponding true trajectory $(x_t)_{t\in I'}$ determined by the target operator f, where $I' \subset I$. Formally, the loss function is defined as

$$\ell\left((x_t)_{t\in I'}, f_{\theta_f}\right) \coloneqq \|(x_t)_{t\in I'} - (x_t^{\theta_f})_{t\in I'}\|^2_{L^2(I')} = \int_{I'} (x_t - x_t^{\theta_f})^2 dt \tag{7.5}$$

where $\|\cdot\|_{L^2(I')}$ denotes the L^2 norm over the time interval I'.

The learning task being defined, a generalisation error (called risk) needs to be introduced. To this end, a distribution over pieces of trajectories is considered and denoted by $\mathbb{P}_{(x_t)_{t\in I'}}$.

$$\mathcal{L}\left(\mathbb{P}_{(x_t)_{t\in I'}}, f_{\theta_f}\right) \coloneqq \mathbb{E}^{\mathbb{P}_{(x_t)_{t\in I'}}}\left[\ell\left((x_t)_{t\in I'}, f_{\theta_f}\right)\right] \tag{7.6}$$

Of course, in practice the distribution $\mathbb{P}_{(x_t)_{t\in I'}}$ is unknown. A frequentist estimation (see Section 3.2.3) of this distribution is obtained by sampling pieces of trajectories from the true dynamics. These trajectories form a dataset $\mathcal{D} = \{(x_{t,i})_{t\in I'}\}_{i=1}^K$ where $K \in \mathbb{N}^*$ is the number of samples. An empirical distribution $\widehat{\mathbb{P}}_{(x_t)_{t\in I'}} \coloneqq \frac{1}{K}\sum_{i=1}^K \delta_{(x_{t,i})_{t\in I'}}$ is then obtained. Moreover, the integral over time characterising the L^2 metric in Eq. (7.5) is approximated by a Riemann sum (see Remark 2.4.3) with $N \in \mathbb{N}^*$ rectangles of width $\delta_{t_k} = t_{k+1} - t_k$. The empirical risk is then defined as

$$\widehat{\mathcal{L}}\left(\widehat{\mathbb{P}}_{(x_t)_{t\in I'}}, f_{\theta_f}\right) \coloneqq \frac{1}{K}\sum_{i=1}^N \sum_{k=1}^K \delta_{t_k}(x_{k,i} - x_{k,i}^{\theta_f})^2 \tag{7.7}$$

where $x_{k,i}$ denotes the state of the system at time $t_k \in I'$ for any $i \in [\![1, K]\!]$ and partition $(t_k)_{k=0}^N$ (more precisely, this sampling and discretisation scheme corresponds to the procedure described in Section 2.3 with the compatibility condition of Section 2.5). Similarly, the point $x_{k,i}^{\theta_f} \in \mathcal{X}$ is the state of the system

at time t_k for the neural vector field f_{θ_f}. In practice, trajectories are obtained by numerical integration (specifically delay differential equation solvers if the system is not Markovian).

Remark 7.2.2. *The optimisation procedure has been explained for the state dynamics f_{θ_f}. The same procedure is applied for the observation dynamics g_{θ_g}. Depending on the context, whether the state or observation dynamics are learned, the loss function $\mathcal{L}$ is defined accordingly. In practice, the observation dynamics are learned since the state dynamics are unknown. There is no work yet on learning both the state and observation dynamics simultaneously.*

Now, the question of the optimisation of the neural differential equation is addressed. The optimisation is performed by ordinary gradient descent.

The Adjoint State Method

There exists an elegant way due to R. T. Q. Chen et al. 2018, and extended by Zhu, Guo, and W. Lin 2021 in the delayed case, for deriving the risk gradient without the need for the backpropagation $\nabla_{\theta_f}\mathcal{L}(\widehat{\mathbb{P}}_{(x_t)_{t\in I'}}, f_{\theta_f})$ with respect to the weights θ_f of the neural operator f_{θ_f}. Indeed, Remark 2.2.12 already discussed how a control problem can be seen as a constrained minimisation problem where the objective function is the generalisation error and the constraints are the differential equations. Expanding this point of view, a Lagrangian formulation of the problem is obtained. This Lagrangian is defined as

$$\mathscr{L}\left((x_t)_{t\in I'}, (\lambda_t^{\mathscr{L}})_{t\in I'}, \theta_f\right) := \mathcal{L}\left((x_t)_{t\in I'}, f_{\theta_f}\right) + \int_{I'} \lambda_t^{\mathscr{L}}\left(\partial_t x_t - f_{\theta_f}\left(x_t, x_{t-\tau_X}, u_t\right)\right)dt \tag{7.8}$$

Without going too much into details, a dual formulation of this constrained minimisation problem can be obtained (McNamara et al. 2004; Stephany et al. 2024). The Lagrange multipliers $(\lambda_t^{\mathscr{L}})_{t\in I'}$ describe trajectories that satisfy some differential equation with terminal condition. Thus, these equations are solved backwards in time. The variable $\lambda_t^{\mathscr{L}}$ for any $t \in I'$ is called the adjoint state.[64] Finally, the gradient $\nabla_{\theta_f}\mathcal{L}(\widehat{\mathbb{P}}_{(x_t)_{t\in I'}}, f_{\theta_f})$ is a function of the adjoint state trajectories and the state trajectories.

Additionally, this method can also be obtained from a result in optimal control theory called the Pontryagin Maximum Principle (PMP) (Trélat 2005). In this case, the weights θ_f are considered as another control parameter $(u'_t)_{t\in I'} \equiv \theta_f$, and the adjoint equations arise as a consequence of the PMP.

The advantage of this method, which is underlined in R. T. Q. Chen et al. 2018, is the non-necessity of storing the trajectory values during the forward computation of the trajectory to compute the objective gradient, while standard backpropagation requires the storage of these values. A drawback is the

[64] The term adjoint refers to the adjoint operator linked to the dual problem formulation (Hiriart-Urruty and Lemarechal 2013).

need to solve the adjoint equation, that depends on the backward integration of the state dynamics. This leads two choices: either storing the state values during the forward computation or recompute them during the backward integration. If the recomputation is chosen, the difference between the backward and forward integration of the state dynamics induces inaccuracies in the gradient computation. An extensive description and comparison with the more standard backpropagation through solver method is given in Kidger 2021. The backpropagation through solver method is introduced in the next section.

Backpropagation through solver

On the other hand, if the numerical solver used to integrate the differential equation is differentiable (is a composition of differentiable operations), then any automatic differentiation library can be used to compute the risk gradient $\nabla_{\theta_f} \mathcal{L}(\widehat{\mathbb{P}}_{(x_t)_{t\in I'}}, f_{\theta_f})$. This method is called *backpropagation through solver* and despite being less memory-efficient than the adjoint state method, it is more stable and faster because of the advances in automatic differentiation libraries (Bradbury et al. 2018; Ansel et al. 2024) and the exact computation of the gradient.

7.3 A Data-Driven Approach to Continuous-Time Flow Control

A promising data-driven approach in continuous-time data-driven control can be built on the concepts and models presented in the first sections of this chapter (Sections 7.1.1-7.1.2-7.1.3) . Surely, the resulting method would be model-based, *i.e.* leveraging the features carried by the neural differential model to perform reliable control procedures.

Here, a reliable control should be understood as an algorithm based on a neural differential model that handles most of the main challenges of Flow Control enumerated in the introduction of this manuscript 1. Ideally, the method covers all those challenges. Though, the work presented here deals only with one of these issues: the presence of a delay in the state or observation dynamics. The other challenges are left for future work.

7.3.1 Programme for a Neural Differential Control Algorithm

As stated in Section 7.1, this project builds principally on two previous studies on learning-based continuous time control.

Neural Differential Continuous-Time Reinforcement Learning

First, the article Yildiz, Heinonen, and Lähdesmäki 2021 where a model based continuous-time reinforcement learning approach is defined to improve robustness to irregular sampling times. From that work, two essential methodological steps are retained in order to construct a programme for a reliable neural differential control algorithm. Of course, modelling the system dynamics with a neural differential equation is the first step. The second step, which is more involved, is to solve the Bellman equation with an actor-critic scheme. The resolution of this continuous-time Bellman equation (*a.k.a.* the *Dynamics Programming Principle* (DPP) in continuous time, see Section 2.2.6) is the most challenging part of the programme since the formulation of the DPP is not straightforward for time-delayed systems.[65]

A Neural Operator for Delayed Systems and Model Predictive Control

Second, the work of S. I. Holt, Qian, and Schaar 2022 treats systems with observation delays and learns offline a neural differential representation of the dynamics to perform model predictive control. The model predictive control part is retained here as a way to ensure the quality of the learnt model, without caring about actor or critic training. However, this approach is inherently less computationally efficient than an actor-critic scheme because it requires solving an optimisation problem at each decision instant. On the other hand, it may require less data since no reinforcement learning is involved. Thus, those two approaches are complementary and can be somehow combined to build a reliable control algorithm.

Programme Details

Accordingly, a programme for a neural differential control algorithm can be defined as follows:

1. Modelling delayed and partially observed systems with a neural controlled delay differential equation.
2. Apply Model Predictive Control with the learnt model.
3. Adopt the continuous-time reinforcement learning approach to learn a policy and obtain an end-to-end learning-based control algorithm.

Only the first step is treated in this thesis while the two others are left for future work. The first item naturally extends previous studies on NDDE (Zhu, Guo, and W. Lin 2021; Monsel et al. 2024; Stephany et al. 2024) by introducing

[65] Not to mention that there is no Q-learning in continuous time since the state-action Q-function collapses to the state value function (Tallec, Blier, and Ollivier 2019; Wiltzer et al. 2024).

a exogenous control input to the learning procedure. The last two elements of the programme trigger many questions and challenges. For instance, the MPC procedure needs to be adapted to delayed systems and continuous-time control. Also, the reinforcement learning approach should be adapted to the delayed and partially observed case while Deep Reinforcement Learning is mostly designed for Markov Decision Processes (MDP). Moreover, Neural Differential Models exhibit complex training dynamics (Kidger 2021) and the convergence of the learning algorithm is not guaranteed.

7.3.2 Modelling Delayed and Partially Observed Systems

This part of the thesis is dedicated to the first item of the programme for a neural differential control algorithm (see Section 7.3.1). The construction of a proper neural model for delayed and partially observed systems requires a progressive curriculum of questions that aim to validate the underlying features that are supposed to be captured by the model.

Question and Hypothesis

The following questions are addressed:

- Are the NDDE models (not necessarily controlled) more accurate approximators for standard dynamical systems than the classical NODE model?
- Does the information provided by the control input improve the approximation of the controlled dynamics?
- Do the NDDE models handle the delay in the state dynamics better than the NODE models?
- How the neural differential models perform against Flow Control sensors signals?

The first point verifies the claims of the seminal paper Zhu, Guo, and W. Lin 2021 that proves NDDE are better approximators than NODE (in the sense of the quantity of functions that can be approximated). The second aims to validate the natural hypothesis that the control information improves the approximation of the dynamics. This serves also as a sanity check for the control input implementation which is not trivial The third question verifies the hypothesis that the delay in the state dynamics is better handled by NDDE models than NODE models. The last question is a first step towards an application in Flow Control.

7.4 Experiments

In order to answer the questions raised in the previous section, a series of experiments are conducted. Basically, they consist in training neural differential models on trajectory data generated by a controlled dynamical system under different configurations (*e.g.* the time delay value).

7.4.1 Ablation Study

All the optimisation procedures (*a.k.a.* training or learning processes) are carried out with the Adam optimiser (Kingma and Ba 2015). When the system is partially observed, the learning task is defined on the observation dynamics. Four variants of the Neural Controlled Delay Differential Equation (NCDDE) model are trained in order to marginally extract information on the effect of the control input and the delay in the state dynamics. Typically, this approach is called an *ablation study* in the machine learning community. Concretely, the following models are considered:

- the Neural Ordinary Differential Equation (NODE) model which considers a Markovian system without delay, and no control input in Eq. (7.1) for the state dynamics or Eq. (7.2) for the observation dynamics.
- the Neural Controlled Differential Equation (NCDE) model which considers a markovian system without delay, and a control input.
- the Neural Delay Differential Equation (NDDE) model which considers a non-markovian system with delay, and no control input.
- the Neural Controlled Delay Differential Equation (NCDDE) model which considers a non-markovian system with delay, and a control input.

In view of answering the above questions by training neural differential models on well-chosen dynamical systems, multiple time series datasets are generated from different dynamical systems.

Finally, note that the objects and quantity considered here, in particular the delays, are considered continuous. The value of the dynamics for incompatible sampling times are estimated by linear interpolation (see Remark 2.4.2).

7.4.2 Time Series Dataset

A sampling procedure is performed to collect a dataset $\mathcal{D} = ((x_{t,i}u_{t,i})_{t\in I})_{i=1}^{m}$ of time series for every dynamical system configuration. All time series are generated by numerical integration (see Section 2.4) of the differential equations associated with the dynamical systems. In practice the elements of $\mathcal{D}$ are finite dimensional vectors of dimension $K \times (d_X + d_U)$ where $d_X \in \mathbb{N}^*$ is the state

dimension, $d_U \in \mathbb{N}^*$ is the control dimension that may result from the discretisation of the state space $\mathcal{X}$ or the control space $\mathcal{U}$ (see Section 3.4). The integer $K \in \mathbb{N}^*$ is the number of measurements extracted from a continuous signal. The sampling times (see Section 2.3 and Definition 2.3.3) are equidistant with a time step (inter-decision time) of $\eta \in \mathbb{R}^+$. The number of time series samples is $m \in \mathbb{N}^*$. Those trajectories are initialised from a random initial condition determined by a collection of probability distributions $(\mathbb{P}_{X_0,i})_{i=1}^m$. on the initial conditions of the dynamical system. Indeed, in practice the distribution of the m-th initial condition depends on the distribution of the previous trajectory. The typical example used both in the literature and this work is when the next initial condition is the last state of the previous trajectory. In fact, using the latter distribution could be a way to ensure the ergodicity (Benoist and Paulin 2000; Leroux 2019) of the dataset and the dynamical system.

Here, except for the Cavity Flow that is computationally expensive, the number of measurements is $K = 200$ (regularly spaced in time with a time step of $\eta = 10^{-2}$ that depends on the dynamical system) and the number of trajectories in the dataset $\mathcal{D}$ is $m = 400$. Finally, each of the configurations is trained over two random seeds. Qualitatively, the variation of the training dynamics over independent runs are much less important than what can be observed in deep reinforcement learning.

The different dynamical systems and their configuration is now presented.

Oscillators with Observation Delays

Two oscillators are considered: the Pendulum (see Section 3.4.4) and the Van der Pol oscillator (see Section 3.4.5). For each environment, $m = 400$ trajectories are generated. The distribution of the initial point is the standard distribution used throughout the thesis (see Section 5.3.2): $\mathbb{P}_{X_0,i} \sim \mathcal{N}(x_e, \sigma_e^2 \mathbb{I}_{d_X})$ for all $i \in [\![1, m]\!]$. Thus, the initial conditions are *i.i.d.*. The starting equilibrium is the bottom point for the pendulum and the zero point for the Van der Pol oscillator ($x_e = (0, 0)$ in both cases). The noise level on the initial condition is $\sigma_e = 10^{-1}$.

The control signal[66] is generated by a random process $(U_{t,i})_{t \in I}$ where $U_{t,i} \sim$ Unif$(\mathcal{U})$ for all $t \in I$ and $i \in [\![1, m]\!]$.

Each environment comes in several configurations with different observation delays $\tau_Y \in \{0, 10^{-2}, 10^{-1}\}$. The observation operator g of the pendulum is the standard trigonometric representation of the angle and angular velocity. The observation operator of the Van der Pol oscillator is the observation shift operator. At time $t \in I$, the observation operator is defined as (by a slight abuse of notation) $y_t = g(y_{t-\tau_Y})$ where $g = Id_{d_X}$ for the Van der Pol oscillator and similarly for the pendulum (under the trigonometric representation). Thus, three configurations are considered for each oscillator.

[66] This choice of control, which inherits from the discrete time approach, has been identified as a very bad choice *a posteriori*. Indeed, it results in a control signal that is *nowhere* continuous and thus may lead to very inaccurate path approximations within the vector field.

This choice creates a lag between an action and the observation of its effect. It is suitable for the study of the impact of observation delays on the neural differential models. A visualisation of inference for all the models on a trajectory drawn from the dataset is given in Figures 7.2 and 7.3.

Delayed Differential Equation

It also interesting to consider a simple delayed differential equation. The Mackey Glass equation (see Section 3.4.6) is a standard example of this kind of system. Instead of taking observation delays, the system state differential is now a function of a past state prescribed by a delay $\tau_X = 1$. The parameters of the Mackey-Glass equation are defined given in Section 3.4.6. Hence, no direct action lag should characterise the dynamics but only a feedback effect from a past state at a fixed delay.

Two choices of initial conditions are considered. First, $\mathbb{P}_{X_0,i} \sim \mathcal{N}(x_e, \sigma_e^2 \mathbb{I}_{d_X})$ with $x_e \in \mathcal{X} = \mathbb{R}$ the non-trivial equilibrium point and $\sigma_e = 10^{-2}$. The inter-decision time is $\eta = 10^{-1}$. Second, the initial condition for trajectory $i \in [\![1, m]\!]$ is the last state of trajectory $i-1$ (deterministic law) with the distribution being the same as the previous case for $i = 0$. This choice corresponds to the ergodic hypothesis of the dataset where a long trajectory is considered as a good approximation of the stationary distribution of the dynamical system. Moreover, two choices of action spaces are selected, $\mathcal{U} = \{0\}$ and $\mathcal{U} = [-10^{-1}, 10^{-1}]$.

Figure 7.5 shows the inference of the models on a trajectory drawn from the dataset.

Fluid Flows

The last environments studied are typical 2-dimensional fluid flows that are used in the literature on flow control, namely the Cylinder Flow (see Section 3.4.7), the Fluidic Pinball (see Section 3.4.7) and the Cavity Flow (see Section 3.4.7). All those systems are driven by the Navier-Stokes equations (Eq. 2.13) but their domain of definition and boundary conditions (geometry) differ to match the physical setup of the experiments. Moreover, the control input is embedded in the boundary conditions of the fluid flow to mimic real-world setups.

Fluid flows are governed by the Navier-Stokes equations. From this equation, a dimensionless[67] quantity denoted $\mathrm{Re} \in \mathbb{R}_+^*$ called the Reynolds number is derived (Candel 1995; Chassaing 2000). Broadly, the Reynolds number characterises the ratio of inertial forces (velocity) to viscous forces in the fluid flow. The higher the Reynolds number, the more turbulent (Lumley and Blossey 2003) (thus chaotic and complex), the flow is. Consequently, different Reynolds numbers are considered for each fluid flow. For the Cylinder and the Pinball

[67] Being dimensionless, this number allows for the comparison between the dynamics of different fluid flows.

flows, $\mathrm{Re} \in \{50, 90, 105, 120\}$. This choice is inspired by the route to chaos paper of Deng et al. 2018. For the Cavity flow, $\mathrm{Re} \in \{500, 5000, 7500\}$ which is inspired by the work of Barbagallo, Schmid, and Huerre 2009.

Regarding the initial states, independent trajectories are generated from a random initial condition drawn from the standard normal distribution around the equilibrium flow (called steady-state flow).

In the same way as the Mackey-Glass equation, two choices of control magnitude are selected. The zero control and $\mathcal{U} = [-10^{-2}, 10^{-2}]$.

Note that being computationally expensive, the models for the Cavity Flow are trained with $m = 40$.

7.4.3 Results

Approximation Capability

The first question in Section 7.3.2 discusses the expressive power of the delay differential equation extension of the neural differential model to learn the dynamics of the dynamical systems. For this task, it should be enough to focus on the uncontrolled dynamics whether by analysing the environment configurations where $\mathcal{U} = \{0\}$ or simply restricting the comparison to the NODE and NDDE models.

Despite performing quantitatively better in Figure 7.1, concluding towards an advantage for NDDE is not really fair since the trajectory data for this experiment is perturbed by a control signal which is not intended to be captured by the two models. In addition, the observation delay introduced should bias the comparison in favour of the delay-based model. On the other hand, Fluid Flows (7.4.2) and the Mackey-Glass equation (7.4.2) are more suitable for this comparison as configurations with no control are considered.

The uncontrolled version of the Cylinder Flow in Figure 7.6 and more significantly the uncontrolled version of the Fluidic Pinball in Figure 7.7 show that for uncontrolled trajectories, the NDDE performs better than the NODE model.

Regarding Mackey-Glass (Figure 7.4), the neural model based on ordinary differential equations is not able to capture the dynamics of the system while the NDDE model is able to approximate the system dynamics when the initial conditions are drawn from the stationary distribution. Here, the performance is given in the standard learning theory sense specified in Section 7.2.2.

Hence, this experiment provides empirical evidence on the approximation power of neural differential dynamics incorporating time delays. Those two fluid flows are partially observed. Thus, the arguments developed in Section 7.1.2 and Section 7.1.2 could justify this performance spread.

Control Information

The second question mentioned in the list of hypothesis aims at evaluating the utility of incorporating the control signal to the neural model in order to approximate controlled differential equations.

Recall that the control injected in the dynamics is open-loop (see Section 2.2.1). Consequently, the observation and control signals are independent; they share no information[68] (then their mutual information is equal to zero, by independence). This point of view notably argues that feedback controls (Definitions 2.2.5 and 2.2.10) signals intrinsically add less information than open loop controls. This choice ensures that the control signal is not redundant with the observation signal.

Closing this parenthesis on information, the NODE baseline should be compared with its NCDE extension on controlled dynamics for judging the impact of the control information on the resulting approximations. Comparing dark (NODE) and light (NCDE) green training and validation curves in Figure 7.1 and, with much less importance, in Figure 7.7, the impact of feeding the control signal to the model on the training and validation losses is clearly observed. Additionally, Figures 7.6 and 7.7 show near performances when $\mathcal{U} = \{0\}$. In the case of Mackey-Glass (Figure 7.4), the comparison between the NODE and NCDE models is not relevant since the ordinary differential equation models are not able to capture the dynamics of the system, regardless of the choice of initial conditions. However, considering delayed models, it can be observed that the NDDE model performs better than the NCDD model in the absence of control. An inverted behaviour is observed in the presence of control. This supports the hypothesis that the controlled models capture the control signal information.

Delays

The third line of analysis is devoted to the model performance in the presence of delays over the dynamics.

Figure 7.1 shows that NCDDE, at best, surpasses NCDE in modelling controlled dynamics with observation delays (middle and right columns), and at worst, achieves equal performances. However, it is not clear and probably unlikely that this performance is intrinsically due to the proper delay modelling in NDDE dynamics.

Indeed, recent results show non-interpretable delay values in general settings Monsel et al. 2024. Rather, the observation signal approximation may be improved by the state augmentation design of NDDE (Zhu shows NDDE can approximate functions that are not learnable with NODE).

[68] Again, viewing the information in terms of σ-algebras, this means the σ-algebra generated by the observation process is independant of the one generated by the control process.

However, when the delay is relatively high (top right graph) clearly both NCDDE and NDDE models outperform the others. In this configuration, the NDDE becomes the second best model despite being agnostic to the control signal. This suggests more information is carried by the delay than the control here.

When looking at the results for the delayed Mackey-Glass equation (Figure 7.4), only the delay-based models are able to capture the dynamics of the system, regardless of the choice of initial conditions.

Consequently, while DDE achieve better performances in modelling, the reason for such performance is still not understood and more work is expected in this direction. Definitely, interpretability is a key in the understanding of the neural differential models.

Fluid Flows

The last question deals with the ability of the neural differential models to approximate signals from fluid flow simulations.

Especially in the case of the Pinball flow (Figure 3.4.7), the NCDDE model achieves promising results in the uncontrolled cases. The Cylinder flow signal is also correctly approximated.

In general, all models fail to learn the time series where a non-zero control input is applied. The first hypothesis behind such a behaviour is the irregularity of the control signal (nowhere continuous) which is transmitted to the observation signal.

Investigations show that the control process is not damped from the controller to the sensor (causal relationship). Thus, the resulting fields to be learnt are very irregular. Another round of experiments with a smoother control signal is planned to confirm this hypothesis.

Regarding the Cavity flow, even without control, the observation signal is chaotic and very stiff with high frequencies.

Further work is planned to investigate the impact of the control signal on the observation signal in the fluid flow environments.

7.5 Conclusion

This final part of the thesis presents preliminary results in the domain of continuous time learning based control for partially observed and delayed dynamics.

The elements presented are part of a larger programme devoted to an autonomous learning scheme for fluid flow control. Notably, the Neural Controlled Delay Differential Equations model that is introduced achieves promising results in the presence of observation delays.

The control signal is shown to improve the approximation of the perturbed dynamics. However, the impact of the delay in the state dynamics is not yet

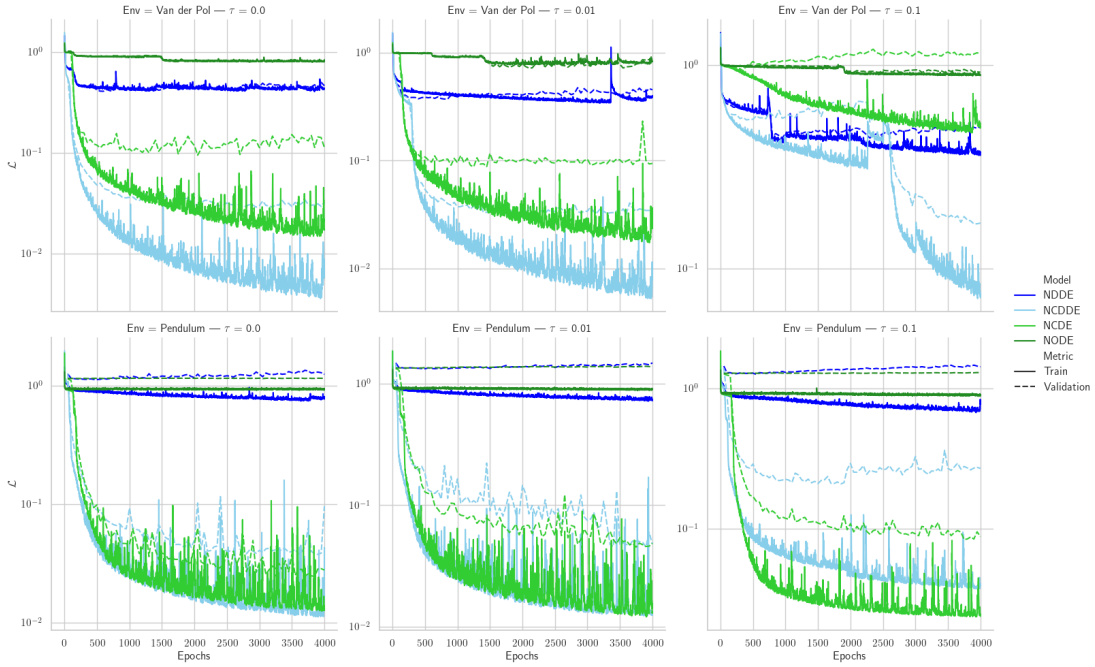
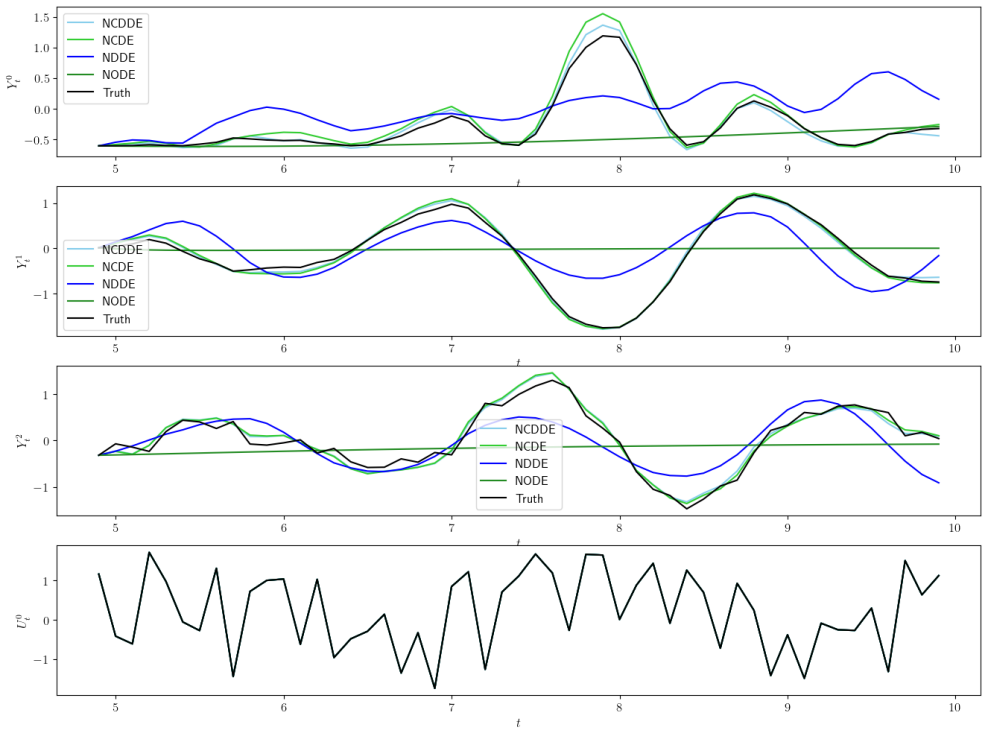
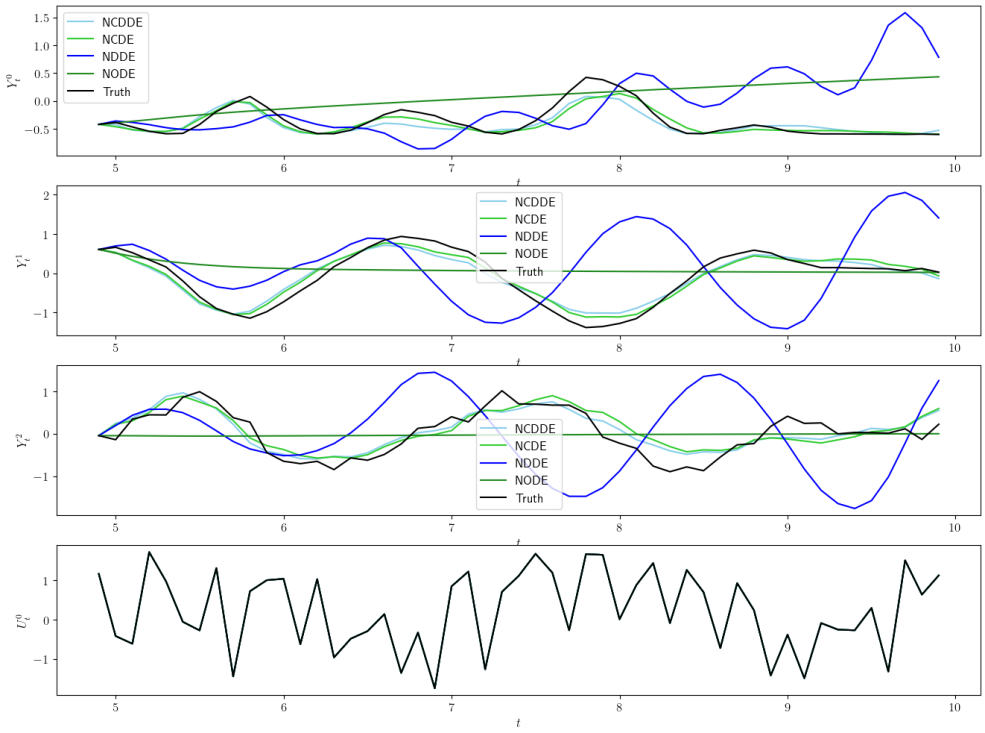


Figure 7.1: Training dynamics of the neural differential models under several observation delays τ_Y for the Van der Pol and the Pendulum environments. The x-axis represents the training epochs. The y-axis is the empirical training loss $\widehat{\mathcal{L}}$. Dashed lines represent the empirical validation loss. Rows: Van der Pol (top); Pendulum (bottom). Columns: $\tau_Y \in \{0, 10^{-2}, 10^{-1}\}$ (from left to right). Green curves are delay-based models. Blue curves are the baseline models. Lighter tones are their controlled version.

fully understood. Thus, more investigations should be carried out on the sole modelling part. In particular, understanding the real behaviour of the learnable delays, reducing the sample complexity.

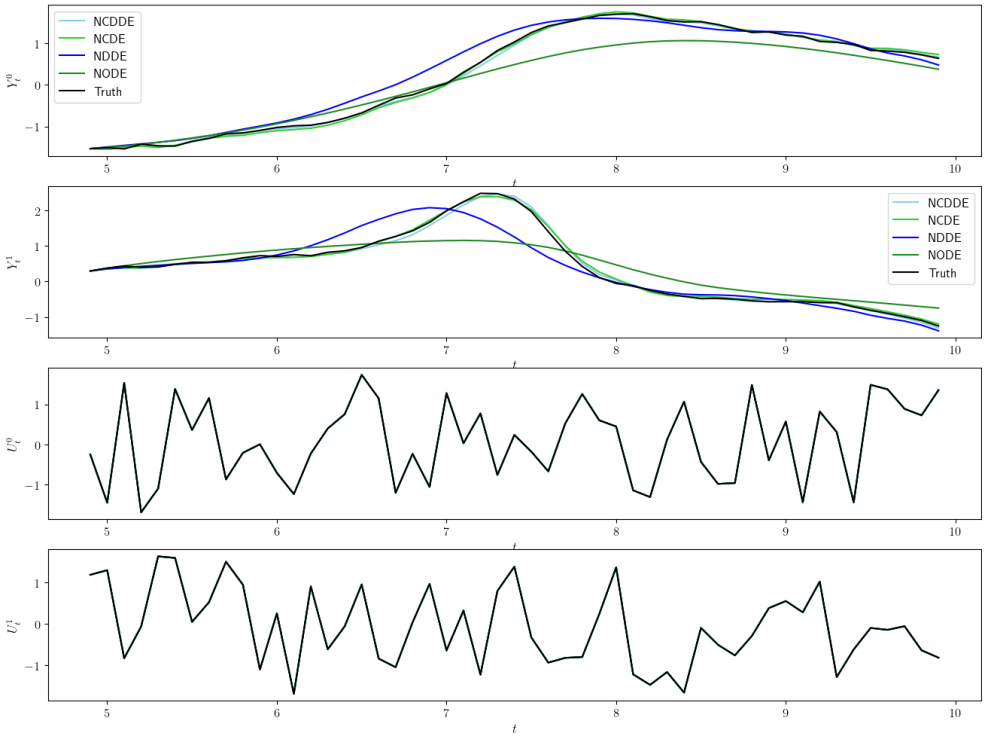


(a) $\tau_Y = 0$

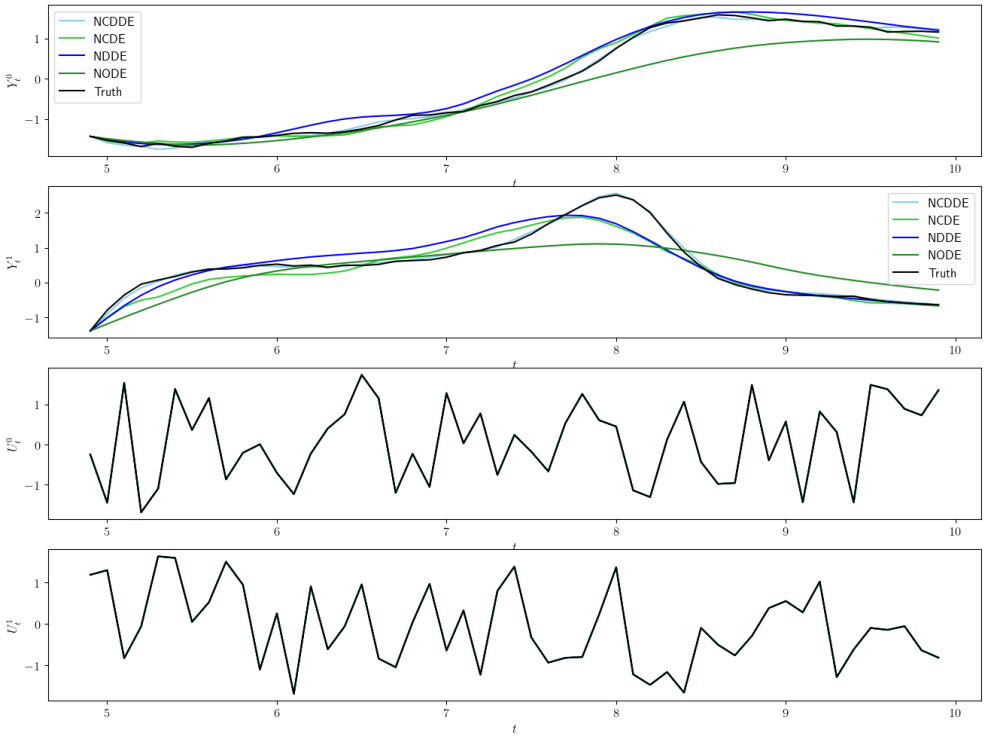


(b) $\tau_Y = 10^{-1}$

Figure 7.2: Inference on the training data for the Pendulum environment with different observation delays τ_Y. For each case, the last row is the control process.



(a) Case $\tau_Y = 0$



(b) Case $\tau_Y = 10^{-1}$

Figure 7.3: Inference on the training data for the Van der Pol environment with different observation delays τ_Y. For each case, the last two rows are the control signal.

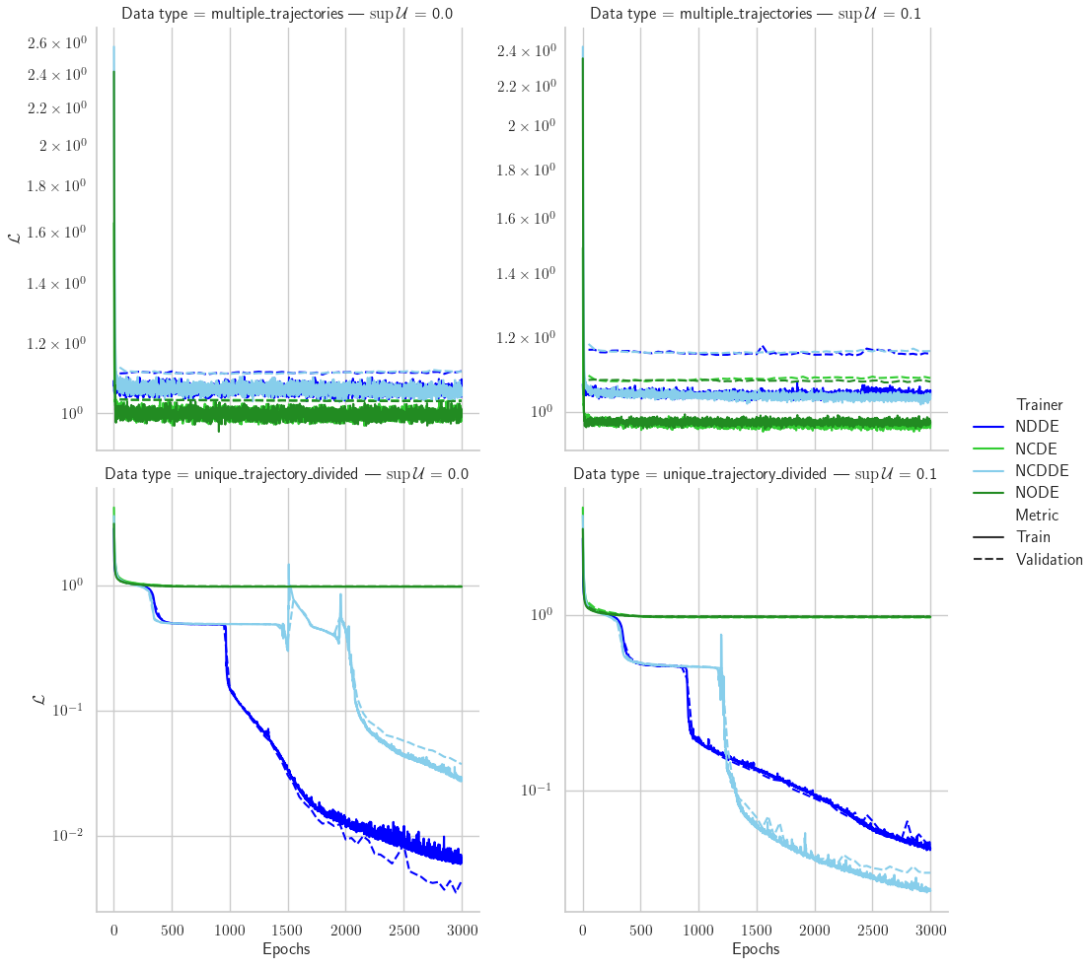
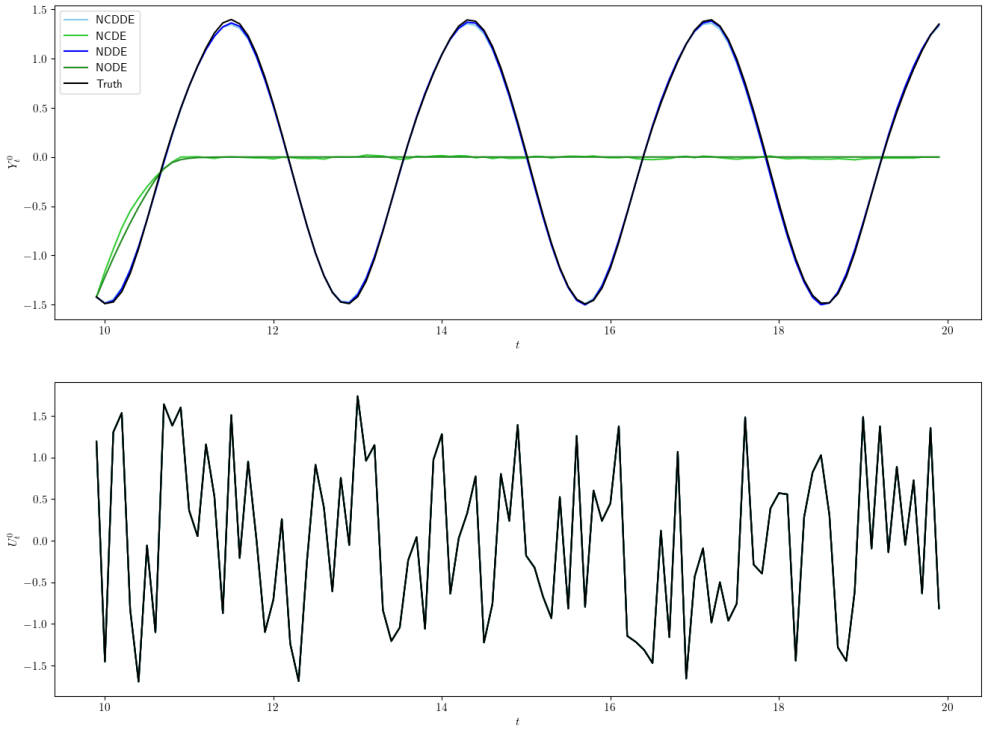
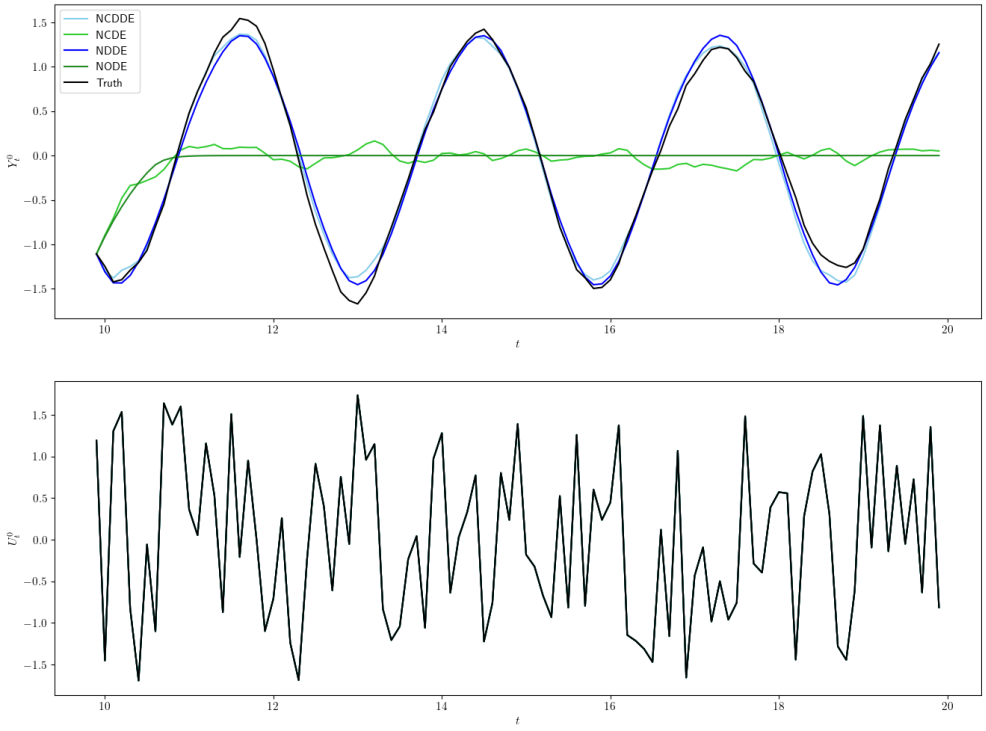


Figure 7.4: Training dynamics of the neural differential models under two control magnitudes ($\mathcal{U} = \{0\}$ and $\mathcal{U} = \left[-10^{-1}, 10^{-1}\right]$) for the Mackey-Glass environment and two kinds of initial conditions distribution. The x-axis represents the training epochs. The y-axis is the empirical training loss $\widehat{\mathcal{L}}$. Dashed lines represent the empirical validation loss. Rows: $\mathbb{P}_{X_0,i} \sim \mathcal{N}(x_e, \sigma_e^2 \mathbb{I}_{d_X})$ (top); $\mathbb{P}_{X_0,i} = \delta_{X_T,i-1}$ (bottom) for all $i \in [\![1, m]\!]$. Green curves are delay-based models. Blue curves are the baseline models. Lighter tones are their controlled version.



(a) Case $\mathcal{U} = \{0\}$



(b) Case $\mathcal{U} = \left[-10^{-1}, 10^{-1}\right]$

Figure 7.5: Inference on the training data for the Mackey-Glass environment with different control magnitudes on $\mathcal{U}$. Regarding the $\mathcal{U} = \{0\}$ case (a), the control signal is displayed but is not injected in the dynamics. For each case, the last row is the control process.

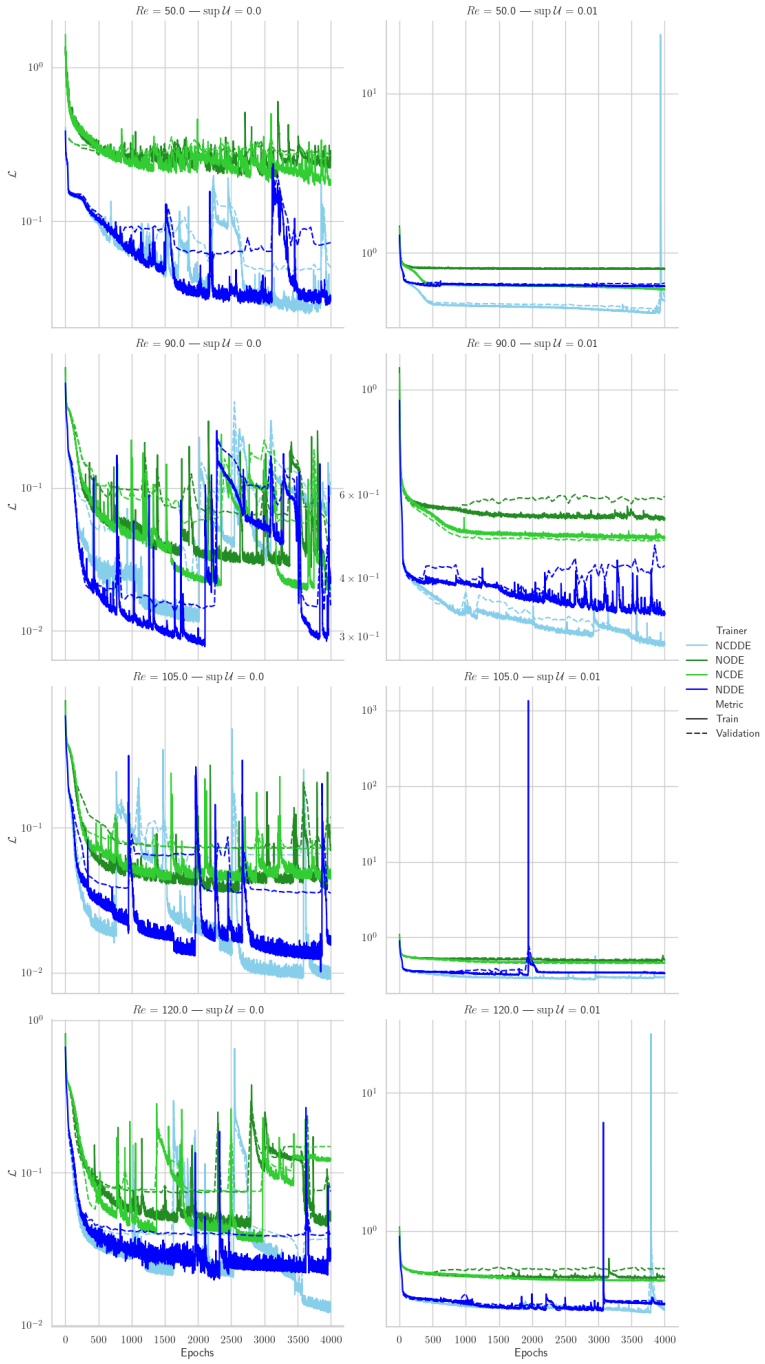


Figure 7.6: Training dynamics of the neural differential models for the Cylinder Flow under several Reynolds numbers Re and two control magnitudes. The x-axis represents the training epochs. The y-axis is the empirical training loss $\widehat{\mathcal{L}}$. Dashed lines represent the empirical validation loss. Rows: $\mathrm{Re} \in \{50, 90, 105, 120\}$ (from top to bottom). Columns: $\mathcal{U} = \{0\}$ and $\mathcal{U} = \left[-10^{-2}, 10^{-2}\right]$ from (left to right). Green curves are delay-based models. Blue curves are the baseline models. Lighter tones are their controlled version.

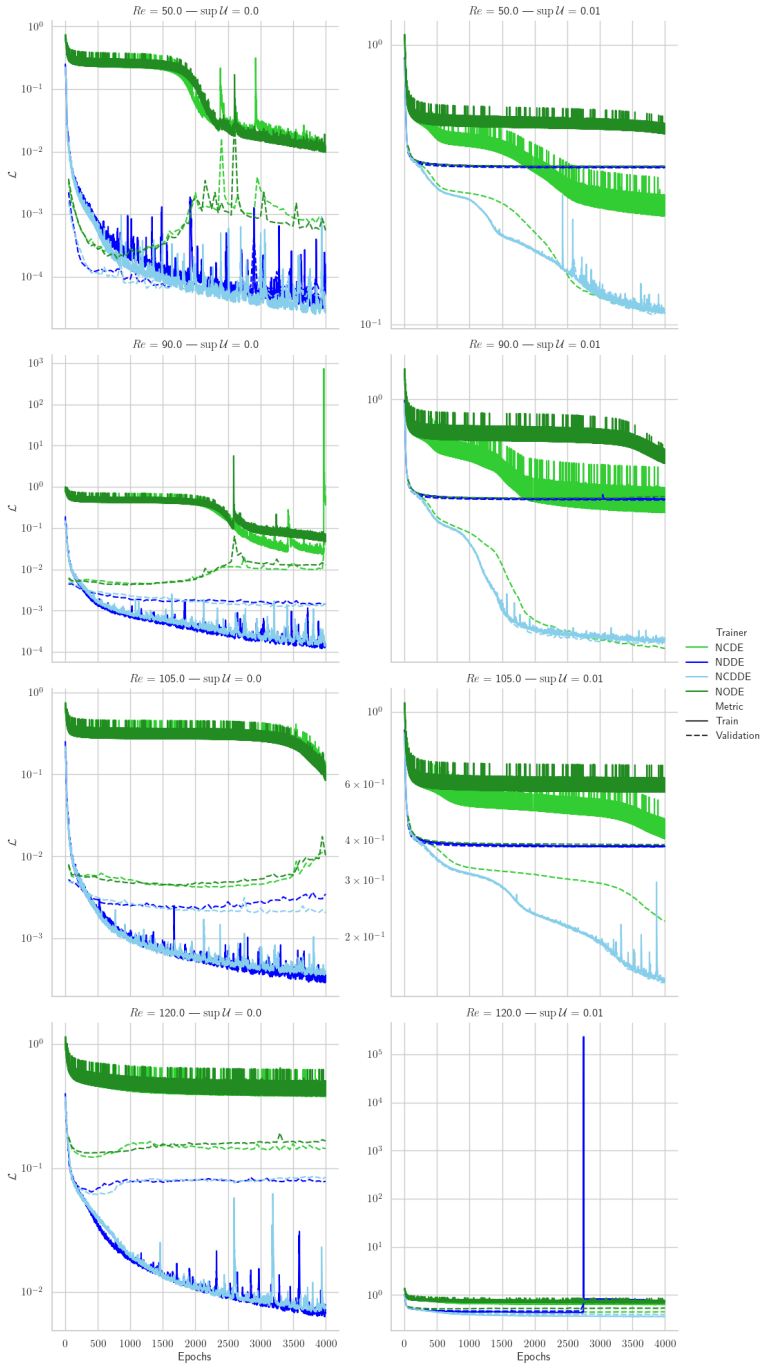


Figure 7.7: Training dynamics of the neural differential models for the Fluidic Pinball under several Reynolds numbers Re and two control magnitudes. The x-axis represents the training epochs. The y-axis is the empirical training loss $\widehat{\mathcal{L}}$. Dashed lines represent the empirical validation loss. Rows: $\mathrm{Re} \in \{50, 90, 105, 120\}$ (from top to bottom). Columns: $\mathcal{U} = \{0\}$ and $\mathcal{U} = \left[-10^{-2}, 10^{-2}\right]$ from (left to right). Green curves are delay-based models. Blue curves are the baseline models. Lighter tones are their controlled version.

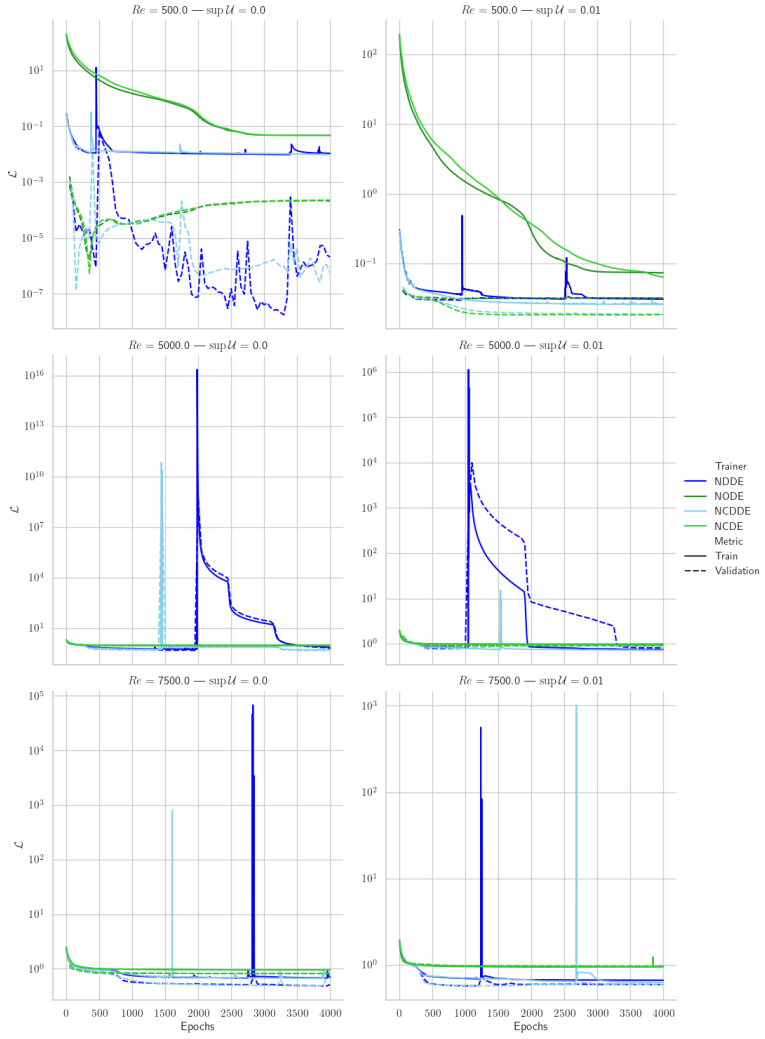


Figure 7.8: Training dynamics of the neural differential models for the Cavity Flow under several Reynolds numbers Re and two control magnitudes. The x-axis represents the training epochs. The y-axis is the empirical training loss $\widehat{\mathcal{L}}$. Dashed lines represent the empirical validation loss. Rows: $\mathrm{Re} \in \{500, 5000, 7500\}$ (from top to bottom). Columns: $\mathcal{U} = \{0\}$ and $\mathcal{U} = \left[-10^{-2}, 10^{-2}\right]$ from (left to right). Green curves are delay-based models. Blue curves are the baseline models. Lighter tones are their controlled version.

8 Conclusion

8.1 Addressing the Open Challenges in Learning based Control for Fluid Flows

The work presented in this thesis addresses a significant part of the open challenges in Learning-based control for fluid flows (see Section 1.4.1).

Notably, it develops the following contributions:

- An extended presentation of the connection between the fields of Stochastic Control and discrete Markov Decision Process (Chapter 2) leading to the discrete Dynamic Programming Principle used in Reinforcement Learning (Chapter 3)
- The introduction of temporal abstraction in the learning process coming from the distinction between decision time and the physical time (Chapter 3, Chapter 5 and Chapter 7)
- The study of the impact of the maximum policy entropy principle on the robustness to noise and the regularisation of the policy distribution (Chapter 4)
- An application of a distributional perspective in Reinforcement Learning to chaotic dynamics which increases the learning speed (Chapter 6)
- The modelling of dynamics with state-of-the-art continuous-time neural differential models (Chapter 7)

8.2 Unification of concurrent fields

Furthermore, this work is a step towards the unification of Stochastic Control, Reinforcement Learning, Information Theory and Flow Control (Chapters 2 and 3). It aims to provide a modern approach to the control of fluid flows by clarifying the potential links between these domains.

The thesis also showcases the different point of views (deterministic vs. stochastic; discrete vs. continuous) available to perform learning-based control of

dynamical systems. Incorporating methods from these different perspectives could lead to more robust and efficient solutions.

8.3 A Multidisciplinary Approach

Moreover, this thesis tries to incorporate the most recent advances in the field of Learning-based Control and Machine Learning to enlarge the set of tools available for the control of fluid flows. It borrows ideas and algorithms from the robotics-oriented Learning-based Control community[69] to address the specific challenges of fluid flows. Additionally, a wide range of domains is covered: Information Theory (Chapters 4 and 5), Statistical Learning (Chapter 4), Delay Differential Equations (Chapter 7), Optimal Transport (Chapter 6), Control and Fluid Dynamics. The thesis aims to provide the necessary framework to combine these domains and to propose a modern approach to the control of fluid flows.

8.4 Further Research Directions

The presented work opens several research directions. A broad description of those directions is given here. More specific ideas are given at the end of the related chapters.

First, the use of the maximum entropy principle in the context of fluid flows is a promising approach to increase the robustness to noise of the control policies. As stated in the bibliography, other benefits could be expected from the use of this principle in the context of fluid flows where the environments are sensitive to small perturbations. Second, the use of Distributional Reinforcement Learning to control chaotic dynamics is a novel approach that could be extended to more complex systems. Plus, the distributional nature of the model could be used to quantify risk and uncertainty in the context of safety-critical systems. Third, the concepts of information-based acquisition functions for active data selection could be extended to fluid flows to increase the efficiency of the learning process. Last, the use of continuous-time neural differential models for the control of fluid flows is a promising approach which could discard the dependence of the algorithm on the discretisation of the dynamics while adding temporal abstraction. Moreover, the delay differential equations methods could be improved to learn interpretable delays.

Finally, the construction of an algorithm able to combine those features would be a significant step towards the control of fluid flows with Learning-based Control. Ideally, the algorithm would combine safe system exploration of a real-world system, together with a careful data selection from a database

[69] For instance, the use of the maximum entropy principle in Deep Reinforcement Learning and the base paper for Chapter 5 are from resarch groups in robotics.

in order to learn a model of the system efficiently and derive a robust control policy. With a large enough database, the algorithm should first be able to identify the right model for the system, then learn a policy that is robust to noise and uncertainty.

Synthèse en français

Les impératifs environnementaux suscitent un regain d'intérêt pour la recherche sur le contrôle de l'écoulement des fluides afin de réduire la consommation d'énergie et les émissions dans diverses applications telles que l'aéronautique et l'automobile. Les stratégies de contrôle des fluides peuvent optimiser le système en temps réel, en tirant parti des mesures des capteurs et des modèles physiques. Ces stratégies visent à manipuler le comportement d'un système pour atteindre un état souhaité (stabilité, performance, consommation d'énergie).

Dans le même temps, le développement d'approches de contrôle pilotées par les données dans des domaines concurrents tels que les jeux et la robotique a ouvert de nouvelles perspectives pour le contrôle des fluides.

Cependant, l'intégration du contrôle basé sur l'apprentissage en dynamique des fluides présente de nombreux défis, notamment en ce qui concerne la robustesse de la stratégie de contrôle, l'efficacité de l'échantillon de l'algorithme d'apprentissage, et la présence de retards de toute nature dans le système.

Ainsi, cette thèse vise à étudier et à développer des stratégies de contrôle basées sur l'apprentissage en tenant compte de ces défis, dans lesquels deux classes principales de stratégies de contrôle basées sur les données sont considérées : l'apprentissage par renforcement (RL) et la commande prédictive basée sur l'apprentissage (LB-MPC). De multiples contributions sont apportées dans ce contexte.

Tout d'abord, un développement étendu sur la connexion entre les domaines du contrôle stochastique (temps continu) et du processus de décision de Markov (temps discret) est fourni pour unifier les deux approches. Le système en temps discret est alors vu comme un système en temps continu échantilloné. Ce point de vue permet de donner un cadre général à l'étude des problèmes de contrôle sur des systèmes dynamiques en temps continu.

Deuxièmement, des preuves empiriques sur les propriétés de régularisation de l'algorithme d'apprentissage par renforcement par maximum d'entropie sont présentées à travers des concepts d'apprentissage statistique pour mieux comprendre la propriété de robustesse de l'approche par maximum d'entropie. Plus précisément, deux mesures de complexité sont proposées pour prédire la robustesse de la politique obtenue en fin d'apprentissage. La première quantifie la régularité du réseau de neurones caractérisant la politique en majorant

la constante de Lipschitz du modèle neuronal. La seconde évalue la régularité locale du paysage d'optimisation autour des paramètres de la politique en fin de procédure d'optimisation, à l'aide d'une statistique basée sur l'information de Fisher de la politique.

Troisièmement, la notion d'abstraction temporelle est utilisée pour améliorer l'efficacité de l'échantillonnage d'un algorithme de commande prédictive par modèle basé sur l'apprentissage et piloté par une règle d'échantillonnage de la théorie de l'information. De manière plus précise, une fonction d'acquisition de donnée basée sur l'information mutuelle est étendue au cas où le temps d'inter-échantillonnage devient aussi une variable de décision. L'introduction de cette variable de décision permet d'augmenter la quantité d'information acquise par la procédure d'échantillonnage, ce qui améliore la performance de l'algorithme de commande prédictive basé sur l'apprentissage.

Enfin, les modèles différentiels neuronaux sont introduits à travers le concept d'équations différentielles neuronales à retard pour modéliser des systèmes à temps continu avec des retards pour des applications en commande prédictive. Les modèles neuronaux à retard montrent de meilleures performances de regression face aux modèles témoins.

Les différentes études sont développées à l'aide de simulations numériques appliquées à des systèmes minimalistes issus des théories des systèmes dynamiques et du contrôle afin d'illustrer les résultats théoriques. Les expériences de la dernière partie sont également menées sur des simulations d'écoulement de fluides en 2D.

Bibliography

Abbeel, Pieter, Morgan Quigley, and Andrew Y. Ng (2006). "Using Inaccurate Models in Reinforcement Learning". In: *Proceedings of the 23rd International Conference on Machine Learning*. ICML '06. Pittsburgh, Pennsylvania, USA: Association for Computing Machinery, pp. 1–8. ISBN: 1595933832. DOI: 10.1145/1143844.1143845. URL: https://doi.org/10.1145/1143844.1143845.

Acemoglu, D. (2008). *Introduction to Modern Economic Growth*. Princeton University Press. ISBN: 9781400835775.

Agarwal, Alekh, Nan Jiang, and Sham Machandranath Kakade (2019). "Reinforcement Learning: Theory and Algorithms". In.

Agarwal, Mridul and Vaneet Aggarwal (May 2021). "Blind Decision Making: Reinforcement Learning with Delayed Observations". In: *Proceedings of the International Conference on Automated Planning and Scheduling* 31.1, pp. 2–6. DOI: 10.1609/icaps.v31i1.15940. URL: https://ojs.aaai.org/index.php/ICAPS/article/view/15940.

Agarwal, Rishabh et al. (2021). "Deep reinforcement learning at the edge of the statistical precipice". In: *Advances in Neural Information Processing Systems* 34.

Ahmed, N. U. and X. Xiang (1992). "Admissible Relaxation in Optimal Control Problems for Infinite Dimensional Uncertain Systems". In: *International Journal of Stochastic Analysis* 5.3, p. 951897. DOI: https://doi.org/10.1155/S1048953392000194. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1155/S1048953392000194. URL: https://onlinelibrary.wiley.com/doi/abs/10.1155/S1048953392000194.

Ahmed, N.U. (Jan. 2007). "A Relaxation Theorem for Partially Observed Stochastic Control on Hilbert Space". In: *Discussiones Mathematicae. Differential Inclusions, Control and Optimization* 27. DOI: 10.7151/dmdico.1086.

Ahmed, Zafarali et al. (June 2019). "Understanding the Impact of Entropy on Policy Optimization". In: *Proceedings of the 36th International Conference on Machine Learning*. Ed. by Kamalika Chaudhuri and Ruslan Salakhutdinov. Vol. 97. Proceedings of Machine Learning Research. PMLR, pp. 151–160. URL: https://proceedings.mlr.press/v97/ahmed19a.html.

Allaire, G. (2005). *Analyse numérique et optimisation: une introduction à la modélisation mathématique et à la simulation numérique*. Ecole polytechnique: Mathématiques appliquées. Editions de l'Ecole polytechnique. ISBN: 9782730212557.

Alnaes, Martin S. et al. (2013). *Unified Form Language: A domain-specific language for weak formulations of partial differential equations*. arXiv: 1211.4047 [cs.MS]. URL: https://arxiv.org/abs/1211.4047.

Alt, Bastian, Matthias Schultheis, and Heinz Koeppl (2020). "POMDPs in Continuous Time and Discrete Spaces". In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle et al. Vol. 33. Curran Associates, Inc., pp. 13151–13162. URL: https://proceedings.neurips.cc/paper_files/paper/2020/file/992f0fed0720dbb9d4e060d03ed531ba-Paper.pdf.

Altman, Eitan and Philippe Nain (1992). "Closed-loop control with delayed information". In: *Proceedings of the 1992 ACM SIGMETRICS Joint International Conference on Measurement and Modeling of Computer Systems*. SIGMETRICS '92/PERFORMANCE '92. Newport, Rhode Island, USA: Association for Computing Machinery, pp. 193–204. ISBN: 0897915070. DOI: 10.1145/133057.133106. URL: https://doi.org/10.1145/133057.133106.

Amari, Shun-ichi (Feb. 1998). "Natural Gradient Works Efficiently in Learning". In: *Neural Computation* 10.2, pp. 251–276. ISSN: 0899-7667. DOI: 10.1162/089976698300017746. eprint: https://direct.mit.edu/neco/article-pdf/10/2/251/813415/089976698300017746.pdf. URL: https://doi.org/10.1162/089976698300017746.

Andrieu, Christophe and Arnaud Doucet (2002). "Particle Filtering for Partially Observed Gaussian State Space Models". In: *Journal of the Royal Statistical Society. Series B (Statistical Methodology)* 64.4, pp. 827–836. ISSN: 13697412, 14679868. (Visited on 12/03/2024).

Ansel, Jason et al. (Apr. 2024). "PyTorch 2: Faster Machine Learning Through Dynamic Python Bytecode Transformation and Graph Compilation". In: *29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2 (ASPLOS '24)*. ACM. DOI: 10.1145/3620665.3640366. URL: https://pytorch.org/assets/pytorch2-2.pdf.

Ashill, P. R., J. L. Fulker, and K. C. Hackett (2005). "A review of recent developments in flow control". In: *The Aeronautical Journal* 109.1095, pp. 205–232. DOI: 10.1017/S0001924000005200.

Åström, K.J (1965). "Optimal control of Markov processes with incomplete state information". In: *Journal of Mathematical Analysis and Applications* 10.1, pp. 174–205. ISSN: 0022-247X. DOI: https://doi.org/10.1016/0022-247X(65)90154-X.

Åström, K.J. and R. Murray (2021). *Feedback Systems: An Introduction for Scientists and Engineers, Second Edition*. Princeton University Press. ISBN: 9780691193984.

Åström, K.J. and B. Wittenmark (1989). *Adaptive Control*. Dover Books on Electrical Engineering. Dover Publications. ISBN: 9780486462783.

Aswani, Anil et al. (2013). "Provably safe and robust learning-based model predictive control". In: *Automatica* 49.5, pp. 1216–1226. ISSN: 0005-1098. DOI: https://doi.org/10.1016/j.automatica.2013.02.003. URL: https://www.sciencedirect.com/science/article/pii/S0005109813000678.

Atay, Fatihcan M. (1998). "Van Der Pol's Oscillator Under Delayed Feedback". In: *Journal of Sound and Vibration* 218 (2), pp. 333–339.

Ayed, Ibrahim et al. (2019). *Learning Dynamical Systems from Partial Observations*. arXiv: 1902.11136 [cs.SY]. URL: https://arxiv.org/abs/1902.11136.

Bach, F. (2024). *Learning Theory from First Principles*. Adaptive Computation and Machine Learning series. MIT Press. ISBN: 9780262381369.

Bander, J. L. and C. C. White (1999). "Markov Decision Processes with Noise-Corrupted and Delayed State Observations". In: *The Journal of the Operational Research Society* 50.6, pp. 660–

668. ISSN: 01605682, 14769360. URL: http://www.jstor.org/stable/3010623 (visited on 12/21/2024).
Banse, Adrien et al. (June 2023). "Data-driven memory-dependent abstractions of dynamical systems". In: *Proceedings of The 5th Annual Learning for Dynamics and Control Conference*. Ed. by Nikolai Matni, Manfred Morari, and George J. Pappas. Vol. 211. Proceedings of Machine Learning Research. PMLR, pp. 891–902. URL: https://proceedings.mlr.press/v211/banse23a.html.
Barański, Krzysztof, Yonatan Gutman, and Adam Śpiewak (July 2020). "A probabilistic Takens theorem". In: *Nonlinearity* 33.9, p. 4940. DOI: 10.1088/1361-6544/ab8fb8. URL: https://dx.doi.org/10.1088/1361-6544/ab8fb8.
Baratta, Igor A. et al. (Dec. 2023). *DOLFINx: The next generation FEniCS problem solving environment*. DOI: 10.5281/zenodo.10447666. URL: https://doi.org/10.5281/zenodo.10447666.
Barbagallo, A., P. J. Schmid, and P. Huerre (2009). "Closed-loop control of an open cavity flow using reduced-order models". In: *Journal of Fluid Mechanics* 641, pp. 1–50. DOI: 10.1017/S0022112009991418.
Bardi, M. and I. Capuzzo-Dolcetta (2008). *Optimal Control and Viscosity Solutions of Hamilton-Jacobi-Bellman Equations*. Modern Birkhäuser Classics. Birkhäuser Boston. ISBN: 978-0-8176-4754-4.
Barra, J.R. (1971). *Notions fondamentales de statistique mathematique.Maitrise de mathematiques*. Dunod.
Bélanger, J., P. Venne, and J. N. Paquin (2010). *The What, Where and Why of Real-Time Simulation*. OPAL-RT Technologies White Paper. URL: https://blobdevweb.opal-rt.com/medias/L00161_0436.pdf.
Bellemare, Marc G., Will Dabney, and Rémi Munos (2017). "A Distributional Perspective on Reinforcement Learning". In: *International Conference on Machine Learning*.
Bellemare, Marc G., Ivo Danihelka, et al. (2017). *The Cramer Distance as a Solution to Biased Wasserstein Gradients*. arXiv: 1705.10743 [cs.LG]. URL: https://arxiv.org/abs/1705.10743.
Bellen, A. and M. Zennaro (2013). *Numerical Methods for Delay Differential Equations*. Numerical Mathematics and Scientific Computation. OUP Oxford. ISBN: 0 19 850654 6.
Bellman, Richard (1957). "A Markovian Decision Process". In: *Journal of Mathematics and Mechanics* 6.5, pp. 679–684. ISSN: 00959057, 19435274. (Visited on 08/20/2024).
Bellman, Richard E. (1957). *Dynamic Programming*. Princeton: Princeton University Press. ISBN: 9781400835386. DOI: doi:10.1515/9781400835386. URL: https://doi.org/10.1515/9781400835386.
Benoist, Yves and Frédéric Paulin (2000). *Systèmes dynamiques élémentaires*. French.
Bensoussan, Alain (1993). *Representation and Control of Infinite Dimensional Systems*. Systems & Control, Foundations & Applications S v. 1. Birkhauser, Switzerland.
Bensoussan, Alain, Yiqun Li, et al. (2020). *Machine Learning and Control Theory*. arXiv: 2006.05604 [cs.LG]. URL: https://arxiv.org/abs/2006.05604.
Bensoussan, Alain and Roger Temam (1973). "Equations stochastiques du type Navier-Stokes". In: *Journal of Functional Analysis* 13.2, pp. 195–222.

Bensoussan, Alain and Michel Viot (1975). “Optimal Control of Stochastic Linear Distributed Parameter Systems”. In: *SIAM Journal on Control* 13.4, pp. 904–926. DOI: 10.1137/0313056. eprint: https://doi.org/10.1137/0313056. URL: https://doi.org/10.1137/0313056.
Berger, J.O. (1985). *Statistical Decision Theory and Bayesian Analysis*. Springer Series in Statistics. Springer. ISBN: 9780387960982.
Bertsekas, Dimitri Panteli (2000). *Dynamic Programming and Optimal Control*. Athena Scientific. ISBN: 1886529094.
Bertsekas, Dimitri Panteli and John Nikolaos Tsitsiklis (Jan. 1996). *Neuro-Dynamic Programming*. Vol. 27. ISBN: 978-0-387-74758-3. DOI: 10.1007/978-0-387-74759-0_440.
Blanchard, O.J. and D.R. Johnson (1991). *Macroeconomics*. Pearson. ISBN: 9780133780581.
Bonzanini, Angelo Domenico and Ali Mesbah (June 2020). “Learning-based Stochastic Model Predictive Control with State-Dependent Uncertainty”. In: *Proceedings of the 2nd Conference on Learning for Dynamics and Control*. Ed. by Alexandre M. Bayen et al. Vol. 120. Proceedings of Machine Learning Research. PMLR, pp. 571–580. URL: https://proceedings.mlr.press/v120/bonzanini20a.html.
Bourguignon, J.P. (2007). *Calcul variationnel*. Mathématiques (École Polytechnique (Paris))). Éditions de l’École Polytechnique. ISBN: 9782730214155.
Boutilier, Craig and Richard Dearden (1994). “Using Abstractions for Decision-Theoretic Planning with Time Constraints”. In: *AAAI Conference on Artificial Intelligence*. URL: https://api.semanticscholar.org/CorpusID:6124617.
Bradbury, James et al. (2018). *JAX: composable transformations of Python+NumPy programs*. Version 0.3.13. URL: http://github.com/jax-ml/jax.
Bradtke, Steven and Michael Duff (1994). “Reinforcement Learning Methods for Continuous-Time Markov Decision Problems”. In: *Advances in Neural Information Processing Systems*. Ed. by G. Tesauro, D. Touretzky, and T. Leen. Vol. 7. MIT Press. URL: https://proceedings.neurips.cc/paper_files/paper/1994/file/07871915a8107172b3b5dc15a6574ad3-Paper.pdf.
Brémaud, P. (2001). *Mathematical Principles of Signal Processing: Fourier and Wavelet Analysis*. Springer New York. ISBN: 978-1-4419-2956-3.
Brockman, Greg et al. (2016). *OpenAI Gym*. eprint: arXiv:1606.01540.
Bucci, M. A. et al. (2019). “Control of Chaotic Systems by Deep Reinforcement Learning”. In: *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* 475.2231, p. 20190351. DOI: 10.1098/rspa.2019.0351. eprint: https://royalsocietypublishing.org/doi/pdf/10.1098/rspa.2019.0351. URL: https://royalsocietypublishing.org/doi/abs/10.1098/rspa.2019.0351.
Buckwar, Evelyn (Dec. 2000). “Introduction to the Numerical Analysis of Stochastic Delay Differential Equations”. In: *Journal of Computational and Applied Mathematics* 125, pp. 297–307. DOI: 10.1016/S0377-0427(00)00475-1.
Candel, S. (1995). *Mécanique des fluides: cours*. Dunod université. Dunod. ISBN: 9782100025855.
Cartan, H. (1971). *Calcul différentiel: Calcul differentiel dans les espaces de Banach. Equations differentielles*. Cours de mathématiques. Hermann.

Cassandra, Anthony R. (1998). “Exact and Approximate Algorithms for Partially Observable Markov Decision Processes”. PhD thesis. Brown University.
Chassaing, P. (2000). *Mécanique des fluides: éléments d’un premier parcours*. Collection Polytech. Cépaduès. ISBN: 9782854285093.
Chaudhari, Pratik et al. (Dec. 2019). “Entropy-SGD: biasing gradient descent into wide valleys”. In: *Journal of Statistical Mechanics: Theory and Experiment* 2019.12, p. 124018. DOI: 10.1088/1742-5468/ab39d9. URL: https://dx.doi.org/10.1088/1742-5468/ab39d9.
Chen, Baiming et al. (Apr. 2021). “Delay-Aware Model-Based Reinforcement Learning for Continuous Control”. In: *Neurocomputing* 450. DOI: 10.1016/j.neucom.2021.04.015.
Chen, Ricky T. Q. et al. (2018). “Neural Ordinary Differential Equations”. In: *Advances in Neural Information Processing Systems*. Ed. by S. Bengio et al. Vol. 31. Curran Associates, Inc. URL: https://proceedings.neurips.cc/paper_files/paper/2018/file/69386f6bb1dfed68692a24c8686939b9-Paper.pdf.
Chen, Xiaoyu et al. (July 2022). “Flow-based Recurrent Belief State Learning for POMDPs”. In: *Proceedings of the 39th International Conference on Machine Learning*. Ed. by Kamalika Chaudhuri et al. Vol. 162. Proceedings of Machine Learning Research. PMLR, pp. 3444–3468. URL: https://proceedings.mlr.press/v162/chen22q.html.
Choi, HeeSun et al. (2021). “On the use of simulation in robotics: Opportunities, challenges, and suggestions for moving forward”. In: *Proceedings of the National Academy of Sciences* 118.1, e1907856118. DOI: 10.1073/pnas.1907856118. eprint: https://www.pnas.org/doi/pdf/10.1073/pnas.1907856118. URL: https://www.pnas.org/doi/abs/10.1073/pnas.1907856118.
Chorin, A.J. and J.E. Marsden (2013). *A Mathematical Introduction to Fluid Mechanics*. Texts in Applied Mathematics. Springer New York. ISBN: 9781461208839.
Chua, Kurtland et al. (2018). “Deep Reinforcement Learning in a Handful of Trials using Probabilistic Dynamics Models”. In: *Advances in Neural Information Processing Systems*. Ed. by S. Bengio et al. Vol. 31. Curran Associates, Inc. URL: https://proceedings.neurips.cc/paper_files/paper/2018/file/3de568f8597b94bda53149c7d7f5958c-Paper.pdf.
Colas, Cédric, Olivier Sigaud, and Pierre-Yves Oudeyer (2018). *How Many Random Seeds? Statistical Power Analysis in Deep Reinforcement Learning Experiments*. arXiv: 1806.08295 [cs.LG].
Copp, David A. and João P. Hespanha (2017). “Simultaneous nonlinear model predictive control and state estimation”. In: *Automatica* 77, pp. 143–154. ISSN: 0005-1098. DOI: https://doi.org/10.1016/j.automatica.2016.11.041. URL: https://www.sciencedirect.com/science/article/pii/S0005109816304848.
Cornejo Maceda, Guy Y. et al. (2021). “Stabilization of the fluidic pinball with gradient-enriched machine learning control”. In: *Journal of Fluid Mechanics* 917, A42. DOI: 10.1017/jfm.2021.301.
Coudène, Yves (2013). *Théorie ergodique et systèmes dynamiques*. EDP Sciences. ISBN: 978-2-7598-0760-4.
Cover, Thomas M. and Joy A. Thomas (July 2006). *Elements of Information Theory 2nd Edition*. Wiley-Interscience. ISBN: 0471241954.

Cox, S.M. and P.C. Matthews (2002). "Exponential Time Differencing for Stiff Systems". In: *Journal of Computational Physics* 176.2, pp. 430–455. ISSN: 0021-9991. DOI: https://doi.org/10.1006/jcph.2002.6995. URL: https://www.sciencedirect.com/science/article/pii/S0021999102969950.

Croissant, Lorenzo (2023). "Diffusive Limit Control and Reinforcement Learning". 2023UPSLD027. PhD thesis.

Cvitanović, Predrag, Ruslan L. Davidchack, and Evangelos Siminos (2010). "On the State Space Geometry of the Kuramoto–Sivashinsky Flow in a Periodic Domain". In: *SIAM Journal on Applied Dynamical Systems* 9.1, pp. 1–33. DOI: 10.1137/070705623. eprint: https://doi.org/10.1137/070705623. URL: https://doi.org/10.1137/070705623.

Da Prato, Giuseppe and Arnaud Debussche (2000). "Dynamic programming for the Stochastic Navier-Stokes Equations". In: *ESAIM: Mathematical Modelling and Numerical Analysis* 34.2, pp. 459–475.

Da Prato, Guiseppe and Jerzy Zabczyk (1992). *Stochastic Equations in Infinite Dimensions*. Encyclopedia of Mathematics and its Applications. Cambridge University Press, pp. i–vi.

Dabney, Will et al. (2018). "Distributional reinforcement learning with quantile regression". In: *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence*. AAAI'18/IAAI'18/EAAI'18. New Orleans, Louisiana, USA: AAAI Press. ISBN: 978-1-57735-800-8.

Dearden, Richard, Nir Friedman, and Stuart J. Russell (1998). "Bayesian Q-Learning". In: *AAAI/IAAI*.

Degrave, Jonas et al. (Feb. 2022). "Magnetic control of tokamak plasmas through deep reinforcement learning". In: *Nature* 602.7897, pp. 414–419. ISSN: 1476-4687. DOI: 10.1038/s41586-021-04301-9. URL: https://doi.org/10.1038/s41586-021-04301-9.

Deisenroth, Marc and Carl Rasmussen (Jan. 2011). "PILCO: A Model-Based and Data-Efficient Approach to Policy Search." In: pp. 465–472.

Deisenroth, Marc Peter and Jan Peters (2012). "Solving Nonlinear Continuous State-Action-Observation POMDPs for Mechanical Systems with Gaussian Noise". In: *European Workshop on Reinforcement Learning*.

Deng, Nan et al. (July 2018). "Route to Chaos in the Fluidic Pinball". In: *Volume 1: Flow Manipulation and Active Control; Bio-Inspired Fluid Mechanics; Boundary Layer and High-Speed Flows; Fluids Engineering Education; Transport Phenomena in Energy Conversion and Mixing; Turbulent Flows; Vortex Dynamics; DNS/LES and Hybrid RANS/LES Methods; Fluid Structure Interaction; Fluid Dynamics of Wind Energy; Bubble, Droplet, and Aerosol Dynamics*. FEDSM2018. American Society of Mechanical Engineers. DOI: 10.1115/fedsm2018-83359. URL: http://dx.doi.org/10.1115/fedsm2018-83359.

Derman, Esther, Matthieu Geist, and Shie Mannor (2021). "Twice regularized MDPs and the equivalence between robustness and regularization". In: *Advances in Neural Information Processing Systems*. Ed. by M. Ranzato et al. Vol. 34. Curran Associates, Inc., pp. 22274–22287. URL: https://proceedings.neurips.cc/paper_files/paper/2021/file/bb1443cc31d7396bf73e7858cea114e1-Paper.pdf.

Dinh, Laurent et al. (Aug. 2017). "Sharp Minima Can Generalize For Deep Nets". In: *Proceedings of the 34th International Conference on Machine Learning*. Ed. by Doina Precup and Yee Whye Teh. Vol. 70. Proceedings of Machine Learning Research. PMLR, pp. 1019–1028. URL: https://proceedings.mlr.press/v70/dinh17b.html.

Doya, Kenji (Feb. 2000). "Reinforcement Learning in Continuous Time and Space". In: *Neural computation* 12, pp. 219–45. DOI: 10.1162/089976600300015961.

Doyle, J. (1996). "Robust and optimal control". In: *Proceedings of 35th IEEE Conference on Decision and Control*. Vol. 2, 1595–1598 vol.2. DOI: 10.1109/CDC.1996.572756.

Du, Jianzhun, Joseph Futoma, and Finale Doshi-Velez (2020). "Model-based Reinforcement Learning for Semi-Markov Decision Processes with Neural ODEs". In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle et al. Vol. 33. Curran Associates, Inc., pp. 19805–19816. URL: https://proceedings.neurips.cc/paper_files/paper/2020/file/e562cd9c0768d5464b64cf61da7fc6bb-Paper.pdf.

Duan, Jingliang et al. (2022). "Distributional Soft Actor-Critic: Off-Policy Reinforcement Learning for Addressing Value Estimation Errors". In: *IEEE Transactions on Neural Networks and Learning Systems* 33.11, pp. 6584–6598. DOI: 10.1109/TNNLS.2021.3082568.

Duan, Yaqi, Chi Jin, and Zhiyuan Li (July 2021). "Risk Bounds and Rademacher Complexity in Batch Reinforcement Learning". In: *Proceedings of the 38th International Conference on Machine Learning*. Ed. by Marina Meila and Tong Zhang. Vol. 139. Proceedings of Machine Learning Research. PMLR, pp. 2892–2902. URL: https://proceedings.mlr.press/v139/duan21a.html.

Duflo, M. (1997). *Random Iterative Models*. Applications of Mathematics : Stochastic Modelling and Applied Probability. Springer. ISBN: 9783540571001.

Dulac-Arnold, Gabriel, Daniel Mankowitz, and Todd Hester (May 2019). "Challenges of Real-World Reinforcement Learning". In: *Proceedings of the ICML 2019 Workshop on RL4RealLife*. Last modified: October 14, 2024. URL: https://openreview.net/forum?id=rl4reallife.

Duriez, Thomas, Steven Brunton, and Bernd Noack (Nov. 2016). *Machine Learning Control – Taming Nonlinear Dynamics and Turbulence*. Vol. 116. ISBN: 978-3-319-40623-7. DOI: 10.1007/978-3-319-40624-4.

Dynkin, E.B. and A.A. Yushkevich (1979). *Controlled Markov Processes*. Grundlehren der mathematischen Wissenschaften in Einzeldarstellungen mit besonderer Berücksichtigung der Anwendungsgebiete. Springer New York. ISBN: 9783540903871.

E, Weinan (2000). "Stochastic PDEs in Turbulence Theory". In: *First International Congress of Chinese Mathematicians*. Princeton University Press, pp. 27–46. URL: https://www.worldcat.org/oclc/61285753.

El Karoui, Nicole (1987). "Partially observable control of diffusions with correlated noise". In: *Stochastic Differential Systems*. Ed. by Hans Jürgen Engelbert and Wolfgang Schmidt. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 272–285. ISBN: 978-3-540-47245-2.

El Karoui, Nicole, Nguyen Du Huu, and Monique Jeanblanc-Picqué (Mar. 1987). "Compactification methods in the control of degenerate diffusions: Existence of an optimal control". In: *Stochastics: an international journal of probability and stochastic processes* 20, pp. 169–219. DOI: 10.1080/17442508708833443.

El Karoui, Nicole, Du Huu Nguyen, and Monique Jeanblanc-Picqué (1988). "Existence of an Optimal Markovian Filter for the Control under Partial Observations". In: *SIAM Journal on Control and Optimization* 26.5, pp. 1025–1061. DOI: 10.1137/0326057. eprint: https://doi.org/10.1137/0326057. URL: https://doi.org/10.1137/0326057.

Elsanosi, Ismail, Bernt Øksendal, and Agnès Sulem (Dec. 2000). "Some Solvable Stochastic Control Problems With Delay". In: *Stochastics: An International Journal of Probability and Stochastic Processes* 71, pp. 69–89. DOI: 10.1080/17442500008834259.

Elsgolts, Lev Ėrnestovich (1964). *Qualitative Methods in Mathematical Analysis*. Translations of mathematical monographs. American Mathematical Society.

Evans, L.C. (1998). *Partial Differential Equations*. Graduate studies in mathematics. American Mathematical Society. ISBN: 9780821849743.

Eysenbach, Benjamin and Sergey Levine (2022). "Maximum Entropy RL (Provably) Solves Some Robust RL Problems". In: *International Conference on Learning Representations*. URL: https://openreview.net/forum?id=PtSAD3caaA2.

Fabbri, Giorgio, Fausto Gozzi, and Andrzej Swiech (2017). "Stochastic Optimal Control in Infinite Dimension". In: *Probability and Stochastic Modelling. Springer*.

Fejoz, J. (2017). *Calcul Différentiel et Optimisation*. Cet ouvrage est sous licence Creative Commons Attribution 4.0 International. Pour accéder à une copie de cette licence, rendez-vous à l'adresse http://creativecommons.org/licenses/by/4.0/. Paris, France: Université Paris-Dauphine. URL: http://creativecommons.org/licenses/by/4.0/.

Feldstein, Morley Alan (1964). "Discretization methods for retarded ordinary differential equations". Ph.D. dissertation. University of California, Los Angeles.

Findeisen, Rolf et al. (2003). "State and Output Feedback Nonlinear Model Predictive Control: An Overview". In: *European Journal of Control* 9.2, pp. 190–206. ISSN: 0947-3580. DOI: https://doi.org/10.3166/ejc.9.190-206. URL: https://www.sciencedirect.com/science/article/pii/S0947358003702751.

Fleming, Peter and Robin Purshouse (May 2002). "Genetic Algorithms In Control Systems Engineering". In.

Fleming, W. H. and M. Nisio (1984). "On Stochastic Relaxed Control for Partially Observed Diffusions". In: *Nagoya Mathematical Journal* 93, pp. 71–108. DOI: 10.1017/S0027763000020742.

Fleming, W.H. and R.W. Rishel (1975). *Deterministic and Stochastic Optimal Control*. Applications of mathematics. Springer.

Forti, Davide and Luca Dedè (2015). "Semi-implicit BDF time discretization of the Navier–Stokes equations with VMS-LES modeling in a High Performance Computing framework". In: *Computers & Fluids* 117, pp. 168–182. ISSN: 0045-7930. DOI: https://doi.org/10.1016/j.compfluid.2015.05.011. URL: https://www.sciencedirect.com/science/article/pii/S0045793015001668.

Fujimoto, Scott, Herke van Hoof, and David Meger (July 2018). "Addressing Function Approximation Error in Actor-Critic Methods". In: *Proceedings of the 35th International Conference on Machine Learning*. Ed. by Jennifer Dy and Andreas Krause. Vol. 80. Proceedings of Machine Learning Research. PMLR, pp. 1587–1596. URL: https://proceedings.mlr.press/v80/fujimoto18a.html.

Gatarek, D. and B. Goldys (1994). "On Uniqueness in Law of Solutions to Stochastic Evolution Equations in Hilbert Spaces". In: *Stochastic Analysis and Applications* 12.2, pp. 193–203. DOI: 10.1080/07362999408809346. eprint: https://doi.org/10.1080/07362999408809346. URL: https://doi.org/10.1080/07362999408809346.

Gawarecki, Leszek and Vidyadhar Mandrekar (July 2015). *Stochastic Differential Equations in Infinite Dimensions with Applications to Stochastic Partial Differential Equations*. ISBN: 978-3-642-16193-3. DOI: 10.1007/978-3-642-16194-0.

Gihman, I.I. and A.V. Skorohod (1979). *Controlled Stochastic Processes*. Springer.

Glass, L. and M. Mackey (2010). "Mackey-Glass equation". In: *Scholarpedia* 5.3. revision #186443, p. 6908. DOI: 10.4249/scholarpedia.6908.

Gogianu, Florin et al. (July 2021). "Spectral Normalisation for Deep Reinforcement Learning: An Optimisation Perspective". In: *Proceedings of the 38th International Conference on Machine Learning*. Ed. by Marina Meila and Tong Zhang. Vol. 139. Proceedings of Machine Learning Research. PMLR, pp. 3734–3744. URL: https://proceedings.mlr.press/v139/gogianu21a.html.

Golowich, Noah, Alexander Rakhlin, and Ohad Shamir (July 2018). "Size-Independent Sample Complexity of Neural Networks". In: *Proceedings of the 31st Conference On Learning Theory*. Ed. by Sébastien Bubeck, Vianney Perchet, and Philippe Rigollet. Vol. 75. Proceedings of Machine Learning Research. PMLR, pp. 297–299. URL: https://proceedings.mlr.press/v75/golowich18a.html.

Grabski, Franciszek (Sept. 2016). "Concept of Semi -Markov Process". In: *Zeszyty Naukowe Akademii Marynarki Wojennej* 206, pp. 25–36. DOI: 10.5604/0860889X.1224743.

Grüne, Lars and Jürgen Pannek (Jan. 2011). *Nonlinear Model Predictive Control: Theory and Algorithms*. ISBN: 978-0-85729-500-2. DOI: 10.1007/978-0-85729-501-9.

Ha, David and Jürgen Schmidhuber (2018). "Recurrent World Models Facilitate Policy Evolution". In: *Advances in Neural Information Processing Systems*. Ed. by S. Bengio et al. Vol. 31. Curran Associates, Inc. URL: https://proceedings.neurips.cc/paper_files/paper/2018/file/2de5d16682c3c35007e4e92982f1a2ba-Paper.pdf.

Haarnoja, Tuomas, Haoran Tang, et al. (Aug. 2017). "Reinforcement Learning with Deep Energy-Based Policies". In: *Proceedings of the 34th International Conference on Machine Learning*. Ed. by Doina Precup and Yee Whye Teh. Vol. 70. Proceedings of Machine Learning Research. PMLR, pp. 1352–1361. URL: https://proceedings.mlr.press/v70/haarnoja17a.html.

Haarnoja, Tuomas, Aurick Zhou, Pieter Abbeel, et al. (July 2018). "Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor". In: *Proceedings of the 35th International Conference on Machine Learning*. Ed. by Jennifer Dy and Andreas Krause. Vol. 80. Proceedings of Machine Learning Research. PMLR, pp. 1861–1870. URL: https://proceedings.mlr.press/v80/haarnoja18b.html.

Haarnoja, Tuomas, Aurick Zhou, Kristian Hartikainen, et al. (2019). *Soft Actor-Critic Algorithms and Applications*. arXiv: 1812.05905 [cs.LG].

Hairer, E., S.P. Nørsett, and G. Wanner (2008). *Solving Ordinary Differential Equations I: Nonstiff Problems*. Springer Series in Computational Mathematics. Springer Berlin Heidelberg. ISBN: 9783540566700.

Hale, J.K. (1971). *Functional Differential Equations*. Applied mathematical sciences. Springer-Verlag. ISBN: 9780387900230.
Ham, David A. et al. (May 2023). *Firedrake User Manual*. First edition. Imperial College London et al. DOI: 10.25561/104839.
Harlamov, B. P. (2004). "Continuous Semi-Markov Processes and Their Applications". In: *Communications in Statistics - Theory and Methods* 33.3, pp. 569–589. DOI: 10.1081/STA-120028685. eprint: https://doi.org/10.1081/STA-120028685. URL: https://doi.org/10.1081/STA-120028685.
Hasselt, Hado van (2010). "Double Q-learning". In: *Proceedings of the 23rd International Conference on Neural Information Processing Systems - Volume 2*. NIPS'10. Vancouver, British Columbia, Canada: Curran Associates Inc., pp. 2613–2621.
Hasselt, Hado van et al. (2018). *Deep Reinforcement Learning and the Deadly Triad*. arXiv: 1812.02648 [cs.AI].
Hastie, T., R. Tibshirani, and J. Friedman (2013). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer Series in Statistics. Springer New York. ISBN: 9780387216065.
Haussler, David and Manfred Opper (1997). "Mutual Information, Metric Entropy and Cumulative Relative Entropy Risk". In: *The Annals of Statistics* 25.6, pp. 2451–2492. ISSN: 00905364. URL: http://www.jstor.org/stable/2959041 (visited on 07/18/2023).
He, Kaiming et al. (2015). *Deep Residual Learning for Image Recognition*. arXiv: 1512.03385 [cs.CV]. URL: https://arxiv.org/abs/1512.03385.
Henderson, Peter et al. (2018). "Deep Reinforcement Learning That Matters". In: *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence*. AAAI'18/IAAI'18/EAAI'18. New Orleans, Louisiana, USA: AAAI Press. ISBN: 978-1-57735-800-8.
Hernández-Lerma, O. (1989). *Adaptive Markov Control Processes*. Applied mathematical sciences. Springer-Verlag. ISBN: 9780387969664.
Hernández-Lerma, Onésimo and Jean B. Lasserre (1996). *Discrete-Time Markov Control Processes: Basic Optimality Criteria*. 1st ed. Springer New York. DOI: 10.1007/978-1-4612-0729-0.
Hewing, Lukas et al. (2020). "Learning-Based Model Predictive Control: Toward Safe Learning in Control". In: *Annual Review of Control, Robotics, and Autonomous Systems* 3.1, pp. 269–296. DOI: 10.1146/annurev-control-090419-075625. eprint: https://doi.org/10.1146/annurev-control-090419-075625. URL: https://doi.org/10.1146/annurev-control-090419-075625.
Hinton, Geoffrey E., Simon Osindero, and Yee Whye Teh (2006). "A Fast Learning Algorithm for Deep Belief Nets". In: *Neural Computation* 18, pp. 1527–1554.
Hiriart-Urruty, J.B. and C. Lemarechal (2013). *Convex Analysis and Minimization Algorithms I: Fundamentals*. Grundlehren der mathematischen Wissenschaften. Springer Berlin Heidelberg. ISBN: 9783662027967.

Hochreiter, Sepp and Jürgen Schmidhuber (Jan. 1997). "Flat Minima". In: *Neural Computation* 9.1, pp. 1–42. ISSN: 0899-7667. DOI: 10.1162/neco.1997.9.1.1. eprint: https://direct.mit.edu/neco/article-pdf/9/1/1/813385/neco.1997.9.1.1.pdf. URL: https://doi.org/10.1162/neco.1997.9.1.1.

Höfer, Sebastian et al. (2021). "Sim2Real in Robotics and Automation: Applications and Challenges". In: *IEEE Transactions on Automation Science and Engineering* 18.2, pp. 398–400. DOI: 10.1109/TASE.2021.3064065.

Hoffman, M. (Jan. 2015). *Introduction aux méthodes statistiques*. Course notes, pages 10–19.

Holmes, Philip et al. (2012). *Turbulence, Coherent Structures, Dynamical Systems and Symmetry*. 2nd ed. Cambridge Monographs on Mechanics. Cambridge University Press.

Holt, Samuel et al. (Apr. 2023). "Neural Laplace Control for Continuous-time Delayed Systems". In: *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*. Ed. by Francisco Ruiz, Jennifer Dy, and Jan-Willem van de Meent. Vol. 206. Proceedings of Machine Learning Research. PMLR, pp. 1747–1778. URL: https://proceedings.mlr.press/v206/holt23a.html.

Holt, Samuel I, Zhaozhi Qian, and Mihaela van der Schaar (2022). "Neural Laplace: Learning diverse classes of differential equations in the Laplace domain". In: *International Conference on Machine Learning*. PMLR, pp. 8811–8832.

Hosseinkhan Boucher, Rémy, Stella Douka, et al. (July 2024). "Increasing information for model predictive control with semi-Markov decision processes". In: *Proceedings of the 6th Annual Learning for Dynamics & Control Conference*. Ed. by Alessandro Abate et al. Vol. 242. Proceedings of Machine Learning Research. PMLR, pp. 1400–1414. URL: https://proceedings.mlr.press/v242/hosseinkhan-boucher24a.html.

Hosseinkhan Boucher, Rémy, Onofrio Semeraro, and Lionel Mathelin (2025). "Evidence on the Regularisation Properties of Maximum-Entropy Reinforcement Learning". In: *Optimization and Learning*. Ed. by Bernabé Dorronsoro, Martin Zagar, and El-Ghazali Talbi. Cham: Springer Nature Switzerland, pp. 123–139. ISBN: 978-3-031-77941-1.

Howard, R.A. (1960). *Dynamic Programming and Markov Processes*. Technology Press of Massachusetts Institute of Technology.

IPCC Core Writing Team, H. Lee, and J. Romero, eds. (2023). *Climate Change 2023: Synthesis Report. Contribution of Working Groups I, II and III to the Sixth Assessment Report of the Intergovernmental Panel on Climate Change*. Geneva, Switzerland: IPCC, pp. 35–115. DOI: 10.59327/IPCC/AR6-9789291691647.

Jastrzebski, Stanislaw et al. (July 2021). "Catastrophic Fisher Explosion: Early Phase Fisher Matrix Impacts Generalization". In: *Proceedings of the 38th International Conference on Machine Learning*. Ed. by Marina Meila and Tong Zhang. Vol. 139. Proceedings of Machine Learning Research. PMLR, pp. 4772–4784. URL: https://proceedings.mlr.press/v139/jastrzebski21a.html.

Jumper, John et al. (Aug. 2021). "Highly accurate protein structure prediction with AlphaFold". In: *Nature* 596.7873, pp. 583–589. ISSN: 1476-4687. DOI: 10.1038/s41586-021-03819-2. URL: https://doi.org/10.1038/s41586-021-03819-2.

Kakade, Sham Machandranath (2001). “A Natural Policy Gradient”. In: *Advances in Neural Information Processing Systems*. Ed. by T. Dietterich, S. Becker, and Z. Ghahramani. Vol. 14. MIT Press. URL: https://proceedings.neurips.cc/paper_files/paper/2001/file/4b86abe48d358ecf194c56c69108433e-Paper.pdf.

— (2003). “On the Sample Complexity of Reinforcement Learning”. PhD thesis.

Kakade, Sham Machandranath and John Langford (2002). “Approximately Optimal Approximate Reinforcement Learning”. In: *Proceedings of the Nineteenth International Conference on Machine Learning*. ICML ’02. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., pp. 267–274. ISBN: 1558608737.

Kalman, R. E. and R. S. Bucy (Mar. 1961). “New Results in Linear Filtering and Prediction Theory”. In: *Journal of Basic Engineering* 83.1, pp. 95–108. ISSN: 0021-9223. DOI: 10.1115/1.3658902. eprint: https://asmedigitalcollection.asme.org/fluidsengineering/article-pdf/83/1/95/5503549/95_1.pdf. URL: https://doi.org/10.1115/1.3658902.

Kálmán, Róbert (1958). “Design of a Self-Optimizing Control System”. In: *Journal of Fluids Engineering*.

Kamthe, Sanket and Marc Deisenroth (Apr. 2018). “Data-Efficient Reinforcement Learning with Probabilistic Model Predictive Control”. In: *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*. Ed. by Amos Storkey and Fernando Perez-Cruz. Vol. 84. Proceedings of Machine Learning Research. PMLR, pp. 1701–1710. URL: https://proceedings.mlr.press/v84/kamthe18a.html.

Karakida, Ryo, Shotaro Akaho, and Shun-ichi Amari (Apr. 2019). “Universal Statistics of Fisher Information in Deep Neural Networks: Mean Field Approach”. In: *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*. Ed. by Kamalika Chaudhuri and Masashi Sugiyama. Vol. 89. Proceedings of Machine Learning Research. PMLR, pp. 1032–1041. URL: https://proceedings.mlr.press/v89/karakida19a.html.

Karniadakis, George Em et al. (June 2021). “Physics-informed machine learning”. In: *Nature Reviews Physics* 3.6, pp. 422–440. ISSN: 2522-5820. DOI: 10.1038/s42254-021-00314-5. URL: https://doi.org/10.1038/s42254-021-00314-5.

Katsikopoulos, K.V. and S.E. Engelbrecht (2003). “Markov decision processes with delays and asynchronous cost collection”. In: *IEEE Transactions on Automatic Control* 48.4, pp. 568–574. DOI: 10.1109/TAC.2003.809799.

Kearns, Michael et al. (1994). “On the learnability of discrete distributions”. In: *Proceedings of the Twenty-Sixth Annual ACM Symposium on Theory of Computing*. STOC ’94. Montreal, Quebec, Canada: Association for Computing Machinery, pp. 273–282. ISBN: 0897916638. DOI: 10.1145/195058.195155. URL: https://doi.org/10.1145/195058.195155.

Kelly, Cónall and Kate O’Donovan (2024). *Adaptive Mesh Construction for the Numerical Solution of Stochastic Differential Equations with Markovian Switching*. arXiv: 2408.14931 [math.NA]. URL: https://arxiv.org/abs/2408.14931.

Keskar, Nitish Shirish et al. (2017). “On large-batch training for deep learning: Generalization gap and sharp minima”. English (US). In: 5th International Conference on Learning Representations, ICLR 2017 ; Conference date: 24-04-2017 Through 26-04-2017.

Khalil, H.K. (2002). *Nonlinear Systems*. Pearson Education. Prentice Hall. ISBN: 9780130673893.

Kidger, Patrick (2021). "On Neural Differential Equations". PhD thesis. University of Oxford.
Kidger, Patrick et al. (2020). "Neural Controlled Differential Equations for Irregular Time Series". In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle et al. Vol. 33. Curran Associates, Inc., pp. 6696–6707. URL: https://proceedings.neurips.cc/paper_files/paper/2020/file/4a5876b450b45371f6cfe5047ac8cd45-Paper.pdf.
Kim, Soung Hie and Byung Ho Jeong (1987). "A Partially Observable Markov Decision Process with Lagged Information". In: *The Journal of the Operational Research Society* 38.5, pp. 439–446. ISSN: 01605682, 14769360. (Visited on 12/21/2024).
Kingma, Diederik P. and Jimmy Ba (2015). "Adam: A Method for Stochastic Optimization". In: *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. Ed. by Yoshua Bengio and Yann LeCun.
Kiss, Gábor and Gergely Röst (Oct. 2017). "Controlling Mackey–Glass chaos". In: *Chaos: An Interdisciplinary Journal of Nonlinear Science* 27.11, p. 114321. ISSN: 1054-1500. DOI: 10.1063/1.5006922. eprint: https://pubs.aip.org/aip/cha/article-pdf/doi/10.1063/1.5006922/14003146/114321_1_online.pdf. URL: https://doi.org/10.1063/1.5006922.
Klenke, Achim (2007). *Probability Theory: A Comprehensive Course*. Universitext. Springer London. ISBN: 978-3-030-56401-8.
Kloeden, P.E. and E. Platen (1992). *Numerical Solution of Stochastic Differential Equations*. Springer-Verlag.
Koenker, Roger (1994). "Confidence Intervals for Regression Quantiles". In: *Asymptotic Statistics*. Ed. by Petr Mandl and Marie Hušková. Heidelberg: Physica-Verlag HD, pp. 349–359. ISBN: 978-3-642-57984-4.
— (2005). *Quantile Regression*. Econometric Society Monographs. Cambridge University Press.
Koller, Torsten et al. (2019). *Learning-based Model Predictive Control for Safe Exploration and Reinforcement Learning*. arXiv: 1906.12189 [eess.SY].
Kolmogoroff, A. (1931). "Über die analytischen Methoden in der Wahrscheinlichkeitsrechnung". In: *Mathematische Annalen* 104, pp. 415–458. URL: http://eudml.org/doc/159476.
Konda, Vijay and John Nikolaos Tsitsiklis (1999). "Actor-Critic Algorithms". In: *Advances in Neural Information Processing Systems*. Ed. by S. Solla, T. Leen, and K. Müller. Vol. 12. MIT Press. URL: https://proceedings.neurips.cc/paper_files/paper/1999/file/6449f44a102fde848669bdd9eb6b76fa-Paper.pdf.
Koos, Sylvain, Jean-Baptiste Mouret, and Stéphane Doncieux (2010). "Crossing the Reality Gap in Evolutionary Robotics by Promoting Transferable Controllers". In: *Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation*. GECCO '10. Portland, Oregon, USA: Association for Computing Machinery, pp. 119–126. ISBN: 9781450300728. DOI: 10.1145/1830483.1830505. URL: https://doi.org/10.1145/1830483.1830505.
Kuang, Y. (1993). *Delay Differential Equations: With Applications in Population Dynamics*. ISSN. Elsevier Science. ISBN: 9780080960029.
Küchler, Uwe and Beatrice Mensch (1992). "Langevins Stochastic Differential Equation Extended by a Time-Delayed Term". In: *Stochastics and Stochastic Reports* 40.1-2, pp. 23–42. DOI: 10.1080/17442509208833780. eprint: https://doi.org/10.1080/17442509208833780. URL: https://doi.org/10.1080/17442509208833780.

Kuksin, Sergei and Armen Shirikyan (2012). *Mathematics of Two-Dimensional Turbulence*. Cambridge Tracts in Mathematics. Cambridge University Press.
Kuss, Malte and Carl Rasmussen (2003). "Gaussian Processes in Reinforcement Learning". In: *Advances in Neural Information Processing Systems*. Ed. by S. Thrun, L. Saul, and B. Schölkopf. Vol. 16. MIT Press. URL: https://proceedings.neurips.cc/paper_files/paper/2003/file/7993e11204b215b27694b6f139e34ce8-Paper.pdf.
Kuznetsov, Arsenii et al. (July 2020). "Controlling Overestimation Bias with Truncated Mixture of Continuous Distributional Quantile Critics". In: *Proceedings of the 37th International Conference on Machine Learning*. Ed. by Hal Daumé III and Aarti Singh. Vol. 119. Proceedings of Machine Learning Research. PMLR, pp. 5556–5566. URL: https://proceedings.mlr.press/v119/kuznetsov20a.html.
Ladosz, Pawel et al. (Sept. 2022). "Exploration in deep reinforcement learning: A survey". In: *Inf. Fusion* 85.C, pp. 1–22. ISSN: 1566-2535. DOI: 10.1016/j.inffus.2022.03.003. URL: https://doi.org/10.1016/j.inffus.2022.03.003.
Lakshminarayanan, Aravind, Sahil Sharma, and Balaraman Ravindran (Feb. 2017). "Dynamic Action Repetition for Deep Reinforcement Learning". In: *Proceedings of the AAAI Conference on Artificial Intelligence* 31.1. DOI: 10.1609/aaai.v31i1.10918. URL: https://ojs.aaai.org/index.php/AAAI/article/view/10918.
Lamboley, Jimmy (2022). *Cours d'analyse pour l'Agrégation Externe de Mathématiques*. Avec un chapitre de probabilités par Quentin Berger. Version du 20 décembre 2021, 2021-2022.
Lancewicki, Tal, Aviv Rosenberg, and Yishay Mansour (June 2022). "Learning Adversarial Markov Decision Processes with Delayed Feedback". In: *Proceedings of the AAAI Conference on Artificial Intelligence* 36.7, pp. 7281–7289. DOI: 10.1609/aaai.v36i7.20690. URL: https://ojs.aaai.org/index.php/AAAI/article/view/20690.
Le Gall, Jean-François (2006). *Intégration, Probabilités et Processus Aléatoires*. URL: https://www.math.ens.fr/~legall/enseignement.html.
Le Gall, Jean-Francois (2013). *Mouvement brownien, martingales et calcul stochastique*. Springer.
Leahy, James-Michael et al. (July 2022). "Convergence of Policy Gradient for Entropy Regularized MDPs with Neural Network Approximation in the Mean-Field Regime". In: *Proceedings of the 39th International Conference on Machine Learning*. Ed. by Kamalika Chaudhuri et al. Vol. 162. Proceedings of Machine Learning Research. PMLR, pp. 12222–12252. URL: https://proceedings.mlr.press/v162/leahy22a.html.
Legendre, Guillaume (June 2021). *Méthodes numériques : Introduction à l'analyse numérique et au calcul scientifique*. Course notes, 800 pages.
Leroux, François (2019). *Systèmes dynamiques*. French.
Lévy, Paul (1954). "Processus semi-markoviens". In: *Proceedings of the International Congress of Mathematicians (ICM)*. Vol. III. Amsterdam: North-Holland, pp. 416–426.
Li, Gen et al. (2021). "Breaking the Sample Complexity Barrier to Regret-Optimal Model-Free Reinforcement Learning". In: *Advances in Neural Information Processing Systems*. Ed. by M. Ranzato et al. Vol. 34. Curran Associates, Inc., pp. 17762–17776. URL: https://proceedings.neurips.cc/paper_files/paper/2021/file/94739e5a5164b4d2396e253a11d57044-Paper.pdf.

Lillicrap, Timothy P. et al. (2019). *Continuous Control with Deep Reinforcement Learning*. arXiv: 1509.02971 [cs.LG].
Lin, Zhixuan et al. (July 2020). "Improving Generative Imagination in Object-Centric World Models". In: *Proceedings of the 37th International Conference on Machine Learning*. Ed. by Hal Daumé III and Aarti Singh. Vol. 119. Proceedings of Machine Learning Research. PMLR, pp. 6140–6149. URL: https://proceedings.mlr.press/v119/lin20f.html.
Lindley, D. V. (1956). "On a Measure of the Information Provided by an Experiment". In: *The Annals of Mathematical Statistics* 27.4, pp. 986–1005. ISSN: 00034851. URL: http://www.jstor.org/stable/2237191 (visited on 12/04/2023).
Lions, Jacques-Louis (1971). *Optimal Control of Systems Governed by Partial Differential Equations*. Grundlehren der mathematischen Wissenschaften in Einzeldarstellungen mit besonderer Berücksichtigung der Anwendungsgebiete. Springer-Verlag. ISBN: 3-540-05115-5.
Ljung, L. (1999). *System Identification: Theory for the User*. Prentice Hall information and system sciences series. Prentice Hall PTR. ISBN: 9780136566953.
Loève, Michel (1977). *Probability Theory*. Graduate texts in mathematics v. 1. Springer. ISBN: 9783540902102.
Lumley, John and Peter Blossey (Nov. 2003). "Control of turbulence". In: *Annual Review of Fluid Mechanics* 30, pp. 311–327. DOI: 10.1146/annurev.fluid.30.1.311.
Machado, Marlos C. et al. (2023). "Temporal Abstraction in Reinforcement Learning with the Successor Representation". In: *Journal of Machine Learning Research* 24.80, pp. 1–69. URL: http://jmlr.org/papers/v24/21-1213.html.
MacKay, David J. C. (July 1992). "Information-Based Objective Functions for Active Data Selection". In: *Neural Computation* 4.4, pp. 590–604. ISSN: 0899-7667. DOI: 10.1162/neco.1992.4.4.590. eprint: https://direct.mit.edu/neco/article-pdf/4/4/590/812354/neco.1992.4.4.590.pdf. URL: https://doi.org/10.1162/neco.1992.4.4.590.
Mackey, Michael C. and Leon Glass (1977). "Oscillation and Chaos in Physiological Control Systems". In: *Science* 197.4300, pp. 287–289. DOI: 10.1126/science.267326. eprint: https://www.science.org/doi/pdf/10.1126/science.267326. URL: https://www.science.org/doi/abs/10.1126/science.267326.
Mandt, Stephan, Matthew D. Hoffman, and David M. Blei (2017). "Stochastic Gradient Descent as Approximate Bayesian Inference". In: *Journal of Machine Learning Research* 18.134, pp. 1–35. URL: http://jmlr.org/papers/v18/17-214.html.
Matheron, Guillaume, Olivier Sigaud, and Nicolas Perrin (2020). *The problem with {DDPG}: understanding failures in deterministic environments with sparse rewards*.
McAllester, David (2003). "Simplified PAC-Bayesian Margin Bounds". In: *Learning Theory and Kernel Machines*. Ed. by Bern hard Schölkopf and Manfred K. Warmuth. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 203–215. ISBN: 978-3-540-45167-9.
McNamara, Antoine et al. (2004). "Fluid control using the adjoint method". In: *ACM SIGGRAPH 2004 Papers*. SIGGRAPH '04. Los Angeles, California: Association for Computing Machinery, pp. 449–456. ISBN: 9781450378239. DOI: 10.1145/1186562.1015744. URL: https://doi.org/10.1145/1186562.1015744.

Mehta, Viraj, Ian Char, et al. (2022). "Exploration via Planning for Information about the Optimal Trajectory". In: *Advances in Neural Information Processing Systems*.
Mehta, Viraj, Biswajit Paria, et al. (2022). "An Experimental Design Perspective on Model-Based Reinforcement Learning". In: *International Conference on Learning Representations*.
Melo, F. S. (2001). *Convergence of Q-learning: A simple proof*. Technical Report. Institute of Systems and Robotics.
Meng, Lingheng, Rob Gorbet, and Dana Kulić (2021). "Memory-based Deep Reinforcement Learning for POMDPs". In: *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5619–5626. DOI: 10.1109/IROS51168.2021.9636140.
Menier, Emmanuel et al. (May 2023). "CD-ROM: Complemented Deep - Reduced order model". In: *Computer Methods in Applied Mechanics and Engineering* 410, p. 115985. DOI: 10.1016/j.cma.2023.115985. URL: https://doi.org/10.1016%2Fj.cma.2023.115985.
Meyn, Sean (2022). *Control Systems and Reinforcement Learning*. Cambridge University Press. ISBN: 9781316511961.
Mézard, Marc and Andrea Montanari (Jan. 2009). *Information, Physics, and Computation*. Oxford University Press. ISBN: 9780198570837. DOI: 10.1093/acprof:oso/9780198570837.001.0001. URL: https://doi.org/10.1093/acprof:oso/9780198570837.001.0001.
Miyato, Takeru et al. (2018). "Spectral Normalization for Generative Adversarial Networks". In: arXiv: 1802.05957 [cs.LG].
Moerland, Thomas M. et al. (2022). *Model-based Reinforcement Learning: A Survey*. arXiv: 2006.16712 [cs.LG].
Mohammed, S.E.A. (1984). *Stochastic Functional Differential Equations*. Chapman & Hall/CRC research notes in mathematics series. Pitman Advanced Pub. Program. ISBN: 0-273-08593-X.
Mohammed, Salah-Eldin A. (1998). "Stochastic Differential Systems with Memory: Theory, Examples and Applications". In: *Stochastic Analysis and Related Topics VI*, pp. 1–77. URL: https://www.springer.com/gp/book/9780817640676.
Mohri, Mehryar, Afshin Rostamizadeh, and Ameet Talwalkar (2018). *Foundations of Machine Learning*. 2nd ed. Adaptive Computation and Machine Learning. Cambridge, MA: MIT Press. 504 pp. ISBN: 978-0-262-03940-6.
Monsel, Thibault et al. (2024). *Neural DDEs with Learnable Delays for Partially Observed Dynamical Systems*. arXiv: 2410.02843 [cs.LG]. URL: https://arxiv.org/abs/2410.02843.
Morimura, Tetsuro et al. (2010). "Nonparametric return distribution approximation for reinforcement learning". In: *Proceedings of the 27th International Conference on International Conference on Machine Learning*. ICML'10. Haifa, Israel: Omnipress, pp. 799–806. ISBN: 9781605589077.
Mulayoff, Rotem and Tomer Michaeli (2020). "Unique Properties of Flat Minima in Deep Networks". In: *Proceedings of the 37th International Conference on Machine Learning*. ICML'20. JMLR.org.
Munos, Rémi and Paul Bourgine (1997). "Reinforcement Learning for Continuous Stochastic Control Problems". In: *Advances in Neural Information Processing Systems*. Ed. by M. Jordan,

M. Kearns, and S. Solla. Vol. 10. MIT Press. URL: https://proceedings.neurips.cc/paper_files/paper/1997/file/186a157b2992e7daed3677ce8e9fe40f-Paper.pdf.
Nagabandi, Anusha et al. (2018). "Neural Network Dynamics for Model-Based Deep Reinforcement Learning with Model-Free Fine-Tuning". In: *2018 IEEE International Conference on Robotics and Automation (ICRA)*. Brisbane, Australia: IEEE Press, pp. 7559–7566. DOI: 10.1109/ICRA.2018.8463189. URL: https://doi.org/10.1109/ICRA.2018.8463189.
Neiswanger, Willie, Ke Alexander Wang, and Stefano Ermon (July 2021). "Bayesian Algorithm Execution: Estimating Computable Properties of Black-box Functions Using Mutual Information". In: *Proceedings of the 38th International Conference on Machine Learning*. Ed. by Marina Meila and Tong Zhang. Vol. 139. Proceedings of Machine Learning Research. PMLR, pp. 8005–8015. URL: https://proceedings.mlr.press/v139/neiswanger21a.html.
Neu, Gergely, Anders Jonsson, and Vicenç Gómez (2017). *A unified view of entropy-regularized Markov decision processes*. arXiv: 1705.07798. URL: http://arxiv.org/abs/1705.07798.
Neveu, J. (1970). *Bases mathématiques du calcul des probabilités*. Masson.
Neyshabur, Behnam, Srinadh Bhojanapalli, et al. (2017). "Exploring Generalization in Deep Learning". In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon et al. Vol. 30. Curran Associates, Inc. URL: https://proceedings.neurips.cc/paper_files/paper/2017/file/10ce03a1ed01077e3e289f3e53c72813-Paper.pdf.
Neyshabur, Behnam, Ryota Tomioka, and Nathan Srebro (July 2015). "Norm-Based Capacity Control in Neural Networks". In: *Proceedings of The 28th Conference on Learning Theory*. Ed. by Peter Grünwald, Elad Hazan, and Satyen Kale. Vol. 40. Proceedings of Machine Learning Research. Paris, France: PMLR, pp. 1376–1401. URL: https://proceedings.mlr.press/v40/Neyshabur15.html.
Nilim, Arnab and Laurent Ghaoui (2003). "Robustness in Markov Decision Problems with Uncertain Transition Matrices". In: *Advances in Neural Information Processing Systems*. Ed. by S. Thrun, L. Saul, and B. Schölkopf. Vol. 16. MIT Press. URL: https://proceedings.neurips.cc/paper_files/paper/2003/file/300891a62162b960cf02ce3827bb363c-Paper.pdf.
Noakes, Lyle (1991). "The Takens Embedding Theorem". In: *International Journal of Bifurcation and Chaos* 01, pp. 867–872.
Øksendal, B. (2010). *Stochastic Differential Equations: An Introduction with Applications*. Universitext. Springer Berlin Heidelberg. ISBN: 9783642143946.
Paehler, Ludger et al. (Nov. 2023). "HydroGym: A Reinforcement Learning Control Framework for Fluid Dynamics". In: *Bulletin of the American Physical Society*.
Pan, Yangchen et al. (July 2018). "Reinforcement Learning with Function-Valued Action Spaces for Partial Differential Equation Control". In: *Proceedings of the 35th International Conference on Machine Learning*. Ed. by Jennifer Dy and Andreas Krause. Vol. 80. Proceedings of Machine Learning Research. PMLR, pp. 3986–3995. URL: https://proceedings.mlr.press/v80/pan18a.html.
Parr, R. E. (1998). "Hierarchical Control and Learning for Decision Processes". PhD thesis. University of California Berkeley.
Peitz, Sebastian, Samuel E. Otto, and Clarence W. Rowley (2020). "Data-Driven Model Predictive Control using Interpolated Koopman Generators". In: *SIAM Journal on Applied Dynami-*

cal Systems 19.3, pp. 2162–2193. DOI: 10.1137/20M1325678. eprint: https://doi.org/10.1137/20M1325678. URL: https://doi.org/10.1137/20M1325678.
Peitz, Sebastian, Jan Stenner, et al. (2024). "Distributed Control of Partial Differential Equations Using Convolutional Reinforcement Learning". In: *Physica D: Nonlinear Phenomena* 461, p. 134096. ISSN: 0167-2789. DOI: https://doi.org/10.1016/j.physd.2024.134096. URL: https://www.sciencedirect.com/science/article/pii/S0167278924000472.
Peypouquet, J. (2015). *Convex Optimization in Normed Spaces: Theory, Methods and Examples*. SpringerBriefs in Optimization. Springer International Publishing. ISBN: 9783319137100.
Pham, H. (2009). *Continuous-time Stochastic Control and Optimization with Financial Applications*. Stochastic Modelling and Applied Probability. Springer Berlin Heidelberg. ISBN: 978-3-540-89500-8.
Pinneri, Cristina et al. (Nov. 2021). "Sample-efficient Cross-Entropy Method for Real-time Planning". In: *Proceedings of the 2020 Conference on Robot Learning*. Ed. by Jens Kober, Fabio Ramos, and Claire Tomlin. Vol. 155. Proceedings of Machine Learning Research. PMLR, pp. 1049–1065. URL: https://proceedings.mlr.press/v155/pinneri21a.html.
Pinsker, M. S. (1964). *Information and information stability of random variables and processes / by M.S. Pinsker. Translated and edited by Amiel Feinstein.* eng;und. Holden-Day series in time series analysis. San Francisco: Holden-Day.
Plaat, Aske (2022). *Deep Reinforcement Learning*. Springer Nature Singapore. ISBN: 978-981-19-0638-1. DOI: 10.1007. URL: http://dx.doi.org/10.1007/978-981-19-0638-1.
Polyak, B. T. and A. B. Juditsky (1992). "Acceleration of Stochastic Approximation by Averaging". In: *SIAM Journal on Control and Optimization* 30.4, pp. 838–855. DOI: 10.1137/0330046. eprint: https://doi.org/10.1137/0330046. URL: https://doi.org/10.1137/0330046.
Precup, Doina (2000). "Temporal Abstraction in Reinforcement Learning". PhD thesis. University of Massachusetts Amherst.
Precup, Doina and Richard Stuart Sutton (1997). "Multi-time Models for Temporally Abstract Planning". In: *Advances in Neural Information Processing Systems*. Ed. by M. Jordan, M. Kearns, and S. Solla. Vol. 10. MIT Press. URL: https://proceedings.neurips.cc/paper_files/paper/1997/file/a9be4c2a4041cadbf9d61ae16dd1389e-Paper.pdf.
Puterman, Martin L (2014). *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons.
Rabault, Jean et al. (2019). "Artificial neural networks trained through deep reinforcement learning discover control strategies for active flow control". In: *Journal of Fluid Mechanics* 865, pp. 281–302. DOI: 10.1017/jfm.2019.62.
Rackauckas, Christopher et al. (2021). *Universal Differential Equations for Scientific Machine Learning*. arXiv: 2001.04385 [cs.LG]. URL: https://arxiv.org/abs/2001.04385.
Raffin, Antonin et al. (2021). "Stable-Baselines3: Reliable Reinforcement Learning Implementations". In: *Journal of Machine Learning Research* 22.268, pp. 1–8. URL: http://jmlr.org/papers/v22/20-1364.html.
Ramstedt, Simon et al. (2020). "Reinforcement Learning with Random Delays". In: *ArXiv* abs/2010.02966. URL: https://api.semanticscholar.org/CorpusID:222177494.

Rasmussen, Carl Edward and Christopher K. I. Williams (2006). *Gaussian processes for machine learning*. Adaptive computation and machine learning. MIT Press, pp. I–XVIII, 1–248. ISBN: 026218253X.
Recht, Benjamin (2018). *A Tour of Reinforcement Learning: The View from Continuous Control*. arXiv: 1806.09460 [math.OC].
Redjil, Amel and Salah Eddine Choutri (2017). *On Relaxed Stochastic Optimal Control for Stochastic Differential Equations Driven by G-Brownian Motion*. arXiv: 1702.08735 [math.PR]. URL: https://arxiv.org/abs/1702.08735.
Revuz, Daniel and Marc Yor (1999). *Continuous Martingales and Brownian Motion*. Springer.
Robert, C. (2007). *The Bayesian Choice: From Decision-Theoretic Foundations to Computational Implementation*. Springer Texts in Statistics. Springer New York. ISBN: 9780387715988.
Rousseau, Judith (2009). *Statistique Bayésienne : Notes de cours*. Rédigé par Mathias André et Alexis Eidelman, relu par Julyan Arbel. Basé sur l'ouvrage de Christian P. Robert, *Le Choix Bayésien*.
Rowland, Mark et al. (2024). "An Analysis of Quantile Temporal-Difference Learning". In: *Journal of Machine Learning Research* 25.163, pp. 1–47. URL: http://jmlr.org/papers/v25/23-0154.html.
Rubinstein, Reuven Y. and Dirk P. Kroese (2004). *The Cross Entropy Method: A Unified Approach To Combinatorial Optimization, Monte-Carlo Simulation (Information Science and Statistics)*. Berlin, Heidelberg: Springer-Verlag. ISBN: 038721240X.
Sagun, Levent, Leon Bottou, and Yann LeCun (2017). "Eigenvalues of the Hessian in Deep Learning: Singularity and Beyond". In: arXiv: 1611.07476 [cs.LG].
Salomon, J. (2010). *Traitement numérique du signal*.
Santambrogio, F. (2015). *Optimal Transport for Applied Mathematicians: Calculus of Variations, PDEs, and Modeling*. Progress in Nonlinear Differential Equations and Their Applications. Springer International Publishing. ISBN: 9783319208282.
Satia, Jay K. and Roy E. Lave (1973). "Markovian Decision Processes with Uncertain Transition Probabilities". In: *Operations Research* 21.3, pp. 728–740. ISSN: 0030364X, 15265463. URL: http://www.jstor.org/stable/169381.
Schmidhuber, Jürgen (2015). "Deep learning in neural networks: An overview". In: *Neural Networks* 61, pp. 85–117. ISSN: 0893-6080. DOI: https://doi.org/10.1016/j.neunet.2014.09.003. URL: https://www.sciencedirect.com/science/article/pii/S0893608014002135.
Schulman, John, Sergey Levine, et al. (July 2015). "Trust Region Policy Optimization". In: *Proceedings of the 32nd International Conference on Machine Learning*. Ed. by Francis Bach and David Blei. Vol. 37. Proceedings of Machine Learning Research. Lille, France: PMLR, pp. 1889–1897. URL: https://proceedings.mlr.press/v37/schulman15.html.
Schulman, John, Filip Wolski, et al. (2017). "Proximal Policy Optimization Algorithms". In: *CoRR*. arXiv: 1707.06347. URL: http://arxiv.org/abs/1707.06347.
Shalev-Shwartz, Shai and Shai Ben-David (2014). *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press. DOI: 10.1017/CBO9781107298019.
Shalizi, Cosma (May 2007). *36-754, Advanced Probability II: Almost None of the Theory of Stochastic Processes*. Course materials, Carnegie Mellon University.

Sharma, Sahil, Aravind S. Lakshminarayanan, and Balaraman Ravindran (2017). "Learning to Repeat: Fine Grained Action Repetition for Deep Reinforcement Learning". In: *International Conference on Learning Representations*. URL: https://openreview.net/forum?id=B1GOWV5eg.

Shen, Zebang et al. (June 2019). "Hessian Aided Policy Gradient". In: *Proceedings of the 36th International Conference on Machine Learning*. Ed. by Kamalika Chaudhuri and Ruslan Salakhutdinov. Vol. 97. Proceedings of Machine Learning Research. PMLR, pp. 5729–5738. URL: https://proceedings.mlr.press/v97/shen19d.html.

Sigaud, Olivier and Olivier Buffet (2010). *Markov Decision Processes in Artificial Intelligence*. Wiley. ISBN: 978-1-848-21167-4.

Sigaud, Olivier and Freek Stulp (2019). "Policy search in continuous action domains: An overview". In: *Neural Networks* 113, pp. 28–40. ISSN: 0893-6080. DOI: https://doi.org/10.1016/j.neunet.2019.01.011. URL: https://www.sciencedirect.com/science/article/pii/S089360801930022X.

Silver, David et al. (Jan. 2016). "Mastering the game of Go with deep neural networks and tree search". In: *Nature* 529.7587, pp. 484–489. ISSN: 1476-4687. DOI: 10.1038/nature16961. URL: https://doi.org/10.1038/nature16961.

Simonyan, Karen and Andrew Zisserman (2015). *Very Deep Convolutional Networks for Large-Scale Image Recognition*. arXiv: 1409.1556 [cs.CV]. URL: https://arxiv.org/abs/1409.1556.

Sipp, Denis and Anton Lebedev (2007). "Global stability of base and mean flows: a general approach and its applications to cylinder and open cavity flows". In: *Journal of Fluid Mechanics* 593, pp. 333–358. DOI: 10.1017/S0022112007008907.

Smith, Hal (2010). *An Introduction to Delay Differential Equations with Applications to the Life Sciences*. Texts in Applied Mathematics. Springer New York. ISBN: 9781441976468.

Steinhauser, Martin Oliver (2013). *Computer Simulation in Physics and Engineering*. Berlin, Boston: De Gruyter. ISBN: 9783110256062. DOI: doi:10.1515/9783110256062. URL: https://doi.org/10.1515/9783110256062.

Steinwart, Ingo, Don Hush, and Clint Scovel (2009). "Learning from dependent observations". In: *Journal of Multivariate Analysis* 100.1, pp. 175–194. ISSN: 0047-259X. DOI: https://doi.org/10.1016/j.jmva.2008.04.001. URL: https://www.sciencedirect.com/science/article/pii/S0047259X08001097.

Stephany, Robert et al. (2024). "Learning The Delay in Delay Differential Equations". In: *ICLR 2024 Workshop on AI4DifferentialEquations In Science*. URL: https://openreview.net/forum?id=VTYhJLoOaR.

Stoehr, Julien (2017). *Méthodes de Monte Carlo*. Master 1, 2017–2018.

Stratonovich, R. L. (1960). "Conditional Markov Processes". In: *Theory of Probability & Its Applications* 5.2, pp. 156–178. DOI: 10.1137/1105015. eprint: https://doi.org/10.1137/1105015. URL: https://doi.org/10.1137/1105015.

Subramanian, Jayakumar et al. (2022). "Approximate Information State for Approximate Planning and Reinforcement Learning in Partially Observed Systems". In: *Journal of Machine Learning Research* 23.12, pp. 1–83. URL: http://jmlr.org/papers/v23/20-1165.html.

Sutton, Richard Stuart (1995). "TD Models: Modeling the World at a Mixture of Time Scales". In: *International Conference on Machine Learning*.
Sutton, Richard Stuart and Andrew G. Barto (2018). *Reinforcement Learning: An Introduction*. Second. The MIT Press. URL: http://incompleteideas.net/book/the-book-2nd.html.
Sutton, Richard Stuart, David McAllester, et al. (1999). "Policy Gradient Methods for Reinforcement Learning with Function Approximation". In: *Advances in Neural Information Processing Systems*. Ed. by S. Solla, T. Leen, and K. Müller. Vol. 12. MIT Press. URL: https://proceedings.neurips.cc/paper_files/paper/1999/file/464d828b85b0bed98e80ade0a5c43b0f-Paper.pdf.
Sutton, Richard Stuart, Doina Precup, and Satinder Singh (Aug. 1999). "Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning". In: *Artificial Intelligence* 112.1-2, pp. 181–211. ISSN: 0004-3702. DOI: 10.1016/s0004-3702(99)00052-1.
Takens, Floris (1981). "Detecting strange attractors in turbulence". In: *Dynamical Systems and Turbulence, Warwick 1980*. Ed. by David Rand and Lai-Sang Young. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 366–381. ISBN: 978-3-540-38945-3.
Tallec, Corentin, Léonard Blier, and Yann Ollivier (June 2019). "Making Deep Q-learning methods robust to time discretization". In: *Proceedings of the 36th International Conference on Machine Learning*. Ed. by Kamalika Chaudhuri and Ruslan Salakhutdinov. Vol. 97. Proceedings of Machine Learning Research. PMLR, pp. 6096–6104. URL: https://proceedings.mlr.press/v97/tallec19a.html.
Thrun, Sebastian and Anton Schwartz (1999). "Issues in Using Function Approximation for Reinforcement Learning". In.
Touati, Ahmed and Yann Ollivier (2021). "Learning One Representation to Optimize All Rewards". In: *Advances in Neural Information Processing Systems*. Ed. by M. Ranzato et al. Vol. 34. Curran Associates, Inc., pp. 13–23. URL: https://proceedings.neurips.cc/paper_files/paper/2021/file/003dd617c12d444ff9c80f717c3fa982-Paper.pdf.
Towers, Mark et al. (2024). *Gymnasium: A Standard Interface for Reinforcement Learning Environments*. arXiv: 2407.17032 [cs.LG]. URL: https://arxiv.org/abs/2407.17032.
Trélat, Emmanuel (Jan. 2005). *Contrôle optimal : théorie et applications*.
— (2023). *Control in Finite and Infinite Dimension*. 1st ed.
Tsitsiklis, John Nikolaos (1994). "Asynchronous Stochastic Approximation and Q-Learning". In: *Machine Learning* 16.3, pp. 185–202. ISSN: 1573-0565. DOI: 10.1023/A:1022689125041. URL: https://doi.org/10.1023/A:1022689125041.
Tsitsiklis, John Nikolaos and Benjamin Van Roy (1997). "An analysis of temporal-difference learning with function approximation". In: *IEEE Transactions on Automatic Control* 42.5, pp. 674–690. DOI: 10.1109/9.580874.
Tsybakov, A.B. (2008). *Introduction to Nonparametric Estimation*. Springer Series in Statistics. Springer New York. ISBN: 9780387790527.
Valiant, Leslie G. (1984). *Probably Approximately Correct: Nature's Algorithms for Learning and Prospering in a Complex World*. ISBN: 978-0-465-03271-6.
Vernet, J. et al. (June 2014). "Flow separation delay on trucks A-pillars by means of Dielectric Barrier Discharge actuation". In: *First International Conference in Numerical and Experimental*

Aerodynamics of Road Vehicles and Trains (Aerovehicles 1). Aerovehicles, Aerovehicles 1-2014–53.
Villani, C. (2008). *Optimal Transport: Old and New*. Grundlehren der mathematischen Wissenschaften. Springer Berlin Heidelberg. ISBN: 9783540710509.
Vincent, Thomas L. and Jianzu Yu (Mar. 1991). "Control of a chaotic system". In: *Dynamics and Control* 1.1, pp. 35–52. ISSN: 1573-8450. DOI: 10.1007/BF02169423. URL: https://doi.org/10.1007/BF02169423.
Viquerat, J. et al. (Nov. 2022). "A review on deep reinforcement learning for fluid mechanics: An update". In: *Physics of Fluids* 34.11, p. 111301. DOI: 10.1063/5.0128446. URL: https://doi.org/10.1063%2F5.0128446.
Viterbo, Claude (2009). *Systèmes dynamiques et Équations différentielles*. Cours de Mathématiques de deuxième année à l'École Polytechnique. Palaiseau, France: Éditions de l'École Polytechnique.
Walsh, Thomas et al. (Feb. 2008). "Learning and planning in environments with delayed feedback". In: *Autonomous Agents and Multi-Agent Systems* 18, pp. 83–105. DOI: 10.1007/s10458-008-9056-7.
Wang, Haoran, Thaleia Zariphopoulou, and Xun Yu Zhou (2020). "Reinforcement Learning in Continuous Time and Space: A Stochastic Control Approach". In: *Journal of Machine Learning Research* 21.198, pp. 1–34. URL: http://jmlr.org/papers/v21/19-144.html.
Wang, Wei et al. (2024). "Addressing Signal Delay in Deep Reinforcement Learning". In: *The Twelfth International Conference on Learning Representations*. URL: https://openreview.net/forum?id=Z8UfDs4J46.
Watkins, Christopher J.C.H. and Peter Dayan (1992). "Technical Note: Q-Learning". In: *Machine Learning* 8.3, pp. 279–292. ISSN: 1573-0565. DOI: 10.1023/A:1022676722315. URL: https://doi.org/10.1023/A:1022676722315.
White III, Chelsea C. (1988). "Note on "A Partially Observable Markov Decision Process with Lagged Information": Reply". In: *Journal of the Operational Research Society* 39.2, pp. 217–218. DOI: 10.1057/jors.1988.37. eprint: https://doi.org/10.1057/jors.1988.37. URL: https://doi.org/10.1057/jors.1988.37.
Williams, Ronald J., Jing Peng, and Hong Li (1991). "Function Optimization using Connectionist Reinforcement Learning Algorithms". In: *Connection Science* 3.3, pp. 241–268. DOI: 10.1080/09540099108946587.
Wiltzer, Harley et al. (Dec. 2024). "Action Gaps and Advantages in Continuous-Time Distributional Reinforcement Learning". In: *Advances in Neural Information Processing Systems (NeurIPS)*. Vancouver, BC, Canada: Curran Associates, Inc.
Xie, Zeke, Issei Sato, and Masashi Sugiyama (2021). "A Diffusion Theory For Deep Learning Dynamics: Stochastic Gradient Descent Exponentially Favors Flat Minima". In: arXiv: 2002.03495 [cs.LG].
Yildiz, Cagatay, Markus Heinonen, and Harri Lähdesmäki (July 2021). "Continuous-time Model-based Reinforcement Learning". In: *Proceedings of the 38th International Conference on Machine Learning*. Ed. by Marina Meila and Tong Zhang. Vol. 139. Proceedings of Machine

Learning Research. PMLR, pp. 12009–12018. URL: https://proceedings.mlr.press/v139/yildiz21a.html.
Yoshida, Yuichi and Takeru Miyato (2017). *Spectral Norm Regularization for Improving the Generalizability of Deep Learning*.
Zhao, Yang, Hao Zhang, and Xiuyuan Hu (July 2022). "Penalizing Gradient Norm for Efficiently Improving Generalization in Deep Learning". In: *Proceedings of the 39th International Conference on Machine Learning*. Ed. by Kamalika Chaudhuri et al. Vol. 162. Proceedings of Machine Learning Research. PMLR, pp. 26982–26992. URL: https://proceedings.mlr.press/v162/zhao22i.html.
Zhou, K., J.C. Doyle, and K. Glover (1996). *Robust and Optimal Control*. Feher/Prentice Hall Digital and. Prentice Hall. ISBN: 9780134565675.
Zhu, Qunxi, Yao Guo, and Wei Lin (2021). *Neural Delay Differential Equations*. arXiv: 2102.10801 [cs.LG].
Ziebart, Brian D. (2010). "Modeling Purposeful Adaptive Behavior with the Principle of Maximum Causal Entropy". AAI3438449. PhD thesis. USA. ISBN: 9781124414218.
Zuily, Claude (2002). *Éléments de distributions et d'équations aux dérivées partielles*. Dunod. ISBN: 2-10-005735-9.

Index