UCLouvain
Institute for Information
and Communication Technologies,
Electronics and Applied Mathematics

Data-driven memory-dependent abstractions of dynamical systems

Adrien Banse

Thesis submitted in partial fulfillment
of the requirements for the degree of
Doctor in Applied Sciences

Dissertation committee:

Prof. Raphaël Jungers (UCLouvain, *advisor*)
Prof. Alessandro Abate (University of Oxford, *advisor*)
Prof. Duarte Antunes (TU Eindhoven, *reader*)
Prof. Gianluca Bianchin (UCLouvain, *reader*)
Prof. Manuel Mazo Jr. (TU Delft, *reader*)
Prof. Sadegh Soudjani (University of Birmingham and Max Planck Institute for Software Systems, *reader*)
Prof. Philippe Lefèvre (UCLouvain, *chair*)

Version of June, 2026.

This thesis was supported by the French Community of Belgium in the framework of a FNRS/FRIA grant.
© Adrien Banse, UCLouvain (2026), all rights reserved.

Finding the occasional straw of truth awash in a great ocean of confusion and bamboozle requires vigilance, dedication, and courage. But if we don't practice these tough habits of thought, we cannot hope to solve the serious problems that face us.

– Carl Sagan

Acknowledgments

WRITING a Ph.D. thesis undoubtedly involves a certain amount of solitude. Paradoxically, this process allows you to meet many inspiring people from diverse backgrounds, and helps you realize just how important your loved ones are. This section and, more broadly, this dissertation are dedicated to all of these people.

First of all, I would like to thank Raphaël Jungers, my supervisor, for the trust he placed in me from the very beginning of my research journey, nearly five years ago now. Raphaël is an advisor who is extremely dedicated to his researchers: he proofreads your drafts in record time, never hesitates to connect you with his network, and always makes it possible for you to seize opportunities. Most importantly, I would like to thank him for always supporting me in my decisions, whether they involved research topics, visits abroad, or side projects – always maintaining a seamless balance between mentoring me and granting me autonomy. What I will remember most about him is that the only limits to professional opportunities are the ones we impose on ourselves. I would also like to thank Alessandro Abate for his support, advice and kindness throughout our collaboration. In particular, I would like to thank him for the warm welcome I received in Oxford during my first year of Ph.D., which helped make this month-long visit as productive as it was.

My deepest gratitude also goes to the members of my committee: Duarte Antunes, Gianluca Bianchin, Manuel Mazo Jr. and Sadegh Soudjani. Thank you very much for your invaluable feedback on the draft version of this manuscript, and for the fascinating discussion we had during the private defense; I could tell that you had read my thesis in detail, and I am very grateful for that. I would also like to thank Philippe Lefèvre for chairing my committee and, more broadly, for helping me organising the defense.

Throughout my Ph.D. thesis, I had the tremendous good fortune to collaborate with brilliant researchers, without whom this dissertation would not exist in its current form: Licio, for the long evenings spent in Oxford; Giannis, for helping me find the right tools to answer my questions; and Luca and Manuel for their warm welcome at TU Delft and for the smooth collaboration. I am already looking forward to working with each of you again. I also had the opportunity to engage in many interesting discussions (whether related to research or not) with many fellow researchers; I am thinking, for example, of Luca's team composed of Eduardo, Frederik, and Steven, but also Ben, Gabriel, Rudi, Morteza, Ibon, Karan, etc. It would be an honor for me to work with one of you in the future (if not already in the pipeline!).

I was also fortunate enough to work in the wonderful environment that is the Euler building. First of all, I would like to thank the administrative and technical staff for their tremendous help. In particular, I would like to thank Marie-Christine, Pascale, François W., and John, who work tirelessly every day to make our research activities possible. Second, I would like to thank Raphaël's team, where I felt welcome right from the start of my master's thesis. In particular, I am thinking of Adrien J., Alexis, Amir B., Antoine A., Benoît, Guillaume, Julien, Léa, Lucas, Margot, Matteo, Riccardo, Sébastien M., Virginie, Wouter and Zheming. I have always felt that we make a great team, especially thanks to our group meetings and research retreats. I would also like to thank other Euler's researchers with whom I have crossed paths: Alexandre, Amir M., Antonin, Arthur., Astrid, Bastien, Charles, Clémence, Donatien, François L., François W., Guillaume, Hari, John, Jonas, Martina, Nizar, Renato, Riccardo, Roy, Sébastien C., Simon M., Simon V., Sophie, Théo, Viktoriia, Vincent (and everyone I forgot). A special thanks to Julien, Philémon, and Yassine, who are true friends and with whom we have shared many wild afterworks and nights out. I am confident that future generations will continue to foster this environment, and I am very proud to have been part of it.

I would also like to thank my family and friends for their unfailing support. First of all, I would like to thank my parents, without whom none of this would be possible. Second, I would like to thank my broader family, including my cousins, uncles, aunts, and grandparents. A special thanks goes to Camille and Emilie, with whom I was able to speak openly about the challenges of pursuing a Ph.D. thesis, and Xavier, for the interesting discussions about research during our family gatherings. Finally, I would like to say just how happy I am to be surrounded by such incredible

friends. My life would be very different if I had not been surrounded by such adventurous, enterprising and caring friends, who were always there for me during tough times.

Last but not least, I would like to thank Adèle for her infinite amount of support and love. Every day I tell myself that I am the luckiest man on earth to be able to share my life with you, and I just cannot wait to see what the future holds.

Abstract

WITH the increasing ubiquity of *safety-critical* autonomous systems operating in uncertain environments, there is a need for methods that provide behavioural guarantees in the presence of *epistemic uncertainty*, i.e. lack of knowledge about the system, and potentially *aleatoric uncertainty*, i.e. intrinsic uncertainty of the system. Toward formally verifying properties of complex systems, methods based on system approximations – *abstractions* – have surged in recent years.

This thesis studies memory-dependent abstraction-based methods for both deterministic and stochastic systems (modelling aleatoric uncertainty) where only partial observations are available (modelling epistemic uncertainty). Instead of representing the system through memoryless discrete states, the proposed approach builds abstractions whose states encode observation histories. For deterministic systems, two frameworks are developed: uniform memory, where abstractions encode fixed-length output sequences, and adaptive memory, a more flexible setting in which memory is allocated locally through a refinement procedure. The thesis then extends memory-dependent abstractions to stochastic systems, a technically challenging extension that necessitates different theoretical tools.

Additionally, we introduce the *Cantor-Kantorovich* distance, a novel metric between labelled Markov chains. This distance arises from the natural topology over infinite symbolic traces, and is useful for our memory refinement procedure. Beyond its direct role in this thesis, the Cantor-Kantorovich distance exhibits strong theoretical properties and holds potential for broader applications, making it a transversal yet central contribution of this thesis.

Contents

⋆ **Introduction** 1

⋆ **List of publications** 13

⋆ **List of symbols** 17

I Background

1 Symbolic dynamics 27
1.1 Language theory 27
1.1.1 Definition 28
1.1.2 The natural topology 28
1.1.3 Specifications 30
1.2 Labelled Markov chains 32
1.2.1 Definition 32
1.2.2 Qualitative and quantitative comparison of LMCs 36

2 Dynamical systems 41
2.1 Deterministic dynamical systems 41
2.2 Stochastic dynamical systems 44

II Abstractions of deterministic systems

3 Uniform memory-dependent abstractions 51
3.1 Framework 53
3.2 Method 55

3.3 Guarantee 59
3.4 Numerical experiments 63
3.5 Summary and further research directions 64

4 Adaptive memory refinement 69
4.1 Framework 72
4.2 Adaptive memory abstractions 73
4.3 Refinement procedure 79
4.4 Numerical experiments 84
4.4.1 Smart partitioning 84
4.4.2 Comparison with uniform memory approach 86
4.4.3 Application to safety analysis 87
4.5 Summary and further research directions 90

III Abstractions of stochastic systems

5 Memory-dependent abstractions of stochastic systems via transfer operators 97
5.1 Framework 100
5.2 Preliminaries 103
5.2.1 Transfer operator 103
5.2.2 Galerkin approximation 106
5.3 Method 108
5.4 Abstractions as Galerkin approximations 113
5.5 Total Variation guarantees 116
5.6 Numerical experiments 123
5.7 Summary and further research directions 127

IV The Cantor-Kantorovich distance

6 Theoretical foundations of the Cantor-Kantorovich distance . 133
6.1 Definition and link with Total Variation 136
6.2 Continuity properties 149
6.3 Approximation 153
6.4 Computational complexity 155
6.5 Summary and further research directions 160

⋆ **Conclusions and perspectives 163**

Appendices

A Topological and metric spaces 171
A.1 Topological spaces 171
A.2 (Ultra-)metric spaces 172

B Measure theory 175

C Linear Algebra 181
C.1 Norms and Banach spaces 181
C.2 Inner products and Hilbert spaces 182
C.3 $L^p(\mu)$ spaces 183

D Probability theory 185
D.1 Probability measures and densities 185
D.2 Distances between probability measures 189

E Computational complexity 193
E.1 Basics 193
E.2 Decision problems 195
E.3 Counting problems 196

Bibliography 199

Introduction

THE broad field of control theory aims at developing methods for analysing and controlling dynamical systems under potential uncertainty in their behaviour and environment [AJP24]. Among the extensive range of applications of control theory are the so-called *safety-critical systems*, i.e. systems for which guarantees that the system exhibits a desired behaviour are needed. Prominent examples of safety-critical systems include autonomous driving, robots in rescue missions, smart grids, smart buildings, etc. [Ala22, PHC+22] – these are all systems for which a failure may result in major human and/or environmental harm. Towards safe deployment of such physical systems, control theorists need advanced mathematical tools to formally verify if they meet pre-specified requirements. There exists a rich set of methods to analyse and control dynamical systems, among which (control) Lyapunov analysis, barrier functions (see e.g. [LWY+23, ATH+23, BGK21]), and optimal control (see e.g. [RMD17, LVS12, BV04]). However, these methods require an accurate model of the systems to provide robust guarantees.

In parallel, recently, due to major technological upheavals, the complexity of many dynamical systems has dramatically increased. This paradigm shift has been coined by the academic community under the name of the *Cyber-Physical revolution* [Lee15, HJT12, KK12, Alu15, LS16]. At the centre of the Cyber-Physical revolution is the notion of *uncertainty*, which we split into two categories [KD09, Mat07]:

- On the one hand, *epistemic uncertainty* is the uncertainty linked to the lack of knowledge about the system, and can therefore vanish with increasing knowledge of the system. Following the Cyber-Physical revolution, the systems are becoming so complex that it is very challenging (if not impossible) to model them accurately, increasing their epistemic uncertainty. It is therefore crucial to develop robust methods that take into account epistemic uncertainty, or reduce it – unlike the classical methods cited above.

- On the other hand, *aleatoric uncertainty* is the intrinsic uncertainty of the system. In the context of control theory, it is often the consequence of an uncertain environment – think of wind hitting a drone, or unpredictable human behaviour influencing a stock market. This type of uncertainty is naturally present in the system, which makes it impossible for any method to make it disappear. However, in regard to the recent increase in epistemic uncertainty, it is central to develop methods that are able to provide guarantees in the presence of both epistemic and aleatoric uncertainty.

In this thesis, we model aleatoric uncertainty through *stochasticity*. Specifically, we analyse two distinct formulations:

$$x_{k+1} = f(x_k), \quad \text{(Without aleatoric uncertainty)}$$
$$x_{k+1} \sim \tau(\cdot|x_k). \quad \text{(With aleatoric uncertainty)}$$

Although widely used to account for aleatoric uncertainty, stochastic models are not the only popular models. Other dynamical systems, such as switched or distribution-free noisy systems (see e.g. [Jun09, BWJ24]), also incorporate aleatoric uncertainty, although without stochasticity.

To provide guarantees in the presence of both epistemic and potentially aleatoric uncertainty, methods based on discrete approximations, called *abstractions*, have recently surged [Tab09, LSAZ22]. Arising from the computer science community and inspired from *symbolic dynamics* (see e.g. [LM95, PP04]), abstraction-based methods consist of three steps. First, the state space is discretised (often by gridding it), and the original continuous system is modelled as a discrete mathematical object (e.g. an automaton, or a Markov Decision Process). Second, an analysis is performed on the coarse model. Possible analyses include designing a controller that satisfies pre-specified conditions, formulated as solving a *game* (see e.g. [BYAG17] for deterministic models), or assessing the safety of autonomous stochas-

tic systems using computer science tools such as Prism [KNP11]. Third, depending on the *relation* between the abstraction and the original system, the analysis or controller can be transferred back to the original system, together with the guarantees (see [CGJ24] for a characterisation of such relations).

Most abstraction-based techniques directly rely on the discretisation of the state space, and construct a model whose discrete states correspond to the different regions of the discretisation. To reduce epistemic uncertainty, a classical and intuitive method is to refine the discretisation, thereby increasing the number of states in the abstraction. In fact, it was proven that if the original system is *incrementally stable*, then there exists a fine enough discretisation such that the abstraction represents exactly the original system (see e.g. [GPT10]) – a situation in which all the epistemic uncertainty has disappeared. However, this approach has two major flaws. First, it relies on incremental stability, a rather restrictive assumption and, second, it suffers from the curse of dimensionality, as the number of abstract states grows exponentially with the dimension. In response to these problems, the community has investigated smarter methods to abstract systems as discrete mathematical objects.

This thesis explores approaches that use *memory* rather than refining the state space directly. Specifically, we focus on abstractions that represent sequences of observations rather than individual outputs, as depicted in Figure ⋆.1. These abstractions are structured as graphs or Markov chains, depending on whether the framework is purely deterministic or stochastic (due to aleatoric uncertainty or sampling). Each node captures a sequence of observations, while the edges define transitions based on historical data. By expanding the memory, we therefore enhance the precision of the approximation.

Our approach is mainly motivated by the two following use cases.

Motivation 1 (Partially observable)**.** In the first case, one does not have access to the states, but only to a region in which the current state belongs – think of a black-box system that only gives the information that the state belongs to the left or right part of the state space, similarly as in Figure ⋆.1. These systems are referred to as *partially observable systems*. In this case, to reduce epistemic uncertainty, one cannot use the classical refinement approach. Increasing memory is therefore a way to leverage observations to reduce epistemic uncertainty.

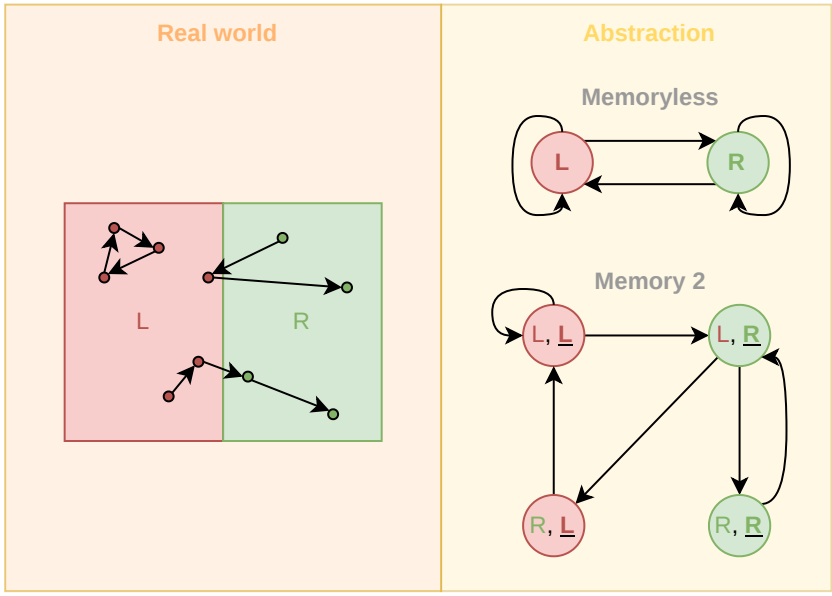


Fig. ⋆.1 Illustration of memory-dependent abstractions. The left part shows the original system, with its state space discretized into two distinct regions (left and right). On the right, two abstractions are presented: the first is a memoryless model, where all transitions between states are permitted, resulting in a coarse and less accurate approximation of the original system. The second abstraction incorporates memory by encoding sequences of length 2, restricting certain transitions and yielding a finer, more precise approximation of the real system.

Motivation 2 (Finite abstraction)**.** The second case is the classical *finite abstractions* framework – one can refine the state space – but where the curse of dimensionality limits the analysis in the sense that the abstraction has reached a maximal size (e.g. for computational reasons), while still suffering from too much epistemic uncertainty to be analysed. In this case, memory offers an alternative way to increase the size of abstraction, resulting in a potentially better approximation that can be accurately analysed. In this thesis, we will provide multiple examples in which such a situation occurs.

Throughout this thesis, memory-dependent abstractions are analysed with these two motivations in mind. For Motivation 1, we increase the memory length without constraining the size of the resulting abstraction.

For Motivation 2, by contrast, we fix a budget on the abstraction size and therefore compensate for longer memory by considering coarser discretisations of the state space. These motivations are summarised in Figure ⋆.2.

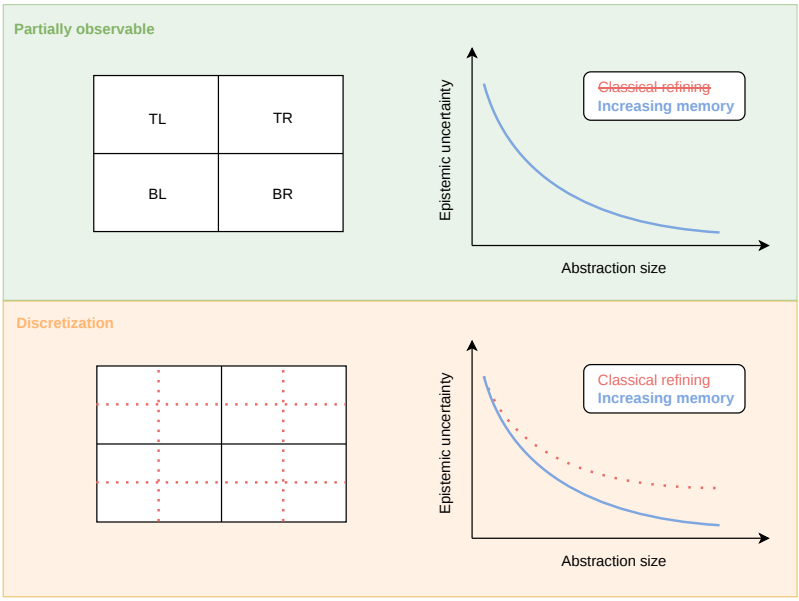


Fig. ⋆.2 Illustration of the two main motivations of memory-dependent abstractions. Above: the partially observable case. In that case, only the region of the state is known (e.g. top-left, top-right, etc.), and one cannot refine the state space. Increasing memory is therefore a way to reduce epistemic uncertainty. Below: the classical framework in which the state space is discretised. In that case, memory provides an alternative way of reducing epistemic uncertainty, possibly better for the same abstraction size.

An alternative motivation of our approach, valid for both partially observable systems and the classical framework, is that our guarantees depend on assumptions other than incremental stability.

This thesis also introduces a novel theoretical tool, the *Cantor-Kantorovich distance* (or *CK distance* for short), as a rigorous means of quantifying differences between labelled Markov chains (*LMCs* for short). This tool was developed in the context of one of the memory-dependent abstraction techniques, described in Chapter 4. Distances between Markov models have

been at the centre of a substantial research effort [DGJP04], with applications ranging from biology [LPN99] to reinforcement learning [GVF22]. Beyond its immediate relevance to our work, the CK distance exhibits compelling theoretical properties and holds potential for broader applications, making it a transversal yet substantial contribution of this thesis.

Related works

The concept of memory has long been used and recognized in control theory, although differently or in other contexts than in this thesis. For instance, dynamical systems exhibiting *hysteresis* – where the current state depends on the history of previous states – have been extensively studied in the literature (see, e.g. [May03]). While *time-invariant* systems states cannot directly depend on past states, their current state can often be reconstructed from output histories, which forms the basis of the classical notion of *observability*. This phenomenon has been thoroughly explored for linear systems, leading to foundational tools such as the *observability matrix* and the *Luenberger observer* [Lue64] – concepts that have since been extended to nonlinear systems [MH89]. Many other examples of how memory can be leveraged exist for nonlinear systems, e.g. for Lyapunov analysis of piecewise linear systems [Lee08]. Beyond deterministic frameworks, memory has also been examined in stochastic settings, particularly within the literature on *partially observable Markov decision processes* (*POMDPs*), where it is known that the current state cannot only be inferred from the last output – a phenomenon known as *non-Markovianity* (see e.g. [GCW+24, CSBD24]). A famous method, called *experience replay* [LWY+23], leverages memory to circumvent non-Markovianity, and has been applied to popular *Reinforcement Learning* (*RL* for short) applications such as Atari playing [MKS+13]. In the RL community, *predictive state representations* [SLJ+03] also leverage memory in POMDPs by representing the state as a vector of predictions over future observation sequences, sharing the intuition underlying our approach, though oriented towards learning and prediction rather than formal verification.

As explained above, this thesis explores the role of memory within abstraction-based approaches for complex systems. However, other techniques have investigated smarter abstraction techniques than the classical state space refinement for circumventing the curse of dimensionality. To mitigate this curse of dimensionality, an alternative to classical state space refinement is *adaptive partitioning* of the state and input spaces. This

targeted refinement keeps the number of abstract states manageable, preserving tractability. For deterministic systems, several scalable abstraction techniques have emerged. For example, *lazy abstractions* [HJMS02, HMMS19] have been at the centre of a research effort recently. Other methods aim at using other objects than hyperrectangles to cover the state space, showcasing better performance in some cases (see e.g. an ellipsoid-based covering in [CEJ24] and another method using zonotopes in [RCJ21]). Other methods include *hierarchical abstractions* [CLEJ21], *state-feedback abstractions* [ELJ22], *compositional approaches* [MGW18], or combinations of these techniques, all of which intend to smarter discretise the state and input spaces. Some smart state space discretisation techniques also exist in the stochastic setting. For example, [EZSA13, LAB15, ZTA16] propose smart refinements of the state/input spaces, and [SMM24] proposes a lazy abstraction algorithm for Markov Decision Processes. To address the loss of Markovianity, the community introduced *Interval Markov Decision Processes* [GLD00, DLMJL23], where transition probabilities are uncertain and lie within intervals. These models preserve the Markov property, enabling safety verification and controller synthesis [DHC22, BLW13, LAB15, LLA+21] with efficient worst-case algorithms [GLD00, NEG05], supported by open-source and fast implementations [MLL24]. However, IMDPs remain conservative and limited due to interval uncertainty and the need to approximate transition kernels. A thorough comparison of our methods with respect to these other smart abstraction methods is left for further work.

Memory has also been explored within abstraction-based methods, although always in frameworks distinct from the ones presented in this thesis. Early work [MR99, MRO02] introduced the concept of ℓ-complete abstractions within the context of *supervisory control*, a model-based control method pioneered by Ramadge and Wonham (see e.g. [RW87, RW89]). The guarantees derived in this thesis differ fundamentally from those in these studies, both in nature and underlying assumptions (more details will be provided in Chapter 3). More recently, ℓ-complete abstractions were extended in [STR15] and applied to chaos detection in [GM23], though still within model-based frameworks. While [PM23, CPM23, Cop26] have examined data-driven ℓ-complete abstractions, they focus on the data-driven side by providing finite-sample guarantees, whereas our focus is on qualitative guarantees on abstraction performance – making these works complementary. Finally, to the best of our knowledge, memory-dependent abstractions have not been studied in the stochastic framework.

Finally, while the Kantorovich metric has been extensively used within

the computer science community [DD09], our application to LMCs with the underlying Cantor distance is novel. Our construction on the metric between Markov chains resembles the one presented in [RAC04], however with another measure. Kantorovich metrics for Markov models have been studied in [DGJP04, vBSW07, MS10, RS18], but their underlying distance is different from ours. In Chapter 6, we will review existing families of metrics between labelled Markov chains, situate the CK distance within this landscape, and highlight its distinctive properties and advantages over prior approaches.

Outline

In the following we review the outline of the thesis, and briefly summarise each chapter. An illustration of the outline and the dependencies between parts and chapters can be found in Figure ⋆.3.

Part I – Background

This part provides the theoretical foundations required by the rest of the manuscript.

Chapter 1: Symbolic dynamics

This chapter presents language-based representation of behaviours over finite alphabets. It formalises the natural topology on infinite words via the Cantor distance, and introduces temporal-logic specifications. Second, it defines labelled Markov chains (or LMCs for short), which constitute the main abstraction object used in this thesis.

Chapter 2: Dynamical systems

This chapter defines the deterministic and stochastic dynamical systems considered in Chapter 3, Chapter 4 and Chapter 5. More specifically, it puts the emphasis on the partially observable framework, which models both motivations described above and depicted in Figure ⋆.2. It formalises the output partition of the state space and discusses the induced loss of the Markov property on observations, which motivates the use of memory-dependent abstractions.

Part II – Abstractions of deterministic systems

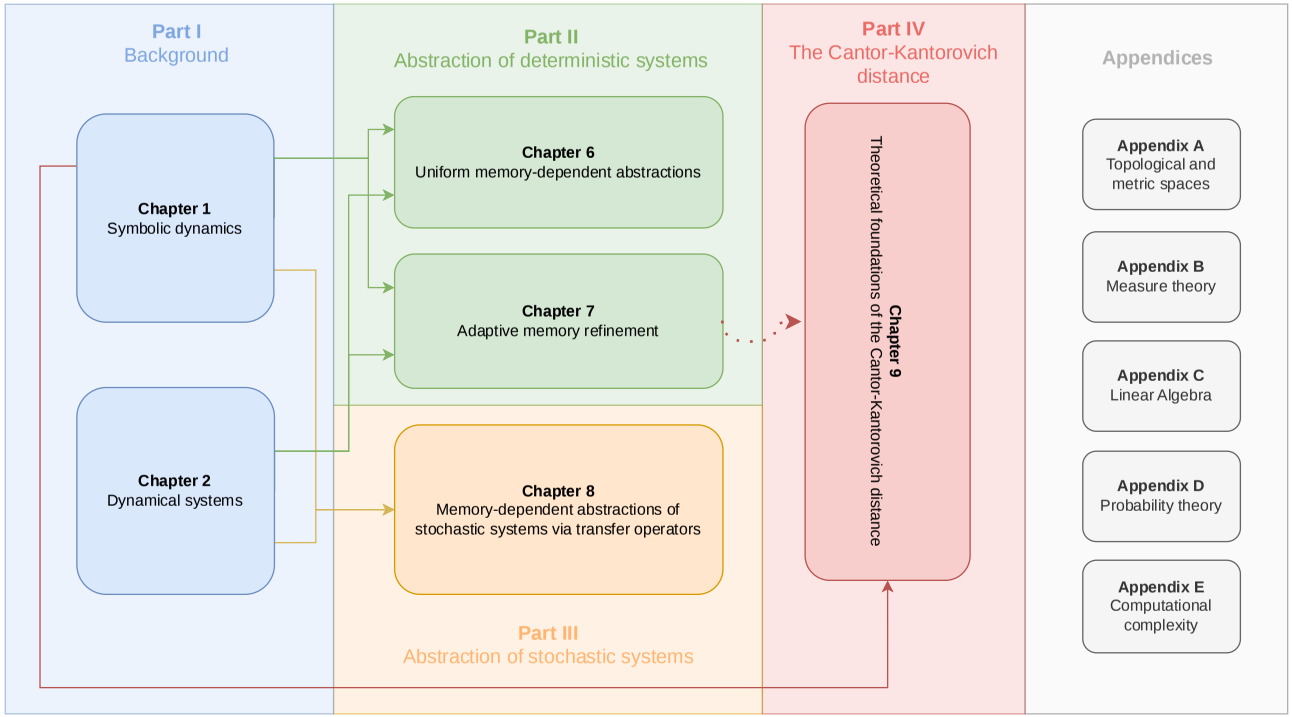


Fig. ⋆.3 Part I establishes the foundational background. Parts II and III present the core contributions: memory-dependent abstraction techniques for deterministic and stochastic systems, respectively. Part IV follows-up on the techniques developed in Chapter 4 by introducing the Cantor-Kantorovich metric, a transversal yet substantial contribution of this thesis. Finally, the appendices collect classical background concepts and results used throughout the manuscript, included for completeness so that the main chapters can focus on the novel contributions of this thesis.

This section presents the first core contribution of the thesis: the development of memory-dependent abstraction techniques for deterministic systems. It introduces two complementary strategies: a uniform-memory construction with asymptotic guarantees, and an adaptive-memory refinement strategy that improves precision while controlling model size.

Chapter 3: Uniform memory-dependent abstractions

This chapter introduces ℓ-memory Markov models whose states encode fixed-length output sequences. A data-driven construction method is proposed and analysed, and the main theoretical result shows asymptotic correctness as memory grows, together with monotonic improvement of the approximation quality.

Chapter 4: Adaptive memory refinement

This chapter extends the uniform approach to adaptive memory schemes, where memory length depends on the local dynamics. It introduces adaptive-memory Markov models based on timed sequences, proves behavioural inclusion under suitable assumptions, and proposes a refinement algorithm that iteratively improves abstraction accuracy.

Part III – Abstractions of stochastic systems

This part extends memory-dependent abstractions to partially observable stochastic systems, where both epistemic and aleatoric uncertainty are present. The main challenge is the non-Markovian nature of observation processes, which is mitigated through the increase of memory.

Chapter 5: Memory-dependent abstractions of stochastic systems via transfer operators

This chapter builds ℓ-memory abstractions for stochastic systems and interprets them as Galerkin approximations of a lifted transfer operator. It provides explicit approximation guarantees in Total Variation, and numerical case studies showing that memory can significantly improve abstraction fidelity compared with memoryless constructions.

Part IV – The Cantor-Kantorovich distance

Following-up on Chapter 4, this part introduces the Cantor-Kantorovich distance, a metric for quantifying differences between labelled Markov chains and their probabilistic behaviours over infinite symbolic traces.

Chapter 6: Theoretical foundations of the Cantor-Kantorovich distance

This chapter formalises the Cantor-Kantorovich distance (or CK distance for short) and relates it to the natural topology on symbolic trajectories. It also proves that the CK distance is a discounted sum of Total Variation terms. It then studies key theoretical properties, including continuity results, exact and approximate computation aspects, and associated computational-complexity hardness results.

Appendices

The following appendices collect classical mathematical results and background material, included for completeness so that the main chapters can focus on the novel contributions of this thesis.

Appendix A: Topological and metric spaces

This appendix introduces fundamental concepts of topology and metric spaces. These structures underpin the development of the CK distance (explored in Chapter 6) and provide the topological foundation for the symbolic dynamics framework presented in Chapter 1.

Appendix B: Measure theory

This chapter develops the theory of measure spaces and measures. These foundational concepts are essential for the probabilistic framework used throughout the thesis.

Appendix C: Linear algebra

This appendix develops basic linear algebra notions that are used throughout the thesis. It introduces norms and normed spaces, completeness and Banach spaces, inner products and Hilbert spaces, and then presents $L^p(\mu)$ spaces with a focus on $L^2(\mu)$ and its Hilbert space structure. These concepts are important for manipulating probability density functions for the analysis of memory-dependent abstractions of stochastic systems in Chapter 5.

Appendix D: Probability theory

This chapter reviews central notions within probability theory, including probability measures and densities, Markov kernels, and distances between probability measures. This appendix relies on the measure theory concepts developed in the previous appendix. Probability theory is central to this thesis, as it is involved in all the core contribution chapters, whether for formalising the data-driven framework, for modelling aleatoric uncertainty, or for the development of the CK distance in Chapter 6.

Appendix E: Computational complexity

This chapter recalls the computational-complexity notions needed for complexity analysis of the CK distance introduced in Chapter 6. It reviews decision and counting problems, polynomial-time reductions, and complexity classes such as `P`, `NP`, and `#P`.

List of publications

HERE is the list of journal, conference, tool and abstract papers written during this PhD thesis, including those whose content might not be included in this manuscript. The list also includes technical reports.

Journal papers

[BWJ24] **Adrien Banse**, Zheming Wang, Raphaël M. Jungers. *Learning stability guarantees for constrained switching linear systems from noisy observations*. Nonlinear Analysis: Hybrid Systems, 2024.

[BRAJ25] **Adrien Banse**, Licio Romao, Alessandro Abate, Raphaël M. Jungers. *Data-driven memory-dependent abstractions of dynamical systems via a Cantor-Kantorovich metric*. IEEE Transactions on Automatic Control, 2025.

Conference papers

[BWJ22a] **Adrien Banse**, Zheming Wang, Raphaël M. Jungers. *Black-box stability analysis of hybrid systems with sample-based multiple Lyapunov functions*. In Proceedings of the 61st IEEE Conference on Decision and Control (Cancún, Mexico), 2022.

[BRAJ23b] **Adrien Banse**, Licio Romao, Alessandro Abate, Raphaël M. Jungers. *Data-driven memory-dependent abstractions of dynamical systems*. In Proceedings of the 5th Learning for Dynamics and Control Conference (Philadelphia, USA), 2023.

[BRAJ23a] **Adrien Banse**, Licio Romao, Alessandro Abate, Raphaël M. Jungers. *Data-driven Abstractions via Adaptive Refinements and a Kantorovich Metric*. In Proceedings of the 62nd IEEE Conference on Decision and Control (Singapore), 2023.
IEEE Hybrid Systems TC Outstanding Student Paper Prize.

[BDL+25] **Adrien Banse**, Giannis Delimpaltadakis, Luca Laurenti, Manuel Mazo Jr., Raphaël M. Jungers. *Memory-dependent abstractions of stochastic systems through the lens of transfer operators*. In Proceedings of the 28th ACM International Conference on Hybrid Systems: Computation and Control, 2025.

[JBCJ25b] Adrien Janssens, **Adrien Banse**, Julien Calbert, Raphaël M. Jungers. *Agile Temporal Discretization for Symbolic Optimal Control*. In Proceedings of the 64th IEEE Conference on Decision and Control (Rio de Janeiro, Brazil), 2025.

[NBBJ26] Léa Ninite, **Adrien Banse**, Guillaume Berger, Raphaël M. Jungers. *A path-complete approach for optimal control of switched systems*. Accepted for presentation at the 29th ACM International Conference on Hybrid Systems: Computation and Control, 2026.

Tool papers

[CBLJ24] Julien Calbert, **Adrien Banse**, Benoît Legat, Raphaël M. Jungers. *Dionysos.jl: a Modular Platform for Smart Symbolic Control*. In the JuliaCon Proceedings, 2024.

[RB25] Samuel Rincé, **Adrien Banse**. *EcoLogits: Evaluating the Environmental Impacts of Generative AI*. The Journal of Open Source Software, 2025.

Abstracts

[BWJ22b] **Adrien Banse**, Zheming Wang, Raphaël M. Jungers. *Learning stability guarantees for data-driven constrained switching linear sys-*

tems. 41st Benelux Meeting on Systems and Control (Brussels, Belgium), 2022.

[BWJ22c] **Adrien Banse**, Zheming Wang, Raphaël M. Jungers. Learning stability guarantees for data-driven constrained switching linear systems. 25th International Symposium on Mathematical Theory of Networks and Systems (Bayreuth, Germany), 2022.

[BRAJ24] **Adrien Banse**, Licio Romao, Alessandro Abate, Raphaël M. Jungers. *Smart abstraction of dynamical systems via a Cantor-Kantorovich distance between Markov chains*. 43rd Benelux Meeting on Systems and Control (Blankenberge, Belgium), 2024.

[BRJ24] **Adrien Banse**, Venkatraman Renganathan, Raphaël M. Jungers. *A Cantor-Kantorovich Metric Between Markov Decision Processes with Application to Transfer Learning*. 26th International Symposium on Mathematical Theory of Networks and Systems (Cambridge, UK), 2024.

[BDMJJ25] **Adrien Banse**, Giannis Delimpaltadakis, Luca Laurenti, Manuel Mazo Jr., Raphaël M. Jungers. *Smart abstraction of stochastic systems using memory*. 44th Benelux Meeting on Systems and Control (Egmond aan Zee, Netherlands), 2025.

[JBCJ25a] Adrien Janssens, **Adrien Banse**, Julien Calbert, Raphaël M. Jungers. *Abstractions with Adaptive Time-step for Reachability Analysis*. 44th Benelux Meeting on Systems and Control (Egmond aan Zee, Netherlands), 2025.

Technical reports

[BAJ25] **Adrien Banse**, Alessandro Abate, Raphaël M. Jungers. *Comparing Labeled Markov Chains: A Cantor-Kantorovich Approach*. Under review.

[FBH+26] Eduardo Figueiredo, **Adrien Banse**, Qi Heng Ho, Raphaël M. Jungers, Morteza Lahijanian, Luca Laurenti. *Learn-To-Propagate: Distributionally Robust Motion Planning for Nonlinear Uncertain Systems*. Under review.

List of symbols

Spaces and measures

$\mathbb{N}$	Set of non-negative integers
$\mathbb{R}$	Real numbers
$\mathbb{R}_{\geq 0}$	Non-negative real numbers
$\emptyset$	Empty set
X	Abstract set or vector space
$\mathcal{T}$	Topology
$(X, \mathcal{T})$	Topological space
$\bigotimes_{i \in I} \mathcal{T}_i$	Product topology
D	Abstract distance or ultradistance
(X, D)	Metric or ultrametric space
$B(x; \varepsilon)$	Open ball of center x and radius ε
$\mathcal{A}$	Algebra or σ-algebra
μ	Pre-measure or measure
$(X, \mathcal{A})$	Measurable space
$\mathcal{B}$	Topology or σ-algebra basis

$\sigma(\mathcal{T})$	Borel σ-algebra
2^X	Power set
$f : X \to Y$	Function
$f^{-1}(B)$	Pre-image of B
$\nu \ll \mu$	The measure ν is absolutely continuous with respect to μ
$\mathrm{d}\nu/\mathrm{d}\mu$	Radon-Nikodym derivative
$f\#\mu$	Pushforward measure by a function
τ	Transition kernel
$\tau\#\mu$	Pushforward measure by a kernel
$\|\cdot\|$	Norm
$(X, \|\cdot\|)$	Normed or Banach space
$\langle\cdot,\cdot\rangle$	Inner product
$(X, \langle\cdot,\cdot\rangle)$	Inner product or Hilbert space
$X^\perp$	Orthogonal complement
$L^p(\mu)$	Space of integrable functions
$\|\cdot\|_p$	Functions p-norm

Probability theory

μ	Probability measure
$(X, \mathcal{A}, \mu)$	Probability space
x	Random variable
$\mathbb{P}(\mathsf{x} \in A)$	Probability that x belongs to A
γ	Coupling
$\mathcal{C}(\mu, \nu)$	Set of all couplings of the probability measures μ and ν
$\mathbb{P}(\mathsf{x} \in A, \mathsf{y} \in B)$	Joint probability
$\mathbb{P}(\mathsf{x} \in A \vert \mathsf{y} \in B)$	Conditional probability
$\mathcal{P}(X, \mathcal{A})$	Set of all probability measures on $(X, \mathcal{A})$
TV	Total Variation distance
K_D	D-Kantorovich distance

Symbolic dynamics

A	Alphabet or labels
$w := a_1 \dots a_k$	Word
A^k	k-long language
$A^\star$	Finite language or Kleene closure
A^ω	Infinite language
$w_1 w_2$	Concatenation of w_1 and w_2
$\mathrm{pref}(w_1)$	Set of all finite prefixes of w
T^ω	Natural topology
wA^ω	Cylinder set
$\mathcal{C}$	Set of all cylinder sets
$\mathsf{C} := \mathsf{C}_\ell$	Cantor distance
LTL	Linear Temporal Logic
B-LTL	Bounded LTL
ϕ	(B-)LTL formula
`true`	Logical true
$\wedge$	Logical conjunction
$\neg$	Logical negation
$\bigcirc$	"Next" temporal modality
$\mathcal{U}$	"Until" temporal modality
ϕ	Property (language)
Φ	Class of properties
LMC	Labelled Markov chain
Γ	LMC
S	LMC states
ι	LMC initial distribution
τ	LMC transition kernel
l	LMC labeling function
$[\![a]\!]$	Equivalence class of label a
μ^k	Probability measure generated by LMC on $(A^k, 2^{A^k})$
μ^ω	Probability measure generated by LMC on $(A^\omega, \sigma(\mathcal{T}^\omega))$

Dynamical systems

Σ	(Partially observable) dynamical system
$X \subseteq \mathbb{R}^d$	State space
$\mathcal{T}_X$	State space topology
$\sigma(X) := \sigma(\mathcal{T}_X)$	State space σ-algebra
x_k	State at time k
f	Transition function
A	Output space
y_k	Output at time k
h	Output function
$[\![a]\!]$	Equivalence class of output a
τ	Transition kernel
λ_k	State probability measure at time k
x_k	State random variable at time k
w_k	Noise at time k
w	Noise random variable
$\mathbf{A} \in \mathbb{R}^{n \times n}$	Linear system transition matrix
$\mathcal{N}(\mathbf{m}, \mathbf{C})$	Gaussian law with mean $\mathbf{m}$ and covariance $\mathbf{C}$
$\mu_{\mathcal{N}(\mathbf{m},\mathbf{C})}$	Probability measure associated with Gaussian law
ν_k	Output probability measure at time k
y_k	Output random variable at time k

Deterministic abstractions

ℓ	Memory length
λ_k^ℓ	State sequence probability from time k to $k + \ell - 1$
ν_k^ℓ	Output sequence probability from time k to $k + \ell - 1$
$\mathcal{B}_k(\Sigma, \lambda_0)$	k-long behaviour of system Σ sampled with λ_0
$\mathcal{B}(\Sigma, \lambda_0)$	Behaviour of system Σ sampled with λ_0
Γ_ℓ	ℓ-memory Markov model/abstraction
S_ℓ	ℓ-memory Markov model states

ι_ℓ	ℓ-memory Markov model initial distribution
τ_ℓ	ℓ-memory Markov model transition kernel
l_ℓ	ℓ-memory Markov model labeling function
$\mathcal{B}_k(\Gamma_\ell)$	k-long behaviour of ℓ-memory Markov model
$\mathcal{B}(\Gamma_\ell)$	Behaviour of ℓ-memory Markov model
F_μ^N	Frequency from dataset of N observations sampled from μ
F_μ	Frequency from dataset of an infinity of observations sampled from μ
ε	Perturbation magnitude
W_ε	Perturbation space
$\tilde{\Sigma}_\varepsilon$	Perturbation of system Σ
$\tilde{\lambda}_k^\ell$	Perturbation state sequence probability from k to $k+\ell-1$
$\tilde{\nu}_k^\ell$	Perturbation output sequence probability from k to $k+\ell-1$
Δ_k^ℓ	Spurious k-long trajectories given by ℓ-memory abstraction
$w_{[-s,t]}$	Timed sequence from $-s$ to t
$[\![w_{[-s,t]}]\!]$	Block of states whose output history matches $w_{[-s,t]}$
W	Set of timed sequences that define an adaptive memory scheme partition
Γ_W	Adaptive memory Markov model/abstraction
S_W	Adaptive memory Markov model states
ι_W	Adaptive memory Markov model initial distribution
τ_W	Adaptive memory Markov model transition kernel
l_W	Adaptive memory Markov model labeling function
$\mathcal{B}_k(\Gamma_W)$	k-long behaviour of adaptive memory Markov model
$\mathcal{B}(\Gamma_W)$	Behaviour of adaptive memory Markov model
REFINE	Adaptive memory abstraction refinement algorithm
Δ_k^W	Spurious k-long trajectories given by adaptive memory abstraction

Stochastic abstractions

T	Transfer operator
$\mathbb{1}$	Unit function
Q	Projection operator
$P := QTQ$	Galerkin approximation
$\mathbf{a}$	Vector representation of an element a
$\mathbf{P}$	Matrix representation of an operator
Σ^ℓ	Lifted system
X^ℓ	Lifted system state space
τ^ℓ	Lifted system transition kernel
A^ℓ	Lifted system alphabet
h^ℓ	Lifted system output function
μ^ℓ	Lifted system invariant measure
λ_k^ℓ	Lifted system joint state distribution from k to $k+\ell-1$
f_k^ℓ	Density of λ_k^ℓ with respect to μ^ℓ
T_ℓ	Lifted transfer operator
$\psi_{a_1,\dots,a_\ell}$	Basis function associated with output sequence $(a_1,\dots,a_\ell)$
D^ℓ	Finite-dimensional subspace of $L^2(\mu^\ell)$ spanned by basis functions $\psi_{a_1,\dots,a_\ell}$
Q_ℓ	Orthogonal projection onto D^ℓ.
$P_\ell := Q_\ell T_\ell Q_\ell$	Galerkin approximation of T_ℓ in D^ℓ
$\mathbf{P}_\ell$	Matrix representation of P_ℓ
$\tilde{\lambda}_k^\ell$	ℓ-memory approximation of the joint measure λ_k^ℓ
$\tilde{f}_k^\ell$	ℓ-memory approximation of the joint density f_k^ℓ
$\tilde{\lambda}^{\ell,k}$	ℓ-memory approximation of the joint measure $\lambda_{\ell,k}$
$\tilde{f}_{\ell,k}$	ℓ-memory approximation of the density f_k
$\overline{\mathsf{TV}}_{\mathrm{inc}}(k)$	Increasing bound on the TV error $\mathsf{TV}(\lambda_k, \tilde{\lambda}_{\ell,k})$
$\overline{\mathsf{TV}}_{\mathrm{dec}}(k)$	Decreasing bound on the TV error $\mathsf{TV}(\lambda_k, \tilde{\lambda}_{\ell,k})$
$e_{\ell,i}$	Eigenvalues of T_ℓ, ordered by magnitude
$g_{\ell,i}$	Eigenfunctions of T_ℓ
δ_ℓ	Maximal projection error on the spectrum

Computational complexity

P	Problem
I	Instances
S	Solutions
$e(\cdot)$	Encoding of an input instance (e.g. binary)
A	Algorithm
$\mathcal{O}(k^c)$	Polynomial-time
`P`	Complexity class of polynomial problems
V	Verification algorithm
`NP`	Complexity class of polynomial nondeterministic polynomial problems
r	Reduction function
`#P`	Complexity class of counting problems associated with `NP`

Cantor-Kantorovich distance

m	Number of labels
$\mathsf{K}_\mathsf{C}(\mu_1^k, \mu_2^k)$	Kantorovich distance with underlying Cantor metric between μ_1^k and μ_2^k
CK	Cantor-Kantorovich distance
c^k	Coupling when solving $\mathsf{K}_\mathsf{C}(\mu_1^k, \mu_2^k)$
M_k	Quantity $M_k = \sum_{w \in A^i} \min\{\mu_1^k(w), \mu_2^k(w)\}$
S_k	Truncated sum approximation
π	Product distribution on $\{0,1\}^k$ defined by parameters $(p_i)_{i=1}^k$
$f_{\mathsf{TV,prod}}$	Counting function encoding TV distance for product distributions
f_{CK}	Function mapping encodings of two LMCs to their CK distance
f_{S_k}	Function mapping encodings of two LMCs to truncated sum S_k
p_i, q_i	Bernoulli parameters defining product distributions π_1, π_2

M	Common denominator factor used in the encoding of $\mathsf{TV}(\pi_1, \pi_2)$ as a counting problem

PART I
Background

1

Symbolic dynamics

THIS chapter introduces *symbolic dynamics*, a framework that encodes the behaviour of dynamical systems as sequences of symbols drawn from a finite alphabet. In this chapter we formalise the structure of symbolic sequences and their *natural topology*, arising from the *Cantor distance*. We also explore specifications defined via Linear Temporal Logic (LTL) and Bounded LTL (B-LTL). We conclude by defining *labelled Markov chains* (or *LMCs* for short). LMCs will be a central tool in this thesis, as it will be the main tool that we use for abstracting dynamical systems. From a more abstract perspective, LMCs extend symbolic dynamics to probabilistic systems. We will study the question of *comparing* LMCs (both qualitatively and quantitatively) via *distances*, *probabilistic bisimilarity relations*, and *trace equivalence*. We will conclude by introducing *continuity* notions for distances between LMCs.

1.1 Language theory

In this first section we formally define languages, i.e. sets of words composed from a finite alphabet. We also study the structure of the infinite language, endowed with the *natural topology* (see Appendix A for an introduction to topological and metric spaces). We conclude by defining the Cantor distance, which generates the natural topology. This section is mostly based on the textbooks [LM95, PP04].

1.1.1 Definition

An *alphabet* A is defined as a finite set of *symbols*. A *word* of length $k \in \mathbb{N} \cup \{+\infty\}$ is a sequence $w := a_1 \dots a_k$, with $a_i \in A$ for $i = 1, \dots, k$.

Definition 1.1 (Language)**.** Let A be an alphabet.

- For any $k \in \mathbb{N} \cup \{0\}$, the *k-long language* is noted A^k, and contains all words of length k formed from A.
- The *finite language* (or *Kleene closure*) is noted $A^\star$, and is defined as

$$A^\star = \bigcup_{k \geq 0} A^k.$$

- The *infinite language* is noted A^ω, and contains all words of infinite length formed from A.

Note that any subset of such languages is also called a language.

We conclude this section by defining the two following concepts, which will be useful for further definitions.

Definition 1.2 (Concatenation)**.** Let A be an alphabet, and let $w_1 \in A^\star$, and $w_2 \in A^\star \cup A^\omega$. Suppose $w_1 := a_1 \dots a_{k_1}$ and $w_2 := b_1 \dots b_{k_2}$, the *concatenation* of w_1 and w_2 is defined as $w_1 w_2 := a_1 \dots a_{k_1} b_1 \dots b_{k_2}$.

Definition 1.3 (Prefix)**.** Let A be an alphabet, and let $w_1 \in A^\star \cup A^\omega$. A finite word $w_2 \in A^\star$ is a *prefix* of w_1 if there exists a word $w_3 \in A^\star \cup A^\omega$ such that $w_1 = w_2 w_3$. The language $\mathrm{pref}(w_1)$ is defined as the set of all finite prefixes of w_1.

1.1.2 The natural topology

In this section, we study the properties of the so-called natural topology on A^ω, defined as follows. This definition is based on the notions of *discrete* and *product* topologies, defined in Appendix A.

Definition 1.4 (Natural topology)**.** Let A be an alphabet, and let $\mathcal{T}$ be the discrete topology for A. Then the product topology

$$\mathcal{T}^\omega := \bigotimes_{i \in \mathbb{N}} \mathcal{T}$$

is called the natural topology for the language A^ω.

The term natural topology arises from the fact that taking the product of discrete topologies is the most intuitive and direct way of defining a topology on A^ω. Moreover, this topology exhibits many interesting properties, e.g. it is the largest topology such that the shift map defined as

$$s : A^\omega \to A^\omega : a_1 a_2 a_3 \ldots \cdot \cdot \cdot \mapsto a_2 a_3 a_4 \ldots$$

is continuous.

In the following, we will investigate two alternative definitions for the natural topology, which are

1. it arises from the set of *cylinder sets*, and
2. it is metrizable and generated by the Cantor distance.

Definition 1.5 (Cylinder set)**.** Let $w \in A^\star$. The set

$$wA^\omega := \{w' \in A^\omega : w \in \mathrm{pref}(w')\}$$

is called a *cylinder set*. The set of all cylinder sets is denoted by $\mathcal{C}$.

Proposition 1.6 ([PP04, Proposition 3.1])**.** *Let A be an alphabet. The collection of all cylinder sets form a basis for the natural topology.*

Proposition 1.6 implies that the natural topology is the smallest topology that contains all the cylinder sets (see Appendix A for a formal definition of *basis*). Cylinder sets are the simplest open sets one can define for infinite words, reinforcing the *natural* characteristic of the topology.

Definition 1.7 (Cantor distance)**.** Let A be an alphabet. For any $\ell > 1$, the function $\mathsf{C}_\ell : A^\omega \times A^\omega \to \mathbb{R}_{\geq 0}$ defined as, for any two words $w_1 = a_1 a_2 \ldots$ and $w_2 = b_1 b_2 \ldots$,

$$\mathsf{C}_\ell(w_1, w_2) := \ell^{-\inf\{i-1 : a_i \neq b_i\}},$$

is called the Cantor distance.

The Cantor distance is entirely characterised by the length of the common prefix between two words. Note that this distance extends naturally to the same notion for k-long languages A^k and for the finite language $A^\star$.

Proposition 1.8 ([PP04, Proposition 3.2])**.** *For all $\ell > 1$, the Cantor distance C_ℓ induces the natural topology.*

Proposition 1.8 implies that the open balls

$$B(w;\varepsilon) := \{w' \in A^\omega : \mathsf{C}_\ell(w,w') < \varepsilon\}$$

generate the natural topology, and therefore that this topology is *metrizable* (see Appendix A for a formal definition of *metrizable topology*).

Remark 1.9. As shown by Proposition 1.8, the choice of the scalar ℓ is arbitrary since all such distances induce the same topology on the language A^ω. This also explains the name Cantor distance, without any dependency on ℓ. In this thesis, for brevity of the involved mathematical expressions, we will consider $\ell = |A| = m$ such as in [Fog02], although $\ell = 2$ is a common choice in the literature (see e.g. [PP04]). From now on, we will simply refer to C_ℓ as C.

To conclude, we highlight the fact that the Cantor distance defines an ultrametric space[1], i.e. that the Cantor distance satisfies the *strong triangular inequality*

$$\mathsf{C}_\ell(w_1,w_3) \leq \max\{\mathsf{C}_\ell(w_1,w_2), \mathsf{C}_\ell(w_2,w_3)\}.$$

A formal definition of *ultradistance* can be found in Appendix A.

Proposition 1.10. *The Cantor distance* C *is an ultradistance.*

Proof. Let $w_1, w_2, w_3 \in A^\omega$. Suppose that $\mathsf{C}(w_1,w_2) = \ell^{-i}$ and $\mathsf{C}(w_2,w_3) = \ell^{-j}$ for two integers i and j, and let $k := \min(i,j)$. It means that w_1 and w_3 are equal up to at least index k, implying $\mathsf{C}(w_1,w_3) \leq \ell^{-k} = \max(\ell^{-i}, \ell^{-j})$. This yields $\mathsf{C}(w_1,w_3) \leq \max(\mathsf{C}(w_1,w_2), \mathsf{C}(w_2,w_3))$, which concludes the proof. □

1.1.3 Specifications

In this section we introduce two families of languages, called the LTL and B-LTL classes of properties. This section is based on [BYAG17, BK08, JMGLM14].

We first introduce the *linear temporal logic* (or *LTL* for short) syntax.

Definition 1.11 (LTL syntax)**.** Given an alphabet A, an *LTL formula* φ over A is recursively defined as

$$\varphi ::= \mathtt{true} \mid a \mid \varphi_1 \wedge \varphi_2 \mid \neg\varphi_1 \mid \bigcirc \varphi_1 \mid \varphi_1 \mathcal{U} \varphi_2,$$

[1] We do not claim that this result is novel, although we provide a proof.

where $a \in A$, φ_1, φ_2 are LTL formulas, $\wedge$ is the *conjunction*, $\neg$ is the *negation*, and $\bigcirc$ and $\mathcal{U}$ are respectively the "next" and the "until" temporal modalities.

LTL syntax is very useful to describe properties (e.g. of dynamical systems) over infinite traces (see e.g. [BYAG17] for examples and further explanation).

The following makes the link between the syntax and the *property* (i.e. the language) it defines.

Definition 1.12 (LTL-definable property)**.** Given an alphabet A, a language $\phi \subseteq A^\omega$ is said to be an *LTL-definable property* if there exists an LTL formula φ such that $\phi = \{w \in A^\omega : w \models \varphi\}$.

The set of languages definable by the LTL syntax is called the *LTL class of properties*, and is defined as follows.

Definition 1.13 (LTL class of properties)**.** The LTL class of properties is defined as

$$\Phi_{\text{LTL}} := \{\phi \subseteq A^\omega : \phi \text{ is a LTL-definable property}\}.$$

In this thesis (and particularly in Chapter 6), we will focus on a particular subset of LTL-definable properties, defined by the so-called *bounded LTL* syntax (or *B-LTL* for short), defined as follows.

Definition 1.14 (B-LTL syntax)**.** Given a finite alphabet A, a *B-LTL formula* φ over A is recursively defined similarly as an LTL formula, however omitting the $\mathcal{U}$ temporal modality.

The notions of *(B-LTL)-definable property* and *B-LTL class of properties* are defined similarly as the notions of LTL-definable and LTL class of properties, respectively. The B-LTL class of properties is denoted as $\Phi_{\text{B-LTL}}$. Following the Definition 1.14, it holds that

$$\Phi_{\text{B-LTL}} \subseteq \Phi_{\text{LTL}}.$$

Remark 1.15. The B-LTL class of properties is different from the *sc-LTL* class of properties (*syntactically co-safe LTL*, see e.g. [BYAG17, Definition 2.3]) and LTL_f (LTL over finite traces [DGV13]). Indeed B-LTL specifications require a bounded lookahead (that one can encode only with "next" operators) whereas both sc-LTL and LTL_f specifications require unbounded lookahead.

B-LTL properties have as central property that they can be written as union of cylinder sets.

Proposition 1.16 ([JMGLM14])**.** *Any (B-LTL)-definable property $\phi \subseteq A^\omega$ is a finite union of cylinder sets.*

This property will be useful for our purpose.

1.2 Labelled Markov chains

Labelled Markov Chains (or *LMC*s for short) are Markov chains whose states are *labelled* by symbols that belong to a finite alphabet A. LMCs are flexible and powerful tools to describe distributions over finite and infinite words, denoted respectively as $A^\star$ and A^ω, making them particularly useful in many applications ranging from computational biology [DEKM98, Chu89, SS10, KLvHS01] to speech/gesture recognition [Rab89, CFH03]. More recently, LMCs have been used in the context of *abstraction* of continuous stochastic systems into simpler discrete models, for controller design and verification goals [LSAZ22].

In this section, we define such objects, and we develop notions that allow us to *compare* two LMCs, both qualitatively and quantitatively. These objects are central in this thesis since

1. they will be used as *abstractions* of (both deterministic and stochastic) dynamical systems in Chapter 3, Chapter 4 and Chapter 5;
2. we will develop a novel *distance between LMCs* in Chapter 6.

1.2.1 Definition

We start by defining LMCs. Note that all subsequent definitions rely on notions from *measure* and *probability* theories, recalled in Appendix B and Appendix D, respectively.

Definition 1.17 (Labelled Markov chain)**.** A *labelled Markov chain* (or LMC for short) is a tuple $\Gamma = (S, A, \iota, \tau, l)$, where

- S is the finite set of *states*,
- A is a finite alphabet called the set of *labels*,
- $\iota : 2^S \to [0,1]$ is a probability measure on the measurable space $(S, 2^S)$ called the *initial measure*,

- $\tau : 2^S \times S \to [0,1]$ is a Markov kernel from $(S, 2^S)$ to itself called the *transition kernel*,
- $l : S \to A$ is a function called the *labelling function*.

Note that, in this thesis, we restrict to discrete and finite state spaces Markov chains. In order to lighten the notations, for all $s_1, s_2 \in S$, we will write $\iota(s) := \iota(\{s\})$ and $\tau(s_2|s_1) := \tau(\{s_2\}|s_1)$.

As highlighted in Definition 1.17, LMCs are Markov chains (as defined in e.g. [Nor97]), but whose nodes are labelled with a symbol belonging to a finite alphabet. An illustration of a LMC is given in Figure 1.1. These labels define *equivalence classes*, defined as follows.

Definition 1.18 (Label equivalence class)**.** Let $\Gamma = (S, A, \iota, \tau, l)$ be a LMC. For all $a \in A$, the set

$$[\![a]\!] := \{s \in S : l(s) = a\}$$

is called an *equivalence class* of the label a.

In the LMC depicted in Figure 1.1, the equivalence classes are $[\![1]\!] = \{s_1\}$ and $[\![2]\!] = \{s_2, s_3\}$.

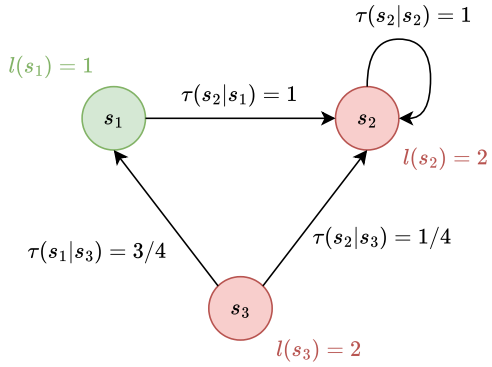


Fig. 1.1 An example of three states, two labels LMC. This LMC is defined as $\Gamma = (S, A, \iota, \tau, l)$ with $S = \{s_1, s_2, s_3\}$ and $A = \{1, 2\}$. The node s such that $l(s) = 1$ is in green, and the nodes s such that $l(s) = 2$ are in red.

A key property of LMCs is that they generate a probability measure on

the languages A^k (for all $k \in \mathbb{N}$), as well as for A^ω.[2] The finite languages A^k are finite, and are therefore endowed with the power set 2^{A^k}, and the infinite language A^ω is endowed with the Borel σ-algebra $\sigma(\mathcal{T}^\omega)$, where $\mathcal{T}^\omega$ is the natural topology such as defined in Definition 1.4. The corresponding *measurable spaces* are noted $(A^k, 2^{A^k})$ and $(A^\omega, \sigma(\mathcal{T}^\omega))$ (see Appendix B for a formal definition of measurable space). It can be shown that, similarly as for the topology, $\sigma(\mathcal{T}^\omega)$ is the σ-algebra whose basis is the set of cylinder sets. The two following propositions define such generated measures.[3]

Proposition 1.19. *Let $\Gamma = (S, A, \iota, \tau, l)$ be a LMC, and let $k \in \mathbb{N}$. Let the function $\mu^k : 2^{A^k} \to [0,1]$ be defined as, for all $w = a_1 \dots a_k \in A^k$,*

$$\mu^k(w) := \sum_{s_1 \in [\![a_1]\!]} \iota(s_1) \sum_{s_2 \in [\![a_2]\!]} \tau(s_2|s_1) \cdots \sum_{s_k \in [\![a_k]\!]} \tau(s_k|s_{k-1}).$$

Then μ^k is a probability measure on $(A^k, 2^{A^k})$.

Proof. Following Remark D.2, it suffices to show that $\mu^k(w) \geq 0$ for all $w \in A^k$, and that $\mu^k(A^k) = 1$. The first follows trivially from the definition ι and τ. We prove the second by recurrence. First, note that

$$\begin{aligned}
\mu^k(A^k) &= \sum_{a_1 \in A} \cdots \sum_{a_k \in A} \mu^k(a_1 \dots a_k) \\
&= \sum_{a_1 \in A} \cdots \sum_{a_k \in A} \sum_{s_1 \in [\![a_1]\!]} \iota(s_1) \sum_{s_2 \in [\![a_2]\!]} \tau(s_2|s_1) \cdots \sum_{s_k \in [\![a_k]\!]} \tau(s_k|s_{k-1}) \\
&= \sum_{a_1 \in A} \sum_{s_1 \in [\![a_1]\!]} \iota(s_1) \sum_{s_2 \in [\![a_2]\!]} \tau(s_2|s_1) \cdots \sum_{a_k \in A} \sum_{s_k \in [\![a_k]\!]} \tau(s_k|s_{k-1}) \\
&= \sum_{s_1 \in S} \iota(s_1) \sum_{s_2 \in S} \tau(s_2|s_1) \cdots \sum_{s_k \in S} \tau(s_k|s_{k-1}).
\end{aligned}$$

The initial case $\mu^1(A^1) = \sum_{s_1 \in S} \iota(s_1) = 1$ directly follows from the definition of ι. Now, for any $k \in \mathbb{N}$, assume that $\mu^k(A^k) = 1$. Then it holds

[2] According to Remark D.2, for the finite language A^k it suffices to define this probability measure on singletons $w \in A^k$.

[3] Similarly as above, while we do not claim that these propositions are novel, we include their proof for completeness as we were unable to locate them in the existing literature.

that

$$\begin{aligned}\mu^{k+1}(A^{k+1}) &= \sum_{s_1\in S}\iota(s_1)\sum_{s_2\in S}\tau(s_2|s_1)\cdots\sum_{s_k\in S}\tau(s_k|s_{k-1})\sum_{s_{k+1}\in S}\tau(s_{k+1}|s_k)\\ &= \sum_{s_1\in S}\iota(s_1)\sum_{s_2\in S}\tau(s_2|s_1)\cdots\sum_{s_k\in S}\tau(s_k|s_{k-1})\\ &= \mu^k(A^k)\\ &= 1,\end{aligned}$$

where the second equality follows from the definition of τ. □

Coming back to the example depicted in Figure 1.1, with $\iota(s_1) = \iota(s_2) = \iota(s_3) = 1/3$, it holds that

$$\mu^1(1) = \iota(s_1) = 1/3 \text{ and } \mu^1(2) = \iota(s_2) + \iota(s_3) = 2/3,$$

for $k = 1$, and

$$\begin{aligned}&\mu^2(11) = \iota(s_1)\tau(s_1|s_1) = 0,\\ &\mu^2(12) = \iota(s_1)(\tau(s_2|s_1) + \tau(s_3|s_1)) = 1/3,\\ &\mu^2(21) = \iota(s_2)\tau(s_1|s_2) + \iota(s_3)\tau(s_1|s_3) = 1/4,\\ &\mu^2(22) = \iota(s_2)(\tau(s_2|s_2) + \tau(s_3|s_2)) + \iota(s_3)(\tau(s_2|s_3) + \tau(s_3|s_3)) = 5/12.\end{aligned}$$

for $k = 2$. One can indeed verify that, for a fixed k, the values above sum to 1.

Proposition 1.20. *Let $\Gamma = (S, A, \iota, \tau, l)$ be a LMC. Let the function $\mu^\omega : \sigma(\mathcal{T}^\omega) \to [0,1]$ be defined as, for any $k \in \mathbb{N}$, for all $w \in A^k$,*

$$\mu^\omega(wA^\omega) := \mu^k(w)$$

and extended to any subset $W \in \sigma(\mathcal{T}^\omega)$ in the sense of the Carathéodory's extension theorem (see Theorem B.11). Then μ^ω is a probability measure on $(A^\omega, \sigma(\mathcal{T}^\omega))$.

Proof. First, we prove that μ^ω is a measure. The latter follows from the Carathéodory's extension theorem. Indeed, the function $\mu_0 : \mathcal{C} \to \mathbb{R} : wA^\omega \mapsto \mu^{|w|}(w)$ is a finite pre-measure on the algebra $\mathcal{C}$, implying that μ^ω is a measure, and is uniquely defined. It remains to show that $\mu^\omega(A^\omega) = 1$. Let k be any natural number, we know that the collection $\{wA^\omega\}_{w\in A^k}$

contains only disjoint sets, and is such that $\bigcup_{w\in A^k} wA^\omega = A^\omega$. Therefore, it holds that

$$\mu^\omega(A^\omega) = \mu^\omega\left(\bigcup_{w\in A^k} wA^\omega\right) = \sum_{w\in A^k} \mu^\omega(wA^\omega) = \sum_{w\in A^k} \mu^k(w) = 1,$$

where the last equality follows from the fact that μ^k is a probability measure. Finally, because μ^k are uniquely defined and finite, then μ^ω is uniquely defined by Theorem B.11, which concludes the proof. □

1.2.2 Qualitative and quantitative comparison of LMCs

In this section we focus on notions that allow to compare two labelled Markov chains. These notions will be crucial in the development of the *Cantor-Kantorovich distance* in Chapter 6. We will start by defining both *bisimilarity* and *trace-based distances* between LMCs. We will conclude by introducing notions of continuity of such distances.

Remark 1.21. From now on, we will make the assumption that every time we compare two LMCs Γ_1 and Γ_2, they must be defined over the same alphabet A. However, if not the case, one can consider a joint alphabet $A = A_1 \cup A_2$, and define probability distributions accordingly by setting a zero probability to any trace containing symbols in $A \setminus A_1$ (resp. $A \setminus A_2$) for Γ_1 (resp. for Γ_2).

We start by defining the common foundation of both *trace-based* and *bisimilarity distances*.

Definition 1.22 (Pseudo-distance between LMCs)**.** Let $\mathcal{M}$ be the set of all LMCs. A function $\mathsf{D} : \mathcal{M} \times \mathcal{M} \to \mathbb{R}_{\geq 0}$ is said to be a *pseudo-distance* if

1. for any LMC $\Gamma \in \mathcal{M}$, $\mathsf{D}(\Gamma, \Gamma) = 0$,
2. for any two LMCs $\Gamma_1, \Gamma_2 \in \mathcal{M}$, $\mathsf{D}(\Gamma_1, \Gamma_2) = \mathsf{D}(\Gamma_2, \Gamma_1)$,
3. for any three LMCs $\Gamma_1, \Gamma_2, \Gamma_3 \in \mathcal{M}$, $\mathsf{D}(\Gamma_1, \Gamma_3) \leq \mathsf{D}(\Gamma_1, \Gamma_2) + \mathsf{D}(\Gamma_2, \Gamma_3)$.

We first define the relaxed notion of *ε-approximate probabilistic bisimilarity relation*. In the following, $R \subseteq A \times B$ will refer to a binary relation. By abuse of notation, for any $\tilde{A} \subseteq A$, let $R(\tilde{A}) := P_B((\tilde{A} \times B) \cap R) \subseteq B$, where $P_B : A \times B \to B$ is the projection operator on B. And let $R(\tilde{B}) \subseteq A$ be defined similarly for any $\tilde{B} \subseteq B$. A subset $\tilde{A} \times \tilde{B}$ is said to be *R-closed* if $R(\tilde{A}) \subseteq \tilde{B}$ and $R(\tilde{B}) \subseteq \tilde{A}$.

Definition 1.23 (ε-approximate probabilistic bisimilarity relation)**.** Given two LMCs $\Gamma_1 = (S_1, A, \iota_1, \tau_1, l_1)$ and $\Gamma_2 = (S_2, A, \iota_2, \tau_2, l_2)$ defined on the same set of labels, and $\varepsilon \in (0, 1)$, an *ε-approximate probabilistic bisimulation relation* between Γ_1 and Γ_2 is a relation $R_\varepsilon \subseteq S_1 \times S_2$ such that the following conditions hold:

1. For all $(s_1, s_2) \in R_\varepsilon$, $l(s_1) = l(s_2)$.
2. For all R_ε-closed sets $\tilde{S}_1 \times \tilde{S}_2$

$$\left| \sum_{\tilde{s}_1 \in \tilde{S}_1} \iota(\tilde{s}_1) - \sum_{\tilde{s}_2 \in \tilde{S}_2} \iota(\tilde{s}_2) \right| \leq \varepsilon.$$

3. For all $(s_1, s_2) \in R_\varepsilon$ and for all R_ε-closed sets $\tilde{S}_1 \times \tilde{S}_2$,

$$\left| \sum_{\tilde{s}_1 \in \tilde{S}_1} \tau(\tilde{s}_1 | s_1) - \sum_{\tilde{s}_2 \in \tilde{S}_2} \tau(\tilde{s}_2 | s_2) \right| \leq \varepsilon.$$

We say that two LMCs are *ε-approximately bisimilar* to each other if there exists an ε-approximate probabilistic bisimulation relation between them. If $\varepsilon = 0$, then there exists a *probabilistic bisimilarity relation* between the two LMCs (such as defined in [LS91]), and they are said to be *bisimilar* to each other.

Definition 1.24 (Bisimilarity distance between LMCs)**.** A function $\mathsf{D} : \mathcal{M} \times \mathcal{M} \to \mathbb{R}_{\geq 0}$ is said to be a *bisimilarity distance* if

1. it is a pseudo-distance, and
2. for all Γ_1, Γ_2, $\mathsf{D}(\Gamma_1, \Gamma_2) = 0$ if and only if Γ_1 and Γ_2 are bisimilar to each other.

Bisimilarity distance is a quantitative notion arising from the qualitative notion of probabilistic bisimilarity.

Example 1.25. The following is an example of bisimilarity distance:

$$\mathsf{D}(\Gamma_1, \Gamma_2) = \inf\{\varepsilon : \Gamma_1 \text{ and } \Gamma_2 \text{ are } \varepsilon\text{-approximately bisimilar}\}.$$

We now define trace equivalence.

Definition 1.26 (Trace-equivalence)**.** Two LMCs Γ_1 and Γ_2 are said to be *trace equivalent* if $\mu_1^\omega \equiv \mu_2^\omega$, where μ_1^ω and μ_2^ω are the two probabilities generated by Γ_1 and Γ_2 respectively on $(A^\omega, \sigma(\mathcal{T}^\omega))$.

Definition 1.27 (Trace-based distance between LMCs)**.** A function $\mathsf{D} : \mathcal{M} \times \mathcal{M} \to \mathbb{R}_{\geq 0}$ is said to be a trace-based distance if

1. it is a pseudo-distance, and
2. for all Γ_1, Γ_2, $\mathsf{D}(\Gamma_1, \Gamma_2) = 0$ if and only if Γ_1 and Γ_2 are trace equivalent.

Similarly as for bisimilarity distance, trace-based distance is a quantitative notion arising from the qualitative notion of trace equivalence.

Although similar, these two notions are not equivalent. Trace equivalence imposes equivalence on what one can observe, whereas bisimilarity imposes equivalence on the state space and transition probabilities, meaning that the LMCs must match not only in their observable behaviour but also in their internal dynamics.[4] This observation is formalised in the following proposition.

Proposition 1.28. *Any two bisimilar LMCs are also trace equivalent.*

Proof. For any $k \in \mathbb{N}$, it is straightforward from the definition of bisimilarity that $\mu_1^k(w) = \mu_2^k(w)$ for all $w \in A^k$. Therefore, for all cylinder set wA^ω, it holds that

$$\mu_1^\omega(wA^\omega) = \mu_1^{|w|}(w) = \mu_2^{|w|}(w) = \mu_2^\omega(wA^\omega). \tag{1.1}$$

By the Carathéodory's extension theorem, μ_1^ω and μ_2^ω are uniquely defined from the probabilities above, and therefore give equal probability to any set in $\sigma(\mathcal{T}^\omega)$. □

We conclude by defining notions of continuity between LMCs.

Definition 1.29 (Parameter continuity)**.** A pseudo-distance D between LMCs is said to be *parameter* continuous if, for all $\varepsilon > 0$, there exists $\delta > 0$ such that, for any two LMCs Γ_1 and Γ_2,

$$\Gamma_1 \text{ and } \Gamma_2 \text{ are } \delta\text{-approximately bisimilar} \implies \mathsf{D}(\Gamma_1, \Gamma_2) \leq \varepsilon.$$

[4] For more information, [BA17] studied the relation between ε-approximate bisimilarity and trace equivalence.

Definition 1.30 (Φ-continuity)**.** Let $\phi \subseteq A^\omega$ be a property. A pseudo-distance D between LMCs is said to be *ϕ-continuous* if, for all $\varepsilon > 0$, there exists $\delta > 0$ such that, for any two LMCs Γ_1 and Γ_2,

$$\mathsf{D}(\Gamma_1, \Gamma_2) \leq \delta \implies \sup_{w \in \phi} |\mu_1^\omega(w) - \mu_2^\omega(w)| \leq \varepsilon.$$

Let Φ be a class of properties, a distance D is said to be *Φ-continuous* if it is ϕ-continuous for all $\phi \in \Phi$.

Remark 1.31. These notions are inspired from [JMGLM14], although parameter continuity as defined here is less restrictive than the notion defined in [JMGLM14, Definition 5].

As highlighted in [JMGLM14], these notions are conflicting. Indeed, the minimal distance $\mathsf{D}(\Gamma_1, \Gamma_2) = 0$ for all pairs of LMCs is parameter continuous, whereas the maximal distance $\mathsf{D}(\Gamma_1, \Gamma_2) = \infty$ for any two different LMCs is Φ-continuous for any class of properties Φ. It is therefore understood that a "good distance" balances these two notions. In Chapter 6, we will show that it is the case of the Cantor-Kantorovich distance introduced in this thesis.

2

Dynamical systems

THIS chapter introduces the framework of *dynamical systems*, focusing on continuous state spaces (e.g., physical systems) and discrete-time models. Our emphasis is on autonomous systems (e.g. uncontrolled), where the evolution of states is governed by internal dynamics, independent of external inputs.

The chapter explores *partially observable systems*, where an output function partitions the state space. In this thesis, *epistemic uncertainty* is modelled with the latter concept: we do not know exactly in which state we are, but we can observe regions of the state space. We begin by defining *deterministic* dynamical systems, where state trajectories are uniquely determined by initial conditions and the transition function. We then extend this framework to *stochastic* dynamical systems, incorporating *aleatoric uncertainty* via the transition kernel. These concepts will serve as the basis for Chapter 3, Chapter 4 and Chapter 5, where we will *abstract* such systems with memory-based labelled Markov chains.

2.1 Deterministic dynamical systems

In Chapter 3 and Chapter 4, we will study deterministic dynamical systems. In such systems, the dynamics of the states is described as

$$x_{k+1} = f(x_k),$$

with $f : X \to X$ a function. This is formalised as follows.

Definition 2.1 (Deterministic dynamical system)**.** A *deterministic dynamical system* of dimension $n \in \mathbb{N}$ is a tuple $\Sigma := (X, f)$, where X is called the *state space*, and $f : X \to X$ the *transition function*.

All along this thesis, we consider that the state space X is a subset of $\mathbb{R}^d$, the *Euclidean* space. This space is a Banach space endowed with the so-called *Euclidean* norm, defined as

$$\|\cdot\| : X \to \mathbb{R}_{\geq 0} : x \mapsto \sqrt{\sum_{i=1}^{d} x_i^2}.$$

We invite the reader to read Appendix C for a formal definition of *Banach space*. This norm induces a distance, which itself induces a metrizable topology $\mathcal{T}$ for $\mathbb{R}^d$. The topology for X is defined to be any set of the form $T \cap X$, where $T \in \mathcal{T}$, and is noted $\mathcal{T}_X$. In this thesis, the state space is assumed to be endowed with the Borel σ-algebra $\sigma(\mathcal{T}_X)$. For the sake of simplicity, we note $\sigma(X) := \sigma(\mathcal{T}_X)$. Again, we refer the reader to Appendix A and Appendix B for the definitions of all concepts related to topological/metric spaces and measure theory, respectively.

Definition 2.2 (State trajectory)**.** Given a system $\Sigma = (X, f)$ and some time horizon $K \in \mathbb{N} \cup \{+\infty\}$, a sequence $(x_0, \ldots, x_K) \in X^K$ is called a *state trajectory* for Σ if $x_{k+1} = f(x_k)$ for all $k = 0, \ldots, K-1$.

Note that a state trajectory is uniquely determined by the so-called *initial state* $x_0 \in X$.

As highlighted in the introduction of this chapter, in this thesis we model epistemic uncertainty by considering that we cannot directly observe the states, but rather regions in which they belong, i.e. we have access to outputs y_k defined as

$$\begin{cases} x_{k+1} = f(x_k) \\ y_k = h(x_k) \in \{1, \ldots, m\}, \end{cases}$$

with $h : X \to \{1, \ldots, m\}$. This is formalised as follows.

Definition 2.3 (Partially observable deterministic dynamical system)**.** A *partially observable* discrete-time deterministic dynamical system is a tuple (Σ, A, h), where Σ is a discrete-time deterministic dynamical system, A is

any space of finite cardinality ($|A| < +\infty$) and is called the *output space*, and $h : X \to A$ is called the *output function*.

By abuse of notation, and when clear from the context, we will write $\Sigma := (X, f, A, h)$ instead of (Σ, A, h) to refer to partially observable deterministic dynamical systems. Also, without loss of generality, we will consider all along this thesis that $A := \{1, \dots, m\}$ for some $m \in \mathbb{N}$.

For all $a \in A$, let

$$[\![a]\!] := \{x \in X : h(x) = a\}.$$

These sets define a *partition* of X, since

$$\text{(covering)} \quad \bigcup_{a \in A} [\![a]\!] = X,$$
$$\text{(pairwise disjoint)} \quad \forall a, b \in A \text{ s.t. } a \neq b : [\![a]\!] \cap [\![b]\!] = \emptyset.$$

The sets $[\![a]\!]$ are called *regions* of the partition.

Definition 2.4 (Output trajectory)**.** Given a system $\Sigma = (X, f, A, h)$ and some time horizon $K \in \mathbb{N} \cup \{+\infty\}$, a sequence $(y_0, \dots, y_K) \in A^K$ is called an *output trajectory* for Σ if there exists a state trajectory $(x_0, \dots, x_K)$ for Σ such that $y_k = h(x_k)$ for all $k = 1, \dots, K$.

Although *partially* observable, deterministic systems can be *asymptotically* observable, meaning that two infinite output trajectories lead to the same state. This is formally defined as follows.

Definition 2.5 (Asymptotically observable system)**.** Let $\Sigma = (X, f, A, h)$ be a partially observable system, where X is endowed with a distance D. The system Σ is said to be *asymptotically observable* if, for any two infinite length state trajectories $(x_0, x_1, \dots)$ and $(x'_0, x'_1, \dots)$ such that $h(x_k) = h(x'_k)$ for all $k = 0, 1, \dots$, one has that

$$\lim_{k \to \infty} \mathsf{D}(x_k, x'_k) = 0.$$

At the centre of our motivation to use *memory* to abstract partially observable systems is the phenomenon of *loss of the Markov property*, described below.

Remark 2.6 (Loss of Markov property)**.** A key property of dynamical systems is that, since the state follows $x_{k+1} = f(x_k)$, then the current state

only depends on the previous state, and not on the whole history of states. In this thesis we refer to this property as the *Markov property*. Indeed consider the sequence of states (x_k, x_{k+1}, x_{k+2}). The fact that the system is at state x_{k+2} at time $k+2$ depends only on x_{k+1}, regardless of x_k.

However, partially observable systems lose this property. Indeed, consider the sequence of outputs (y_k, y_{k+1}, y_{k+2}). It can be easily verified that the fact that the system outputs y_{k+2} at time $k+2$ depends on both y_k and y_{k+1}. This is illustrated in Example 2.7.

Example 2.7 (Sturmian dynamics)**.** Let φ be an irrational number. Consider the system $\Sigma = (X, f, A, h)$ with

- the state space $X = [0, 1)$,
- the transition function $f : X \to X : x \mapsto x + \varphi \mod 1$,
- the output space $A = \{0, 1\}$,
- the output function is defined as $h : X \to A : x \mapsto \begin{cases} 1 & \text{if } x \in [0, \varphi) \\ 0 & \text{otherwise.} \end{cases}$

This system is known as the *Sturmian* dynamical system, and is studied e.g. in [Fog02]. It is illustrated in Figure 2.1. Following-up on Remark 2.6, at any time $k+2$, the fact that $y_{k+2} = 1$ indeed depends on the value of y_k. Indeed, if $y_k = 1$, it is impossible that $y_{k+2} = 1$, whereas this is not the case if $y_k = 0$. This illustrates the fact that considering outputs instead of states makes the Markov property vanish. In fact, it can be shown that, if φ is irrational, there exists no *finite* memory sufficient to characterise the output [Fog02].

2.2 Stochastic dynamical systems

In Chapter 5, we will study stochastic dynamical systems. In such systems, the dynamics of the states is described as

$$x_{k+1} \sim \tau(\cdot|x_k),$$

with $\tau : \sigma(X) \times X \to [0, 1]$ a *Markov kernel*, i.e. a map such that, for all $x \in X$, $\tau(\cdot|x)$ is a *probability measure* on $(X, \sigma(X))$, and where $\sigma(X)$ is the σ-algebra derived from the Euclidean distance such as defined above. We refer the reader to Appendix D for all notions related to probability theory.

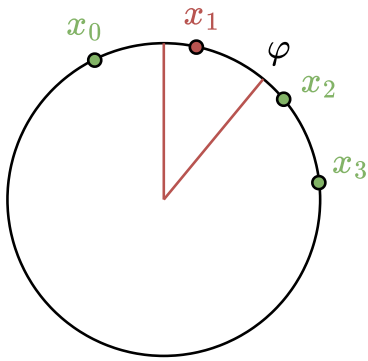


Fig. 2.1 Illustration of the Sturmian dynamics. The statespace $[0,1)$ is represented as a circle. An example of trajectory (x_0, x_1, x_2, x_3) is given. The states in green output 0, whereas the states in red output 1.

Remark 2.8. As discussed in the introduction, stochastic dynamical systems are a popular framework for modelling aleatoric uncertainty. However, other types of dynamical systems also incorporate aleatoric uncertainty without resorting to stochasticity, such as switched or distribution-free noisy systems [Jun09, BWJ24]. Extending our results to such settings is left as an interesting direction for future work.

The notion of stochastic dynamical system is formalised as follows.

Definition 2.9 (Stochastic dynamical system)**.** A *stochastic dynamical system* of dimension $d \in \mathbb{N}$ is a tuple $\Sigma := (X, \tau)$, where $X \subseteq \mathbb{R}^d$ is called the state space, and $\tau : \sigma(X) \times X \to [0,1]$ is a Markov kernel called the transition kernel of the system.

In stochastic systems, sequences of probability measures play a role analogous to that of state trajectories in deterministic systems.

Definition 2.10 (State probability measures)**.** Given a system $\Sigma = (X, \tau)$ and a horizon $K \in \mathbb{N} \cup \{+\infty\}$, a sequence $(\lambda_0, \dots, \lambda_K)$ is called a sequence of *state probability measures* if

1. for all $k = 0, \dots, K$, $\lambda_k : \sigma(\mathcal{T}_X) \to [0,1]$ is a probability measure on $(X, \sigma(\mathcal{T}_X))$,
2. for all $k = 0, \dots, K-1$, λ_{k+1} is the *pushforward measure* of λ_k by the

kernel τ, noted $\lambda_{k+1} := \tau\#\lambda_k$, i.e.

$$\lambda_{k+1}(A) = \int_{x\in X} \tau(A|x)\lambda_k(\mathrm{d}x)$$

for all $A \in \sigma(X)$.

We invite the reader to read Appendix B for a formal definition of push-forward measure. Similarly as for deterministic systems, a sequence of state probability measures is uniquely determined by the so-called *initial state probability measure* (often shortened to initial measure) λ_0.

The notion of state trajectory is defined as follows.

Definition 2.11 (State trajectory)**.** Given a system $\Sigma = (X, \tau)$ and some time horizon $K \in \mathbb{N} \cup \{+\infty\}$, let $(\mathsf{x}_0, \ldots, \mathsf{x}_K)$ be a sequence of random variables associated with the sequence of state probability measures $(\lambda_0, \ldots, \lambda_K)$, called the *state random variables*. A sequence $(x_0, \ldots, x_K) \in X^K$ is called a state trajectory for Σ if, for every $k = 0, \ldots, K$, x_k is a realisation of x_k.

Example 2.12 (Linear system with Gaussian noise)**.** Consider a dynamical system of the form

$$x_{k+1} = \mathbf{A}x_k + w_k,$$

where $\mathbf{A} \in \mathbb{R}^{d\times d}$ is a matrix, and w_k is the realisation of a Gaussian random variable $\mathsf{w} \sim \mathcal{N}(\mathbf{m}, \mathbf{C})$, where $\mathbf{m} \in \mathbb{R}^d$ is the mean and $\mathbf{C} \in \mathbb{R}^{d\times d}$ is the covariance matrix. The state space of the system is $\mathbb{R}^d$, and the considered probability space is therefore $(\mathbb{R}^d, \sigma(\mathbb{R}^d))$, where $\sigma(\mathbb{R}^d)$ is the Borel σ-algebra generated by the usual open sets in the Euclidean space. This system can be re-written as

$$x_{k+1} \sim \tau(\cdot|x_k),$$

where $\tau : \sigma(\mathbb{R}^d) \times \mathbb{R}^d \to [0, 1]$ is defined as

$$\tau(A|x) = \mu_{\mathcal{N}(\mathbf{A}x+\mathbf{m},\mathbf{C})}(A),$$

for all $A \in \sigma(\mathbb{R}^d)$, with $\mu_{\mathcal{N}(\mathbf{A}x+\mathbf{m},\mathbf{C})} : \sigma(\mathbb{R}^d) \to [0, 1]$ being the probability measure associated with the Gaussian distribution centred in $\mathbf{A}x + \mathbf{m}$ and with covariance matrix $\mathbf{C}$. Suppose that this system is initialised at a Gaussian initial measure $\lambda_0 : \sigma(\mathbb{R}^d) \to \mathbb{R}$ defined as

$$\lambda_0 := \mu_{\mathcal{N}(\mathbf{m}_0,\mathbf{C}_0)},$$

for some means and covariance matrices $\mathbf{m}_0 \in \mathbb{R}^d$ and $\mathbf{C}_0 \in \mathbb{R}^{d\times d}$. Then it can be shown that

$$x_k \sim \lambda_k = \mu_{\mathcal{N}(\mathbf{m}_k, \mathbf{C}_k)},$$

where $\mathbf{m}_k$ and $\mathbf{C}_k$ satisfy the recurrences

$$\mathbf{m}_{k+1} = \mathbf{A}\mathbf{m}_k + \mathbf{m}, \quad \mathbf{C}_{k+1} = \mathbf{A}\mathbf{C}_k\mathbf{A}^\top + \mathbf{C}.$$

If, in addition, $\mathbf{A}$ is *stable*[1], then the kernel of the system admits a unique *invariant measure*

$$\mu := \mu_{\mathcal{N}(\mathbf{m}_\star, \mathbf{C}_\star)},$$

where $\mathbf{m}_\star$ and $\mathbf{C}_\star$ satisfy

$$\mathbf{m}_\star = \mathbf{A}\mathbf{m}_\star + \mathbf{m}, \quad \mathbf{C}_\star = \mathbf{A}\mathbf{C}_\star\mathbf{A}^\top + \mathbf{C}.$$

The notion of *partial observability*, extends naturally to stochastic systems as follows.

Definition 2.13 (Partially observable stochastic dynamical system)**.** A partially observable discrete-time stochastic dynamical system is a tuple (Σ, τ, h), where Σ is a discrete-time stochastic dynamical system, A is any space of finite cardinality ($|A| < +\infty$) and is called the output space, and $h : X \to A$ is called the output function.

Again, by abuse of notation, and when clear from the context, we will write $\Sigma := (X, \tau, A, h)$ instead of (Σ, A, h) to refer to partially observable stochastic dynamical systems. Similarly to for deterministic systems, the output function h defines a partition of X, noted $\{[\![a]\!]\}_{a\in A}$.

As opposed to deterministic systems, outputs are now associated with a probability measure, defined as follows.

Definition 2.14 (Output probability measures)**.** Given a system $\Sigma = (X, \tau, A, h)$ and a horizon $K \in \mathbb{N} \cup \{+\infty\}$, let $(\lambda_0, \ldots, \lambda_K)$ be the sequence of state probability measures. The sequence of functions $(\nu_0, \ldots, \nu_K)$ defined as

$$\nu_k(a) = \lambda_k\left([\![a]\!]\right)$$

for all $k = 0, \ldots, K$ is called the sequence of *output probability measures*.

Note that the definition above implies that $\nu_k : 2^A \to [0, 1]$ is a probability measure on the measurable space $(A, 2^A)$ where we set $\nu_k(a) := \nu_k(\{a\})$

[1] In the sense that the modulus of all eigenvalues is strictly smaller than 1.

by abuse of notation.

Finally, the notion of output trajectory is extended in the following way.

Definition 2.15 (Output trajectory)**.** Given a system $\Sigma = (X, \tau, A, h)$ and some time horizon $K \in \mathbb{N} \cup \{+\infty\}$, let $(\mathsf{y}_0, \dots, \mathsf{y}_K)$ be a sequence of random variables associated with the sequence of output probability measures $(\nu_0, \dots, \nu_K)$, called the *output random variables*. A sequence $(a_0, \dots, a_K) \in A^K$ is called an output trajectory for Σ if, for every $k = 0, \dots, K$, a_k is a realisation of y_k.

Remark 2.16 (Loss of Markov property – stochastic version)**.** Partially observable stochastic systems also suffer from the loss of the Markov property, such as defined in Remark 2.6 for deterministic systems. Let x_k and y_k be the random variables respectively associated with the states and outputs of Σ, then, by definition of the Markov kernel, for all $B \in \sigma(X)$,

$$\mathbb{P}(\mathsf{x}_k \in B \,|\, \mathsf{x}_{k-1} = x_{k-1}, \mathsf{x}_{k-2} = x_{k-2}, \dots) = \mathbb{P}(\mathsf{x}_k \in B \,|\, \mathsf{x}_{k-1} = x_{k-1}),$$

whereas, for all $C \in 2^A$,

$$\mathbb{P}(\mathsf{y}_k \in C \,|\, \mathsf{y}_{k-1} = y_{k-1}, \mathsf{y}_{k-2} = y_{k-2}, \dots) \neq \mathbb{P}(\mathsf{y}_k \in C \,|\, \mathsf{y}_{k-1} = y_{k-1}).$$

This is a known property, discussed e.g. in [SNS10].

PART II
Abstractions of deterministic systems

3

Uniform memory-dependent abstractions

THE following three chapters explore *data-driven* and *memory-dependent* abstractions for both deterministic and stochastic systems subject to epistemic uncertainty. By modelling unknown systems as *labelled Markov chains* that encode *memory*, we transform *epistemic uncertainty* into purely *aleatoric uncertainty*. For each method, we describe how the abstraction is constructed from output samples, prove the guarantees it provides, and validate our approach through numerical experiments.

We first focus on data-driven memory-dependent abstractions of *deterministic systems*. Especially, in this chapter we focus on the case where the LMC encodes a *uniform* memory scheme, i.e. in which the nodes encode output sequences of a fixed length (see Figure 3.1 for an illustration). We will demonstrate that, when constructed appropriately, these abstractions increasingly approximate the behaviour of the original system, and, as the memory grows, the abstraction converges to the true dynamics, making them *asymptotically correct*.

Similar methods have been investigated in the literature, although always in different frameworks. The works [MR99, MRO02] introduced the similar concept of *ℓ-complete abstractions*, although in the context of *supervisory control*, a model-based control method introduced by Ramadge and

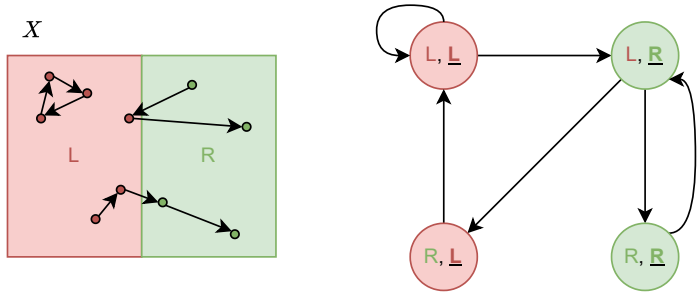


Fig. 3.1 Illustration of a uniform memory-dependent abstraction of a dynamical system. On the left: a partially observable deterministic dynamical system with $A = \{\mathrm{L}, \mathrm{R}\}$. On the right: an LMC whose nodes encode sequences of length 2. Being in the node (R, L) means that the current observation is L and that the previous observation was R.

Wonham (see e.g. [RW87, RW89] for an introduction). Moreover the guarantees obtained in this chapter are of a different nature, and depend on different assumptions. More recently, ℓ-complete abstractions have been extended in [STR15] and applied to chaos detection in [GM23], but always in a model-based context. The works [PM23, CPM23] have investigated data-driven ℓ-complete abstractions, however they focus on the data-driven side by providing finite-sample guarantees, whereas our focus is on qualitative guarantees on abstraction performance – making these works complementary.

Contributions. In this chapter we introduce data-driven and uniform memory abstractions of deterministic dynamical systems. The main contributions are summarised below.

1. **Method.** We propose a method to construct a so-called ℓ-memory Markov model that captures the behaviour of the original partially observable dynamical system.
2. **Guarantee.** We prove that, under some assumptions, our abstraction is asymptotically correct in the sense that increasing the memory to infinity leads to a non-conservative abstraction. We also show that the approximation quality is monotonically better with the memory. To do that, we introduce the notion of *perturbation* of a system, and

show that the behaviour of the abstraction is the one of the original system, but slightly perturbed by a noise whose magnitude goes monotonically to zero as the memory grows.

3. **Numerical experiments.** We provide numerical evidence that increasing the memory leads to a better abstraction.

Outline. Section 3.1 clarifies the data-driven framework and states overall assumptions on the system. The overall abstraction method is described in Section 3.2. In Section 3.3, we prove the asymptotic correctness of our abstraction. To do that, we introduce the notion of perturbation of a system, and give the necessary corresponding formalism. In Section 3.4 we provide numerical evidence that increasing the memory leads to better abstractions, and we provide conclusions and further research directions in Section 3.5.

Publication. This work has been done in collaboration with Licio Romao, Alessandro Abate and Raphaël M. Jungers, and has been published in [BRAJ23b].

3.1 Framework

In this chapter we consider deterministic dynamical systems of the form $\Sigma = (X, f, A, h)$ where $X \subseteq \mathbb{R}^d$ and is endowed with the σ-algebra $\sigma(X)$ as defined in Chapter 2, whose state and output trajectories are driven by

$$\begin{cases} x_{k+1} = f(x_k), \\ y_k = h(x_k). \end{cases}$$

As we will see in Section 3.2 and Section 3.3, a central characteristic of our method is that we leverage the *invariant measure* of the system, defined as the probability measure $\mu : \sigma(X) \to [0, 1]$ such that

$$(f\#\mu)(A) := \mu(A)$$

for all $A \in \sigma(X)$, and where $f\#\mu$ is the pushforward measure by f (see Appendix B for a formal definition). The following is therefore a natural assumption.

Assumption 3.1 (Existence of the invariant measure)**.** There exists an invariant measure $\mu : \sigma(X) \to [0,1]$ for the transition function f.

We claim that this assumption is not restrictive, and is met in many practical situations. Indeed in this setting, if f is continuous, then the assumption is satisfied.[1]

As explained above, we consider a data-driven situation, in which one can only *sample* observations of the system. In this work, this is formalised by the fact that the initial state of Σ is random. More formally, we consider that there exists a probability measure $\lambda_0 : \sigma(X) \to [0,1]$, noted

$$x_0 \sim \lambda_0.$$

This initial measure implies the existence of subsequent probability measures $\lambda_k : \sigma(X) \to [0,1]$ for the dynamical system, recursively defined as

$$\lambda_{k+1} = f\#\lambda_k$$

for all $k \geq 0$, as well as probability measures on the outputs, noted $\nu_k : 2^A \to [0,1]$, and defined as

$$\nu_k(a) := \lambda_k(\llbracket a \rrbracket)$$

for all $a \in A$.

The initial measure also implies the existence of probability measures on *sequences* of outputs. In the following, for any integer $\ell \in \mathbb{N}$, we will set $\sigma(X)^\ell := \bigotimes_{i=1}^{\ell} \sigma(X)$, where $\bigotimes_{i=1}^{\ell}$ denotes the ℓ-th power σ-algebra.[2]

Definition 3.2 (State sequences probability)**.** Let $\Sigma = (X, f, A, h)$ be a deterministic system whose initial state is sampled according to the probability measure $\lambda_0 : \sigma(X) \to [0,1]$. Let $k, \ell \in \mathbb{N}$, then the system Σ induces a probability measure on $(X^\ell, \sigma(X)^\ell)$, noted $\lambda_k^\ell : \sigma(X)^\ell \to [0,1]$, and defined as, for any $A_k \times \dots A_{k+\ell-1} \in \sigma(X)^\ell$,

$$\begin{aligned}&\lambda_k^\ell(A_k \times \cdots \times A_{k+\ell-1}) \\ &\quad = \int_{x_k \in A_k} \cdots \int_{x_{k+\ell-1} \in A_{k+\ell-1}} \left(\prod_{i=k}^{k+\ell-2} \delta_{f(x_i)}(\mathrm{d}x_{i+1}) \right) \lambda_k(\mathrm{d}x_k).\end{aligned}$$

[1] This directly follows from the Krylov-Bogolyubov theorem (see Theorem D.7).

[2] The ℓ-th power σ-algebra is a generalization of the product σ-algebra.

This measure also implies the existence of a probability measure on sequences of outputs, noted $\nu_k^\ell : 2^{A^\ell} \to [0,1]$, and defined as

$$\nu_k^\ell(a_k, \dots, a_{k+\ell-1}) := \lambda_k^\ell([\![a_k]\!] \times \cdots \times [\![a_{k+\ell-1}]\!])$$

for all $(a_k, \dots, a_{k+\ell-1}) \in A^\ell$. This measure on sequences of outputs will be useful to construct and analyse our abstractions.

Definition 3.3 (System behaviour)**.** Let $\Sigma = (X, f, A, h)$ be a deterministic system. For any $k \in \mathbb{N}$, the set $\mathcal{B}_k(\Sigma, \lambda_0) \subseteq A^k$ defined as

$$\mathcal{B}_k(\Sigma, \lambda_0) = \{(a_0, \dots, a_{k-1}) \in A^k : \nu_0^k(a_0, \dots, a_{k-1}) > 0\}$$

is called the *k-long behaviour* of the system. The set $\mathcal{B}(\Sigma, \lambda_0) \subseteq A^\star$ defined as

$$\mathcal{B}(\Sigma, \lambda_0) = \bigcup_{k \in \mathbb{N}} \mathcal{B}_k(\Sigma, \lambda_0)$$

is called the *behaviour* of the system.

In this chapter we will construct an abstraction whose behaviour is close to the one of the original system. Note that this definition of behaviour is inspired from Willem's *behavioural approach*, see e.g. [PW97] for an introduction.

3.2 Method

In this section we provide the method to construct the uniform memory abstraction of a deterministic system. The systems will be abstracted by a particular type of LMC called a *ℓ-memory Markov model*, defined as follows.

Definition 3.4 (ℓ-memory Markov model)**.** Let $\ell \in \mathbb{N}$ be a natural number and A be a finite alphabet. An ℓ-memory Markov model is an LMC $\Gamma_\ell := (S_\ell, A, \iota_\ell, \tau_\ell, l_\ell)$ where

- the states S_ℓ is a subset of ℓ-long output sequences, that is $S_\ell \subseteq A^\ell$,
- $\tau_\ell : 2^{S_\ell} \times S_\ell \to [0,1]$ is such that for all two sequences $(a_0, \dots, a_{\ell-1})$, $(b_1, \dots, b_\ell) \in S_\ell$, if $(a_1, \dots, a_{\ell-1}) \neq (b_1, \dots, b_{\ell-1})$, then

$$\tau_\ell((b_1, \dots, b_\ell) | (a_0, \dots, a_{\ell-1})) = 0,$$

- the labelling function $l_\ell : S_\ell \to A$ is defined as $l_\ell(a_1, \dots, a_\ell) := a_\ell$.

Because of the definition above, we will simplify the notations and note

$$\tau_\ell(a_\ell|a_0,\dots,a_{\ell-1}) := \tau_\ell((a_1,\dots,a_\ell)|(a_0,\dots,a_{\ell-1}))$$

for all $a_0,\dots,a_\ell \in A^{\ell+1}$. Intuitively, a ℓ-memory Markov model is a generalization of a Markov chain, in which the probability of the current observation only depends on the ℓ last observations. Similar ideas are studied e.g. in [WC17].

Definition 3.5 (ℓ-memory Markov model behaviour)**.** Given a ℓ-memory Markov model $\Gamma_\ell = (S_\ell, A, \iota_\ell, \tau_\ell, l_\ell)$, and for any $k \in \mathbb{N}$, the set $\mathcal{B}_k(\Gamma_\ell) \subseteq A^k$ defined as

$$\mathcal{B}_k(\Gamma_\ell) = \{(a_0,\dots,a_{k-1}) \in A^k : \mu^k(a_0,\dots,a_{\ell-1}) > 0\},$$

where μ^k is the k-long probability measure generated by Γ_ℓ such as defined in Chapter 1, is called the k-long behaviour of the ℓ-memory Markov model. The set $\mathcal{B}(\Gamma_\ell) \subseteq A^\star$ defined as

$$\mathcal{B}(\Gamma_\ell) = \bigcup_{k\in\mathbb{N}} \mathcal{B}_k(\Gamma_\ell)$$

is called the behaviour of the ℓ-memory Markov model.

Remark 3.6. Although both are denoted by $\mathcal{B}$, the objects $\mathcal{B}(\Sigma, \lambda_0)$ and $\mathcal{B}(\Gamma_\ell)$ are different: the former is the behaviour of the sampled deterministic system, while the latter is the behaviour of the Markov model.

As explained above, we will provide a method to construct a ℓ-memory Markov model Γ_ℓ whose behaviour $\mathcal{B}(\Gamma_\ell)$ is close to $\mathcal{B}(\Sigma, \mu_0)$, the behaviour of the original system.

Our data-driven method is provided in Algorithm 1, and consists in sampling output trajectories from the partially observable dynamical system and constructing transition probabilities by empirical counting of output sequences. Formally, we define the frequency as follows.

Definition 3.7 (Frequency)**.** Let Σ be a deterministic system whose initial state is sampled according to a probability measure μ. For any trajectory length $\ell \in \mathbb{N}$, let

$$\omega_N := \{(a_0^i,\dots,a_{\ell-1}^i)\}_{i=1}^N \subseteq A^\ell$$

be a collection of N output trajectories. The function $F_\mu^N : A^\ell \to [0,1]$ is

defined, for any $(a_0, \ldots, a_{\ell-1}) \in A^\ell$, by

$$F_\mu^N(a_0, \ldots, a_{\ell-1}) := \frac{1}{N} \sum_{i=1}^{N} \mathbf{1}\Big\{(a_0^i, \ldots, a_{\ell-1}^i) = (a_0, \ldots, a_{\ell-1})\Big\}.$$

where $\mathbf{1}$ is the *indicator function*. The limit for $N \to \infty$ is noted

$$F_\mu := \lim_{N\to\infty} F_\mu^N. \tag{3.1}$$

Algorithm 1 Construction of a ℓ-memory Markov abstraction

1: Generate output trajectories of the system Σ whose initial state is sampled according to the invariant measure μ and, for all sequence $(a_0, \ldots, a_\ell) \in A^{\ell+1}$, compute

$$\begin{aligned} &F_\mu(a_0, \ldots, a_{\ell-1}) && (\ell\text{-long frequency}) \\ &F_\mu(a_0, \ldots, a_\ell) && ((\ell+1)\text{-long frequency}), \end{aligned}$$

where F_μ is the counting function such as defined above.

2: Let $\overline{S}$ be defined as

$$\overline{S} = \{(a_0, \ldots, a_{\ell-1}) \in A^\ell : F_\mu(a_0, \ldots, a_{\ell-1}) = 0\}.$$

Define

$$S_\ell := A^\ell \setminus \overline{S}.$$

3: Generate output trajectories of the system Σ initialised at the initial measure λ_0, and, for all sequence $(a_0, \ldots, a_{\ell-1}) \in S_\ell$, compute

$$F_{\lambda_0}(a_0, \ldots, a_{\ell-1}) \qquad (\ell\text{-long frequency}).$$

For all ℓ-long sequence $a_0, \ldots, a_{\ell-1}$, let

$$\iota_\ell(a_0, \ldots, a_{\ell-1}) := F_{\lambda_0}(a_0, \ldots, a_{\ell-1}). \tag{3.2}$$

4: For all $(a_0, \ldots, a_{\ell-1})$, $(a_1, \ldots, a_\ell) \in S_\ell$, let

$$\tau_\ell(a_\ell | a_0, \ldots, a_{\ell-1}) := \frac{F_\mu(a_0, \ldots, a_\ell)}{F_\mu(a_0, \ldots, a_{\ell-1})}.$$

return The ℓ-memory Markov model $\Gamma_\ell := (S_\ell, A, \tau_\ell, l_\ell)$.

Assumption 3.8. We have access to infinitely many samples in Algorithm 1. That is, one can compute exactly F_μ, as described in (3.1).

Remark 3.9 (Infinite scenarios)**.** In this thesis, we do not focus on the case when one only has access to a finite number of samples (as formalised in Assumption 3.8), although our abstractions can be constructed in a Monte-Carlo fashion, making them data-driven in that sense. However handling finite observation scenarios ($N < +\infty$) is a key challenge, requiring advanced techniques like the scenario approach [CG19], conformal prediction [CPLM24], or compression learning [MPL15]. This is a dissociated (but very relevant) problem, and relating it with our work is beyond this thesis scope.

Remark 3.10 (Sampling from the invariant)**.** In practice, computing the invariant output probabilities can be done in at least two ways. Either one samples a sufficiently large number of $(\ell+1)$-long output traces initialised at the invariant distribution. Or, employing Birkhoff's theorem (see e.g. [Shi16]), one samples a very long output trace initialised at any initial distribution.

In regard of Assumption 3.8, Algorithm 1 can be interpreted as follows. In step 3, we set

$$\begin{aligned}\iota_\ell(a_0,\dots,a_\ell) &:= \nu_0^\ell(a_0,\dots,a_\ell)\\ &:= \mathbb{P}_{\lambda_0}(\mathsf{y}_0 = a_0,\dots,\mathsf{y}_{\ell-1} = a_{\ell-1}).\end{aligned}$$

In steps 1 and 3, we set

$$\begin{aligned}\tau_\ell(a_\ell|a_0,\dots,a_{\ell-1}) &:= \frac{\mu^{\ell+1}([\![a_0]\!]\times\cdots\times[\![a_\ell]\!])}{\mu^\ell([\![a_0]\!]\times\cdots\times[\![a_{\ell-1}]\!])}\\ &:= \mathbb{P}_\mu\left(\mathsf{y}_\ell = a_\ell|\mathsf{y}_0 = a_0,\dots,\mathsf{y}_{\ell-1} = a_{\ell-1}\right).\end{aligned} \tag{3.3}$$

Our ℓ-memory Markov is therefore initialised at a measure corresponding to the *initial regime* of the original system, but the *invariant regime* is used to transition. In the following section, we will show that this construction leads to asymptotic guarantees.

It becomes direct from (3.3) that the following assumption is necessary.

Assumption 3.11. For all sequences $(a_1,\dots,a_\ell)\in A^\ell$,

$$\mu^\ell([\![a_0]\!]\times\cdots\times[\![a_{\ell-1}]\!]) > 0.$$

Assumption 3.11 exposes a fundamental limitation of our method. If the invariant measure is degenerate, observations at stationarity carry no information about the system. Consider for instance a stable deterministic linear system $x_{k+1} = \mathbf{A}x_k$: its unique invariant measure is the Dirac mass δ_0, so all observations at stationarity are identically zero. Crucially, augmenting memory does not resolve this issue, as the joint output measure ν^ℓ at stationarity remains a Dirac mass $\delta_{(0,\dots,0)}$. The method therefore relies implicitly on the invariant measure being sufficiently non-degenerate, in the sense that output sequences sampled at stationarity are informative about the system dynamics. We believe that Assumption 3.11 may be linked to *mixing* properties of the system: intuitively, a mixing system is one whose trajectories eventually explore the entire invariant set, which would guarantee that the invariant measure has positive mass on all reachable cells. Moreover, the slower the system mixes, the larger the memory ℓ required for the abstraction to be informative. In this thesis, we will not focus on this question, and investigating such links is left as further work.

3.3 Guarantee

In this section we will prove that our abstractions are asymptotically correct, in the sense that a theoretically infinite memory ℓ leads to an exact representation of the partially observable original dynamics Σ. More precisely, we will show that the behaviour of the abstraction is the same as the behaviour of a *perturbation of* Σ (defined below), whose perturbation magnitude decreases with memory.

Definition 3.12 (Perturbation of Σ)**.** Let $\Sigma := (X, f, A, h)$ be a deterministic system, and let $\varepsilon \in \mathbb{R}_{>0}$ be a *perturbation magnitude*. Let $W_\varepsilon \subseteq \mathbb{R}^d$ be such that, for any $w \in W_\varepsilon$, $\|w\| \leq \varepsilon$ and, for all k, let $\eta_k : \sigma(W_\varepsilon) \to [0, 1]$ be a probability measure on $(W_\varepsilon, \sigma(W_\varepsilon))$. The tuple $\tilde{\Sigma}_\varepsilon := (X, f, A, h, W_\varepsilon, (\eta_k)_{k\in\mathbb{N}})$ is called the perturbation of Σ.

Intuitively, given a system Σ, the perturbation of Σ is a partially observable deterministic dynamical system with the same transition function as Σ but whose state is potentially perturbed by an additive noise at each time step.

Definition 3.13 (Perturbed state trajectory)**.** Given a system $\Sigma = (X, f, A, h)$ and a perturbation $\tilde{\Sigma}_\varepsilon := (X, f, A, h, W_\varepsilon, (\eta_k)_{k\in\mathbb{N}})$, let $(\mathsf{w}_k)_{k\in\mathbb{N}}$ be a sequence of random variables associated with the sequence of probability

measures $(\eta_k)_{k\in\mathbb{N}}$. A sequence $(x_0, x_1, \dots)$ is a *perturbed state trajectory* if, for some realisations $(w_k)_{k\in\mathbb{N}}$ of the random variables $(\mathsf{w}_k)_{k\in\mathbb{N}}$,

$$x_{k+1} = f(x_k + w_k) \tag{3.4}$$

for all $k = 0, 1, \dots$.

Assuming that the initial state of a perturbation is sampled according to a probability measure $\lambda_0 : \sigma(X) \to [0,1]$, we can extend the notion of behaviour to *perturbations* as follows. Let $(\tilde{\lambda}_k)_{k\in\mathbb{N}}$ be a sequence of probability measures on $(X, \sigma(X))$ such that $\tilde{\lambda}_0 := \lambda_0$, and such that

$$\tilde{\lambda}_{k+1}(A) := \int_{x_k\in X} \int_{\eta_k\in W_\varepsilon} \delta_{f(x_k+w_k)}(A)\eta(\mathrm{d}w_k)\tilde{\lambda}_k(\mathrm{d}x_k)$$

for all $k = 0, 1, \dots$.

Definition 3.14 (Perturbed state sequences probability)**.** Given a system $\Sigma = (X, f, A, h)$ and a perturbation $\tilde{\Sigma}_\varepsilon := (X, f, A, h, W_\varepsilon, (\eta_k)_{k\in\mathbb{N}})$ whose initial state is sampled according to the probability measure $\lambda_0 : \sigma(X) \to [0,1]$, let $k, \ell \in \mathbb{N}$. The perturbation $\tilde{\Sigma}_\varepsilon$ induces a probability measure on $(X^\ell, \sigma(X)^\ell)$, noted $\tilde{\lambda}_k^\ell : \sigma(X)^\ell \to [0,1]$, and defined as, for any $A_k \times \dots A_{k+\ell-1} \in \sigma(X)^\ell$,

$$\begin{aligned}
&\tilde{\lambda}_k^\ell(A_k \times \cdots \times A_{k+\ell-1}) \\
&\quad = \int_{x_k\in A_k} \cdots \int_{x_{k+\ell-1}\in A_{k+\ell-1}} \int_{w_k\in W_\varepsilon} \cdots \int_{w_{k+\ell-1}\in W_\varepsilon} \\
&\qquad \left(\prod_{i=k}^{k+\ell-2} \delta_{f(x_i+w_i)}(\mathrm{d}x_{i+1})\right)\left(\prod_{i=k}^{k+\ell-2} \eta_i(\mathrm{d}w_i)\right)\tilde{\lambda}_k(\mathrm{d}x_k).
\end{aligned}$$

Similarly as in the non-perturbed case, we can defined the measure on sequences of outputs, noted $\tilde{\nu}_k^\ell : 2^{A^\ell} \to [0,1]$, and defined as

$$\tilde{\nu}_k^\ell(a_k, \dots, a_{k+\ell-1}) = \tilde{\lambda}_k^\ell([\![a_k]\!] \times \cdots \times [\![a_{k+\ell-1}]\!])$$

for all $(a_k, \dots, a_{k+\ell-1}) \in A^\ell$. The notion of behaviour is therefore naturally extended as

$$\mathcal{B}_\ell(\tilde{\Sigma}_\varepsilon, \lambda_0) = \{(a_0, \dots, a_{\ell-1}) \in A^\ell : \tilde{\nu}_0^\ell(a_0, \dots, a_{\ell-1}) > 0\}$$

and

$$\mathcal{B}(\tilde{\Sigma}_\varepsilon, \lambda_0) = \bigcup_{\ell \in \mathbb{N}} \mathcal{B}_\ell(\tilde{\Sigma}_\varepsilon, \lambda_0).$$

We now prove the main result of this chapter.

Theorem 3.15. *Let $\Sigma := (X, f, A, h)$ be an asymptotically observable system (see Definition 2.5) whose initial state is sampled according to some initial probability measure $\lambda_0 : \sigma(X) \to [0,1]$. Let Γ_ℓ be the ℓ-memory Markov abstraction given by Algorithm 1 such that Assumption 3.11 is satisfied. Then there exists a monotonically decreasing sequence of perturbation magnitudes $(\varepsilon_\ell)_{\ell \in \mathbb{N}}$ with $\lim_{\ell \to \infty} \varepsilon_\ell = 0$ and perturbation $\tilde{\Sigma}_{\varepsilon_\ell}$ (see Definition 3.12) such that $\mathcal{B}(\Gamma_\ell) = \mathcal{B}(\tilde{\Sigma}_{\varepsilon_\ell}, \lambda_0)$.*

Proof. Our proof relies on the existence of a lifted partition for the original system. For all $(a_0, \ldots, a_{\ell-1}) \in A^\ell$, let

$$\begin{aligned}&[\![a_0, \ldots, a_{\ell-1}]\!] \\ &= \{x_{\ell-1} \in X : \text{there exists a trajectory } (x_0, \ldots, x_{\ell-1}) \text{ with } h(x_i) = a_i\}.\end{aligned}$$

Let ε_ℓ be the diameter of the largest such set, that is

$$\varepsilon_\ell := \max_{(a_0,\ldots,a_{\ell-1}) \in A^\ell} \sup_{x,y \in [\![a_0,\ldots,a_{\ell-1}]\!]} \|x - y\|. \tag{3.5}$$

By asymptotic observability of Σ and compactness of X, the maximal diameter ε_ℓ decreases with ℓ and tends to zero as ℓ grows.

We now show the existence of measures $(\eta_k)_{k \geq 0}$ such that the corresponding perturbation with dynamics $x_{k+1} = f(x_k + w_k)$, $w_k \sim \eta_k$ (see eq. (3.4)) has the same behaviour as the ℓ-memory Markov abstraction. We will proceed by construction. Let $\tilde{\lambda}_0 := \lambda_0$, which implies that

$$\tilde{\lambda}_0^\ell([\![a_0, \ldots, a_{\ell-1}]\!]) = \nu_0^\ell(a_0, \ldots, a_{\ell-1}) = \iota_\ell(a_0, \ldots, a_{\ell-1}). \tag{3.6}$$

We now show the existence of a measure $\overline{\lambda}_0 : \sigma(X) \to [0,1]$ such that

1. there exists $\eta_0 : \sigma(W_{\varepsilon_\ell}) \to [0,1]$ such that the random variable $\overline{x}_0$ associated to the measure $\overline{\lambda}_0$ can be written as

$$\overline{\mathsf{x}}_0 = \tilde{\mathsf{x}}_0 + \mathsf{w}_0,$$

where $\tilde{\mathsf{x}}_0$ and w_0 are the random variable associated to $\tilde{\lambda}_0$ and η_0, respectively.

2. for all $a_0, \dots, a_{\ell-1} \in A^\ell$,

$$\frac{\tilde{\nu}_0^{\ell+1}(a_0, \dots, a_\ell)}{\tilde{\nu}_0^{\ell}(a_0, \dots, a_{\ell-1})} = \tau^\ell(a_\ell | a_0, \dots, a_{\ell-1}). \tag{3.7}$$

Such measure $\overline{\lambda}_0$ is defined as, for all $A \subseteq [\![a_0, \dots, a_{\ell-1}]\!]$,

$$\overline{\lambda}_0(A) := \mu(A) \frac{\tilde{\nu}_0^{\ell}(a_0, \dots, a_{\ell-1})}{\mu^\ell([\![a_0, \dots, a_{\ell-1}]\!])}.$$

The second point above can be directly verified by computation. The first point follows from the fact that

$$\overline{\lambda}_0([\![a_0, \dots, a_{\ell-1}]\!]) = \tilde{\nu}_0^{\ell}(a_0, \dots, a_{\ell-1}) = \tilde{\lambda}_0([\![a_0, \dots, a_{\ell-1}]\!]),$$

meaning that any realisation $\overline{x}_0$ can be written as $\tilde{x}_0 + w_0$, with $\|w_0\| \leq \varepsilon_\ell$, by (3.5), since ε_ℓ is the maximum diameter of cells $[\![a_0, \dots, a_{\ell-1}]\!]$.

Now, let $\tilde{\lambda}_1 = f\#\overline{\lambda}_0$. Again, we can show the construction of a measure

$$\tilde{\lambda}_1(A) := \mu(A) \frac{\tilde{\nu}_1^{\ell}(a_1, \dots, a_\ell)}{\mu^\ell([\![a_1, \dots, a_\ell]\!])}$$

such that there exists $\eta_1 : \sigma(W_{\varepsilon_\ell}) \to [0, 1]$ for which $\overline{\mathrm{x}}_1 = \tilde{\mathrm{x}}_1 + \mathrm{w}_1$, where $\tilde{\mathrm{x}}_1$ and w_1 are the random variable associated to $\tilde{\lambda}_1$ and η_1, respectively, and such that

$$\frac{\tilde{\nu}_1^{\ell+1}(a_1, \dots, a_{\ell+1})}{\tilde{\nu}_1^{\ell}(a_1, \dots, a_\ell)} = \tau^\ell(a_{\ell+1} | a_1, \dots, a_\ell). \tag{3.8}$$

The same reasoning can be done recursively on $k = 1, 2, \dots$. Equation (3.6), which yields that the perturbation is initialised similarly as the ℓ-memory Markov abstraction, and the transition equalities for all k (in (3.7) and (3.8) for $k = 0$ and $k = 1$, respectively) imply that the behaviours of the perturbation and the ℓ-memory Markov abstraction are the same. □

Theorem 3.15 is central in our work because it justifies the method presented in Algorithm 1. It indeed guarantees that the behaviour of Γ_ℓ is the same as the behaviour of a perturbation whose noise is decreasing (and goes to zero) in magnitude, thereby making it closer and closer to the behaviour of the original system. An illustration of this result is provided in Figure 3.2. While this result is asymptotic in nature and does not provide explicit error bounds for any fixed memory ℓ, it offers a theoretical founda-

tion for a practically appealing heuristic: the more memory is incorporated into the abstraction, the more faithful it becomes to the original system.

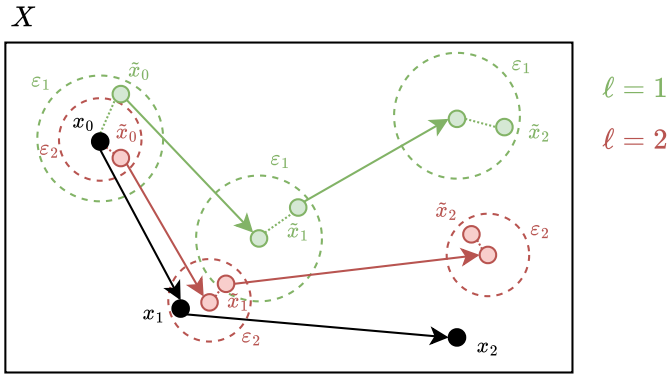


Fig. 3.2 Illustration of Theorem 3.15. The black trajectory is an example of trajectory of the original system. Let Γ_1 be the 1-memory Markov abstraction of the original system, Theorem 3.15 guarantees that the behaviour of this abstraction is the same as the perturbation $\tilde{\Sigma}_{\varepsilon_1}$, for which an example of perturbed trajectory is depicted in green. We also illustrate a perturbed trajectory of $\tilde{\Sigma}_{\varepsilon_2}$, the perturbation corresponding to memory $\ell = 2$, and whose perturbation has therefore a maximum norm of $\varepsilon_2 \leq \varepsilon_1$.

3.4 Numerical experiments

In this section we give numerical evidence that the behaviour of the ℓ-memory abstraction is closer and closer to the true behaviour $\mathcal{B}(\Sigma, \lambda_0)$.

We perform our analysis on the Sturmian dynamics, because it is a typical example of a system that necessitates infinite memory to be exactly represented (see Example 2.7). We consider an angle $\varphi = \sqrt{2}/5$, and we assume that the states are sampled according to a uniform distribution on $X = [0, 1)$, that is λ_0 is the Lebesgue measure on X. We perform the following experiment:

- for $k = 1, \ldots, 20$, we compute $\mathcal{B}_k(\Sigma, \lambda_0)$,[3]

[3]This amounts to computing all the possible *Sturmian words* of length k [Fog02].

- for memories $\ell = 1, \dots, 6$:
  - we compute Γ_ℓ according to Algorithm 1,
  - for $k = 1, \dots, 20$, we compute $\mathcal{B}_k(\Gamma_\ell)$ and compute

$$\Delta_k^\ell := |\mathcal{B}_k(\Gamma_\ell) \setminus \mathcal{B}_k(\Sigma, \lambda_0)| . \tag{3.9}$$

Δ_k^ℓ is a measure of the abstraction quality for a fixed horizon k. Indeed, by construction, it holds that $\mathcal{B}_k(\Sigma, \lambda_0) \subseteq \mathcal{B}_k(\Gamma_\ell)$ for all k – the abstraction is an *over-approximation* of the original dynamics – and therefore the set $\mathcal{B}_k(\Gamma_\ell) \setminus \mathcal{B}_k(\Sigma, \lambda_0)$ is the set of all *spurious* trajectories in the abstraction.[4] Note that, again by construction, it holds that $\Delta_k^\ell = 0$ for all $k \leq \ell + 1$.

The values of Δ_k^ℓ for the Sturmian dynamics are collected in Table 3.1. Our results confirm that increasing the memory length ℓ generally improves the quality of the abstraction, though the improvement is not strictly monotonic (for instance, the 3-, 4-, and 5-memory Markov abstractions exhibit identical behaviour across all values of k). Notably, even a modest increase in memory can yield substantial improvements: at $k = 20$, the number of spurious trajectories drops from over 2000 with memory 2 to fewer than 100 with memory 3.

3.5 Summary and further research directions

Chapter 3 – Summary

In this chapter, we provided a method to construct a data-driven and uniform memory abstraction of deterministic dynamical systems, and proved its correctness.

Section 3.1 – Framework

This section establishes the theoretical foundation for constructing uniform memory-dependent abstractions of deterministic dynamical systems. We describe the exact framework considered in this chapter, that is partially observable deterministic dynamical systems whose initial state is sampled. A key assumption is the existence of an invariant measure μ (Assumption 3.1), which ensures that the abstraction that we will construct is well-defined. Finally the section introduces probability measures on state and output sequences arising from the initial

[4] This result will be formally proven in Chapter 4 in a more general framework.

	$\ell = 1$ (2)	$\ell = 2$ (4)	$\ell = 3$ (8)	$\ell = 4$ (16)	$\ell = 5$ (32)	$\ell = 6$ (64)
$k = 1$	**0**	**0**	**0**	**0**	**0**	**0**
$k = 2$	**0**	**0**	**0**	**0**	**0**	**0**
$k = 3$	1	**0**	**0**	**0**	**0**	**0**
$k = 4$	3	1	**0**	**0**	**0**	**0**
$k = 5$	7	3	0	**0**	**0**	**0**
$k = 6$	14	6	0	0	**0**	**0**
$k = 7$	26	11	1	1	1	**0**
$k = 8$	46	19	2	2	2	0
$k = 9$	79	31	3	3	3	0
$k = 10$	133	49	5	5	5	0
$k = 11$	221	76	8	8	8	1
$k = 12$	364	116	11	11	11	2
$k = 13$	596	175	15	15	15	3
$k = 14$	972	262	21	21	21	4
$k = 15$	1581	390	28	28	28	6
$k = 16$	2567	578	36	36	36	8
$k = 17$	4163	854	47	47	47	10
$k = 18$	6746	1259	61	61	61	13
$k = 19$	10926	1853	77	77	77	17
$k = 20$	17690	2724	97	97	97	21

Table 3.1 Values of Δ_k^ℓ as defined in (3.9) (number of spurious output trajectories) for the Sturmian dynamics, for $\ell = 1, \ldots, 6$ and $k = 1, \ldots, 20$. The zeros in bold could have been predicted by construction of Γ_ℓ. The numbers in parenthesis are the size (i.e. the number of nodes) of the resulting abstraction.

measure, which are critical for defining the behaviour of the system, crucial to analyse the abstraction quality.

Section 3.2 – Method

We propose a method to construct a ℓ-memory Markov model Γ_ℓ that approximates the behaviour of the original system. The abstraction is built in three steps: (1) sampling output trajectories from the system, (2) counting frequencies of sequences, and (3) using these frequencies to define the initial and transition probabilities of the ℓ-memory Markov model.

Section 3.3 – Guarantee

This section proves the asymptotic correctness of the abstraction. We introduce the concept of a perturbation $\tilde{\Sigma}_\varepsilon$, where the state is subject to additive noise bounded by ε. The main result (Theorem 3.15) shows that the behaviour of Γ_ℓ matches that of a perturbation $\tilde{\Sigma}_{\varepsilon_\ell}$, with $(\varepsilon_\ell)_{\ell\geq 1}$ monotonically decreasing to zero as ℓ increases. This implies that as memory ℓ grows, the abstraction behaviour converges to the true system dynamics, with the perturbation vanishing in the limit. The proof relies on the asymptotic observability of the system and the shrinking diameter of state sets corresponding to longer output sequences, ensuring the abstraction becomes arbitrarily precise.

Section 3.4 – Numerical experiments

Numerical experiments on the Sturmian dynamics (see Example 2.7) validate the theoretical guarantees. We compute the number of spurious trajectories (trajectories in Γ_ℓ but not possible in the original system) for varying memories ℓ and horizons k. Results in Table 3.1 demonstrate that increasing ℓ significantly reduces the number of spurious trajectories, confirming that longer memory yields more accurate abstractions.

Chapter 3 – Further research directions

In general, we believe that this method is a genuine alternative to classical state space refinement methods, and offers a practical alternative to decrease epistemic uncertainty. The following directions can be considered for future research.

Relate true and perturbed behaviours

While Theorem 3.15 establishes that the behaviour of the ℓ-memory Markov abstraction Γ_ℓ matches that of a perturbation $\tilde{\Sigma}_{\varepsilon_\ell}$ with vanishing noise, a deeper theoretical connection between the perturbed and true system behaviours remains to be explored. One promising direction is to quantify how the perturbation ε_ℓ propagates through the system dynamics, particularly under additional regularity assumptions such as Lipschitz continuity of the transition function f: if f is Lipschitz, the effect of the perturbation on the state trajectory could be bounded explicitly over time. This would allow for rigorous error

bounds between the trajectories of the true system and those of its perturbation, providing a better understanding of how the abstraction error decays with increasing memory ℓ. Another direction is to consider contraction or incremental stability of the system, which would imply that the long-term error scales with ε_ℓ or vanishes completely. Such results would not only strengthen the theoretical guarantees but also offer practical insights into the robustness of the abstraction method.

Finite number of observations

The current framework assumes access to an infinite number of samples (Assumption 3.8), which is unrealistic in practice. A critical extension is to develop finite-sample guarantees for the ℓ-memory Markov abstraction. As highlighted in Remark 3.9, techniques such as the scenario approach [CG19], conformal prediction [CPLM24], or compression learning [MPL15] could be leveraged to provide a probabilistic version of Theorem 3.15.

Adding control inputs

The current work focuses on autonomous deterministic systems, but many applications involve controlled systems of the form $x_{k+1} = f(x_k, u_k)$, where u_k is a control input. Extending the framework to such systems would require incorporating control-dependent transitions into the ℓ-memory Markov model, making it a *Markov Decision Process* (or *MDP* for short, see [SB] for an introduction).

4

Adaptive memory refinement

In the previous chapter, we introduced uniform memory-dependent abstractions for deterministic dynamical systems, where the abstraction memory length ℓ was fixed and applied uniformly across all states. While this approach guarantees asymptotic correctness as ℓ increases, it may lead to unnecessary computational complexity, especially when only certain regions of the state space require longer memory to accurately capture the system behaviour.

This chapter extends those results by developing *adaptive memory* abstractions, where the memory length is not uniform but instead tailored to the local dynamics of the system, as illustrated in Figure 4.1. Instead of encoding fixed-length output sequences, the abstraction now uses timed sequences of varying lengths, allowing for a more efficient and precise representation of the system behaviour. By refining the memory adaptively (only where needed), we aim to balance accuracy and computational tractability, ensuring that the abstraction remains both data-driven and *sound* (in the sense that it is an over-approximation of the original dynamics). Note that our focus in this chapter is therefore different than in Chapter 3: instead of guaranteeing *asymptotic correctness* as the memory length grows uniformly, we aim to *quantify behavioural richness* and to prove that the resulting abstractions are sound.

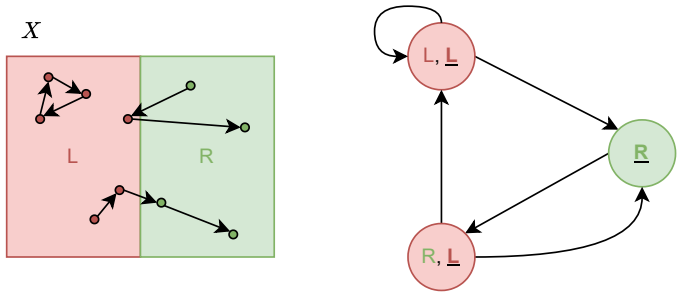


Fig. 4.1 Illustration of an adaptive memory-dependent abstraction of a dynamical system. On the left: a partially observable deterministic dynamical system with $A = \{\mathrm{L}, \mathrm{R}\}$. On the right: an LMC whose nodes encode an adaptive memory scheme. When the current output is L, the previous output is encoded. However, when the current output is R, the past is forgotten, and no memory is considered.

The adaptive memory scheme is determined by a greedy refinement algorithm, which iteratively adjusts the memory depth in regions where the abstraction is insufficiently precise. The algorithm relies on a notion of distance between labelled Markov chains (LMCs), quantifying the improvement of approximation quality from iteration to iteration. Even though, in principle, any distance may be used for our purposes, we propose in Chapter 6 a new distance, which we call the *Cantor-Kantorovich distance*.

Iterative refinement of the state space has been explored in the literature within abstraction-based methods, for instance in [EZSA13]. However, these existing approaches do not incorporate memory-dependent abstractions. The work in [YMR20] is the most closely related to ours, as it also introduces adaptive memory schemes through sets of finite-length sequences of varying lengths. However, [YMR20] adopts a model-based approach, relying on explicit knowledge of the system dynamics and a specification-driven refinement process. In contrast, our work presents a data-driven and specification-free framework. We construct abstractions directly from sampled trajectories, allowing the memory length to adapt locally to the system's behaviour without requiring a predefined model or control objective.

Contributions. In this chapter we introduce data-driven and adaptive memory abstractions of deterministic dynamical systems. The main contributions are summarised below.

1. **A general memory-dependent framework.** We formalise a general framework for abstractions that encode both past and future memory sequences, possibly of different lengths, in their states.
2. **Guarantee.** We prove that, under some assumptions on the dynamics and sampling procedure, such abstractions are sound in the sense that they over-approximate the original dynamics. We further show that this soundness guarantee extends to infinite output trajectories: any B-LTL specification (see Chapter 1) verified almost surely on the abstraction is certified to hold for all infinite output trajectories of the original system.
3. **Method.** We propose a practical method to construct a so-called *adaptive-memory Markov model* that captures the behaviour of the original partially observable dynamical system, as well as a refinement procedure to decide a memory scheme for the abstraction. We will prove that this method leads to the guarantee presented earlier.
4. **Numerical experiments.** We provide empirical evidence that increasing the memory leads to a better abstraction. We show this in three ways: (1) we show that our method is able to create an abstraction whose partition aligns well with the dynamics, (2) we compare our method to the uniform memory presented in Chapter 3 and (3) we apply our method to the safety analysis of a real-life system.

Outline. Section 4.1 introduces the mathematical framework for adaptive memory abstractions, extending the deterministic dynamical systems setting from Chapter 3. Section 4.2 formalises the concept of *adaptive memory abstractions*, defining *timed sequences*, *adaptive memory partitions*, and the corresponding labelled Markov chains. It also establishes the theoretical guarantees of soundness (behavioural inclusion) under appropriate assumptions, and derives a corollary showing that any B-LTL property verified on the abstraction certifies all infinite output trajectories of the original system. Section 4.3 presents the REFINE algorithm, a data-driven procedure for iteratively refining the memory scheme to improve abstraction accuracy. The correctness of the algorithm is proven. Section 4.4 demonstrates the practical benefits of the approach through three numerical ex-

periments. Finally, Section 4.5 summarises the contributions and discusses potential directions for future research.

Publication. This work has been done in collaboration with Licio Romao, Alessandro Abate and Raphaël M. Jungers, and has been published in [BRAJ23a, BRAJ25].

4.1 Framework

In this chapter we consider a similar framework as in Chapter 3, namely deterministic dynamical systems of the form $\Sigma = (X, f, A, h)$ where $X \subseteq \mathbb{R}^d$ and is endowed with the σ-algebra $\sigma(X)$, whose state and output trajectories are driven by

$$\begin{cases} x_{k+1} = f(x_k), \\ y_k = h(x_k). \end{cases}$$

Again, we consider that we can sample observations, that is, we assume that the initial state of Σ is random, and therefore that there exists a probability measure $\lambda_0 : \sigma(X) \to [0, 1]$.

In this chapter, we will formalise a general notion of *memory-dependent* abstraction for such dynamical systems. Our main focus is on proving that *behavioural inclusion* holds for these abstractions, that is, whether

$$\mathcal{B}(\Sigma, \lambda_0) \subseteq \mathcal{B}(\Gamma)$$

holds, where Γ is a memory-dependent abstraction of the system. Such abstractions are called sound abstractions, because any "safety" property proven on those implies that the system exhibits a similar property, although conservativeness can be observed. More precisely, we will show that behavioural inclusion implies that any B-LTL specification (see Chapter 1) verified almost surely on Γ holds for all infinite output trajectories of Σ. In order to be able to both (1) write a common formalism and (2) prove behavioural inclusion, we restrict our analysis to *invertible* dynamics.

Assumption 4.1. The transition function f is invertible.

This assumption is needed to handle abstractions that incorporate *past* memory, ensuring that every state has a unique, well-defined past trajectory. It is not required when restricting to future memory only; see Remark 4.4 for a more formal discussion.

4.2 Adaptive memory abstractions

In this section we introduce a general framework for memory-dependent abstractions of sampled deterministic dynamical systems. This framework includes both past and future memory sequences, possibly of varying lengths.

We first introduce timed sequences, defined as follows.

Definition 4.2 (Timed sequence)**.** Let A be an alphabet, and let $s, t \in \mathbb{N}_0$. A *timed sequence* over the time window $[-s, t]$ is a map

$$w : \{-s, -s+1, \dots, t\} \to A.$$

We write it as

$$w_{[-s,t]} = (a_{-s}, a_{-s+1}, \dots, a_t),$$

where $a_i := w(i)$ for each $i \in \{-s, \dots, t\}$.

Remark 4.3. The ℓ-memory sequences of Chapter 3 are a special case of timed sequences: a fixed memory length ℓ corresponds to restricting all timed sequences to the time window $[-\ell+1, 0]$, i.e. $s = \ell - 1$ and $t = 0$, using only past outputs and no future lookahead.

As in Chapter 3, an abstraction is defined by a partition of the state space. A timed sequence $w_{[-s,t]} = (a_{-s}, \dots, a_t)$ defines a subset of the state space as follows:

$$\left[\!\left[w_{[-s,t]} \right]\!\right] := \left\{ x_0 \in X \,\middle|\, \exists x_{-s} \in X : \begin{array}{l} f^s(x_{-s}) = x_0, \\ h\Big(f^{i+s}(x_{-s})\Big) = a_i, \end{array} \quad \forall i \in \{-s, \dots, t\} \right\}.$$

Intuitively, a timed sequence $w_{[-s,t]}$ is a sequence of outputs $w \in A^{s+t+1}$ linked to a time window from $k = -s$ to $k = t$. The index 0 corresponds to the current time, negative indices denote past outputs and positive indices denote future outputs. The corresponding $\left[\!\left[w_{[-s,t]} \right]\!\right]$ is the set of states at $k = 0$ compatible with this trajectory. An illustration is provided in Figure 4.2.

Remark 4.4. Assumption 4.1 is needed precisely because blocks $[\![w_{[-s,t]}]\!]$ with $s > 0$ encode *past* observations. If f were not injective, a given state x_0 could admit two distinct pre-images $x_{-1} \neq x'_{-1}$ with $h(x_{-1}) \neq h(x'_{-1})$, so x_0 would simultaneously belong to two different past-memory blocks,

violating the partition property. In contrast, if all timed sequences are restricted to future memory only (i.e., $s = 0$), the blocks reduce to

$$\left[\!\left[w_{[0,t]}\right]\!\right] = \left\{x_0 \in X \;\middle|\; h\left(f^i(x_0)\right) = a_i,\ \forall i \in \{0,\dots,t\}\right\},$$

which are well-defined and non-overlapping for any f. In this case, Assumption 4.1 is not required, and behavioural inclusion (Proposition 4.10) holds under Assumption 4.9 alone.

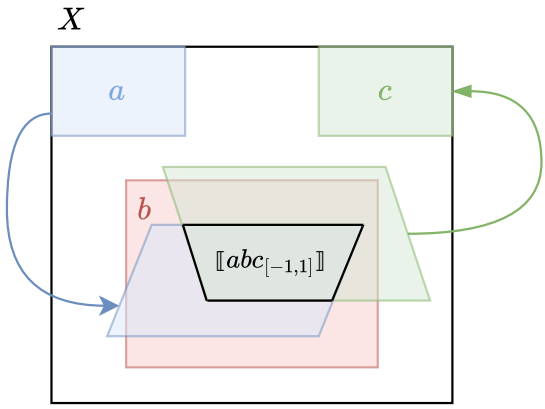


Fig. 4.2 Illustration of a subset $[\![abc_{[-1,1]}]\!]$. The sets of states $A := \{x : h(x) = a\}$ and $f(A) := \{x : h(f^{-1}(x)) = a\}$ are depicted in blue, the set of states $B := \{x : h(x) = b\}$ is depicted in red, and the sets of states $C := \{x : h(x) = c\}$ and $f^{-1}(C) := \{x : h(f(x)) = c\}$ are in green. The intersection of $f(A)$, B and $f^{-1}(C)$ is the set of states x_0 such that, previously, $h(x_{-1}) = a$, currently $h(x_0) = b$ and, subsequently, $h(x_1) = c$.

The partition corresponding to a set of words is defined as follows.

Definition 4.5 (Adaptive memory partition)**.** A set of timed sequences

$$W = \left\{w^1_{[-s_1,t_1]}, \dots, w^K_{[-s_K,t_K]}\right\}$$

defines a *partition* of the state space if

$$\bigcup_{k=1,\dots,K} \left[\!\left[w^k_{[-s_k,t_k]} \right]\!\right] = X,$$

$$\forall k \neq l : \left[\!\left[w^k_{[-s_k,t_k]} \right]\!\right] \cap \left[\!\left[w^l_{[-s_l,t_l]} \right]\!\right] = \emptyset.$$

In this case, a subset $\left[\!\left[w^k_{[-s_k,t_k]} \right]\!\right]$ is called a *block*.

Given any set of timed sequences defining a partition, we are now able to define its corresponding abstraction. The resulting abstractions are called *adaptive memory Markov models*, because they encode memory schemes of *adaptive* lengths.

Definition 4.6 (Adaptive memory Markov model)**.** Let W be a set of timed sequences. An *adaptive memory Markov model* is a LMC

$$\Gamma_W := (S_W, A, \iota_W, \tau_W, l_W)$$

where

- the states S_W are the timed sequences, that is $S_W := W$.
- $\tau_W : 2^W \times W \to [0,1]$ is such that, for any two timed sequences $w^1_{[-s_1,t_1]}, w^2_{[-s_2,t_2]} \in W$, if

  $$\tau_W \left(w^2_{[-s_2,t_2]} \middle| w^1_{[-s_1,t_1]} \right) \geq 0,$$

  then $w^1_{[-s_1,t_1]}$ and $w^2_{[-s_2,t_2]}$ are of the form

  $$w^1_{[-s_1,t_1]} = (a_{-s_1}, \dots, a_{-s_3-1}, c_{-s_3}, \dots, c_{t_3})$$
  $$w^2_{[-s_2,t_2]} = (c_{-s_3}, \dots, c_{t_3}, b_{t_3+1}, \dots, b_{t_2}),$$

  i.e. they overlap.
- the labelling function $l_W : W \to A$ is defined as, for all $w_{[-s,t]} = (a_{-s}, \dots, a_t) \in W$, $l_W(w_{[-s,t]}) := a_0$.

This model generalises the ℓ-memory Markov model of Definition 3.4, as noted in Remark 4.3.

We are now able to define adaptive memory abstractions.

Definition 4.7 (Adaptive memory abstraction)**.** Let $\Sigma = (X, f, A, h)$ be a deterministic dynamical system whose initial state is sampled according to λ_0, and let W be an adaptive memory partition that satisfies Assumption 4.8. An adaptive memory abstraction for Σ is an adaptive memory Markov model $\Gamma_W := (S_W, A, \iota_W, \tau_W, l_W)$ (defined in Definition 4.6) such that, for all $w_{[-s,t]} \in W$,

$$\iota_W\left(w_{[-s,t]}\right) := \lambda_0\left(\left[\!\left[w_{[-s,t]}\right]\!\right]\right)$$

and, for any two

$$w^1_{[-s_1,t_1]} = (a_{-s_1}, \ldots, a_{-s_3-1}, c_{-s_3}, \ldots, c_{t_3})$$
$$w^2_{[-s_2,t_2]} = (c_{-s_3}, \ldots, c_{t_3}, b_{t_3+1}, \ldots, b_{t_2}),$$

the transition kernel τ_W is defined as

$$\tau_W\left(w^2_{[-s_2,t_2]} \middle| w^1_{[-s_1,t_1]}\right) := \frac{\lambda_0\left(\left[\!\left[w^{1,2}_{[-s_1,t_2]}\right]\!\right]\right)}{\lambda_0\left(\left[\!\left[w^1_{[-s_1,t_1]}\right]\!\right]\right)}. \tag{4.1}$$

with

$$w^{1,2}_{[-s_1,t_2]} = (a_{-s_1}, \ldots, c_{-s_3}, \ldots, c_{t_3}, \ldots, b_{t_2}).$$

Intuitively, an adaptive memory abstraction groups initial states by the output sequences they produce over some (both past and future) time window: two initial states belong to the same block if and only if they generate identical outputs over that window. Transitions between blocks are governed by the conditional probability (4.1), which, given that a trajectory currently lies in block $w^1_{[-s_1,t_1]}$, measures the likelihood of it moving into a block $w^2_{[-s_2,t_2]}$ that, with respect to $w^1_{[-s_1,t_1]}$, encodes new future observations and erases some past observations.

In the following, we will show that such abstractions are sound in the sense that behavioural inclusion holds. Because this abstraction is only constructed from sampled observations, some assumptions on the sampling measure are needed.

Assumption 4.8. The set of timed sequences W is such that, for all $w_{[-s,t]} \in W$,

$$\lambda_0\left(\left[\!\left[w_{[-s,t]}\right]\!\right]\right) > 0.$$

Assumption 4.8 is a natural necessary condition in the context of data-driven methods, and ensures that the conditional probability (4.1) is well-defined.

We will also need the pair (f, λ_0) to satisfy the so-called *Lusin N^{-1} condition* (see e.g. [HK14, Section 4.4]), defined as follows.

Assumption 4.9 (Lusin N^{-1} condition)**.** For all $A \in \sigma(X)$, if $\lambda_0(A) = 0$, then $\lambda_0(f^{-1}(A)) = 0$.

Assumption 4.9 is commonly satisfied in many practical situations. This includes *homeomorphisms*, i.e. continuous and invertible mappings whose inverse is continuous (such as most linear systems), though it is not restricted to these [HK14].

We are now able to formally prove behavioural inclusion for such abstractions.

Proposition 4.10. *Let $\Sigma = (X, f, A, h)$ be a deterministic dynamical system whose initial state is sampled according to λ_0, and let Γ_W be an adaptive memory abstraction for Σ given an adaptive memory partition W. Then, if Assumption 4.8 and Assumption 4.9 are satisfied, then it holds that $\mathcal{B}(\Sigma, \lambda_0) \subseteq \mathcal{B}(\Gamma_W)$.*

Proof. It suffices to show that, for all $k \in \mathbb{N}$, for all timed sequences $w_{[0,k-1]} = (b_0 \dots b_{k-1})$, if $\lambda_0([\![w_{[0,k-1]}]\!]) > 0$ in the original system, then $\mu^k(w) > 0$ in the LMC Γ_W, where $\mu^k : 2^{A^k} \to [0,1]$ is the probability measure on the k-long language A^k generated by Γ_W, and where $w = (b_0, \dots, b_{k-1}) \in A^k$. Since Assumption 4.8 is satisfied, the condition $\mu^k(w) > 0$ holds if there exists a sequence of timed sequences $w^0_{[-s_0,t_0]}, \dots, w^{k-1}_{[-s_{k-1},t_{k-1}]} \in W$ such that $l_W\left(w^i_{[-s_i,t_i]}\right) = b_i$ for all $i = 0, \dots, k-1$, and

$$\lambda_0\left(\left[\!\left[w^i_{[-s_i,t_i]}\right]\!\right] \cap f^{-1}\left(\left[\!\left[w^{i+1}_{[-s_{i+1},t_{i+1}]}\right]\!\right]\right)\right) > 0$$

for all $i = 0, \dots, k-2$. We will therefore prove this property.

First, it holds that

$$\left[\!\left[w_{[0,k-1]}\right]\!\right] = \bigcap_{i=0,\dots,k-1} f^{-i}\left(\left[\!\left[(b_i)_{[0,0]}\right]\!\right]\right),$$

where $(b)_{[0,0]}$ denotes the timed sequence composed only of the letter b.

Moreover, since W defines a partition, it holds that

$$\left[\!\left[(b_i)_{[0,0]}\right]\!\right] = \bigcup_{s_i : l_W(s_i) = b_i} [\![s_i]\!]$$

for all $i = 0, \dots, k-1$. Therefore, one can write

$$\begin{aligned}\left[\!\left[w_{[0,k-1]}\right]\!\right] &= \bigcap_{i=0,\dots,k-1} f^{-i}\left(\bigcup_{s_i : l_W(s_i) = b_i} [\![s_i]\!]\right) \\ &= \bigcap_{i=0,\dots,k-1} \bigcup_{s_i : l_W(s_i) = b_i} f^{-i}\left([\![s_i]\!]\right).\end{aligned} \tag{4.2}$$

Now, since $\lambda_0\left(\left[\!\left[w_{[0,k-1]}\right]\!\right]\right) > 0$, then there exists at least a sequence $s_0, \dots, s_{k-1} \in W$ such that $l(s_i) = b_i$ and such that

$$\lambda_0\left(\bigcap_{i=0,\dots,k-1} f^{-i}\left([\![s_i]\!]\right)\right) > 0.$$

Furthermore, one can say that the inequality above implies that

$$\lambda\left(f^{-i}\left([\![s_i]\!]\right) \cap f^{-(i+1)}\left([\![s_{i+1}]\!]\right)\right) > 0 \tag{4.3}$$

for all $i = 0, \dots, k-2$. This is explained by the fact that, for any three sets A, B, C, if $\lambda_0(A \cap B \cap C) > 0$, then it holds that $\lambda_0(A \cap B) > 0$, $\lambda_0(A \cap C) > 0$ and $\lambda_0(B \cap C) > 0$. By invertibility of f, (4.3) implies that $\lambda\left([\![s_i]\!] \cap f^{-1}\left([\![s_{i+1}]\!]\right)\right) > 0$ for all $i = 0, \dots, k-2$, and the proof is completed. □

While Proposition 4.10 establishes inclusion of *finite* output words, it also carries implications for *infinite* output trajectories. To formalise this, we first extend the notion of system behaviour to infinite traces.

Definition 4.11 (System infinite behaviour)**.** Let $\Sigma = (X, f, A, h)$ be a deterministic system whose initial state is sampled according to λ_0. The set $\mathcal{B}^\omega(\Sigma, \lambda_0) \subseteq A^\omega$ defined as

$$\mathcal{B}^\omega(\Sigma, \lambda_0) = \{(a_0, a_1, \dots) \in A^\omega : \nu_0^k(a_0, \dots, a_{k-1}) > 0 \text{ for all } k \in \mathbb{N}\},$$

is called the *infinite behaviour* of the system. Equivalently, $\mathcal{B}^\omega(\Sigma, \lambda_0)$ is the set of infinite output traces all of whose finite prefixes belong to $\mathcal{B}(\Sigma, \lambda_0)$

(see Definition 3.3).

We now turn to the specification-level implications of behavioural inclusion. Recall from Chapter 1 that a *(B-LTL)-definable property* (Definition 1.14) is a temporal specification whose syntax is that of LTL but restricted to the $\bigcirc$ (next) modality, without the $\mathcal{U}$ (until) operator. A key structural result (Proposition 1.16) states that any such property is a *finite union of cylinder sets*, meaning its satisfaction depends only on bounded-length output prefixes. This is precisely the class of specifications for which the finite behavioural inclusion of Proposition 4.10 transfers to a guarantee on infinite trajectories.

Corollary 4.12. *Under the conditions of Proposition 4.10, for any (B-LTL)-definable property $\phi \subseteq A^\omega$ (see Definition 1.14), if the probability measure μ^ω on A^ω generated by Γ_W satisfies $\mu^\omega(\phi) = 1$, then $\mathcal{B}^\omega(\Sigma, \lambda_0) \subseteq \phi$.*

Proof. Since B-LTL is closed under negation (Definition 1.14), $\neg\phi$ is also (B-LTL)-definable. By Proposition 1.16, $\neg\phi$ is a finite union of cylinder sets, i.e. there exists finite words $w_1, \dots, w_m \in A^\star$ such that $\neg\phi = \bigcup_{i=1}^m w_i A^\omega$. Since $\mu^\omega(\phi) = 1$, we have $\mu^\omega(\neg\phi) = 0$, and therefore, by definition of μ^ω, $\mu^{|w_i|}(w_i) = 0$ for all $i = 1, \dots, m$. This yields all words $w_i \notin \mathcal{B}(\Gamma_W)$, and therefore $w_i \notin \mathcal{B}(\Sigma, \lambda_0)$ by Proposition 4.10. By definition of the behaviour, this means that, for all $i = 1, \dots, m$, $\nu_0^{|w_i|}(w_i) = 0$, and therefore $\mathcal{B}^\omega(\Sigma, \lambda_0) \cap w_i A^\omega = \emptyset$, equivalent to $\mathcal{B}^\omega(\Sigma, \lambda_0) \cap \neg\phi = \emptyset$, which proves the claim. □

Corollary 4.12 gives a concrete verification certificate: it suffices to check that a safety B-LTL property holds almost surely in the finite-state abstraction Γ_W to certify that it holds for every infinite output trajectory of the original sampled system Σ. This covers practically relevant specifications such as bounded safety and reachability.

4.3 Refinement procedure

In the previous section, the choice of a convenient set W is not discussed. In this section, we present a data-driven procedure that tackles this problem, and we will prove its correctness. We first show in Algorithm 2 how to construct an adaptive memory Markov abstraction from sampled data.

Assumption 4.13. We have access to infinitely many samples in Algorithm 2. That is, one can compute exactly F_{λ_0}, as described in (3.1).

Algorithm 2 Construction of an adaptive memory Markov abstraction Γ_W

1: Generate output trajectories of the system Σ whose initial state is sampled according to the invariant measure λ_0 and, for all timed sequences $w_{[0,t]} = (a_0, \ldots, a_t) \in W$, compute

$$F_{\lambda_0}(a_0, \ldots, a_t),$$

where F_{λ_0} is the *frequency* as defined in Chapter 3. Set

$$\iota_W(w_{[0,t]}) := F_{\lambda_0}(a_0, \ldots, a_t),$$

2: For all pairs $w^1_{[0,t_1]} = (a_0, \ldots, a_{t_1})$ and $w^2_{[0,t_2]} = (b_0, \ldots, b_{t_2})$ such that $t_2 \geq t_1$ and such that

$$(a_0, \ldots, a_{t_1}) = (b_0, \ldots, b_{t_1}),$$

compute

$$F_{\lambda_0}(a_0, \ldots, a_{t_1}, b_{t_1+1}, \ldots, b_{t_2}),$$

and set

$$\tau_W\left(w^2_{[0,t_2]} \middle| w^1_{[0,t_1]}\right) = \frac{F_{\lambda_0}(a_0, \ldots, a_{t_1}, b_{t_1+1}, \ldots, b_{t_2})}{F_{\lambda_0}(a_0, \ldots, a_{t_1})}.$$

return The adaptive memory Markov model $\Gamma_W := (W, A, \tau_W, l_W)$.

Algorithm 2 and Assumption 4.13 are the generalizations of Algorithm 1 and Assumption 3.8 from Chapter 3. We invite the reader to read Remark 3.9 for a discussion of Assumption 4.13.

We now describe our refinement procedure. This algorithm, called REFINE, takes as input a dynamical system Σ, an initial measure λ_0, a distance D between labelled Markov chains (see Section 1.2.2 for a formal definition of distance between LMCs), and a stopping criterion. The latter can correspond to a maximal number of iterations, or a maximal abstraction size, for instance; the choice is arbitrary, though the procedure should terminate. It is described in Algorithm 3, and an illustration of this procedure is given in Figure 4.3.

Algorithm 3 works as follows. We start in step 1 from the initial partition $\{(a)_{[0,0]}\}_{a \in A}$. We then sample output trajectories from the system in step 2, and discard subsets of outputs that are never sampled (step 3). Finally, we construct the adaptive memory abstraction Γ_W by counting the

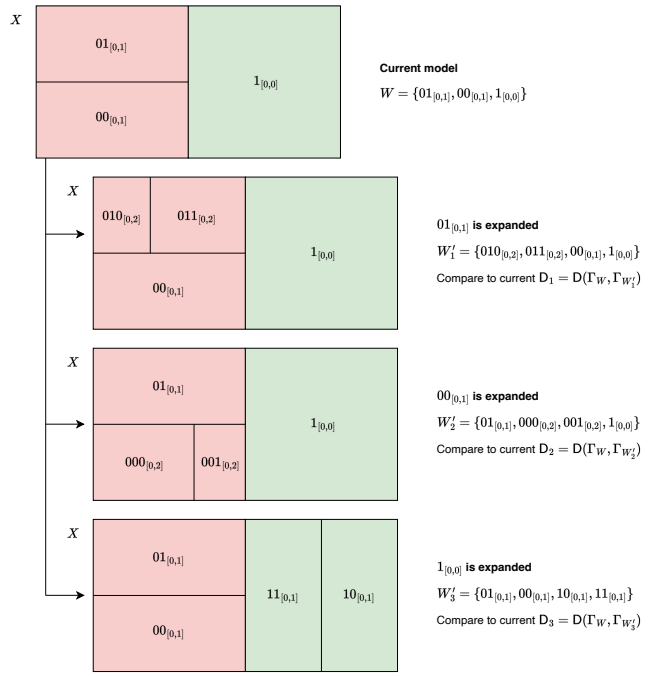


Fig. 4.3 Illustration of the REFINE algorithm for a system with $A = \{0,1\}$. The current model contains three states $W = \left\{01_{[0,1]}, 00_{[0,1]}, 1_{[0,0]}\right\}$. Three sets W_1', W_2', W_3' are then constructed, each time expanding one block of the partition into $|A| = 2$ sub-blocks. For each abstraction $\Gamma_{W_i'}$, the distance with the current is computed. The current model is updated with the model for which the distance is the largest.

initial observations and the transitions (step 4). The rest of the algorithm follows similar steps: we try to expand the partition (steps 7 and 8); we sample output trajectories and discard the unsampled blocks (steps 9 and 10); we construct the adaptive memory abstraction by counting the initial observations and transitions (step 11). For each considered partition, we compute the distance with the current partition (step 12), and we update the partition by choosing the one that has the largest distance with the current one (steps 14, 15 and 16).

The output W from the REFINE algorithm depends on the chosen distance between the abstractions. In Chapter 6, we will introduce a novel distance between LMCs, the so-called *Cantor-Kantorovich metric*, and dis-

Algorithm 3 REFINE

$W \leftarrow \{(a)_{[0,0]}\}_{a\in A}$ — Start with a coarse partition
Find $\overline{W}$ s.t. $\lambda_0([\![(a)_{[0,0]}]\!]) = 0 \; \forall (a)_{[0,0]} \in \overline{W}$
$W \leftarrow W \setminus \overline{W}$
Construct Γ_W following Algorithm 2.
while `stopping criterion` **do**
  for $i = 1, \ldots, |W|$ **do**
    $W_i' \leftarrow W \setminus \left\{w^i_{[0,t_i]}\right\}$ — Try to refine each block
    $W_i' \leftarrow W_i' \cup \left\{(w^i a)_{[0,t_i+1]}\right\}_{a\in A}$
    Find $\overline{W}$ s.t. $\lambda_0([\![w_{[0,t]}]\!]) = 0 \; \forall w_{[0,t]} \in \overline{W}$
    $W \leftarrow W \setminus \overline{W}$
    Construct $\Gamma_{W_i'}$ following Algorithm 2.
    Compute $\mathsf{D}_i \leftarrow \mathsf{D}(\Gamma_W, \Gamma_{W_i'})$
  end for
  $j \leftarrow \arg\max_{i=1,\ldots,|W|} \mathsf{D}_i$ — Greedy choice
  $W \leftarrow W_j'$
  $\Gamma_W \leftarrow \Gamma_{W_j'}$
end while
  return W — Return a refined partition

cuss the interpretation of the corresponding output in Remark 6.11.

Remark 4.14. Many variants of the REFINE algorithm can be considered. For example, one could expand the memory in the past by expanding each word $w^i_{[-s_i,0]}$ into $\{(aw^i)_{[-(s_i+1),0]}\}_{a\in A}$. Also, inspired by algorithms from reinforcement learning (cf. TD learning scheme in [SB]), one could compare the current model with the $|A|^n$ models possible after n steps instead of choosing between the possible $|A|$ different models, and take the one for which the distance is the largest.

As shown in Proposition 4.16, assuming any output trace can be sampled with non-zero measure, REFINE produces a valid partition.

Assumption 4.15. The system Σ and measure λ_0 are such that, for all k, for all $w \in A^k$, if $\lambda_0([\![w_{[0,k-1]}]\!]) = 0$, then $[\![w_{[0,k-1]}]\!] = \emptyset$.

Proposition 4.16. *Suppose that Assumption 4.15 holds. Let Σ be a dynamical system whose initial state is sampled according to λ_0, and let D be a metric be-*

tween labelled Markov chains (as defined in Chapter 1). Then REFINE *outputs an adaptive memory partition of the state space X.*

Proof. We have to show that each current model Γ_W is such that W defines a partition of X and satisfies Assumption 4.8. We prove the claim by induction. First, we prove that the set W as defined after line 3 defines a partition and satisfies Assumption 4.8. The set $W = \{(a)_{[0,0]}\}_{a\in A}$ defines a partition by definition of the output function h. Now, the set $W \setminus \overline{W}$ satisfies Assumption 4.8 by construction. Moreover, it still defines a partition because the sets $[\![a_{[0,0]}]\!]$ for $(a)_{[0,0]} \in \overline{W}$ are all empty by Assumption 4.15. We now assume that W defines a partition and satisfies Assumption 4.8. We prove that, for any $w^i_{[0,t_i]} \in W$, the set W'_i as after 10 is a partition and satisfies Assumption 4.8. We first show that the expanded set $\tilde{W} = (W \setminus \{w^i_{[0,t_i]}\}) \cup \{(w^i a)_{[0,t_i+1]}\}_{a\in A}$ defines a partition. Indeed, it covers the state space because

$$
\begin{aligned}
&\bigcup_{w_{[0,t]}\in\tilde{W}} \left[w_{[0,t]}\right] \\
&= \left(\bigcup_{w_{[0,t]}\in W\setminus\left\{w^i_{[0,t_i]}\right\}} \left[w_{[0,t]}\right] \right) \cup \left(\bigcup_{a\in A} \left[(w^i a)_{[0,t_i+1]}\right] \right) \\
&= \left(\bigcup_{w_{[0,t]}\in W\setminus\left\{w^i_{[0,t_i]}\right\}} \left[w_{[0,t]}\right] \right) \cup \left[w^i_{[0,t_i]}\right],
\end{aligned}
$$

which is the whole state space X by the recursion assumption. Moreover, the sets are pairwise disjoint because

$$
\left[\!\left[(w^i a)_{[0,t_i+1]}\right]\!\right] \subseteq \left[\!\left[w^i_{[0,t_i]}\right]\!\right]
$$

for all $a \in A$. Now, with $\overline{W}$ as defined in line 9, the $\tilde{W} \setminus W_0$ satisfies Assumption 4.8 by construction. Finally, it still defines a partition because the sets $[\![w_{[0,t]}]\!]$ for $w_{[0,t]} \in \overline{W}$ are empty by Assumption 4.15, and the proof is completed. □

Together, Proposition 4.10 and Proposition 4.16 capture the two main theoretical contributions of this chapter. Rather than pursuing asymptotic correctness as the memory length grows uniformly (the central goal in Chapter 3) the focus here is different: proving soundness of our ab-

stractions. Proposition 4.10 guarantees that the abstraction Γ_W overapproximates the original dynamics (behavioural inclusion), while Proposition 4.16 ensures that REFINE produces a valid adaptive memory partition whenever it terminates. These two results together justify both the correctness of the abstraction and the practical reliability of the refinement procedure.

4.4 Numerical experiments

In this section we highlight the practical benefits of our adaptive memory approach through three numerical examples. First, we demonstrate how the REFINE method produces a meaningful abstraction that closely aligns with the system dynamics, thereby creating a "smart" partition of the state space. Next, we benchmark our method against the uniform memory approach introduced in Chapter 3. By revisiting the Sturmian dynamics experiment (see Section 3.4), we show that REFINE achieves comparable approximation accuracy while significantly reducing the size of the abstraction. Finally, we illustrate its real-world applicability by applying the method to the safety analysis of a practical system.

4.4.1 Smart partitioning

In this first experiment, we showcase that our algorithm REFINE is able to create an abstraction that corresponds to a smart partition of the state space.

Example 4.17. Let $\Sigma = (X, f, A, h)$ be a partially observable dynamical system with $X = [0,2] \times [0,1]$, and f be defined as

$$f(x) = \begin{cases} x & \text{if } x \in P_1 \cup P_4 \\ (x_1/2 + 1/2, x_2 + 1/2) & \text{if } x \in P_2 \\ (x_1 - 1/2, x_2) & \text{if } x \in P_3 \\ (2x_1 + 1, 4x_2 - 3/4) & \text{else,} \end{cases}$$

where P_i are depicted in Figure 4.4. Let $A = \{0,1\}$, and $h(x) = 0$ if $x \in P_1$, else $h(x) = 1$. An illustration and interpretation of Σ is given in Figure 4.4.

Consider Example 4.17. In this system, only partial information is known: whether the current state is in the right part of the state space ($y_k = 0$) or on the left part ($y_k = 1$). In the following, we will show that our method

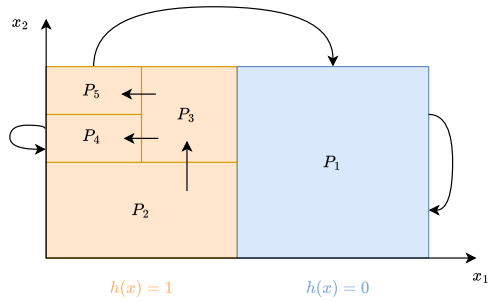


Fig. 4.4 Illustration and description of the system Σ presented in Example 4.17. The function f has to be understood in the following way: P_1 is mapped to itself, P_2 is mapped to P_3, the lower half of P_3 is mapped to P_4, the upper half of P_3 is mapped to P_5, P_5 is mapped to P_1, and P_4 to itself.

recognizes the regions P_i, showcasing its usefulness for constructing smart abstractions. We apply the data-driven algorithm REFINE (Algorithm 3) with D the *Cantor-Kantorovich* distance between LMCs (see Chapter 6). The successive adaptive memory partitions are collected in Table 4.1, and the final partition is illustrated in Figure 4.5. As we can see, the generated partition aligns well with the dynamics, and our algorithm generates an emerging structure which is not trivial, demonstrating its effectiveness.

	W	$\mathsf{D}\left(\Gamma_W, \Gamma_{W'_j}\right)$
$k=0$	$\left\{(0)_{[0,0]}, (1)_{[0,0]}\right\}$	–
$k=1$	$\left\{(0)_{[0,0]}, (10)_{[0,1]}, (11)_{[0,1]}\right\}$	0.006
$k=2$	$\left\{(0)_{[0,0]}, (10)_{[0,1]}, (110)_{[0,2]}, (111)_{[0,2]}\right\}$	0.023
$k=3$	$\left\{(0)_{[0,0]}, (10)_{[0,1]}, (110)_{[0,2]}, (1110)_{[0,3]}, (1111)_{[0,3]}\right\}$	0.015

Table 4.1 Results of the REFINE algorithm for Example 4.17. W is the chosen memory scheme, and $\mathsf{D}(\Gamma_W, \Gamma_{W'_j})$ is the corresponding maximal distance as defined in Algorithm 3.

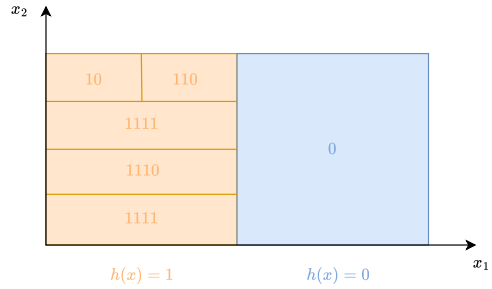


Fig. 4.5 Illustration of the last partition W given by the REFINE procedure for Example 4.17.

4.4.2 Comparison with uniform memory approach

In order to further illustrate our procedure and highlight the difference with the uniform memory procedure described in Chapter 3, we perform a similar experiment as in Section 3.4 on the Sturmian dynamics (described in Example 2.7). More precisely, we perform the following experiment:

- for $k = 1, \ldots, 20$, we compute $\mathcal{B}_k(\Sigma, \lambda_0)$,[1]
- we executed the REFINE procedure with the Cantor-Kantorovich distance and, for each iteration and associated adaptive memory partition W, we compute $\mathcal{B}_k(\Gamma_W)$ and then the quantity

$$\Delta_k^W := |\mathcal{B}_k(\Gamma_W) \setminus \mathcal{B}_k(\Sigma, \lambda_0)| \,. \tag{4.4}$$

At each iteration where a new node is added[2], the quantities Δ_k^W are depicted in Table 4.2 for $k = 1, \ldots, 20$, and where W is the adaptive memory partition of the iteration. Moreover, the associated size of the abstraction is written between parentheses in Table 4.2. Comparing these results to uniform memory in Table 3.1 yields an interesting observation: our algorithm allows to obtain the same abstraction quality with much fewer nodes than the procedure described in Chapter 3. Indeed, the adaptive memory abstractions of size 3, 4, 5, 6 and 7 achieve the same number of spurious

[1] This step is exactly the same as in Section 3.4.

[2] Because we do not account for the nodes with zero initial probability in the REFINE procedure, adding a new node does not necessarily happen at every iteration.

trajectories as the uniform memory abstraction of size 4, 8, 16, 32 and 64, respectively.

	It. 0 (2)	It. 1 (3)	It. 2 (4)	It. 9 (5)	It. 10 (6)	It. 14 (7)
$k = 1$	0	0	0	0	0	0
$k = 2$	0	0	0	0	0	0
$k = 3$	1	0	0	0	0	0
$k = 4$	3	1	0	0	0	0
$k = 5$	7	3	0	0	0	0
$k = 6$	14	6	0	0	0	0
$k = 7$	26	11	1	1	1	0
$k = 8$	46	19	2	2	2	0
$k = 9$	79	31	3	3	3	0
$k = 10$	133	49	5	5	5	0
$k = 11$	221	76	8	8	8	1
$k = 12$	364	116	11	11	11	2
$k = 13$	596	175	15	15	15	3
$k = 14$	972	262	21	21	21	4
$k = 15$	1581	390	28	28	28	6
$k = 16$	2567	578	36	36	36	8
$k = 17$	4163	854	47	47	47	10
$k = 18$	6746	1259	61	61	61	13
$k = 19$	10926	1853	77	77	77	17
$k = 20$	17690	2724	97	97	97	21

Table 4.2 Values of Δ_k^W (number of spurious output trajectories) for different iterations of the REFINE procedure for the Sturmian dynamics, and for $k = 1, \ldots, 20$. The numbers in parenthesis are the size (i.e. the number of nodes) of the resulting abstraction.

4.4.3 Application to safety analysis

In this section we apply our adaptive memory abstraction to safety analysis of a real-life system, namely an electron subject to a Lorentz force. In particular, we compare our method to a naïve method in which an abstraction is constructed from uniformly gridding the state space. We show that, for a comparable abstraction size, our method yields better results.

Example 4.18 (Electron subject to Lorentz force)**.** We are interested in the position of an electron subject to a planar Lorentz force (see e.g. [GP13]).

Its dynamics is given by

$$\begin{pmatrix} \dot{p}_1(t) \\ \dot{p}_2(t) \\ \dot{v}_1(t) \\ \dot{v}_2(t) \end{pmatrix} = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & qB_3/m \\ 0 & 0 & -qB_3/m & 0 \end{pmatrix} \begin{pmatrix} p_1(t) \\ p_2(t) \\ v_1(t) \\ v_2(t) \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ qE_1/m \\ qE_2/m \end{pmatrix}, \quad (4.5)$$

where $p_1(t)$ and $p_2(t)$ represent the positions in [m] along the x-axis and y-axis, respectively, while $v_1(t)$ and $v_2(t)$ represent the corresponding velocities in [m/s]. All other constants and their units are given in Table 4.3.

Symbol	Physical quantity	Unit	Value
m	Mass	[kg]	9.1×10^{-31}
q	Charge	[C]	1.6×10^{-19}
E_1	x-axis electric field	[V/m]	-1.0×10^{-10}
E_2	y-axis electric field	[V/m]	5.0×10^{-11}
B_3	z-axis magnetic field	[T]	1.0×10^{-11}

Table 4.3 Constants in the Lorentz force equation (4.5)

We study this dynamical system in discrete time. To do that, we approximate the derivative with the explicit Euler scheme

$$\dot{x}(t) \approx \frac{x_{t+1} - x_t}{h},$$

with $h = 0.1$, which gives

$$x_{t+1} = (\mathbf{I} + h\mathbf{A})x_t + h\mathbf{b},$$

with $x_t = (p_{1,t}, p_{2,t}, v_{1,t}, v_{2,t}) \in \mathbb{R}^4$, $\mathbf{I} \in \mathbb{R}^{4\times4}$ the identity matrix, and $\mathbf{A} \in \mathbb{R}^{4\times4}$ and $\mathbf{b} \in \mathbb{R}^4$ respectively the matrix and vectors of the continuous-time affine system given in (4.5). Since $f(x) = (\mathbf{I} + h\mathbf{A})x + h\mathbf{b}$ is a homeomorphism, it satisfies Assumption 4.9. We consider an obstacle defined by $O = [0.5, 1.5] \times [-0.5, 0.5] \times \mathbb{R} \times \mathbb{R}$. The output function h is defined as

$$h(x_t) = \begin{cases} 0 & \text{if } x_t \in O, \\ 1 & \text{else if } p_{1,t} \geq 1.5 \\ 2 & \text{otherwise.} \end{cases}$$

Finally, we take the uniform measure on $[-1,4] \times [-1,1] \times [-1,1] \times [-1,1]$ for sampling the initial state, which defines λ_0. An illustration of the dynamical system Σ is given in Figure 4.6.

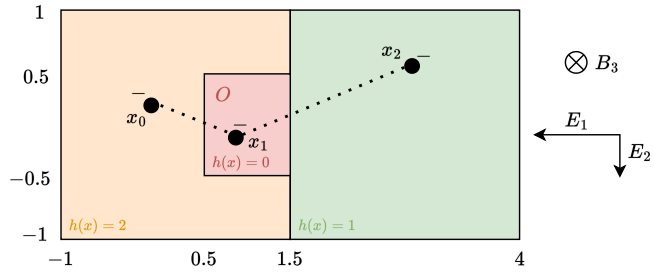


Fig. 4.6 Illustration of the Lorentz force dynamical system Σ.

We are interested in approximating the initial safe set measure

$$P_H = \lambda_0(\{x_0 \in \mathbb{R}^4 : x_t \in \mathbb{R}^4 \setminus O \, \forall t = 0, \dots, H\}),$$

where $H \in \mathbb{N}$ is a given horizon. We show that our procedure REFINE with the Cantor-Kantorovich metric CK (see Chapter 6) yields better results than explicit grid-based approaches.

We proceed as follows. We are given a labelled Markov chain $\Gamma = (S, A, \iota, \tau, l)$ abstracting the dynamical system where each state $s \in S$ corresponds to a block of a partition of the state space. If a block has a non-empty intersection with the obstacle O, we label the corresponding state as unsafe, and the other ones as safe. The set of safe states is denoted as S_{safe}. To approximate P_H, for a given level of confidence $\beta \in (0,1)$, we identify the set of states from which the probability of a safe random walk is greater than $1-\beta$, noted S_β and formally defined as the set of $s_0 \in S$ such that

$$\sum_{s_1 \in S_{\text{safe}}} \tau(s_1|s_0) \cdots \sum_{s_H \in S_{\text{safe}}} \tau(s_H|s_{H-1}) \geq 1 - \beta.$$

Then, for a given β, P_H is approximated as

$$P_H \approx \sum_{s_0 \in S_\beta} \iota_{s_0}.$$

We compare two different approaches. In the first one, we uniformly

grid the state space. If each dimension is divided into p parts, there will be p^4 blocks, noted $P_1, \dots, P_{p^4}$. From this partition, we compute a labelled Markov chain where the states correspond to the p^4 blocks with initial measure $\iota_{P_i} = \lambda_0(P_i)$, and where the transition probability from P_i to P_j is

$$\tau(P_j|P_i) := \frac{\lambda_0(f^{-1}(P_j) \cap P_i)}{\lambda_0(P_i)}$$

We consider two labelled Markov chains constructed in this way, respectively with $p = 2$ (16 states) and $p = 3$ (81 states). The second approach is to compute the adaptive memory abstraction Γ_W with W being decided by the REFINE procedure. Again we study two cases, 6 iterations (15 states) and 14 iterations (31 states). An illustration of the two approaches can be found in Figure 4.7.

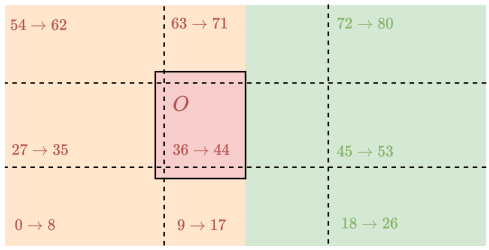
We compare the approximation of P_H with the four abstractions described above for different confidence levels. The approximation results and the true probability P_H can be found in Figure 4.8. We start by discussing the results of the abstractions generated by the REFINE algorithm (green and orange lines on Figure 4.8). When the required level of confidence is too strict, e.g. $\beta = 0.01$, then one needs more states to approximate P_H well (see Figure 4.8a). On the other hand, when the level of confidence is too high, e.g. $\beta = 0.25$, then it is more likely to overapproximate P_H (see Figure 4.8b for $H = 4$ with the 15 states abstraction). For this example, the confidence level is reasonable when it is equal to 0.05, as one can see in Figure 4.8c. In this case, the 31-state abstraction perfectly approximates P_H until $H = 8$. Now, one can see that the approximation is better using the REFINE algorithm compared to the naive approach. Indeed the former with 15 states always estimates better P_H than the latter with 81 states.

4.5 Summary and further research directions

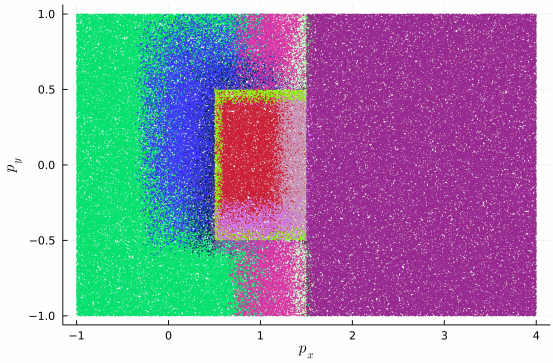
Chapter 4 – Summary

In this chapter we extended the method of Chapter 3 by introducing sound abstractions that encode memory schemes of adaptive lengths, thereby reducing their size for a similar approximation quality.

Section 4.1 – Framework



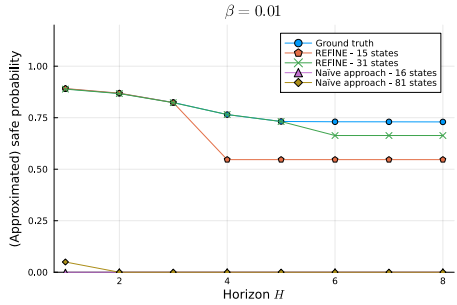
(a) Partition with classical approach.



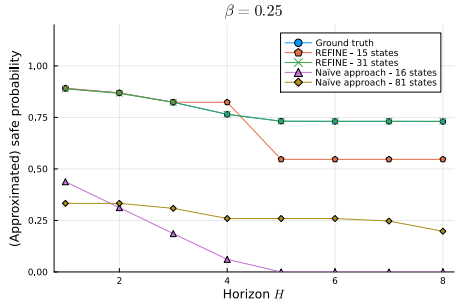
(b) Partition with REFINE.

Fig. 4.7 Comparison between a classical partition and the partition computed with the REFINE procedure on a four-dimensional example (here projected on (p_x, p_y)). Above: the classical approach, where each dimension is divided into three parts, which gives a total of $3^4 = 81$ blocks. Below: our adaptive memory abstraction after 6 iterations of REFINE, which leads to 15 blocks in the partition. In the figure, each color corresponds to a different block of the partition. However, we do not have access to a closed-form of the blocks; rather, for each point, we have access to the label of its block, that is the sequence of observation. Thus, we color points whose trajectories correspond to the sequence of the block. Observe that there are less blocks here than the 81 blocks in the above partition: this fact enhances the efficient computability properties of our technique. Moreover, one sees that the blocks generate a smart partition, as the critical places in the system (close to the obstacles) benefit from a finer resolution than the large regions away from the obstacle.

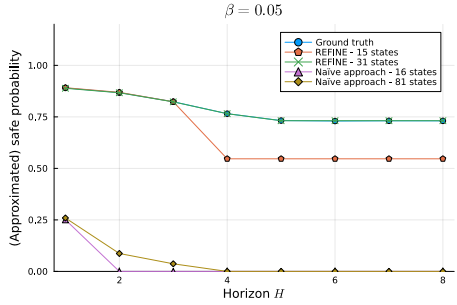
The section defines the class of systems under study and the key as-



(a) Confidence level $\beta = 0.01$. The confidence level is too low, but with 31 states the REFINE abstraction still approximates well P_H.



(b) Confidence level $\beta = 0.25$. The confidence level is too high, and P_H may be overapproximated.



(c) Confidence level $\beta = 0.05$. The confidence level is reasonable, and the REFINE abstractions yield good results with both 15 and 31 states.

Fig. 4.8 Approximation of P_H for $H = 0, \ldots, 8$ with different abstractions and for different confidence levels β.

sumptions, including invertibility of the transition function, which will ensure the theoretical soundness of the developed abstractions.

Section 4.2 – Adaptive memory abstractions

We formalise adaptive memory abstractions using timed sequences of varying lengths to partition the state space. As in Chapter 3, the abstraction is modelled as an LMC, but where each state encodes a sequence of past and future observations, possibly of different lengths. We prove that, under mild assumptions, these abstractions are sound, i.e. they over-approximate the original system behaviour, guaranteeing that any property verified on the abstraction holds for the system, albeit possibly with some conservatism. We further extend this result to infinite output trajectories: we prove that any B-LTL specification (see Chapter 1) verified almost surely on the abstraction is certified to hold for all infinite output trajectories of the original system.

Section 4.3 – Refinement procedure

This section presents the REFINE algorithm, a greedy, data-driven procedure to iteratively improve the abstraction by expanding memory, but only where needed. The algorithm uses a notion of distance between LMCs to guide refinement. We prove its correctness and discuss its specification-free nature, which distinguishes it from existing specification-driven methods.

Section 4.4 – Numerical experiments

We demonstrate the practical advantages of adaptive memory through three experiments:

1. Smart partitioning: the algorithm generates non-trivial, dynamics-aligned partitions, revealing emergent structure.
2. Comparison with uniform memory: adaptive memory achieves comparable accuracy with significantly smaller abstractions, reducing computational overhead.
3. Safety analysis: applied to an electron under Lorentz force, our method outperforms naïve grid-based approaches, offering better precision for safety verification with fewer states.

The experiments illustrate the efficiency and scalability of the approach.

Chapter 4 – Further research directions

For the same reasons as in Chapter 3, we believe that this method is a genuine alternative to classical state space refinement and offers a practical and frugal alternative to decrease epistemic uncertainty. The following directions can be considered for future research.

Approximation quality quantification

For a given memory scheme W, it would be valuable to derive a theorem quantifying approximation quality, analogous to Theorem 3.15 in the previous chapter for uniform memory. Such a result would provide explicit bounds on the error between the original system and its abstraction, depending on the memory depth and the system dynamical properties.

Optimality of the REFINE algorithm

The greedy nature of the REFINE algorithm raises questions about its global optimality. Future work could analyse whether the algorithm converges to an optimal memory scheme (e.g., minimal size for a given precision) or if alternative refinement strategies could yield better performance.

Investigate specification-encoding distances

While our approach is specification-free, one could explore distances between LMCs that explicitly encode temporal logic specifications (e.g. safety, reachability). This would provide a targeted alternative to the work of [YMR20].

Similar further works as in Chapter 3

Many future directions described in Chapter 3 remain interesting in the context of adaptive memory abstractions. For instance, it is interesting to account for a finite number of observations, or to add control inputs in the overall method.

PART III
Abstractions of stochastic systems

5

Memory-dependent abstractions of stochastic systems via transfer operators

The previous two chapters explored memory-dependent abstraction methods for partially observable deterministic systems – systems therefore affected only by epistemic uncertainty. Yet, real-world applications such as autonomous vehicles, rescue robots, smart grids, and intelligent buildings often face environments where uncertainty is inherent [Ala22, PHC+22]. Unlike epistemic uncertainty, this intrinsic unpredictability is known as aleatoric uncertainty.

To formally analyse the aforementioned systems, abstraction-based methods have been developed to approximate stochastic processes (see [LSAZ22] for a survey). These abstractions can be interpreted as an approximation of the discrete stochastic process that arises from the partial observability of the original system. Although the original process is Markovian, the resulting discrete abstraction often is not (see Remark 2.16). Yet, for computational reasons, it is typically modelled as (sometimes robust) Markov chains, introducing inevitable conservatism and/or approximation errors.

5 | Memory-dependent abstractions of stochastic systems via transfer operators

In this chapter, and building on top of the two previous chapters, we investigate memory-dependent abstractions to alleviate approximation errors due to non-Markovianity in partially observable stochastic systems. The introduction of memory aims precisely at capturing memory effects inherent in the non-Markovian discrete process. More precisely, we introduce uniform memory-dependent abstractions for stochastic systems, that are LMCs whose nodes encode ℓ-long sequences of outputs, such as illustrated in Figure 5.1. This chapter therefore marks a significant step toward creating smart memory-dependent abstractions for the analysis and control of complex systems.

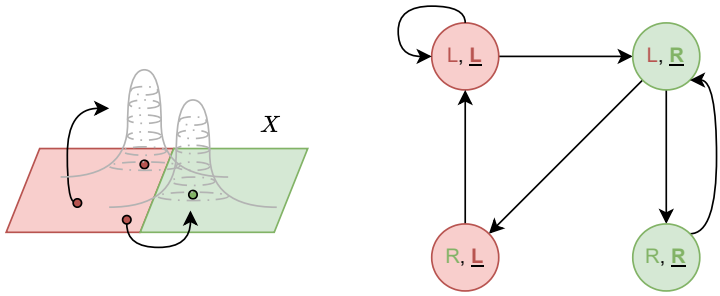


Fig. 5.1 Illustration of a uniform memory-dependent abstraction of a stochastic dynamical system. On the left: a partially observable stochastic dynamical system with $A = \{\mathrm{L}, \mathrm{R}\}$. The dynamics is defined as $x_{k+1} = f(x_k) + w_k$, where w_k for instance is the realisation of a Gaussian random variable (hence the densities). On the right: similarly as in Chapter 3, a LMC whose nodes encode sequences of length $\ell = 2$.

A key difference with the deterministic framework is that, in Chapter 3, we could rely on an underlying partition with blocks

$$\begin{aligned}&[\![a_0, \dots, a_{\ell-1}]\!]\\ &= \{x_{\ell-1} \in X : \text{there exists a trajectory } (x_0, \dots, x_{\ell-1}) \text{ with } h(x_i) = a_i\}\end{aligned}$$

to analyse such abstractions.[1] However this definition is deeply rooted in determinism as, in presence of aleatoric uncertainty, these sets are not well-defined anymore – the same state can lead to different outputs in the next timestep. In this chapter, we will see that analysing such abstrac-

[1] Theorem 3.15 relies on the existence of such a partition.

tions requires non-trivial mathematical tools. In particular, our analysis will rely on the *transfer operator* of a *lifted system*, whose state space is X^ℓ, the ℓ-cartesian power of the state space. More precisely, we will show that abstracting such stochastic systems with memory amounts to compute a *Galerkin approximation* of this lifted transfer operator.

Stochastic systems are commonly abstracted using Markov models in the literature, including robust variants [AKLP10, DLM24, FPLL24, LAB15, LSZ20, ML23]. Among robust Markov models, *Interval Markov Chains* or *Interval Markov Decision Processes* (see e.g. [GLD00]) – where transition probabilities are defined over intervals – are widely adopted. While these models preserve Markovianity, they are inherently conservative. To the best of our knowledge, memory-dependent abstraction methods, whether for classical or robust Markov models, have yet to be explored for stochastic systems. Our work is deeply related to and inspired by the work on *Markov state models* [NFW13, PWS+11, SNS10], which employs Galerkin approximations of the underlying Markov process's transfer operator, to build a finite approximation in a partial observation scenario. Our contribution w.r.t. [NFW13, PWS+11, SNS10] is twofold. First, we introduce memory to the discrete approximation, whereas only the memoryless case is studied in those works. Second, although we use intermediate results proven in [SNS10], our bounds – even in the memoryless case – fundamentally differ from those obtained in that work.

Finally, our work is related to partially observable Markov decision processes (POMDPs for short, see [SBB+24] for a survey). Indeed, the considered dynamical systems can be framed in the formalism of POMDPs, where the continuous state space of the original system and the discrete cells of the partition are the state and observation spaces, respectively, as in, e.g., [ZFM10]. Although the loss of the Markov property for observations is a known phenomenon in the POMDP literature [GCW+23, SLJ+03], our contribution departs from this literature in that we focus on problems that arise in the framework of (safety-critical) abstractions. That is, we study the loss of the Markov property when the state space is continuous and is approximated by cells corresponding to sets of states.

Contributions. In this chapter, we develop memory-dependent abstractions of stochastic systems. Inspired by the two previous chapters, we extend the state space of the stochastic system to create a lifted system, where each state represents an ℓ-long sequence of states, ℓ being the considered memory. Further, akin to work on Markov state models [NFW13, PWS+11,

SNS10], we employ transfer-operator theory and construct an ℓ-memory abstraction, through Galerkin approximations of the lifted process's transfer operator. Critically, we provide an upper bound on the total variation distance between the distribution of the original continuous state system and its discrete approximation, enabling formal verification through the abstract model. Finally, we showcase through examples how memory increases approximation accuracy in various situations.

Outline. This chapter is organised as follows. Section 5.1 introduces the framework for stochastic dynamical systems, while Section 5.2 presents the preliminaries used throughout the chapter. Key assumptions such as those related to ergodicity are stated to ground the analysis. Section 5.3 presents a data-driven method to construct ℓ-memory abstractions. The notion of a lifted system is introduced, and Algorithm 4 is detailed, showing how to approximate state densities using sampled trajectories. Section 5.4 interprets the abstractions as Galerkin approximations of the lifted system transfer operator. This theoretical connection justifies the algorithm steps and enables rigorous error analysis. Section 5.5 derives formal guarantees on the approximation quality. Two complementary bounds (one increasing and one decreasing with time) are proven, ensuring the total variation distance between the original and abstracted distributions remains controlled. Section 5.6 validates the approach through numerical experiments. Two use cases demonstrate that memory-dependent abstractions (1) improve accuracy when only partial observations are available, and (2) outperform classical abstraction techniques even with coarser discretisations. Section 5.7 concludes with a synthesis of contributions and open questions.

Publication. This work has been done in collaboration with Giannis Delimpaltadakis, Luca Laurenti, Manuel Mazo Jr. and Raphaël M. Jungers, and has been published in [BDL+25].

5.1 Framework

We consider stochastic dynamical systems of the form $\Sigma = (X, \tau, A, h)$ where $X \subseteq \mathbb{R}^d$ and is endowed with the Borel σ-algebra $\sigma(X)$, whose state

and output trajectories are driven by

$$\begin{cases} x_{k+1} \sim \tau(\cdot|x_k), \\ y_k = h(x_k). \end{cases}$$

We recall that, for a given initial measure $\lambda_0 : \sigma(X) \to [0,1]$, this system admits a sequence of state random variables $(\mathsf{x}_0, \mathsf{x}_1, \dots)$, associated with probability state measures $(\lambda_0, \lambda_1, \dots)$. It also admits a sequence of output random variables $(\mathsf{y}_0, \mathsf{y}_1, \dots)$, defined as $\mathsf{y}_k(a) = \lambda_k([\![a]\!])$. We refer the reader to Chapter 2 for a complete description of partially observable stochastic dynamical systems.

Similarly as in Chapter 3, our method and analysis relies on the assumptions that the invariant measure exists and is unique, where the invariant measure is the probability measure $\mu : \sigma(X) \to [0,1]$ such that

$$(\tau \# \mu)(A) := \mu(A)$$

for all $A \in \sigma(X)$ (see Appendix B for a complete description of the push-forward measure $\tau\#\mu$).

Assumption 5.1 (Existence and unicity of the invariant measure)**.** There exists a unique invariant measure $\mu : \sigma(X) \to [0,1]$ for the transition kernel τ.

Remark 5.2 (Necessity of Assumption 5.1)**.** Both existence and uniqueness are necessary for our analysis. In the following, we will indeed see that existence is required to define densities $f_k := \mathrm{d}\lambda_k/\mathrm{d}\mu$ and the transfer operator $T : L^2(\mu) \to L^2(\mu)$ (see Section 5.2.1). Uniqueness is equally critical, as already pointed out in [SHD01, Section 4.3]. In the following sections, we will need the transfer operator to admit a *simple* eigenvalue 1 (see Assumption 5.21) – if two distinct invariant measures μ_1, μ_2 existed, both $\mathbb{1}$ and $\mathrm{d}\mu_2/\mathrm{d}\mu_1 \neq \mathbb{1}$ would be eigenfunctions of T for eigenvalue 1, yielding multiplicity ≥ 2 and collapsing the spectral gap (i.e. $e_{\ell,1} = 1$ in Assumption 5.21).

Systems violating this assumption arise naturally in the context of *bifurcation phenomena* (see e.g. [SN90]). A classical example is a stochastic double-well potential: the system exhibits two distinct ergodic components, each supporting its own invariant measure. More generally, a system whose state space decomposes into $n \geq 2$ ergodic components admits at least n invariant measures (each supported on one component), and

eigenvalue 1 of T has multiplicity n. In such settings, the method of this chapter can nonetheless be applied *within each ergodic component separately*, by conditioning the analysis on trajectories confined to a single component – which effectively restores uniqueness locally.

Our method also relies on the fact that the system converges to the invariant measure – a phenomenon known as *ergodicity* in the literature. This is more formally stated as follows.

Assumption 5.3 (Ergodicity)**.** The invariant measure μ is such that, for any initial state probability measure λ_0, it holds that

$$\lim_{k\to\infty} \mathsf{TV}(\lambda_k, \mu) = 0,$$

where TV is the Total Variation distance (see Appendix D for a formal definition).

For any $k \in \mathbb{N} \cup +\{\infty\}$, each state probability measure λ_k admits a *state probability density* with respect to the invariant measure, denoted $f_k := \mathrm{d}\lambda_k/\mathrm{d}\mu$, and satisfying

$$\lambda_k(A) := \int_{x\in A} f_k(x)\mu(\mathrm{d}x).$$

Formally, it is the Radon-Nikodym derivative of λ_k with respect to the invariant measure μ (see Appendix B and Appendix D for technical details). In this chapter, our main goal is to approximate the densities f_k with a memory-dependent abstraction. More precisely, we will show that using the abstractions described in this chapter yields approximate densities $\tilde{f}_k$ such that $\mathsf{TV}(f_k, \tilde{f}_k)$ is bounded from above.

Our analysis relies on the notions of *projection* and *Galerkin approximations*, which are defined below and are well-defined only in Hilbert spaces (see Appendix C for mathematical background on Hilbert spaces). Because $L^2(\mu)$ is the only Hilbert space among the $L^p(\mu)$ spaces (see Section C.3), we impose the following assumption.

Assumption 5.4. We assume that all the densities involved in this chapter are in $L^2(\mu)$.

We claim that this assumption is not restrictive in many practical situations, e.g. dynamical systems of the form $x_{k+1} = f(x_k) + w_k$ with w_k being the realisation of a Gaussian noise. Note that the choice of the Total

Variation distance in Assumption 5.3 is compatible with our framework, as convergence in the $\|\cdot\|_2$ norm implies TV convergence by the Cauchy-Schwarz inequality.

5.2 Preliminaries

Our method relies on the analysis of the Galerkin approximation of a transfer operator. In this section, we formally define those concepts.

5.2.1 Transfer operator

In this section, we define the so-called transfer operator of a Markov kernel, a key concept for Chapter 5, and analyse its properties. Although classical, this chapter is inspired by notions introduced in [SHD01, SNS10], and is based on notions from measure theory (see Appendix B), linear algebra (see Appendix C) and probability theory (see Appendix D).

Definition 5.5 (Transfer operator)**.** Let $(X, \mathcal{A})$ be a measurable space, let $\tau : \mathcal{A} \times X$ be a Markov kernel from $(X, \mathcal{A})$ to itself, and let $\mu : \mathcal{A} \to [0, 1]$ be a probability measure on $(X, \mathcal{A})$. The transfer operator of τ on μ is the operator $T : L^1(\mu) \to L^1(\mu)$ such that

$$\int_{y \in A} (Tf)(y)\mu(\mathrm{d}y) = \int_{x \in X} \tau(A|x) f(x)\mu(\mathrm{d}x)$$

for all functions $f \in L^1(\mu)$ and all sets $A \in \mathcal{A}$.

If, for all $x \in X$, the probability measure $\tau(\cdot|x)$ admits a density $t(\cdot|x) := \mathrm{d}\tau(\cdot|x)/\mathrm{d}\mu$, then the transfer operator is explicitly defined as

$$(Tf)(y) = \int_{x \in X} t(y|x) f(x)\mu(\mathrm{d}x)$$

for all $f \in L^1(\mu)$.

Intuitively, the operator T propagates in time a probability density function in $L^1(\mu)$, and therefore has a similar role to the pushforward measure, but for densities. The following proposition formally states this.[2]

Proposition 5.6. *Let $(X, \mathcal{A})$ be a measurable space. Let τ be a Markov kernel from $(X, \mathcal{A})$ to itself, and let ν, μ be two probability measures on $(X, \mathcal{A})$. Let T*

[2]While we do not claim that this result is novel, we include its proof for completeness as we were unable to locate it in the existing literature.

be the transfer operator of τ on μ. Then, if f is a density for ν with respect to μ, it holds that Tf is a density for $\tau\#\nu$ wih respect to μ.

Proof. We have that

$$\begin{aligned}(\tau\#\nu)(A) &= \int_{x\in X} \tau(A|x)\nu(\mathrm{d}x)\\ &= \int_{x\in X} \tau(A|x)f(x)\mu(\mathrm{d}x)\\ &= \int_{x\in A} (Tf)(x)\mu(\mathrm{d}x),\end{aligned}$$

where the first equality follows from the definition of the pushfoward measure, the second from the definition of the Radon-Nikodym derivative, and the third from the definition of the transfer operator. □

One of the key properties of transfer operators is that their 1-norm is unitary.[3] This norm is defined as

$$\|T\|_1 := \sup_{f\in L^1(\mu):\|f\|_1\leq 1} \|Tf\|_1,$$

with $\|\cdot\|_1$ being the norm of the $L^1(\mu)$ space such as defined in Section C.3.

Proposition 5.7. *Let $(X,\mathcal{A})$ be a measurable space. Let τ be a Markov kernel from $(X,\mathcal{A})$ to itself, and μ be a probability measure on $(X,\mathcal{A})$. Let T be the transfer operator of τ on μ. Then it holds that $\|T\|_1 = 1$.*

Proof. We first prove that $\|T\|_1 \geq 1$. Take any nonnegative function $f : X \to \mathbb{R}_{\geq 0}$ such that $\|f\|_1 = 1$. Then it holds that

$$\|Tf\|_1 = \int_{x\in X} |(Tf)(x)|\mu(\mathrm{d}x).$$

By definition of the transfer operator, if f is nonnegative, then so is Tf. Therefore,

$$\begin{aligned}\|Tf\|_1 &= \int_{x\in X} (Tf)(x)\mu(\mathrm{d}x)\\ &= \int_{x\in X} \tau(X|x)f(x)\mu(\mathrm{d}x)\\ &= \int_{x\in X} f(x)\mu(\mathrm{d}x)\\ &= 1.\end{aligned}$$

[3] Again, we do not claim that this result is novel, but we include its proof for completeness.

By definition of sup, this proves the first claim. Now we prove that $\|T\|_1 \leq 1$. Any function $f : X \to \mathbb{R}$ can be written as $f := f^+ - f^-$, where $f^+ : X \to \mathbb{R}_{\geq 0}$ and $f^- : X \to \mathbb{R}_{\geq 0}$ are respectively the positive and negative parts of f. Therefore, for all functions f such that $\|f\|_1 \leq 1$, it holds that

$$\begin{aligned}
\|Tf\|_1 &= \int_{x\in X} |(Tf)(x)|\mu(\mathrm{d}x) \\
&= \int_{x\in X} |(T(f^+ - f^-))(x)|\mu(\mathrm{d}x) \\
&= \int_{x\in X} |(Tf^+)(x) - (Tf^-)(x)|\mu(\mathrm{d}x) \\
&= \int_{x\in X} (Tf^+)(x) + (Tf^-)(x)\mu(\mathrm{d}x) \\
&= \int_{x\in X} (Tf^+)(x)\mu(\mathrm{d}x) + \int_{x\in X} (Tf^-)(x)\mu(\mathrm{d}x) \\
&= \int_{x\in X} f^+(x)\mu(\mathrm{d}x) + \int_{x\in X} f^-(x)\mu(\mathrm{d}x) \\
&= \int_{x\in X} (f^+(x) + f^-(x))\mu(\mathrm{d}x) \\
&= \int_{x\in X} |f^+(x) - f^-(x)|\mu(\mathrm{d}x) \\
&= \int_{x\in X} |f(x)|\mu(\mathrm{d}x) \\
&\leq 1,
\end{aligned}$$

which concludes the proof. □

The notion of *spectrum* extends naturally to function operators. In the following, we define the spectrum of the transfer operator.

Definition 5.8 (Transfer operator spectrum)**.** Let $\alpha \in \mathbb{C}$ and $f \in L^1(\mu)$. If

$$Tf := \alpha f,$$

then α and f form a pair of *eigenvalue* and *eigenfunction* of T. All such pairs form the spectrum of the transfer operator.

Remark 5.9. Note that, if μ is invariant under the Markov kernel τ, then the pair $(1, \mathbb{1})$, where $\mathbb{1}$ is the constant unitary function defined as

$$\mathbb{1} : X \to \mathbb{R} : x \mapsto 1$$

always belongs to the spectrum of the transfer operator. Indeed, by Proposition 5.6, $\tau\#\mu = \mu$ implies that $T\mathbb{1} = \mathbb{1}$, since $\mathbb{1}$ is the probability density function of μ with respect to itself (i.e. $\mathbb{1} = \mathrm{d}\mu/\mathrm{d}\mu$).

As explained above, because we will need projection operators, we will restrict the domain and co-domain of the transfer operator to the Hilbert space $L^2(\mu)$.[4] Formally, we consider

$$T' : L^2(\mu) \to L^2(\mu) : f \mapsto T'(f) := T(f),$$

where we recall that $L^2(\mu) \subseteq L^1(\mu)$. This operator is well-defined, because one can show that, if $f \in L^2(\mu)$, then $Tf \in L^2(\mu)$ as well. For the sake of conciseness, and when clear from the context, we will refer to this operator as $T : L^2(\mu) \to L^2(\mu)$ and drop the apostrophe. In this case, the notion of *orthonormal eigenfunctions* extends for the transfer operator.

Definition 5.10 (Orthonormal eigenfunctions)**.** Let $T : L^2(\mu) \to L^2(\mu)$, and let F be the set of all eigenfunctions of T. Then F is said to be *orthonormal* if

1. for all $f, g \in F$ such that $f \neq g$, $\langle f, g \rangle = 0$,
2. for all $f \in F$, $\|f\|_2 := \sqrt{\langle f, f \rangle} = 1$.

5.2.2 Galerkin approximation

We now conclude the preliminaries by defining projections and Galerkin approximations.

Definition 5.11 (Projection operator)**.** Let $(X, \langle \cdot, \cdot \rangle)$ be a Hilbert space, and let $\| \cdot \|$ be its associated norm. For a closed subspace $A \subset X$, the surjective map $Q : X \to A$ is a *projection operator* onto A if

1. $Q^2 = Q$,
2. $\sup_{x \in X : \|x\| = 1} \|Qx\| = 1$.

By definition, the image of a projection operator gives the vector that is the closest to the input vector, in the sense of the inner product distance. More formally,

$$Qx \in \arg\min_{y \in A} \mathsf{D}(x, y),$$

with $\mathsf{D}(x, y) = \|x - y\| = \sqrt{\langle x - y, x - y \rangle}$.

The following gives a closed-form formula for Qx.

[4] We recall that $L^2(\mu)$ is the only Hilbert space among $L^p(\mu)$ spaces.

Proposition 5.12. *Let $(X, \langle\cdot,\cdot\rangle)$ be a Hilbert space. Let $A \subset X$ be a closed subspace, and let $\{a_i\}_{i=1}^n$ be an orthonormal basis for A, that is a basis such that, for all $i, j = 1, \ldots, n$ such that $i \neq j$,*

$$\langle a_i, a_j\rangle = 0 \text{ and } \langle a_i, a_i\rangle = 1.$$

Let $Q : X \to A$ be a projection operator onto A, then, for all $x \in X$, it holds that

$$Qx = \sum_{i=1}^{n} \langle x, a_i\rangle a_i.$$

We conclude by defining the so-called Galerkin approximation, defined as follows.

Definition 5.13 (Galerkin approximation)**.** Let $(X, \langle\cdot,\cdot\rangle)$ be a Hilbert space, and let $T : X \to X$ be an operator. For a closed subspace $A \subset X$, let $Q : X \to A$ be the projection operator of $(X, \langle\cdot,\cdot\rangle)$ onto A. The operator $P : A \to A$ defined as $P := QTQ$ is called the Galerkin approximation of T.

Intuitively, the Galerkin approximation of an operator $T : X \to X$ is the projection of an operator onto a subset A. Given an input $x \in X$, it first projects x onto A (to get Qx), then applies the operator to obtain $T(Qx)$. Now, since $T(Qx) \notin A$ in general, it projects it to obtain $Q(T(Qx))$, which is Px.

If A is finitely generated, that is

$$A = \operatorname{span}(\{a_1, \ldots, a_n\}) := \left\{ \sum_{i=1}^{n} \alpha_i a_i : \alpha_i \in \mathbb{R} \right\},$$

then, since $P : A \to A$ is a finite-dimensional operator, it admits a *matrix representation* $\mathbf{P} \in \mathbb{R}^{n\times n}$, defined as follows. Let $\mathbf{a} \in \mathbb{R}^n$ be the *vector representation* of some $a \in A$, that is

$$a = \sum_{i=1}^{n} \mathbf{a}_i a_i.$$

Similarly, let $\mathbf{a}' \in \mathbb{R}^n$ be the vector representation of $a' := Pa$. Then $\mathbf{P}$ satisfies

$$\mathbf{a}' = \mathbf{P}^\top \mathbf{a}$$

for all such $\mathbf{a}$ and $\mathbf{a}'$.

5.3 Method

In this section we provide a method to construct ℓ-memory approximations of f_k, for $k = 1, 2, \ldots$. Analogous to the methods described in Chapter 3 and Chapter 4, this method specifies the data that one needs to sample. The method is described in Algorithm 4. In the latter, we use the quantity $F_\mu(a_0, \ldots, a_\ell)$, defined similarly as in the deterministic case (see Definition 3.7), and $\mathbf{1}_A(x)$ denotes the indicator function.

In the same fashion as in Chapter 3 and Chapter 4, we assume that we can sample an infinite number of trajectories. We refer the reader to Remark 3.9 for a discussion of this assumption.

Assumption 5.14. We have access to infinitely many samples in Algorithm 4. That is, one can compute exactly F_μ, as described in (3.1).

In order to interpret Algorithm 4, we will need the notion of lifted system, as well as its transfer operator. The rest of this section is dedicated to those concepts. In the following, for any integer $\ell \in \mathbb{N}$, we will set $\sigma(X)^\ell := \bigotimes_{i=1}^{\ell} \sigma(X)$, where $\bigotimes_{i=1}^{\ell}$ denotes the ℓ-th power σ-algebra.[5]

Definition 5.15 (Lifted system)**.** Let $\Sigma = (X, \tau, A, h)$ be a stochastic system with initial measure $\lambda_0 : \sigma(X) \to [0, 1]$, and let $\ell \in \mathbb{N}$ be an integer called *the memory*. The stochastic system $\Sigma^\ell := (X^\ell, \tau^\ell, A^\ell, h^\ell)$ is called the lifted system and is defined as follows:

- the state space X^ℓ is the ℓ-th Cartesian power of the original state space,
- the kernel $\tau^\ell : \sigma(X)^\ell \times X^\ell \to [0, 1]$ is defined as, for all $(A_1, \ldots, A_\ell) \in \sigma(X)^\ell$ and $(x_0, \ldots, x_{\ell-1}) \in X^\ell$,

$$\tau^\ell((A_1 \times \cdots \times A_\ell)|(x_0, \ldots, x_{\ell-1})) = \begin{cases} \tau(A_\ell|x_{\ell-1}) & \text{if } x_1 \in A_1, \ldots, x_{\ell-1} \in A_{\ell-1}, \\ 0 & \text{otherwise,} \end{cases} \tag{5.4}$$

- the output space A^ℓ is the ℓ-th Cartesian power of the original output space,
- the output function $h^\ell : X^\ell \to A^\ell$ is defined as

$$h^\ell(x_1, \ldots, x_\ell) = (h(x_1), \ldots, h(x_\ell)),$$

[5] The ℓ-th power σ-algebra is a generalisation of the product σ-algebra.

Algorithm 4 Compute $\tilde{f}_{\ell,k}$, the ℓ-memory approximation of f_k

1: Generate output trajectories of the system Σ initialised at the invariant measure μ, and, for all sequences $(a_0, \dots, a_\ell) \in A^{\ell+1}$, compute

$$\begin{aligned} &F_\mu(a_0) && \text{(1-long frequency)} \\ &F_\mu(a_0, \dots, a_{\ell-1}) && (\ell\text{-long frequency}) \\ &F_\mu(a_0, \dots, a_\ell) && ((\ell+1)\text{-long frequency}), \end{aligned}$$

where F_μ is the counting function such as defined above. For all $(\ell+1)$-long sequence $a_0, \dots, a_\ell$, let

$$(\mathbf{P}_\ell)_{a_0 \dots a_{\ell-1}, a_1 \dots a_\ell} := \frac{F_\mu(a_0, \dots, a_\ell)}{F_\mu(a_0, \dots, a_{\ell-1}),}. \tag{5.1}$$

2: Generate output trajectories of the system Σ initialised at the initial measure λ_0, and, for all sequences $(a_0, \dots, a_{\ell-1}) \in A^\ell$, compute

$$F_{\lambda_0}(a_0, \dots, a_{\ell-1}) \qquad (\ell\text{-long frequency}).$$

For all ℓ-long sequence $a_0, \dots, a_{\ell-1}$, let

$$(\tilde{\mathbf{f}}_0^\ell)_{a_0 \dots a_{\ell-1}} := F_{\lambda_0}(a_0, \dots, a_{\ell-1}). \tag{5.2}$$

3: Compute

$$\tilde{\mathbf{f}}_{k-\ell+1}^\ell = \left(\mathbf{P}_\ell^{k-\ell+1}\right)^\top \tilde{\mathbf{f}}_0^\ell \tag{5.3}$$

4: For all $a_k \in A$, compute

$$(\tilde{\mathbf{f}}_{\ell,k})_{a_k} := \sum_{a_{k-\ell+1}=1}^{n} \cdots \sum_{a_{k-1}=1}^{n} (\tilde{\mathbf{f}}_{k+\ell-1}^\ell)_{a_{k-\ell+1} \dots a_{k-1} a_k}$$

return $\tilde{f}_{\ell,k} := \sum_{a_k=1}^n \frac{(\tilde{\mathbf{f}}_{\ell,k})_{a_k}}{F_\mu(a_k)} \mathbf{1}_{[\![a_k]\!]}$.

for all $(x_1, \dots, x_\ell) \in X^\ell$, and where h is the original output function.

If the system is assumed to be initialised at a state measure $\lambda_0 : \sigma(X) \to [0, 1]$, then the lifted system is assumed to be initialised at the state measure

$\lambda_0^\ell : \sigma(X)^\ell \to [0,1]$ defined as

$$\begin{aligned}&\lambda_0^\ell(A_0 \times \cdots \times A_{\ell-1}) \\ &\quad := \int_{x_0 \in A_0} \cdots \int_{x_{\ell-1} \in A_{\ell-1}} \tau(\mathrm{d}x_{\ell-1}|x_{\ell-2}) \dots \tau(\mathrm{d}x_1|x_0)\lambda_0(\mathrm{d}x_0).\end{aligned} \tag{5.5}$$

for all $(A_0, \dots, A_{\ell-1}) \in \sigma(X)^\ell$. Owing to Assumption 5.1, the lifted system admits a unique invariant measure $\mu^\ell : \sigma(X)^\ell \to [0,1]$ (see e.g. [Hai06, Equation (4.1)]), defined as

$$\begin{aligned}&\mu^\ell(A_0 \times \cdots \times A_{\ell-1}) \\ &= \int_{x_0 \in A_0} \cdots \int_{x_{\ell-1} \in A_{\ell-1}} \tau(\mathrm{d}x_{\ell-1}|x_{\ell-2}) \dots \tau(\mathrm{d}x_1|x_0)\mu(\mathrm{d}x_0).\end{aligned} \tag{5.6}$$

for all $(A_0, \dots, A_{\ell-1}) \in \sigma(X)^\ell$.

In this chapter, we make the following assumption.

Assumption 5.16. The lifted system Σ^ℓ is such that its invariant measure μ^ℓ satisfies

$$\mu^\ell([\![a_1]\!] \times \cdots \times [\![a_\ell]\!]) > 0$$

for all sequences $(a_1, \dots, a_\ell) \in A^\ell$.

Assumption 5.16 is the same assumption as Assumption 3.11 in Chapter 3 but for stochastic systems. We claim that, in the stochastic setting, this assumption is much less restrictive than in the deterministic setting, for at least two reasons. First, it holds in many practical cases such as unbounded noise. Second, it also suffices to assume that $\mu^\ell([\![a_1]\!] \times \cdots \times [\![a_\ell]\!]) = 0$ implies $\lambda_0^\ell([\![a_1]\!] \times \cdots \times [\![a_\ell]\!]) = 0$. That is, zero measure invariant states correspond to zero measure initial conditions.

Similarly as the original system, the lifted stochastic system admits a sequence of state probability measures $(\lambda_0^\ell, \lambda_1^\ell, \dots)$. As formalised in the following proposition, the measure λ_k^ℓ corresponds to the joint probability measure of the states $(x_{k-\ell+1}, \dots, x_k)$.

Proposition 5.17. *Let $\Sigma = (X, \tau, A, h)$ be a stochastic system initialised at the initial measure $\lambda_0 : \sigma(X) \to [0,1]$, and let $\Sigma^\ell = (X^\ell, \tau^\ell, A^\ell, h^\ell)$ be the lifted system for any memory $\ell \in \mathbb{N}$ initialised at the initial measure λ_0^ℓ such as defined in* (5.5)*. Let $(\lambda_0, \lambda_1, \dots)$ and $(\lambda_0^\ell, \lambda_1^\ell, \dots)$ be the sequences of state probability measures for the original system and the lifted system, respectively. Then it holds*

that, for all $(A_{k-\ell+1}, \dots, A_k) \in \sigma(X)^\ell$

$$\lambda_k^\ell(X \times \cdots \times X \times A_i \times X \times \cdots \times X) = \lambda_i(A_i)$$

for all $i = k - \ell + 1, \dots, k$.

Proof. We proceed by induction. Let $k = 0$, it holds by definition that

$$\begin{aligned}&\lambda_0^\ell(X \times \cdots \times X \times A_i \times X \times \cdots \times X)\\ &\quad := \int_{x_0 \in X} \cdots \int_{x_i \in A_i} \cdots \int_{x_{\ell-1} \in X} \tau(\mathrm{d}x_{\ell-1}|x_{\ell-2}) \dots \tau(\mathrm{d}x_1|x_0)\lambda_0(\mathrm{d}x_0).\end{aligned}$$

By definition of a transition kernel, for all $j > i$, it holds that

$$\int_{x_j \in X} \tau(\mathrm{d}x_j|x_{j-1}) = \tau(X|x_{j-1}) = 1,$$

which yields

$$\begin{aligned}&\lambda_0^\ell(X \times \cdots \times X \times A_i \times X \times \cdots \times X)\\ &\quad := \int_{x_0 \in X} \cdots \int_{x_i \in A_i} \tau(\mathrm{d}x_i|x_{i-1}) \dots \tau(\mathrm{d}x_1|x_0)\lambda_0(\mathrm{d}x_0).\end{aligned}$$

Now, for all $j < i$, it holds that

$$\int_{x_j \in X} \tau(\mathrm{d}x_{j+1}|x_j)\lambda_j(\mathrm{d}x_j) = \lambda_{j+1}(\mathrm{d}x_{j+1}).$$

By induction it implies that

$$\begin{aligned}&\lambda_0^\ell(X \times \cdots \times X \times A_i \times X \times \cdots \times X)\\ &\quad := \int_{x_0 \in X} \cdots \int_{x_i \in A_i} \tau(\mathrm{d}x_i|x_{i-1}) \dots \tau(\mathrm{d}x_1|x_0)\lambda_0(\mathrm{d}x_0)\\ &\quad := \int_{x_1 \in X} \cdots \int_{x_i \in A_i} \tau(\mathrm{d}x_i|x_{i-1}) \dots \tau(\mathrm{d}x_2|x_1)\lambda_1(\mathrm{d}x_1)\\ &\quad := \dots\\ &\quad := \int_{x_{i-1} \in X} \int_{x_i \in A_i} \tau(\mathrm{d}x_i|x_{i-1})\lambda_{i-1}(\mathrm{d}x_{i-1})\\ &\quad := \lambda_i(A_i).\end{aligned}$$

We now prove the inductive step, and assume that the proposition is true

for all k. Let

$$B := X \times \cdots \times X \times A_i \times X \times \cdots \times X.$$

By definition of the lifted system, it holds that

$$\lambda_{k+1}^{\ell}(B) = \int_{x^{\ell} \in X^{\ell}} \tau^{\ell}(B|x^{\ell}) \lambda_k^{\ell}(\mathrm{d}x^{\ell}),$$

where $x^{\ell} := (x_{k-\ell+1}, \ldots, x_k)$. By (5.4), we have that

$$\tau^{\ell}(B|x^{\ell}) = \tau(X|x_k)\mathbf{1}_{A_i}(x_i) = \mathbf{1}_{A_i}(x_i),$$

where $\mathbf{1}_A(x)$ denotes the indicator function of a set $A \subseteq X$, which yields

$$\lambda_{k+1}^{\ell}(B) = \int_{x^{\ell} \in X^{\ell}} \mathbf{1}_{A_i}(x_i) \lambda_k^{\ell}(\mathrm{d}x^{\ell}) = \lambda_k^{\ell}(B).$$

By the induction hypothesis, this is $\lambda_i(A_i)$, and the proof is done. □

Similarly as for the original system, for any $k \in \mathbb{N} \cup +\{\infty\}$, each lifted state probability measure λ_k^{ℓ} admits a *lifted state probability density* with respect to the lifted invariant measure μ^{ℓ} defined in (5.6). It is defined as

$$f_k^{\ell} := \frac{\mathrm{d}\lambda_k^{\ell}}{\mathrm{d}\mu^{\ell}}$$

for $k = 1, 2, \ldots$, and belongs to $L^1(\mu^{\ell})$. The lifted system Σ^{ℓ} also admits a *lifted transfer operator* $T_{\ell} : L^1(\mu^{\ell}) \to L^1(\mu^{\ell})$ that is such that $T_{\ell} f_k^{\ell} = f_{k+1}^{\ell}$. Following Assumption 5.4, we will restrict our analysis to $L^2(\mu^{\ell})$ functions and simply consider $T_{\ell} : L^2(\mu^{\ell}) \to L^2(\mu^{\ell})$.

Remark 5.18 (Notations)**.** Study of lifted measures and lifted probability density functions are at the center of this chapter. We therefore draw the reader's attention to the fact that, all along this chapter, we denote lifted measures (resp. densities) on $(X^{\ell}, \sigma(X)^{\ell})$ with a superscript λ^{ℓ} (resp. f^{ℓ}), whereas original measures (resp. densities) on $(X, \sigma(X))$ are without any superscript λ (resp. f). In contrast, superscripts on operators, e.g., T^k, denote powers (or recursive applications of the operator).

We now interpret Algorithm 4 by making the link between the lifted system and steps 1 and 2 of Algorithm 4. In step 1, under Assumption 5.14,

it holds that, for any $(a_0, \dots, a_\ell) \in A^{\ell+1}$,

$$\begin{aligned} F_\mu(a_0) &= \mu([\![a_0]\!]), \\ F_\mu(a_0, \dots, a_{\ell-1}) &= \mu^\ell([\![a_0]\!] \times \cdots \times [\![a_{\ell-1}]\!]), \\ F_\mu(a_0, \dots, a_\ell) &= \mu^{\ell+1}([\![a_0]\!] \times \cdots \times [\![a_\ell]\!]). \end{aligned}$$

In (5.1), the quantity is therefore defined as

$$\begin{aligned} (\mathbf{P}_\ell)_{a_0,\dots,a_{\ell-1},a_1\dots,a_\ell} &= \frac{\mu^{\ell+1}([\![a_0]\!] \times \cdots \times [\![a_\ell]\!])}{\mu^\ell([\![a_0]\!] \times \cdots \times [\![a_{\ell-1}]\!])} \\ &:= \mathbb{P}_\mu \left(\mathsf{y}_\ell = a_\ell | \mathsf{y}_0 = a_0, \dots, \mathsf{y}_{\ell-1} = a_{\ell-1} \right), \end{aligned}$$

where this notation follows from the fact that this quantity is the probability of y_ℓ to take value a_ℓ given that $\mathsf{y}_0 = a_0, \dots, \mathsf{y}_{\ell-1} = a_{\ell-1}$ in the original system initialised at the invariant measure μ. In step 2, following a similar reasoning, the computed quantity is

$$(\tilde{\mathbf{f}}_0^\ell)_{a_0 \dots a_{\ell-1}} = \lambda_0^\ell([\![a_0]\!] \times \cdots \times [\![a_{\ell-1}]\!]).$$

5.4 Abstractions as Galerkin approximations

In the previous section, we developed a method to propagate approximate densities thanks to ℓ-memory Markov models. In this section we show that such Markov models are in fact Galerkin approximations of the transfer operators of the lifted systems introduced above. The goal of this theoretical result is twofold: (1) it will allow us to interpret and explain the final steps of Algorithm 4, and (2) it will be useful to analyse our method in Section 5.5.

In the context of Galerkin approximations, we will consider the subspace of functions $D^\ell \subset L^2(\mu^\ell)$ defined as

$$D^\ell := \operatorname{span}\left(\{\psi_{a_1,\dots,a_\ell}\}_{a_1,\dots,a_\ell \in A^\ell} \right),$$

with $\psi_{a_1,\dots,a_\ell} : X^\ell \to \mathbb{R}$ defined as

$$\psi_{a_1,\dots,a_\ell}(x_1, \dots, x_\ell) := \frac{\mathbf{1}_{[\![a_1]\!]}(x_1) \dots \mathbf{1}_{[\![a_\ell]\!]}(x_\ell)}{\mu^\ell([\![a_1]\!] \times \cdots \times [\![a_\ell]\!])}$$

for all $(x_1, \dots, x_\ell) \in X^\ell$. We see that D^ℓ is a set of *piecewise constant* func-

tions on X^ℓ.

In Algorithm 4, we compute (1) the vector representation of the approximated densities in the basis $\{\psi_{a_1,\dots,a_\ell}\}_{a_1,\dots,a_\ell \in A^\ell}$, and (2) the matrix representation of the Galerkin approximation of the lifted transfer operator. This is formally proven in the two following propositions.

Proposition 5.19. *Let $\lambda^\ell : \sigma(X)^\ell \to [0,1]$ be a probability measure, and let $f^\ell \in L^2(\mu^\ell)$ be defined as*

$$f^\ell := \frac{\mathrm{d}\lambda^\ell}{\mathrm{d}\mu^\ell}.$$

Then, it holds that

$$Q_\ell f^\ell = \sum_{a_1,\dots,a_\ell \in A^\ell} \lambda^\ell([\![a_1]\!] \times \cdots \times [\![a_\ell]\!])\psi_{a_1,\dots,a_\ell},$$

where Q_ℓ is the projection operator onto D^ℓ.

Proof. First, let us note that the functions

$$\varphi_{a_1,\dots,a_\ell} := \frac{\mathbf{1}_{[\![a_1]\!]}(x_1)\dots\mathbf{1}_{[\![a_\ell]\!]}(x_\ell)}{\sqrt{\mu^\ell([\![a_1]\!] \times \cdots \times [\![a_\ell]\!])}}$$

form an orthonormal basis for D^ℓ. Therefore, it holds that

$$Q_\ell f^\ell = \sum_{a_1,\dots,a_\ell \in A^\ell} \langle f^\ell, \varphi_{a_1,\dots,a_\ell}\rangle \varphi_{a_1,\dots,a_\ell}.$$

Now, we compute

$$\begin{aligned}\langle f^\ell, \varphi_{a_1,\dots,a_\ell}\rangle &= \int_{x_1 \in [\![a_1]\!]} \cdots \int_{x_\ell \in [\![a_\ell]\!]} \frac{f^\ell(x_1,\dots,x_\ell)}{\sqrt{\mu^\ell([\![a_1]\!] \times \cdots \times [\![a_\ell]\!])}} \mu^\ell(\mathrm{d}(x_1,\dots,x_\ell)) \\ &= \frac{\lambda^\ell([\![a_1]\!] \times \cdots \times [\![a_\ell]\!])}{\sqrt{\mu^\ell([\![a_1]\!] \times \cdots \times [\![a_\ell]\!])}}.\end{aligned}$$

Therefore,

$$\begin{aligned} Q_\ell f^\ell &= \sum_{a_1,\dots,a_\ell \in A^\ell} \frac{\lambda^\ell([\![a_1]\!] \times \cdots \times [\![a_\ell]\!])}{\sqrt{\mu^\ell([\![a_1]\!] \times \cdots \times [\![a_\ell]\!])}} \varphi_{a_1,\dots,a_\ell} \\ &= \sum_{a_1,\dots,a_\ell \in A^\ell} \lambda^\ell([\![a_1]\!] \times \cdots \times [\![a_\ell]\!]) \psi_{a_1,\dots,a_\ell}, \end{aligned}$$

which concludes the proof. □

Proposition 5.20. *Let $Q_\ell : L^2(\mu^\ell) \to D^\ell$ be the projection operator on D^ℓ such as defined in Definition 5.11, and let*

$$P_\ell := Q_\ell T_\ell Q_\ell$$

be the Galerkin approximation of T_ℓ on D^ℓ. Then it holds that

$$\begin{aligned} &P_\ell \psi_{a_0 \dots a_{\ell-1}} \\ &\quad = \sum_{a_\ell=1}^{n} \mathbb{P}_\mu \left(\mathsf{y}_\ell = a_\ell | \mathsf{y}_0 = a_0, \dots, \mathsf{y}_{\ell-1} = a_{\ell-1} \right) \psi_{a_1 \dots a_\ell} \end{aligned} \tag{5.7}$$

for all $(a_0, \dots, a_{\ell-1}) \in A^\ell$. Therefore $\mathbf{P}_\ell$, as defined in (5.1), is the matrix representation of P_ℓ in the basis $\{\psi_{a_1,\dots,a_\ell}\}_{a_1,\dots,a_\ell \in A^\ell}$.

Proof. In the following, we use the same orthonormal basis $\{\varphi_{a_1,\dots,a_\ell}\}$ as in the proof of Proposition 5.19.

By definition of P_ℓ , for all $(a_0, \dots, a_{\ell-1}) \in A^\ell$,

$$\begin{aligned} P_\ell \psi_{a_0,\dots,a_{\ell-1}} &= Q_\ell T_\ell Q_\ell \psi_{a_0,\dots,a_{\ell-1}} \\ &= Q_\ell \left(T_\ell \psi_{a_0,\dots,a_{\ell-1}} \right) \\ &= \sum_{b_1,\dots,b_\ell \in A^\ell} \langle T_\ell \psi_{a_0,\dots,a_{\ell-1}}, \varphi_{b_1,\dots,b_\ell} \rangle \varphi_{b_1,\dots,b_\ell} \\ &= \sum_{b_1,\dots,b_\ell \in A^\ell} \frac{\left\langle T_\ell \left(\mathbf{1}_{a_0} \dots \mathbf{1}_{a_{\ell-1}} \right), \mathbf{1}_{b_1} \dots \mathbf{1}_{b_\ell} \right\rangle}{\mu^\ell([\![a_0]\!] \times \cdots \times [\![a_{\ell-1}]\!])} \psi_{b_1,\dots,b_\ell}. \end{aligned} \tag{5.8}$$

By (5.4) , the dot product above equals zero if $b_1 \dots b_{\ell-1} \neq a_1 \dots a_{\ell-1}$, and,

otherwise, is equal to

$$\int_{x_0 \in [\![a_0]\!]} \cdots \int_{x_{\ell-1} \in [\![a_{\ell-1}]\!]} \tau([\![b_\ell]\!]|x_{\ell-1})\tau(dx_{\ell-1}|x_{\ell-2}) \dots \mu(\mathrm{d}x_1|x_0)\mu(\mathrm{d}x_0) = \mu^{\ell+1}([\![a_0]\!] \times \cdots \times [\![a_{\ell-1}]\!] \times [\![b_\ell]\!]).$$

Therefore, inserting this in (5.8) yields

$$P_\ell \psi_{b_0,\dots,b_{\ell-1}} = \sum_{b_1,\dots,b_\ell \in A^\ell} \frac{\mu^{\ell+1}([\![a_0]\!] \times \cdots \times [\![a_{\ell-1}]\!] \times [\![b_\ell]\!])}{\mu^\ell([\![a_0]\!] \times \cdots \times [\![a_{\ell-1}]\!])} \psi_{b_1,\dots,b_\ell},$$

which is (5.7), and the proof is completed. □

We now explain the end of Algorithm 4 in light of these results. Following Proposition 5.19 and 5.20, in steps 1 and 2, we compute the matrix representation of the operator $P_\ell := Q_\ell T_\ell Q_\ell$ and the vector representation of the function

$$\tilde{f}_0^\ell := Q_\ell f_0^\ell,$$

where f_0^ℓ is the density of the measure λ_0^ℓ. This implies that, in step 3, we compute the vector representation of the vector representation of the function

$$\tilde{f}_{k-\ell+1}^\ell := (P_\ell)^{k-\ell+1} f_0^\ell,$$

Let $\tilde{f}_{\ell,k} \in L^2(\mu)$ be defined as

$$\tilde{f}_{\ell,k}(x_k) := \int_{x_{\ell-k+1} \in X} \cdots \int_{x_{k-1} \in X} \tilde{f}_{k-\ell+1}^\ell(x_{k-\ell+1}, \dots, x_k)\mu^{\ell-1}(\mathrm{d}(x_{k-\ell+1}, \dots, x_{k-1})).$$

In step 4, we compute the vector representation of this function in the basis $\{\psi_a\}_{a \in A}$, and we return it at the end of the algorithm. Figure 5.2 summarizes this interpretation.

5.5 Total Variation guarantees

In this section we leverage the theoretical result presented in the previous section and we prove that there exists two functions $\overline{\mathsf{TV}}_{\mathrm{inc}}(k)$ and $\overline{\mathsf{TV}}_{\mathrm{dec}}(k)$, respectively increasing and decreasing with k, such that

$$\mathsf{TV}(\lambda_k, \tilde{\lambda}_{\ell,k}) \leq \min\left\{\overline{\mathsf{TV}}_{\mathrm{inc}}(k), \overline{\mathsf{TV}}_{\mathrm{dec}}(k)\right\}.$$

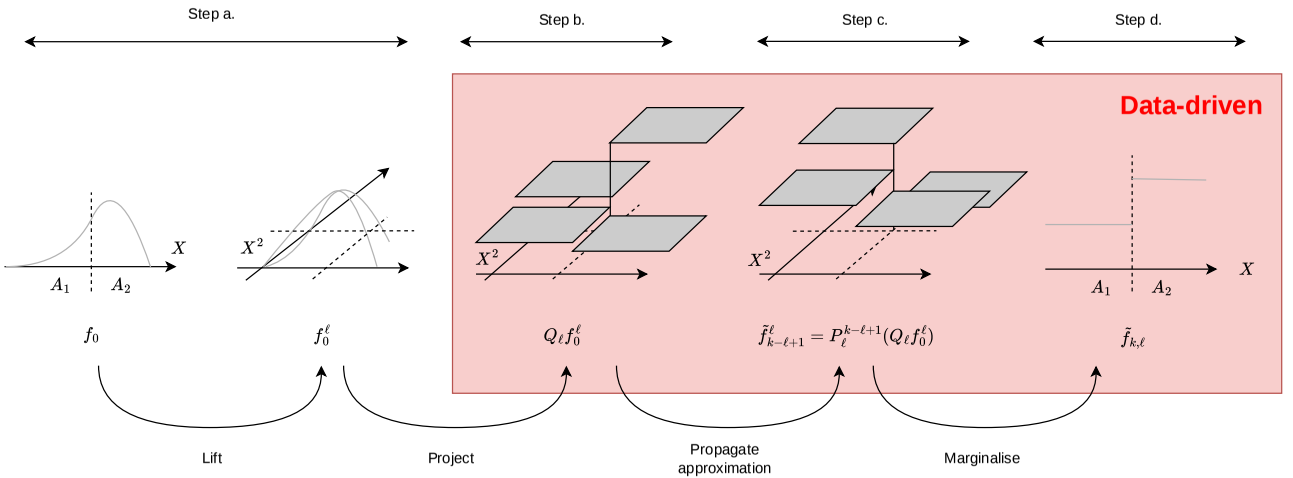


Fig. 5.2 Summary of the method described in Algorithm 4, from left to right. Step a.) We consider the lifted process defined in Definition 5.15. Step b.) We project the x_0-to-$x_{\ell-1}$ joint distribution on a finitely generated space D^ℓ. Step c.) We propagate it with the Galerkin approximation P_ℓ to get the approximate $x_{k-\ell+1}$-to-x_k joint distribution. Step d.) We marginalize it to retrieve the approximate density at x_k. The steps b., c. and d. can be done by sampling trajectories from the system.

This result is crucial for the safety of our abstractions, as it provides a rigorous bridge between the abstraction and the original system. Specifically, the TV bound allows us to transfer safety guarantees from the abstract model to the true system: if we verify that the probability of reaching an unsafe set of states $U \subseteq X$ is at most ε in the abstraction, then the actual probability for the original system is guaranteed to be no greater than $\varepsilon + \mathsf{TV}(\lambda_k, \tilde{\lambda}_{\ell,k})$. Consequently, by ensuring that the abstraction is "safe enough" to account for this error margin, we can perform safety-critical analysis on the abstraction while maintaining the safety certificate for the original dynamics.

To show this overall result, we will prove separately the increasing and decreasing results, respectively in Theorem 5.24 and Theorem 5.25. Both results rely on Lemma 5.23 proven in [SNS10]. The dependencies between these results are summarised in Figure 5.3.

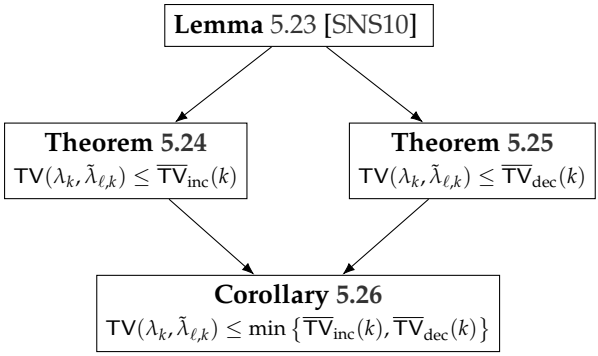


Fig. 5.3 Summary of the main results of Section 5.5.

Similarly as in the works [NFW13, PWS+11, SNS10], our results rely on the spectral properties of the lifted transfer operator.

Assumption 5.21. The lifted transfer operator T_ℓ admits m real eigenvalues $e_{\ell,1}, \ldots, e_{\ell,m}$ such that $1 = e_{\ell,0} > e_{\ell,1} \geq \ldots e_{\ell,m}$, together with an orthonormal set of eigenfunctions $\mathbb{1} = g_0^\ell, g_1^\ell, \ldots, g_m^\ell$ (see Definition 5.8). Moreover, for some $r_\ell < |e_{\ell,m}|$, all remaining eigenvalues $e_{\ell,i} \in \mathbb{C}$ for $i > m$ are such that $|e_{\ell,i}| < r_\ell$. Finally let $\Pi_\ell : L^2(\mu^\ell) \to L^2(\mu^\ell)$ be the operator defined by

$$\Pi_\ell f^\ell = \sum_{i=1}^{m} \langle f^\ell, g_i^\ell \rangle g_i^\ell. \tag{5.9}$$

for any $f^\ell \in L^2(\mu^\ell)$. We assume that $\Pi_\ell T_\ell = \Pi_\ell T_\ell \Pi_\ell$.

Remark 5.22. Assumption 5.21 is related but not the same as Assumption 5.1 and Assumption 5.3. As discussed in [SNS10], sufficient conditions for Assumption 5.21 to hold are *reversibility* and *sufficient ergodicity* (such as defined in [SNS10, Remark 2.1]), which are stronger assumptions than Assumption 5.1 and Assumption 5.21. As stated in [SNS10, KM11], reversibility and sufficient ergodicity are natural and satisfied for a large class of dynamical systems.

Especially, our bound relies on the quantity

$$\delta_\ell := \max_{i=1,\dots,m} \|Q_\ell g_i^\ell - g_i^\ell\|_2, \tag{5.10}$$

where g_i are the eigenfunctions of T_ℓ as described in Assumption 5.21. δ_ℓ therefore quantifies the maximal projection error on the spectrum.

We are now able to prove the results depicted in Figure 5.3.

Lemma 5.23 ([SNS10, Lemma 2.2])**.** *For all initial measure λ_0^ℓ with density $f_0^\ell \in L^2(\mu^\ell)$, and $k \geq \ell - 1$ it holds that*

$$\|P_\ell^{k-\ell+1} f_0^\ell - \mathbb{1}\|_2 \leq \|(T_\ell Q_\ell)^{k-\ell+1} f_0^\ell - \mathbb{1}\|_2 \leq e_{\ell,1}^{k-\ell+1} \|f_0^\ell\|_2,$$

where $e_{\ell,1}$ is the second largest eigenvalue of the transfer operator T_ℓ, as described in Assumption 5.21.

Lemma 5.23 quantifies the difference between an approximated projected joint density of the system, and the density of the invariant measure.[6] We first prove the increasing bound in Theorem 5.24.

Theorem 5.24 (Increasing upper bound)**.** *For any memory $\ell \geq 1$, horizon $k \geq \ell$ and initial measure λ_0^ℓ with density $f_0^\ell \in L^2(\mu^\ell)$, let $\tilde{\lambda}_{k-\ell+1}^\ell$ be the measure whose density is*

$$\tilde{f}_{k-\ell+1}^\ell = P_\ell^{k-\ell+1} f_0^\ell,$$

and similarly for $\tilde{\lambda}_{k-\ell}^\ell$. Then, under Assumption 5.21, it holds that

$$\begin{aligned} \mathsf{TV}(\lambda_{k-\ell+1}^\ell, \tilde{\lambda}_{k-\ell+1}^\ell) \leq \mathsf{TV}(\lambda_{k-\ell}^\ell, \tilde{\lambda}_{k-\ell}^\ell) \\ + \frac{1}{2} \left(|A| e_{\ell,1} \delta_\ell + r_\ell\right) e_{\ell,1}^{k-\ell} \|f_0^\ell\|_2, \end{aligned} \tag{5.11}$$

where δ_ℓ is defined in (5.10)*.*

[6] We recall that $\mathbb{1}$ is the density of the invariant measure, see Remark 5.9.

Proof. First, by the triangular inequality, it holds that

$$\mathsf{TV}(\lambda^\ell_{k-\ell+1}, \tilde{\lambda}^\ell_{k-\ell+1}) \leq \mathsf{TV}(\lambda^\ell_{k-\ell+1}, \overline{\lambda}^\ell_{k-\ell+1}) + \mathsf{TV}(\overline{\lambda}^\ell_{k-\ell+1}, \tilde{\lambda}^\ell_{k-\ell+1}),$$

where $\lambda^\ell_{k-\ell+1}, \overline{\lambda}^\ell_{k-\ell+1}$ and $\tilde{\lambda}^\ell_{k-\ell+1}$ are respectively the measures corresponding to

$$\begin{aligned} f^\ell_{k-\ell+1} &= T^{k-\ell+1}_\ell f^\ell_0, \\ \overline{f}^\ell_{k-\ell+1} &= T_\ell P^{k-\ell}_\ell f^\ell_0, \\ \tilde{f}^\ell_{k-\ell+1} &= P^{k-\ell+1}_\ell f^\ell_0. \end{aligned}$$

First, we show that

$$\mathsf{TV}(\lambda^\ell_{k-\ell+1}, \overline{\lambda}^\ell_{k-\ell+1}) \leq \mathsf{TV}(\lambda^\ell_{k-\ell}, \tilde{\lambda}^\ell_{k-\ell}).$$

It holds that $f^\ell_{k-\ell+1} = T_\ell f^\ell_{k-\ell}$, and that $\overline{f}^\ell_{k-\ell+1} = T_\ell \tilde{f}^\ell_{k-\ell}$. Therefore, by definition of the total variation distance and by Proposition 5.7,

$$\begin{aligned} \mathsf{TV}(\lambda^\ell_{k-\ell+1}, \overline{\lambda}^\ell_{k-\ell+1}) &= \frac{1}{2} \left\| T_\ell \left(f^\ell_{k-\ell} - \tilde{f}^\ell_{k-\ell} \right) \right\|_1 \\ &\leq \frac{1}{2} \|T_\ell\|_1 \|f^\ell_{k-\ell} - \tilde{f}^\ell_{k-\ell}\|_1 \\ &= \frac{1}{2} \|f^\ell_{k-\ell} - \tilde{f}^\ell_{k-\ell}\|_1 \\ &= \mathsf{TV}(\lambda^\ell_{k-\ell}, \tilde{\lambda}^\ell_{k-\ell}). \end{aligned}$$

Second, we show that

$$\mathsf{TV}(\overline{\lambda}^\ell_{k-\ell+1}, \tilde{\lambda}^\ell_{k-\ell+1}) \leq \frac{1}{2} (m e_{\ell,1} \delta_\ell + r_\ell) e^{k-\ell}_{\ell,1} \|f^\ell_0\|_2.$$

It holds that $\overline{f}^\ell_{k-\ell+1} = T_\ell \tilde{f}^\ell_{k-\ell}$, and that $\tilde{f}^\ell_{k-\ell+1} = P_\ell \tilde{f}^\ell_{k-\ell} = Q_\ell T_\ell \tilde{f}^\ell_{k-\ell}$. Therefore, by definition of the total variation distance, and by Hölder's inequality, it holds that

$$\begin{aligned} \mathsf{TV}(\overline{\lambda}^\ell_{k-\ell+1}, \tilde{\lambda}^\ell_{k-\ell+1}) &= \frac{1}{2} \|(T_\ell - Q_\ell T_\ell) \tilde{f}^\ell_{k-\ell}\|_1 \\ &\leq \frac{1}{2} \|(T_\ell - Q_\ell T_\ell) \tilde{f}^\ell_{k-\ell}\|_2. \end{aligned}$$

Now we follow a similar reasoning as in the proof of [SNS10, Theorem 3.1].

Let $Q_\ell^\perp = \mathrm{Id} - Q_\ell$, where Id is the identity operator. Since $Q_\ell^\perp \mathbb{1} = 0$, it holds that

$$\begin{aligned}\|(T_\ell - Q_\ell T_\ell)\tilde{f}_{k-\ell}^\ell\|_2 &= \|Q_\ell^\perp T_\ell \tilde{f}_{k-\ell}^\ell\|_2 \\ &= \|Q_\ell^\perp T_\ell Q_\ell (T_\ell Q_\ell)^{k-\ell} \tilde{f}_0^\ell\|_2 \\ &= \|Q_\ell^\perp T_\ell Q_\ell ((T_\ell Q_\ell)^{k-\ell} \tilde{f}_0^\ell - \mathbb{1})\|_2 \\ &\leq \|Q_\ell^\perp T_\ell Q_\ell\|_2 \|(T_\ell Q_\ell)^{k-\ell} \tilde{f}_0^\ell - \mathbb{1}\|_2.\end{aligned}$$

Following [SNS10, Eq. (63)], the first factor is such that

$$\|Q_\ell^\perp T_\ell Q_\ell\|_2 \leq m e_{\ell,1} \delta_\ell,$$

and, by Lemma 5.23, the second is such that

$$\|(T_\ell Q_\ell)^{k-\ell} \tilde{f}_0^\ell - \mathbb{1}\|_2 \leq e_{\ell,1}^{k-\ell} \|f_0^\ell\|_2,$$

which concludes the proof. □

We now prove the decreasing bound in Theorem 5.25.

Theorem 5.25 (Decreasing upper bound)**.** *For any memory $\ell \geq 1$, horizon $k \geq \ell$ and initial measure λ_0^ℓ with density $f_0^\ell \in L^2(\mu^\ell)$, let $\tilde{\lambda}_{k-\ell+1}^\ell$ be the measure defined as in Theorem 5.24. Then, under Assumption 5.21, it holds that*

$$\mathsf{TV}(\lambda_{k-\ell+1}^\ell, \tilde{\lambda}_{k-\ell+1}^\ell) \leq e_{\ell,1}^{k-\ell+1} \|f_0^\ell\|_2.$$

Proof. By the triangular inequality, it holds that

$$\mathsf{TV}(\lambda_{k-\ell+1}^\ell, \tilde{\lambda}_{k-\ell+1}^\ell) \leq \mathsf{TV}(\lambda_{k-\ell+1}^\ell, \mu^\ell) + \mathsf{TV}(\mu^\ell, \tilde{\lambda}_{k-\ell+1}^\ell).$$

By Hölder's inequality and by Assumption 5.21, it holds that

$$\mathsf{TV}(\lambda_{k-\ell+1}^\ell, \mu^\ell) \leq \frac{1}{2}\|f_{k-\ell+1}^\ell - \mathbb{1}\|_2 \leq \frac{1}{2} e_{\ell,1}^{k-\ell+1} \|f_0^\ell\|_2.$$

By Lemma 5.23, we can follow the exact same reasoning as above to get

$$\mathsf{TV}(\mu^\ell, \tilde{\lambda}_{k-\ell+1}^\ell) \leq \frac{1}{2} e_{\ell,1}^{k-\ell+1} \|f_0^\ell\|_2,$$

which concludes the proof. □

Finally, we can combine the results from 5.24 and 5.25 to obtain the

overall bound in Corollary 5.26.

Corollary 5.26. *For any memory $\ell \geq 1$, horizon $k \geq \ell$ and initial measure λ_0^ℓ with density $f_0^\ell \in L^2(\mu^\ell)$, let*

$$\overline{\mathsf{TV}}_{inc}(k) := \mathsf{TV}(\lambda_0^\ell, \tilde{\lambda}_0^\ell) + (m e_{\ell,1}\delta_\ell + r_\ell)\,\frac{1 - e_{\ell,1}^{k-\ell+1}}{1 - e_{\ell,1}}\|f_0^\ell\|_2,$$
$$\overline{\mathsf{TV}}_{dec}(k) := e_{\ell,1}^{k-\ell+1}\|f_0^\ell\|_2,$$

where δ_ℓ is defined in (5.10)*. Also, let $\tilde{\lambda}_{\ell,k}$ be the measure whose density is $\tilde{f}_{\ell,k}$, the output of Algorithm 4. Then it holds that*

$$\mathsf{TV}(\lambda_k, \tilde{\lambda}_{\ell,k}) \leq \min\left\{\overline{\mathsf{TV}}_{inc}(k), \overline{\mathsf{TV}}_{dec}(k)\right\}.$$

Proof. First, by Theorem 5.24, it holds that

$$\begin{aligned}
\mathsf{TV}(\lambda_{k-\ell+1}^\ell, \tilde{\lambda}_{k-\ell+1}^\ell) &\leq \mathsf{TV}(\lambda_0^\ell, \tilde{\lambda}_0^\ell) + \sum_{i=0}^{k-\ell}\frac{1}{2}\,(m e_{\ell,1}\delta_\ell + r_\ell)\, e_{\ell,1}^i\|f_0^\ell\|_2 \\
&= \mathsf{TV}(\lambda_0^\ell, \tilde{\lambda}_0^\ell) + \frac{1}{2}\,(m e_{\ell,1}\delta_\ell + r_\ell)\left(\sum_{i=0}^{k-\ell} e_{\ell,1}^i\right)\|f_0^\ell\|_2 \\
&= \mathsf{TV}(\lambda_0^\ell, \tilde{\lambda}_0^\ell) + \frac{1}{2}\,(m e_{\ell,1}\delta_\ell + r_\ell)\,\frac{1 - e_{\ell,1}^{k-\ell+1}}{1 - e_{\ell,1}}\|f_0^\ell\|_2.
\end{aligned}$$

Now it remains to show that

$$\mathsf{TV}(\lambda_k, \tilde{\lambda}_{\ell,k}) \leq \mathsf{TV}(\lambda_{k-\ell+1}^\ell, \tilde{\lambda}_{k-\ell+1}^\ell).$$

By definition of the total variation distance,

$$\begin{aligned}
\mathsf{TV}(\lambda_{k-\ell+1}^\ell, \tilde{\lambda}_{k-\ell+1}^\ell) &= \sup_{A_{k-\ell+1},\ldots,A_k \in \sigma(X)} \left|\begin{array}{l}\lambda_{k-\ell+1}^\ell(A_{k-\ell+1}\times\cdots\times A_k) \\ \quad - \lambda_{k-\ell+1}^\ell(A_{k-\ell+1}\times\cdots\times A_k)\end{array}\right| \\
&\geq \sup_{A_k \in \sigma(X)} \left|\begin{array}{l}\lambda_{k-\ell+1}^\ell(E\times\cdots\times E\times A_k) \\ \quad - \lambda_{k-\ell+1}^\ell(E\times\cdots\times E\times A_k)\end{array}\right| \\
&= \sup_{A_k \in \sigma(X)} |\lambda_k(A_k) - \lambda_{\ell,k}(A_k)| \\
&= \mathsf{TV}(\lambda_k, \lambda_{\ell,k}),
\end{aligned}$$

and the proof is completed. □

5.6 Numerical experiments

In this section we connect the method described in Section 5.3 with the two main motivations introduced in the introduction of this thesis, namely Motivation 1 and Motivation 2, through two numerical experiments. The first experiment corresponds to Motivation 1: by nature, only partial information on the state is available and the output alphabet is fixed, so that refining the state-space partition is not an option; we show that increasing memory is an effective lever for obtaining more accurate abstractions. The second experiment addresses Motivation 2: the output results from a chosen discretisation of the state space and we compare abstractions of comparable size obtained either by refining the partition or by augmenting memory starting from a coarser one, highlighting the benefits of the latter. An illustration of these two motivations and their corresponding experiments is provided in Figure 5.4. In both cases, we consider the following dynamical system – a particular case of Example 2.12.

Example 5.27. Consider the 2-dimensional linear system of the form

$$x_{k+1} = \mathbf{A}x_k + w_k,$$

where

$$\mathbf{A} = \begin{pmatrix} 0.995 & 0.005 \\ 0 & 0.98 \end{pmatrix},$$

and w_k is the realisation of a Gaussian random variable $\mathsf{w} \sim \mathcal{N}(\mathbf{m}, \mathbf{C})$ with

$$\mathbf{m} := (0,0)^\top \text{ and } \mathbf{C} := 0.07\mathbf{I}.$$

We suppose that this system is initialised at a Gaussian initial measure $\lambda_0 : \sigma(\mathbb{R}^2) \to [0,1]$ defined as $\lambda_0 := \mu_{\mathcal{N}(\mathbf{m}_0, \mathbf{C}_0)}$, with

$$\mathbf{m}_0 := (-0.4, -0.4)^\top \text{ and } \mathbf{C}_0 := 0.3\mathbf{I}.$$

Since A is stable, this system converges in total variation to a unique invariant distribution $\mu = \mathcal{N}(\mathbf{m}_\star, \mathbf{C}_\star)$, where

$$\mathbf{m}_\star = (0,0)^\top \text{ and } \mathbf{C}_\star \approx \begin{pmatrix} 7.36896 & 0.347856 \\ 0.347856 & 1.76768 \end{pmatrix}. \tag{5.12}$$

The stationary covariance matrix $\mathbf{C}_\star$ was computed by solving the Riccati

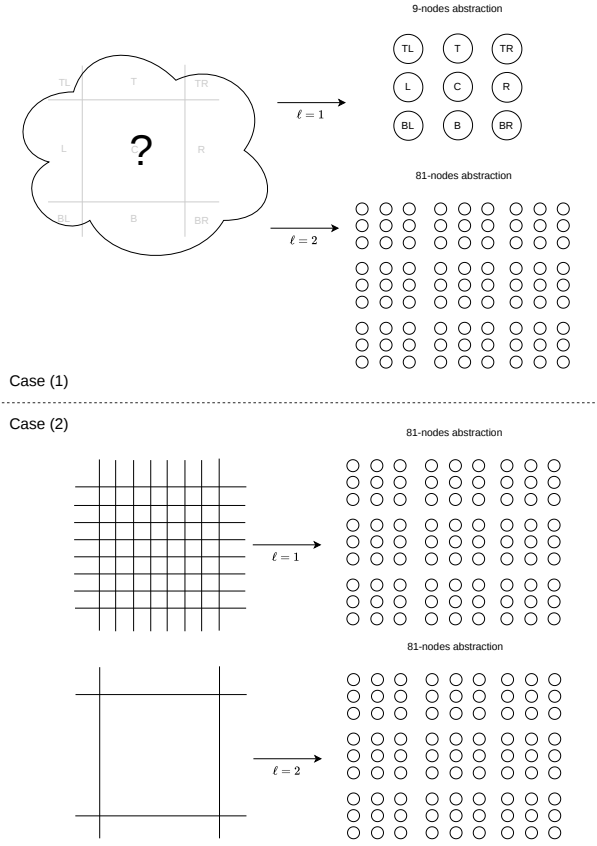


Fig. 5.4 Illustration of the two numerical use cases in direct correspondence with the motivations introduced in the Introduction. In case (1) above (Motivation 1), the system is only partially observable: we have no control over the underlying state space and can only access coarse, fixed observations (e.g. left, top-left, etc.), so that refining the partition is impossible; the associated experiment therefore investigates how increasing memory alone reduces epistemic uncertainty and improves the abstraction. In case (2) below (Motivation 2), we are in the finite-abstraction setting where the discretisation of the state space can be chosen; the corresponding experiment compares abstractions of comparable size obtained either by refining the partition with memory 1 or by augmenting memory starting from a coarser partition, highlighting when memory-based constructions yield better approximations for the same size budget.

equation $\mathbf{C}_\star = \mathbf{A}\mathbf{C}_\star\mathbf{A}^\top + \mathbf{C}$ with the `MatrixEquations.jl` package.[7]

In the experiments below, the matrices $\mathbf{P}_\ell$ have been computed with one very long trajectory $\{y_i\}_{i=1,\dots,10^5}$ (see Remark 3.10), and the initial vectors $\mathbf{f}_0^\ell$ have been computed with $10^5/\ell$ samples of length ℓ.

Case (1) - Partial information is accessible In this case, we only have access to the outputs. The state space is discretized as follows: each dimension is partitioned into $(-\infty, -1)$, $(1, \infty)$, and the interval $[-1, 1]$ is further partitioned into p subintervals of equal size. Thus the partition contains $n = (p+2)^2$ cells. In this case, we fix $p = 3$, leading to a 25-cell partition. For $\ell = 1, 2, 3$, we compute $\tilde{f}_{\ell,k}$ with Algorithm 4, and compute $\mathrm{TV}(\lambda_k, \tilde{\lambda}_{\ell,k})$ for $k \in \{0, \dots, 100\}$.

The results are in Figure 5.5. One can see that increasing memory reduces $\mathrm{TV}(\lambda_k, \tilde{\lambda}_{\ell,k})$ for most horizons k, **thereby increasing the approximation quality**. Moreover, one can see that the observed TV distances seem to follow two regimes, first increasing then decreasing, which is coherent with the theoretical bounds Theorem 5.24 and Theorem 5.25.

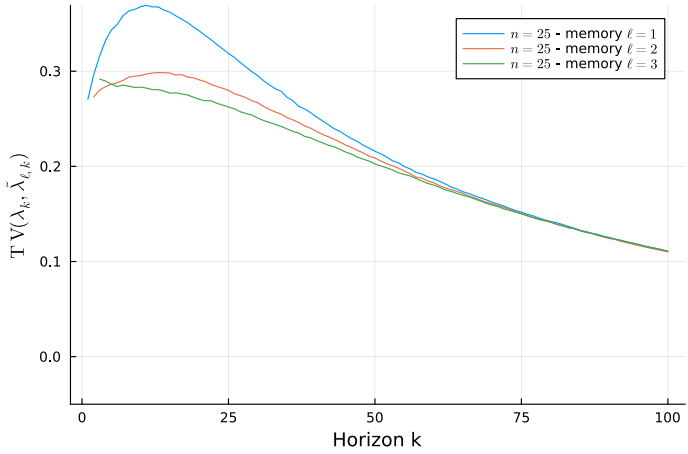


Fig. 5.5 Approximation quality of ℓ-memory Markov models for partially observable systems. Introducing memory improves approximation quality.

[7] See `https://github.com/andreasvarga/MatrixEquations.jl`.

Case (2) - Discretized state space In this case, the state process of the system is available, but the state space is discretized in order to abstract the system. We consider two comparable settings:

1. (Classical Markov chain) The state space is discretized in $n = 729$ blocks in the same fashion as above (with $p = 25$). Memory $\ell = 1$ is considered, leading to $729^2 = 531441$ transition probabilities $(\mathbf{P}_1)_{i_1,i_2}$ with $i_1, i_2 \in \{1, \dots, 729\}$.
2. (2-memory Markov model) The state space is discretized in the same way (uniformly in the square $[-1,1]^2$) in $n = 81$ blocks ($p = 7$). Memory $\ell = 2$ is considered, also leading to $81^3 = 531441$ transition probabilities $(\mathbf{P})_{i_1 i_2, i_2 i_3}$ with $i_1, i_2, i_3 \in \{1, \dots, 81\}$.

The two settings lead to discrete objects of the same size, since one only needs to store 531441 values to save them. For these two settings, we compute $\tilde{f}_{\ell,k}$ with Algorithm 4, and compute $\mathrm{TV}(\lambda_k, \tilde{\lambda}_{k\ell,k})$ for $k \in \{0, \dots, 100\}$. The results are in Figure 5.6. We observe that, for the same final abstraction size, the memory-based approach leads to a better approximation, showcasing the fact that memory allows to construct smarter abstractions than classical approaches.

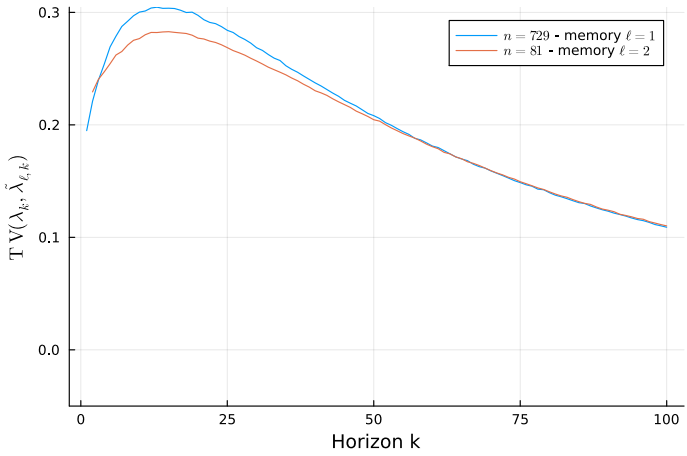


Fig. 5.6 Approximation quality of memory-dependent abstractions of an observable system. One can see that, for the same number of transition probabilities ($n^{\ell+1} = 531441$ in both cases), starting from a coarser partition and increasing the memory leads to better approximations.

5.7 Summary and further research directions

Chapter 5 – Summary

In this chapter, we extended the memory-dependent abstraction framework of Chapter 3 to stochastic systems, where both epistemic and aleatoric uncertainty are present. By integrating tools such as transfer operators and Galerkin approximations, we addressed the fundamental challenge of approximating non-Markovian processes by leveraging memory.

Section 5.1 – Framework

We formalised the class of stochastic dynamical systems under study, emphasising the role of invariant measures and ergodicity.

Section 5.2 – Preliminaries

We introduced the key concepts used in the chapter: transfer operators and their spectrum, projection operators, and Galerkin approximations.

Section 5.3 – Method

A data-driven algorithm (Algorithm 4) was proposed to construct ℓ-memory abstractions. The method leverages sampled trajectories to compute transition probabilities and initial distributions. The lifted system, whose state space encodes ℓ-long sequences, was introduced as a key tool to interpret the method. This approach generalizes deterministic abstractions presented in Chapter 3 by accounting for aleatoric uncertainty through probabilistic state densities.

Section 5.4 – Abstractions as Galerkin approximations

We established that ℓ-memory abstractions correspond to Galerkin approximations of the lifted transfer operator. This connection not only justifies the algorithm steps but also enables the use of functional analytic tools to bound approximation errors.

Section 5.5 – Total Variation guarantees

Rigorous bounds on the total variation distance between the original and abstracted distributions were derived. Two complementary results – (1) an increasing bound (Theorem 5.24) and (2) a decreasing bound (Theorem 5.25) – ensure that the approximation error remains controlled over time, enabling formal safety guarantees for verification and control tasks.

Section 5.6 – Numerical experiments

Numerical case studies demonstrated the practical advantages of memory-dependent abstractions. In scenarios with partial observations, increasing memory consistently improved accuracy. Moreover, for systems with discretized state spaces, memory-based abstractions outperformed classical abstractions of comparable size.

Chapter 5 – Further research directions

We believe that this approach is a novel and efficient manner of propagating uncertainty in partially observable stochastic systems. It provides rigorous guarantees for safety-critical applications, bridging the gap between abstract models and real-world stochastic dynamics. The following directions can be considered for future research.

Characterizing systems for which memory helps

Our numerical experiments suggest that (1) for partially observable systems with a fixed partition, increasing memory allows to improve the approximation quality, and (2) increasing memory may even improve the approximation quality for a constant abstraction size. Identifying sufficient and necessary conditions for which increasing memory guarantees a better approximation – whether in the case (1) or (2) – is an interesting direction for further research.

Reducing Total Variation conservatism

While our current bounds are valid, they suffer from conservatism and often exceed 1, the maximal value of any Total Variation distance. Two research directions are therefore interesting for reducing this conservatism: (1) investigating alternative approaches that directly rely on results on the 1-norm (rather than relying on the 2-norm as we do

here) and (2) using the 2-norm results to directly bound the Wasserstein distance between λ_k and $\tilde{\lambda}_{\ell,k}$, which would then enable the use of distributionally robust optimisation for control and analysis (see e.g. [KENSA19]).

Investigating adaptive memory schemes

In this chapter, the memory length ℓ is fixed a priori, which can lead to unnecessarily large abstractions. In fact, not all memory sequences may contribute equally to the approximation quality. An adaptive approach, where memory is composed of sequences of different lengths (as in Chapter 4), could achieve comparable approximation quality while significantly reducing the abstraction size. Specifically, a stochastic adaptation of the REFINE algorithm from Chapter 4 could be investigated.

Similar further works as in Chapter 3 and Chapter 4

Future directions described in Chapter 3 and Chapter 4 also remain interesting in the context of stochastic abstractions. These include accounting for a finite number of observations, and construct Markov Decision Processes abstractions to add control inputs in the overall method.

PART IV
The Cantor-Kantorovich distance

6

Theoretical foundations of the Cantor-Kantorovich distance

CENTRAL to the adaptive memory method developed in Chapter 4 is the question of *quantifying* the difference between two LMCs Γ and $\tilde{\Gamma}$. Indeed, Algorithm 3 relies on a notion of *distance* between LMCs (such as defined in Chapter 1).

Different distances between LMCs were proposed in the literature. It was mostly driven by the use case in which an LMC $\tilde{\Gamma}$ serves as an *approximation* or *biased* model of another LMC Γ (see, e.g. Example 6.1): it is then of interest to quantify whether $\tilde{\Gamma}$ is a good approximation of Γ or not.

Example 6.1 (Markov's Eugene Onegin text model [Mar13])**.** In the seminal paper [Mar13], Markov counted the transitions between vowels and consonants in one of Eugene Onegin's poems, resulting in the reference Markov chain Σ, depicted in Figure 6.1 with $\varepsilon = 0$. Suppose that such a model was computed only from a subset of the poem, then it could result in an approximate (or biased) Markov chain $\tilde{\Sigma}_\varepsilon$, with an $\varepsilon > 0$. Our goal is to quantify the difference between Σ and its approximation $\tilde{\Sigma}_\varepsilon$.

First, different *bisimilarity distances* (see Definition 1.24) were introduced,

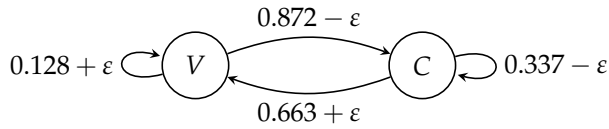


Fig. 6.1 Biased Markov's Eugene Onegin text model [Mar13].

e.g. in [DGJP04, vBSW07, CvBW12, vBW05]. Multiple *trace-based distances* were introduced as well. For example, [CK14, Kie18] consider the TV distance between the asymptotic distributions, while [DHKP16, JMGLM14] expose families of linear distances, sometimes discounted or averaged. Measures that are not pseudo-distances (as defined in Chapter 1) were also introduced, e.g. [RAC04] which measures the *Kullback–Leibler divergence* between the asymptotic distributions.

The research effort introduced in Chapter 4 led to the development of a novel notion of distance that relies on the theory of optimal transport. More precisely, we leverage the Kantorovich (also known as Wasserstein) distance in order to evaluate the difference between two Markov models in terms of the probability distributions that they define on the output language. To define the Kantorovich distance, we equip the language A^ω with the *Cantor distance*, thereby making this novel distance *natural* in the sense that it relies on the distance that generates the *natural topology* $\mathcal{T}^\omega$ on A^ω (see Chapter 1 for more information). More formally, this novel distance is defined as

$$\mathsf{CK}(\Gamma_1, \Gamma_2) = \mathsf{K}_{\mathsf{C}}(\mu_1^\omega, \mu_2^\omega),$$

where $\mu_1^\omega, \mu_2^\omega$ are the distributions generated by Γ_1 and Γ_2 on the measurable space $(A^\omega, \sigma(\mathcal{T}^\omega))$, and is referred to as the *Cantor-Kantorovich distance* (or *CK distance* for short). Other works have investigated Kantorovich-based distances between LMCs, but with another underlying space or distance (see e.g. [MS10, RS18, DGJP04, vBSW07]).

The suitability of a distance for a given application depends on specific characteristics. In this chapter, we focus on two types of characteristics: *continuity* properties (such as defined in Chapter 1) and *computational complexity* (see Appendix E for an overview of computational complexity results). First, we study parameter and (B-LTL)-continuity of the Cantor-Kantorovich metric. Second, we are interested in knowing whether CK is computable and if it is hard to compute. While most distances are either not computable or hard to compute, it is also of the utmost importance to

identify computable approximation schemes, and characterise their computational complexity – see e.g. [CK14] for a proof of NP-hardness of both the exact computation and the approximation of the asymptotic TV distance.

Contributions. The goal of this chapter is to write the theoretical foundations of the Cantor-Kantorovich distance. The main contributions are summarised below.

1. **Definition.** We make the link between the CK distance and the topology generated by the Cantor distance on A^ω, making the CK distance a natural choice for comparing LMCs. We also show that the CK distance is an infinite sum of discounted TV distances, which highlights that it can be seen as an instance of the discounted linear distances studied in [JMGLM14], but one that arises directly from the natural topology.
2. **Exact computation.** We show that the exact computation of the CK distance is #P-hard.
3. **Continuity properties.** We first prove the continuity of the distance by showing that the distance between two similar (in a precise sense) LMCs is upper bounded. We also show that a small CK distance implies a bounded error between probabilities of finite traces and, given the distance $\mathsf{CK}(\Sigma, \tilde{\Sigma})$ and a tolerance ε, we provide a practical method to select an horizon k such that using $\tilde{\Sigma}$ yields an error of at most ε up to that horizon k.
4. **Approximation.** We provide an approximation scheme based on finite sums of discounted TV distances. We show that this scheme is also #P-hard.

Publication. This work has been done in collaboration with Licio Romao, Alessandro Abate and Raphaël M. Jungers. Preliminary results have been published in [BRAJ25], and a pre-print containing additional results can be found in [BAJ25].[1]

[1] The pre-print [BAJ25] is currently under review.

6.1 Definition and link with Total Variation

We first recall the two ingredients used to define the CK distance. Following Chapter 1, for $m := |A|$ and for any two words $w_1 = a_1 a_2 \dots$ and $w_2 = b_1 b_2 \dots$ in A^ω, the Cantor distance is

$$\mathsf{C}(w_1, w_2) = m^{-\inf\{i-1 : a_i \neq b_i\}},$$

and this expression extends in the same way to finite words in A^k. Now, let $\Gamma_1 = (S_1, A, \iota_1, \tau_1, l_1)$ and $\Gamma_2 = (S_2, A, \iota_2, \tau_2, l_2)$ be two LMCs defined over the same alphabet. For a finite horizon k, they respectively generate the measures μ_1^k and μ_2^k on the finite alphabet A^k (see Chapter 1 for more information). Following Appendix D, given the underlying metric space (A^k, C), the Kantorovich distance between μ_1^k and μ_2^k is given by

$$\mathsf{K}_\mathsf{C}(\mu_1^k, \mu_2^k) := \min_{c^k : A^k \times A^k \to [0,1]} \sum_{w_1, w_2 \in A^k} \mathsf{C}(w_1, w_2) c^k(w_1, w_2),$$

subject to

$$\forall w_1 \in A^k, \sum_{w_2 \in A^k} c^k(w_1, w_2) = \mu_1^k(w_1),$$

$$\forall w_2 \in A^k, \sum_{w_1 \in A^k} c^k(w_1, w_2) = \mu_2^k(w_2).$$

In the same setting, the Total Variation distance between μ_1^k and μ_2^k is

$$\mathsf{TV}(\mu_1^k, \mu_2^k) = \frac{1}{2} \sum_{w \in A^k} |\mu_1^k(w) - \mu_2^k(w)|.$$

We refer the reader to Appendix D for a more general definition.

We are now able to introduce the so-called *Cantor-Kantorovich* distance between two LMCs, and provide the link with Total Variation.

Definition 6.2 (Cantor-Kantorovich distance)**.** Let $\Gamma_1 = (S_1, A, \iota_1, \tau_1, l_1)$ and $\Gamma_2 = (S_2, A, \iota_2, \tau_2, l_2)$ be two LMCs defined over the same alphabet. For any $k \in \mathbb{N}$, let μ_1^k and μ_2^k be the probability measures on $(A^k, 2^{A^k})$ generated by Γ_1 and Γ_2, respectively. The Cantor-Kantorovich distance (or CK distance for short) between Γ_1 and Γ_2 is given by

$$\mathsf{CK}(\Gamma_1, \Gamma_2) := \lim_{k \to \infty} \mathsf{K}_\mathsf{C}(\mu_1^k, \mu_2^k). \tag{6.1}$$

Remark 6.3 (On the choice of the underlying metric space)**.** The use of the Cantor metric makes the CK distance very general. However, if one were to use this distance in a specification-driven context – for instance, to find reduced LMCs that are close to the original one in order to verify specific behaviours – it could be interesting as further work to investigate whether other choices of metric space on $A^\star$ would be better suited to the problem at hand.

Following-up on Remark D.15, the computation of the CK distance can be interpreted through the lens of *optimal transport*: for a given horizon k, at each $w \in A^k$, the "supplies" are given by $\mu_1^k(w)$ and the "demands" by $\mu_2^k(w)$. For each $w_1, w_2 \in A^k$, the cost of moving $c^k(w_1, w_2)$ of supply mass from w_1 to w_2 amounts to $c^k(w_1, w_2)\mathsf{C}(w_1, w_2)$. The specific choice of the Cantor distance as the underlying distance allows to visualise this with a tree, as shown in Figure 6.2.

We will now prove the main result of this section, which is that

$$\mathsf{CK}(\Gamma_1, \Gamma_2) = (m-1)\sum_{i=1}^{\infty} m^{-i}\mathsf{TV}(\mu_1^i, \mu_2^i),$$

where m is the number of labels of the LMCs, that is $m = |A|$.

This result (Theorem 6.8) relies on the combination of Proposition 6.7 (from [CK14]), and Proposition 6.6. The latter proposition relies on two important lemmata (Lemma 6.4 and Lemma 6.5). It is worth noting that Lemma 6.4, Lemma 6.5 and Proposition 6.6 all rely on the optimal transport interpretation described above, highlighting the importance of this optimal transport viewpoint. The dependencies between these results are summarised in Figure 6.3.

Lemma 6.4. *Let Γ_1 and Γ_2 be two LMCs defined over the same alphabet A. For any $k \in \mathbb{N}$, let $c^k : A^k \times A^k \to [0,1]$ be the optimal coupling when computing the Kantorovich distance $\mathsf{K}_\mathsf{C}(\mu_1^k, \mu_2^k)$ (see optimal transport problem (D.2)). Then it holds that*

$$c^k(w, w) = \min\{\mu_1^k(w), \mu_2^k(w)\}$$

for all $w \in A^k$.

Proof. We first prove that $c^k(w, w) \leq \min\{\mu_1^k(w), \mu_2^k(w)\}$. Constraint (D.4)

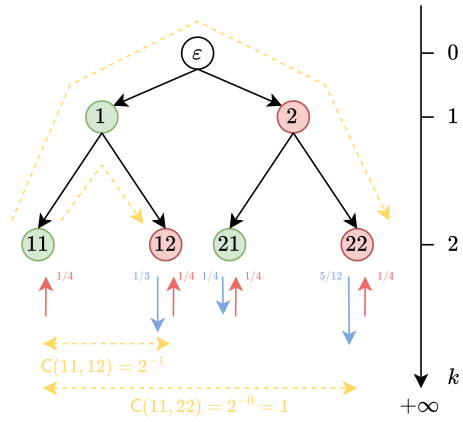


Fig. 6.2 Optimal transport interpretation of the CK distance. Let $k = 2$, and let Γ_1 be a LMC such that $\mu_1^2(w) = 1/4$ for all $w \in \{1,2\}^2$ (depicted in red), and Γ_2 be such that $\mu_2^2(11) = 0$, $\mu_2^2(12) = 1/3$, $\mu_2^2(21) = 1/4$ and $\mu_2^2(22) = 5/12$ (depicted in blue). Computing $\mathsf{K}_\mathsf{C}(\mu_1^2, \mu_2^2)$ amounts to solving the optimal transport problem in which the demands (blue) are satisfied with the supplies (red). To do that, one has to move $(1/3 - 1/4)$ mass from 11 to 12, and $(5/12 - 1/4)$ mass from 11 to 22. These moves are depicted in yellow. The marginal costs of such moves are respectively $\mathsf{C}(11,12) = 2^{-1}$ and $\mathsf{C}(11,22) = 1$; this can be computed by looking at the level in the tree to which the mass has to move ($k = 1$ for the first move, resulting in a 2^{-1} cost, and $k = 0$ for the second move, resulting in a 2^{-0} cost). The CK distance corresponds to this optimal transport problem, but at $k = +\infty$.

imply that

$$\begin{aligned} \mu_1^k(w) &= c^k(w,w) + \sum_{\substack{w' \in A^k \\ w \neq w'}} c^k(w,w') \geq c^k(w,w), \\ \mu_2^k(w) &= c^k(w,w) + \sum_{\substack{w' \in A^k \\ w \neq w'}} c^k(w',w) \geq c^k(w,w), \end{aligned} \tag{6.2}$$

which implies $c^k(w,w) \leq \min\{\mu_1^k(w), \mu_2^k(w)\}$. We now prove that $c^k(w,w) \geq$

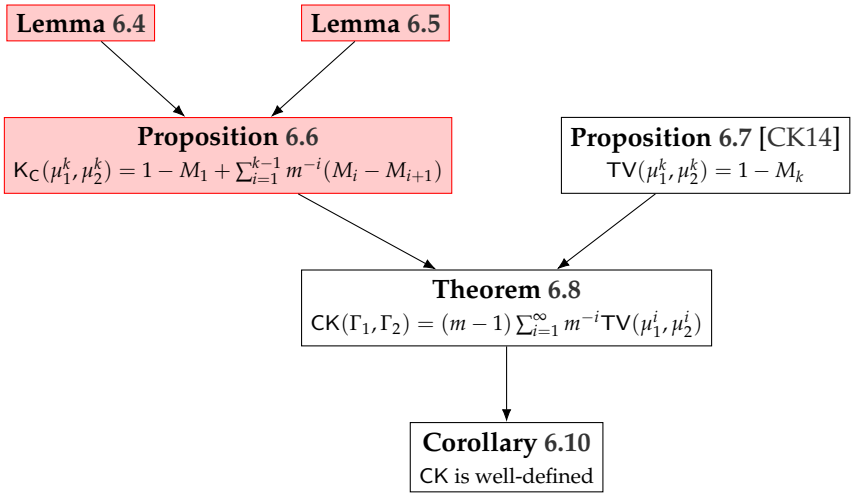


Fig. 6.3 Summary of the main results of Section 6.1. The results leveraging the optimal transport interpretation of the Kantorovich distance are in red.

$\min\{\mu_1^k(w), \mu_2^k(w)\}$. By contradiction, let c^k be an optimal solution such that there exists $w \in A^k$ and $\varepsilon > 0$ with

$$c^k(w, w) = \min\{\mu_1^k(w), \mu_2^k(w)\} - \varepsilon.$$

Assume without loss of generality that $\min\{\mu_1^k(w), \mu_2^k(w)\} = \mu_1^k(w)$, then constraints (D.4) imply that

1. there exists $w' \neq w$ such that $c^k(w, w') = \varepsilon'$ for some $\varepsilon' \in (0, \varepsilon]$, and
2. there exists $w'' \neq w$ such that $c^k(w'', w) = \varepsilon''$ for some $\varepsilon'' \in (0, \varepsilon]$.

Let $\mathsf{K_C}(\mu_1^k, \mu_2^k)$ denote the Kantorovich metric corresponding to such c^k. Now assume, again without loss of generality, that $\varepsilon' \leq \varepsilon''$. Consider then $(c^k)'$ such that $(c^k)'(w_1, w_2) = c^k(w_1, w_2)$ for all $w_1, w_2 \in A^k$ except

1. $(c^k)'(w, w) = c^k(w, w) + \varepsilon'$,
2. $(c^k)'(w, w') = c^k(w, w') - \varepsilon'$,
3. $(c^k)'(w'', w') = c^k(w'', w') + \varepsilon'$,
4. $(c^k)'(w'', w) = c^k(w'', w) - \varepsilon'$.

The joint distribution $(c^k)'$ is feasible since it satisfies constraints (D.3) and (D.4). Now let $\mathsf{K}'_{\mathsf{C}}(\mu_1^k, \mu_2^k)$ denote the solution corresponding to such $(c^k)'$, it holds that

$$\begin{aligned}\mathsf{K}'_{\mathsf{C}}(\mu_1^k, \mu_2^k) &= \mathsf{K}_{\mathsf{C}}(\mu_1^k, \mu_2^k) \\ &\quad - \varepsilon'[\mathsf{C}(w, w') + \mathsf{C}(w', w'') - \mathsf{C}(w', w'')].\end{aligned}$$

By triangular inequality of C, we have that $\mathsf{K}'_{\mathsf{C}}(\mu_1^k, \mu_2^k) < \mathsf{K}_{\mathsf{C}}(\mu_1^k, \mu_2^k)$, which contradicts the fact that c^k is optimal, and the proof is completed. □

Lemma 6.5. *Let Γ_1 and Γ_2 be two LMCs defined over the same alphabet A. For any $k > 1$, let $c^k : A^k \times A^k \to [0, 1]$ be the optimal coupling when computing the Kantorovich distance $\mathsf{K}_{\mathsf{C}}(\mu_1^k, \mu_2^k)$ (see optimal transport problem (D.2)). Then, for all $w \in A^{k-1}$ such that $\mu_1^{k-1}(w) > \mu_2^{k-1}(w)$, it holds that*

$$\sum_{\substack{w' \in A^{k-1} \\ w' \neq w}} \sum_{a_1, a_2 \in A} c^k(wa_1, w'a_2) = \mu_1^{k-1}(w) - \mu_2^{k-1}(w),$$

$$\sum_{\substack{w' \in A^{k-1} \\ w' \neq w}} \sum_{a_1, a_2 \in A} c^k(w'a_1, wa_2) = 0.$$

Else if $\mu_1^{k-1}(w) \leq \mu_2^{k-1}(w)$, then

$$\sum_{\substack{w' \in A^{k-1} \\ w' \neq w}} \sum_{a_1, a_2 \in A} c^k(wa_1, w'a_2) = 0,$$

$$\sum_{\substack{w' \in A^{k-1} \\ w' \neq w}} \sum_{a_1, a_2 \in A} c^k(w'a_1, wa_2) = \mu_2^{k-1}(w) - \mu_1^{k-1}(w).$$

Proof. For some $w \in A^{k-1}$, assume without loss of generality that $\mu_1^{k-1}(w) > \mu_2^{k-1}(w)$. First, by the constraints (D.4), it holds that

$$\sum_{\substack{w' \in A^{k-1} \\ w' \neq w}} \sum_{a_1, a_2 \in A} c^k(wa_1, w'a_2) \geq \mu_1^{k-1}(w) - \mu_2^{k-1}(w).$$

Now, we proceed similarly as for Lemma 6.4. Suppose by contradiction that $\sum_{\substack{w' \in A^{k-1} \\ w' \neq w}} \sum_{a_1, a_2 \in A} c^k(wa_1, w'a_2) > \mu_1^{k-1}(w) - \mu_2^{k-1}(w)$. Then there exists $w' \neq w \in A^{k-1}$, and $a_1, a_2 \in A$ such that $c^k(w'a_1, wa_2) = \varepsilon' > 0$. There

also exists $w'' \in A^{k-1}$ such that $w'' \neq w$ and $w'' \neq w'$, and $a_3, a_4 \in A$ such that $c^k(wa_3, w''a_4) = \varepsilon'' > 0$. Assume w.l.o.g. that $\varepsilon' \leq \varepsilon''$, and consider a solution $(c^k)'$ such that $(c^k)' = c^k$, except for

1. $(c^k)'(wa_3, w''a_4) = c^k(wa_3, w''a_4) - \varepsilon'$,
2. $(c^k)'(w'a_1, wa_2) = c^k(w'a_1, wa_2) - \varepsilon'$,
3. $(c^k)'(w'a_1, w''a_4) = c^k(w'a_1, w''a_4) + \varepsilon'$, and
4. $(c^k)'(wa_3, wa_2) = c^k(wa_3, wa_2) + \varepsilon'$.

The joint distribution $(c^k)'$ is feasible since it still satisfies the constraints (D.3) and (D.4). Note that, since the Cantor distance is an ultradistance (see Proposition 1.10), then

$$\begin{aligned}\mathsf{C}(w'a_1, w''a_4) &\leq \max\{\mathsf{C}(w'a_1, wa_2), \mathsf{C}(wa_2, w''a_4)\} \\ &= \max\{\mathsf{C}(w'a_1, wa_2), \mathsf{C}(wa_3, w''a_4)\}.\end{aligned}$$

Moreover, $\mathsf{C}(wa_3, wa_2) = m^{-k}$. Now let $\mathsf{K}'_{\mathsf{C}}(\mu_1^k, \mu_2^k)$ denote the solution corresponding to such $(c^k)'$, we have that $\mathsf{K}_{\mathsf{C}}(\mu_1^k, \mu_2^k) - \mathsf{K}'_{\mathsf{C}}(\mu_1^k, \mu_2^k)$ is

$$\begin{aligned}\mathsf{K}_{\mathsf{C}}(\mu_1^k, \mu_2^k) - \mathsf{K}'_{\mathsf{C}}(\mu_1^k, \mu_2^k) &= -\varepsilon' \begin{bmatrix} + & \mathsf{C}(w'a_1, wa_2) \\ + & \mathsf{C}(wa_3, w''a_4) \\ - & \mathsf{C}(w'a_1, w''a_4) \\ - & m^{-k} \end{bmatrix} \\ &\leq -\varepsilon' \begin{bmatrix} + & \mathsf{C}(w'a_1, wa_2) + \mathsf{C}(wa_3, w''a_4) \\ - & \max\{\mathsf{C}(w'a_1, wa_2), \mathsf{C}(wa_3, w''a_4)\} \\ - & m^{-k} \end{bmatrix} \\ &\leq -\varepsilon'[m^{-(k-1)} - m^{-k}] \\ &\leq 0,\end{aligned}$$

which contradicts the fact that c^k is optimal. □

Proposition 6.6. *Let Γ_1 and Γ_2 be two LMCs defined over the same alphabet A. For any $i \in \mathbb{N}$, let*

$$M_i := \sum_{w \in A^i} \min\{\mu_1^i(w), \mu_2^i(w)\},$$

where μ_1^i and μ_2^i are the probability measures on $(A^i, 2^{A^i})$ generated by Γ_1 and

Γ_2, *respectively. For any* $k \in \mathbb{N}$, *it holds that*

$$\mathsf{K}_{\mathsf{C}}(\mu_1^k, \mu_2^k) = 1 - M_1 + \sum_{i=1}^{k-1} m^{-i}(M_i - M_{i+1}). \tag{6.3}$$

Proof. For the sake of conciseness, we note $\mathsf{K}_{\mathsf{C}}^k = \mathsf{K}_{\mathsf{C}}(\mu_1^k, \mu_2^k)$. We will prove that

$$\mathsf{K}_{\mathsf{C}}^{k+1} = \mathsf{K}_{\mathsf{C}}^k \quad + m^{-k} \sum_{w \in A^k} \left(r^k(w) - \sum_{a \in A} r^{k+1}(wa) \right), \tag{6.4}$$

where $r^k(w) = \min\{\mu_1^k(w), \mu_2^k(w)\}$, which is equivalent to (6.3). This proof is divided into two parts. First, we prove that the right-hand side of (6.4) is a lower bound for $\mathsf{K}_{\mathsf{C}}^{k+1}$. Second, we prove that it is also an upper bound.

Lower bound First we note that $\mathsf{K}_{\mathsf{C}}^{k+1}$ is equal to

$$\begin{aligned}
&\sum_{w_1, w_2 \in A^k} \sum_{a_1, a_2 \in A} \mathsf{C}(w_1 a_1, w_2 a_2) c^{k+1}(w_1 a_1, w_2 a_2) \\
&= \sum_{\substack{w_1, w_2 \in A^k \\ w_1 \neq w_2}} \mathsf{C}(w_1, w_2) \sum_{a_1, a_2 \in A} c^{k+1}(w_1 a_1, w_2 a_2) \\
&\quad + m^{-(k+1)} \sum_{w \in A^k} \sum_{\substack{a_1, a_2 \in A \\ a_1 \neq a_2}} c^{k+1}(wa_1, wa_2) \\
&\quad := C_1 + C_2.
\end{aligned}$$

In the first part of this proof, we show that the two following expressions hold:

$$C_1 \geq \mathsf{K}_{\mathsf{C}}^k, \tag{6.5}$$

$$C_2 = m^{-k} \sum_{w \in A^k} \left(r^k(w) - \sum_{a \in A} r^{k+1}(wa) \right). \tag{6.6}$$

In the first instance, we show that (6.5) holds. To do this, let $d^k : A^k \times A^k \to [0, 1]$ be defined as $d^k(w_1, w_2) = \sum_{a_1, a_2 \in A} c^{k+1}(w_1 a_1, w_2 a_2)$. We show that d^k satisfies the constraints (D.3) and (D.4). Indeed, $d^k(w_1, w_2) \geq 0$, and

$$\begin{aligned}
\sum_{w_2 \in A^k} d^k(w_1, w_2) &= \sum_{w_2 \in A^k} \sum_{a_1, a_2 \in A} c^{k+1}(w_1 a_1, w_2 a_2) \\
&= \sum_{a_1} \mu_1^{k+1}(w_1 a_1) = \mu_1^k(w_1),
\end{aligned}$$

and similarly for the second condition in (D.4). This implies that d^k is a coupling, thereby a feasible solution. This yields

$$\mathsf{K}_{\mathsf{C}}^k \leq \sum_{a_1,a_2 \in A} \mathsf{C}(w_1, w_2) d^k(w_1, w_2) = C_1.$$

As a second step, we show that (6.6) holds. More precisely, we show that, for all $w \in A^k$, the following holds:

$$\sum_{\substack{a_1,a_2 \in A \\ a_1 \neq a_2}} c^{k+1}(wa_1, wa_2) = r^k(w) - \sum_{a \in A} r^{k+1}(wa),$$

which implies that

$$C_2 = m^{-(k+1)} \sum_{w \in A^k} \left[r^k(w) - \sum_{a \in A} r^{k+1}(wa) \right].$$

We prove the claim. Assume without loss of generality that w is such that $\mu_1^k(w) > \mu_2^k(w)$, then

$$\begin{aligned}
&\sum_{a_1,a_2 \in A} c^{k+1}(wa_1, wa_2) \\
&= \sum_{a_1 \in A} \sum_{w' \in A^k} \sum_{a_2 \in A} c^{k+1}(wa_1, w'a_2) \\
&\qquad - \sum_{\substack{w' \in A^k \\ w' \neq w}} \sum_{a_2 \in A} c^{k+1}(wa_1, w'a_2) \\
&= \sum_{a_1 \in A} \mu_1^k(wa_1) - \sum_{\substack{w' \in A^k \\ w' \neq w}} \sum_{a_1,a_2 \in A} c^{k+1}(wa_1, w'a_2)
\end{aligned}$$

Following Lemma 6.5, this is equal to $\mu_1^k(w) - \left(\mu_1^k(w) - \mu_2^k(w)\right) = r^k(w)$. And the following holds:

$$\begin{aligned}
&\sum_{\substack{a_1,a_2\in A\\ a_1\neq a_2}} c^{k+1}(wa_1, wa_2)\\
&= \sum_{\substack{a_1\neq a_2\in A\\ a_1\neq a_2}} c^{k+1}(wa_1, wa_2) - \sum_{a\in A} c^{k+1}(wa_1, wa_2)\\
&= r^k(w) - \sum_{a\in A} r^{k+1}(wa)
\end{aligned}$$

by Lemma 6.4. This concludes that the right-hand side of (6.4) is a lower bound for $\mathsf{K}_{\mathsf{C}}^{k+1}$.

Upper bound We now move to the second part of this proof. To provide an upper bound, we will show that we can construct a feasible $k+1$ solution feasible d^{k+1} such that

$$\begin{aligned}
&\sum_{w_1,w_2\in A^{k+1}} \mathsf{C}(w_1, w_2) d^{k+1}(w_1, w_2)\\
&\qquad = \mathsf{K}_{\mathsf{C}}^k + \sum_{w\in A^k} \left[r^k(w) - \sum_{a\in A} r^{k+1}(wa) \right].
\end{aligned}$$

Consider c^k, an optimal solution at step k. We will construct d^{k+1} in the following greedy way. Initialise d^{k+1} with only zero elements, and for all $w \in A^k$, $a \in A$, we initialise $\delta(wa) = 0$. We start by updating the blocks $d^{k+1}(w_1a_1, w_2a_2)$ where $w_1 \neq w_2$. For all w such that $\mu_1^k(w) > \mu_2^k(w)$, for all $a \in A$ such that $\mu_1^{k+1}(wa) > \mu_2^{k+1}(wa)$, do the following.

1. Let $\tilde{\delta}(wa) = \mu_1^{k+1}(wa) - \mu_2^{k+1}(wa)$.
   If $\sum_{a'\neq a} \delta(wa') + \tilde{\delta}(wa) > \mu_1^k(w) - \mu_2^k(w)$, let $\delta(wa) = (\mu_1^k(w) - \mu_2^k(w)) - \sum_{a'\neq a} \delta(wa')$.
   Else let $\delta(wa) = \tilde{\delta}(wa)$.

2. Find a $w' \neq w$ such that

$$c^k(w, w') > \sum_{a_1,a_2\in A} d^{k+1}(wa_1, w'a_2).$$

Now, for any $a' \in A$, let

$$\psi(a') = \mu_2^{k+1}(w'a') - \mu_1^{k+1}(w'a') - \sum_{\substack{w'' \in A^k \\ w'' \neq w'}} \sum_{a_1 \in A} d^{k+1}(w''a_1, w'a')$$

Then, find $a' \in A$ such that $\psi(a') > 0$.
Now, if $\delta(wa) > \psi(a')$, then:

- Update $d^{k+1}(wa, w'a') \leftarrow \psi(a')$
- Update $\delta(wa) \leftarrow \delta(wa) - \psi(a')$
- Return to 2.

Else, update $d^{k+1}(wa, w'a') \leftarrow \delta(wa)$.

We claim that, in the procedure above, there always exists such a w' for a given wa. Otherwise, $\sum_{w' \neq w} c^k(w, w') < \mu_1^k(w) - \mu_2^k(w)$, which is impossible by Lemma 6.4. Also, we claim that there also always exists such a' for a given wa and w'. Otherwise, for all $a' \in A$,

$$\sum_{a' \in A} \sum_{\substack{w'' \in A^k \\ w'' \neq w'}} \sum_{a_1 \in A} d^{k+1}(w''a_1, w'a') = \sum_{a' \in A} \mu_2^{k+1}(w'a') - \mu_1^{k+1}(w'a'),$$

which means by construction that

$$\sum_{a' \in A} \sum_{\substack{w'' \in A^k \\ w'' \neq w'}} \sum_{a_1 \in A} c^{k+1}(w''a_1, w'a') > \sum_{a' \in A} \mu_2^{k+1}(w'a') - \mu_1^{k+1}(w'a'),$$

which is $\mu_2^k(w) - \mu_1^k(w) > \mu_2^k(w) - \mu_1^k(w)$ by Lemma 6.5. Moreover, by construction we have that, for all $w \neq w'$,

$$c^k(w, w') = \sum_{a_1, a_2 \in A^k} d^{k+1}(wa_1, w'a_2). \tag{6.7}$$

Now, we construct the diagonal blocks $d^{k+1}(wa_1, wa_2)$. For each $w \in A^k$

and $a \in A$, let

$$\begin{aligned}\tilde{\mu}_1^{k+1}(wa) &= \mu_1^{k+1}(wa) - \sum_{w' \neq w} \sum_{a \in A} d^{k+1}(wa, w'a'),\\ \tilde{\mu}_2^{k+1}(wa) &= \mu_2^{k+1}(wa) - \sum_{w' \neq w} \sum_{a \in A} d^{k+1}(w'a', wa).\end{aligned} \tag{6.8}$$

Now, for a given w, let us solve the following balanced optimal transport problem:

$$\begin{aligned}&\inf_{d^{k+1}} m^{-k} \sum_{\substack{a_1, a_2 \in A \\ a_1 \neq a_2}} d^{k+1}(wa_1, wa_2)\\ &\text{s.t. } \forall a_1 \in A : \sum_{a_2} d^{k+1}(wa_1, wa_2) = \tilde{\mu}_1^{k+1}(wa_1),\\ &\qquad \forall a_2 \in A : \sum_{a_1} d^{k+1}(wa_1, wa_2) = \tilde{\mu}_2^{k+1}(wa_2).\end{aligned}$$

Following the definition of $\tilde{\mu}_1$ and $\tilde{\mu}_2$, this is a balanced optimal transport whose trivial solution is given by

$$m^{-k}\left(r^k(w) - \sum_{a \in A} r^{k+1}(wa)\right). \tag{6.9}$$

Now we conclude the proof. By (6.7) and (6.8), d^{k+1} is a coupling of μ_1^{k+1} and μ_2^{k+1}. Indeed it is positive, and for any $w_1 \in A^k$ and $a_1 \in A$,

$$\begin{aligned}&\sum_{w_2 \in A^k} \sum_{a_2 \in A} d^{k+1}(w_1a_1, w_2a_2)\\ &= \sum_{a_2 \in A} d^{k+1}(w_1a_1, w_1a_2) + \sum_{\substack{w_2 \in A^k \\ w_2 \neq w_1}} \sum_{a_2 \in A} d^{k+1}(w_1a_1, w_2a_2)\\ &= \tilde{\mu}_1^{k+1}(w_1a_1) + \left(\mu_1^{k+1}(w_1a_1) - \tilde{\mu}_1^{k+1}(w_1a_1)\right) = \mu_1^{k+1}(w_1a_1),\end{aligned}$$

and similarly for μ_2^{k+1}. Finally,

$$\begin{aligned}
&\sum_{w_1,w_2\in A^k}\sum_{a_1,a_2\in A}\mathsf{C}(w_1a_1,w_2a_2)d^{k+1}(w_1a_1,w_2a_2)\\
&=\sum_{w_1\neq w_2}\mathsf{C}(w_1,w_2)\sum_{a_1,a_2}d^{k+1}(w_1a_1,w_2a_2)\\
&+m^{-k}\sum_{w}\sum_{\substack{a_1,a_2\\ a_1\neq a_2}}d^{k+1}(wa_1,wa_2).
\end{aligned}$$

By (6.7), the first term is $\mathsf{K}_{\mathsf{C}}^k$, and by (6.9), the second term is

$$\sum_{w\in A^k}\left[r^k(w)-\sum_{a\in A}r^{k+1}(wa)\right].$$

This provides an upper bound on $\mathsf{K}_{\mathsf{C}}^{k+1}$, and the proof is completed. □

Proposition 6.7 ([CK14]). *Let Γ_1 and Γ_2 be two LMCs defined over the same alphabet A. For any $k\in\mathbb{N}$, it holds that*

$$\mathsf{TV}(\mu_1^k,\mu_2^k)=1-M_k,$$

where μ_1^k and μ_2^k are the probability measures on $(A^k,2^{A^k})$ generated by Γ_1 and Γ_2, respectively, and where M_k is defined such as in (6.3).

Proposition 6.6 shows that, for any k, one can compute $\mathsf{K}_{\mathsf{C}}(\mu_1^k,\mu_2^k)$ by computing the minimum between $\mu_1^i(w)$ and $\mu_2^i(w)$ for all $i\leq k$. It turns out that this minimum quantity is also sufficient to compute the TV distance between $\mu_1^k(w)$ and $\mu_2^k(w)$, such as highlighted in [CK14]. The combination of these two observations yields the following result.

Theorem 6.8. *Let Γ_1 and Γ_2 be two LMCs defined over the same alphabet A. It holds that*

$$\mathsf{CK}(\Gamma_1,\Gamma_2)=(m-1)\sum_{i=1}^{\infty}m^{-i}\mathsf{TV}(\mu_1^i,\mu_2^i),$$

with $m:=|A|$, and where μ_1^i and μ_2^i are the probability measures on $(A^i,2^{A^i})$ generated by Γ_1 and Γ_2, respectively.

Proof. First, from Proposition 6.6, it holds that

$$\mathsf{K}_{\mathsf{C}}(\mu_1^k, \mu_2^k) = 1 + \sum_{i=1}^{k-1} \left(\frac{1-m}{m^i} \right) M_i - m^{-(k-1)} M_k.$$

Now, replacing $M_i = 1 - \mathsf{TV}(\mu_1^i, \mu_2^i)$ from Proposition 6.7 yields

$$\begin{aligned}\mathsf{K}_{\mathsf{C}}(\mu_1^k, \mu_2^k) = 1 + (1-m) \sum_{k-1} m^{-i} - m^{-(k-1)} \\ + \sum_{i=1}^{k-1} \left(\frac{m-1}{m^i} \right) \mathsf{TV}(\mu_1^i, \mu_2^i) + m^{1-k} \mathsf{TV}(\mu_1^k, \mu_2^k).\end{aligned}$$

It remains to see that the first three terms of the equation above cancel, which implies

$$\mathsf{K}_{\mathsf{C}}(\mu_1^k, \mu_2^k) = \sum_{i=1}^{k-1} \left(\frac{m-1}{m^i} \right) \mathsf{TV}(\mu_1^i, \mu_2^i) + m^{1-k} \mathsf{TV}(\mu_1^k, \mu_2^k).$$

Since $\mathsf{TV}(\mu_1^k, \mu_2^k)$ converges to a finite value and that $\lim_{k\to\infty} m^{1-k} = 0$, taking the equation to the limit $k \to \infty$ yields the statement, and the proof is concluded. □

Remark 6.9. Theorem 6.8 is central in this paper as it provides both theoretical and practical insights about the CK distance. In particular, it shows that, up to a factor $(m-1)$, the CK distance can be seen as an instance of the discounted TV distances studied in [JMGLM14], but one that arises directly from the natural topology generated by the underlying Cantor metric.

Theorem 6.8 will also be crucial in the development of the rest of this chapter. For instance in Corollary 6.10 we show that the CK distance is well-defined, in the sense that the limit in (6.1) converges, and that the resulting distance is a trace-based distance, i.e.

$$\mathsf{CK}(\Gamma_1, \Gamma_2) = 0 \iff \mu_1^\omega \equiv \mu_2^\omega.$$

We refer the reader to Definition 1.27 for a formal definition of trace-based distance.

Corollary 6.10. *For any two LMCs Γ_1 and Γ_2 defined over the same alphabet A, the CK distance converges, is a trace-based distance between LMCs as defined in Definition 1.27, and is such that $0 \leq \mathsf{CK}(\Gamma_1, \Gamma_2) \leq 1$.*

Proof. We first prove that the distance converges, and is contained in $[0, 1]$. By Theorem 6.8, it suffices to show that the sequence of partial sums $(S_k)_{k\geq 0}$ converges, where

$$S_k := \sum_{i=1}^{k} \left(\frac{m-1}{m^i}\right) \mathsf{TV}(\mu_1^i, \mu_2^i).$$

Let

$$\overline{S}_k := \sum_{i=1}^{k} \left(\frac{m-1}{m^i}\right) = 1 - \frac{1}{m^k}.$$

Since $0 \leq \mathsf{TV}(\mu^i, \mu^i) \leq 1$ for all i, it holds that $0 \leq S_k \leq \overline{S}_k$ for all k. By the comparison test, $(S_k)_{k\geq 0}$ converges to a value that is lower bounded by 0 and upper bounded by $\lim_{k\to\infty} \overline{S}_k = 1$.

We now prove that it is a trace-based distance. First, by Theorem 6.8, CK is a pseudo-distance since it is a discounted sum of distances. We therefore conclude by showing that $\mathsf{CK}(\Gamma_1, \Gamma_2) = 0$ is a necessary and sufficient condition for $\mu_1^\omega \equiv \mu_2^\omega$. The latter follows from the fact that, by Theorem 6.8, $\mathsf{CK}(\Gamma_1, \Gamma_2) = 0$ if and only if $\mathsf{TV}(\mu_1^k, \mu_2^k) = 0$ for all $k \in \mathbb{N}$ which, by definition of μ_1^ω and μ_2^ω, is true if and only if $\mu_1^\omega \equiv \mu_2^\omega$. □

We conclude this section by providing an interpretation of the REFINE algorithm (from Chapter 4) used with the CK distance.

Remark 6.11 (Interpretation of Algorithm 3 with the CK distance)**.** The CK distance defines a metric space in which two LMCs are close if they have similar short-horizon behaviours – this directly follows Theorem 6.8. The procedure REFINE with the CK distance therefore tends to choose the model with a different short-term behaviour.

6.2 Continuity properties

In this section we study the continuity properties of the CK distance, described in Chapter 1. We recall these definitions hereinafter.

Definition 6.12 (Parameter continuity)**.** A pseudo-distance CK between LMCs is said to be *parameter* continuous if, for all $\varepsilon > 0$, there exists $\delta > 0$ such that, for any two LMCs Γ_1 and Γ_2,

$$\Gamma_1 \text{ and } \Gamma_2 \text{ are } \delta\text{-approximately bisimilar} \implies \mathsf{CK}(\Gamma_1, \Gamma_2) \leq \varepsilon,$$

where δ-approximate bisimilarity is defined in Definition 1.23.

Definition 6.13 (Φ-continuity)**.** Let $\phi \subseteq A^\omega$ be a property. A pseudo-distance CK between LMCs is said to be *ϕ-continuous* if, for all $\varepsilon > 0$, there exists $\delta > 0$ such that, for any two LMCs Γ_1 and Γ_2,

$$\mathsf{CK}(\Gamma_1, \Gamma_2) \leq \delta \implies \sup_{w \in \phi} |\mu_1^\omega(w) - \mu_2^\omega(w)| \leq \varepsilon.$$

Let Φ be a class of properties, a distance CK is said to be Φ-*continuous* if it is ϕ-continuous for all $\phi \in \Phi$.

In this section we will show that our distance is (B-LTL)-continuous, where B-LTL is the LTL class of properties, however omitting the "until" $\mathcal{U}$ temporal modality (see Chapter 1 for a formal definition).

Dovetailing on Remark 6.9, Jaeger et al. already showed in [JMGLM14] that this distance is both parameter and (B-LTL)-continuous. In this section, we extend the results in [JMGLM14] in two separate ways:

1. We quantify in Proposition 6.15 the value of ε such that

$$\Gamma_1 \text{ and } \Gamma_2 \text{ are } \delta\text{-approximately bisimilar} \implies \mathsf{CK}(\Gamma_1, \Gamma_2) \leq \varepsilon.$$

2. We quantify in Proposition 6.17 the value of ε such that

$$\mathsf{CK}(\Gamma_1, \Gamma_2) \leq \delta \implies |\mu_1^k(w) - \mu_2^k(w)| \leq \varepsilon,$$

for all finite words $w \in A^k$.

Hence, we recover the less general continuity results shown in [JMGLM14], as corollaries of these two propositions in Corollary 6.16 and Corollary 6.18.

We start by showing that, if two LMCs are δ-approximately bisimilar, then the CK distance is upper bounded by a function of δ. To prove this result, we rely on a result proven by Bian and Abate in [BA17, Theorem 4].

Proposition 6.14 ([BA17])**.** *Given $\delta \in (0, 1)$, let Γ_1 and Γ_2 be two δ-approximately bisimilar LMCs defined on the same set of labels. For all $k \geq 0$, it holds that*

$$\mathsf{TV}(\mu_1^k, \mu_2^k) \leq 1 - (1 - \delta)^k, \tag{6.10}$$

where μ_1^k and μ_2^k are the distributions on A^k respectively generated by Γ_1 and Γ_2.

Proposition 6.15. *Given $\delta \in (0,1)$, let Γ_1 and Γ_2 be two δ-approximately bisimilar LMCs defined over the same alphabet. It holds that*

$$\mathsf{CK}(\Gamma_1, \Gamma_2) \leq \frac{m\delta}{m-1+\delta}. \tag{6.11}$$

Proof. By Theorem 6.8 and applying the bound (6.10), it holds that

$$\begin{aligned}
\mathsf{CK}(\Gamma_1, \Gamma_2) &= \sum_{i=1}^{\infty} \left(\frac{m-1}{m^i}\right) \mathsf{TV}(\mu_1^i, \mu_2^i) \\
&\leq \sum_{i=1}^{\infty} \left(\frac{m-1}{m^i}\right) (1-(1-\delta)^i) \\
&= (m-1) \sum_{i=1}^{\infty} \left(\frac{1}{m}\right)^i - (m-1) \sum_{i=1}^{\infty} \left(\frac{1-\delta}{m}\right)^i \\
&= 1 - \frac{(m-1)(1-\delta)}{m-1+\delta} = \frac{m\delta}{m-1+\delta},
\end{aligned}$$

which proves the claim. □

An illustration of the bound (6.11) can be found in Figure 6.4 for the cases $m = 2, \ldots, 10$. A second illustration is also presented in Section 6.3 (see Figure 6.5).

Proposition 6.15 provides practical insight about the CK distance. Suppose that a lower bound $\mathsf{CK}(\Gamma_1, \Gamma_2) \geq \underline{\mathsf{CK}}$ is known. Then, with

$$\overline{\delta} := (m-1)\underline{\mathsf{CK}}/(m-\underline{\mathsf{CK}}),$$

it is impossible that Γ_1 and Γ_2 are δ-approximately bisimilar for all $\delta \leq \overline{\delta}$. Proposition 6.15 also provides theoretical insight, as it offers a proof for a less restrictive notion of parameter continuity (see Remark 1.31).

Corollary 6.16. *The CK distance is parameter continuous.*

Proof. For all $\varepsilon \in (0,1)$, let $\delta(\varepsilon) = (m-1)\varepsilon/(m-\varepsilon)$. By Proposition 6.15, for all $\varepsilon \in (0,1)$, if Γ_1 and Γ_2 are $\delta(\varepsilon)$-approximately bisimilar to each other, then $\mathsf{CK}(\Gamma_1, \Gamma_2) \leq \varepsilon$, and the proof is completed. □

We now show that, if $\mathsf{CK}(\Gamma_1, \Gamma_2) \leq \delta$, then the TV distance $\mathsf{TV}(\mu_1^k, \mu_2^k)$ is upper bounded by a function of δ and the horizon k.

Proposition 6.17. *Let Γ_1 and Γ_2 be two LMCs defined over the same alphabet. For all $k \geq 1$, it holds that, if $\mathsf{CK}(\Gamma_1, \Gamma_2) \leq \delta$, then $\mathsf{TV}(\mu_1^k, \mu_2^k) \leq m^{k-1}\delta$,*

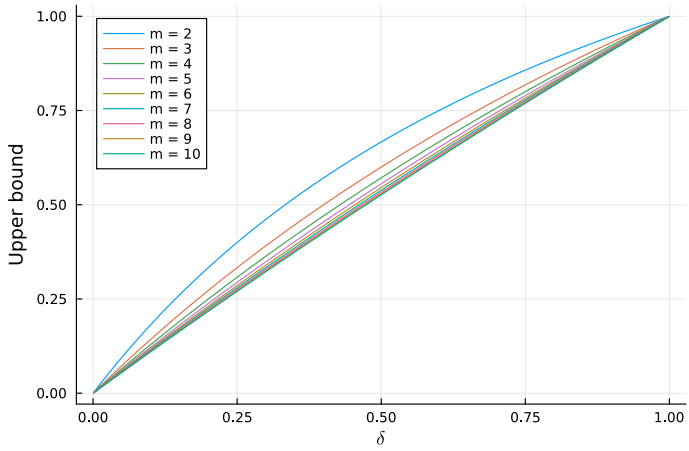


Fig. 6.4 Upper bound (6.11) on the CK distance between two δ-approximately bisimilar LMCs. The upper bound is given as a function $\delta \in (0,1)$, and for different numbers of labels.

with μ_1^k and μ_2^k the probability measures on $(A^k, 2^{A^k})$ generated by Γ_1 and Γ_2, respectively.

Proof. Since $\mathsf{TV}(\mu_1^k, \mu_2^k)$ is increasing with k, by Theorem 6.8, it holds that, for all $k \geq 1$,

$$\begin{aligned}
\mathsf{CK}(\Gamma_1, \Gamma_2) &= \sum_{i=1}^{\infty} \left(\frac{m-1}{m^i} \right) \mathsf{TV}(\mu_1^i, \mu_2^i) \\
&\geq \sum_{i=k}^{\infty} \left(\frac{m-1}{m^i} \right) \mathsf{TV}(\mu_1^i, \mu_2^i) \\
&\geq \mathsf{TV}(\mu_1^k, \mu_2^k) \sum_{i=k}^{\infty} \left(\frac{m-1}{m^i} \right) \\
&= m^{1-k} \mathsf{TV}(\mu_1^k, \mu_2^k).
\end{aligned}$$

The inequality above implies that $m^{1-k}\mathsf{TV}(\mu_1^k, \mu_2^k) \leq \mathsf{CK}(\Gamma_1, \Gamma_2) \leq \delta$, which concludes the proof. □

Again, Proposition 6.17 provides a practical insight about the CK distance. Suppose that an upper bound $\mathsf{CK}(\Gamma_1, \Gamma_2) \leq \overline{\mathsf{CK}}$ is known. Given a

threshold $\varepsilon \in (0,1)$, for all

$$k \leq 1 + \left\lfloor \frac{\log(\varepsilon/\overline{\mathsf{CK}})}{\log(m)} \right\rfloor,$$

then $\mathsf{TV}(\mu_1^k, \mu_2^k) \leq \varepsilon$. Therefore, given a threshold ε and an upper bound $\mathsf{CK}(\Gamma_1, \Gamma_2) \leq \overline{\mathsf{CK}}$, the result above gives a maximal horizon k such that $|\mu_1^k(w) - \mu_2^k(w)| \leq \varepsilon$ for all k-long word $w \in A^k$.

Proposition 6.17 also provides theoretical insights as it provides an alternative proof of (B-LTL)-continuity.

Corollary 6.18. *The CK distance is (B-LTL)-continuous.*

Proof. Let $\phi \subseteq A^\omega$ be any (B-LTL)-definable property. Following Proposition 1.16, there exists a finite number of finite prefixes $W \subseteq A^\star$ such that $\phi = \bigcup_{w \in W} wA^\omega$. This implies that

$$\sup_{w \in \phi} |\mu_1^\omega(w) - \mu_2^\omega(w)| \leq \max_{w \in W} |\mu_1^{k(w)}(w) - \mu_2^{k(w)}(w)|, \tag{6.12}$$

where $k(w) = |w|$. Let

$$w_* := \arg\max_{w \in W} |\mu_1^{k(w)}(w) - \mu_2^{k(w)}(w)|,$$

and $k_* := k(w_*)$. Since W is a finite set of words, such maximum exists. Now, for all $\varepsilon \in (0,1)$, let $\delta(\varepsilon) = (1 - m^{-k_*})\varepsilon$. By Proposition 6.17, if $\mathsf{CK}(\Gamma_1, \Gamma_2) \leq \delta(\varepsilon)$, then $\mathsf{TV}(\mu_1^{k_*}, \mu_2^{k_*}) \leq \varepsilon$. By definition of the TV distance, the latter is an upper bound on (6.12), and the proof is completed. □

In this chapter, we studied continuity with respect to B-LTL properties. It could be interesting as further work to investigate continuity with respect to discounted LTL formulae, where the "eventually" operators of LTL are relaxed into discounting operators, so that the satisfaction value – in $[0,1]$ – decreases the longer it takes to fulfill eventuality requirements (see [ABK14] for an introduction).

6.3 Approximation

In Section 6.1, we showed that the CK distance is an infinite discounted sum of TV distances. In this section, we investigate the approximation

scheme in which we *truncate* this sum. More formally, for any two LMCs Γ_1 and Γ_2 defined over the same alphabet, and for any $k \in \mathbb{N}$, let

$$S_k(\Gamma_1, \Gamma_2) := (m-1) \sum_{i=1}^{k} m^{-i} \mathsf{TV}(\mu_1^i, \mu_2^i), \tag{6.13}$$

where μ_1^i and μ_2^i are the probability measures on $(A^i, 2^{A^i})$ generated by Γ_1 and Γ_2, respectively. We show in Proposition 6.19 that computing $S_k(\Gamma_1, \Gamma_2)$ allows to approximate $\mathsf{CK}(\Gamma_1, \Gamma_2)$ up to an arbitrarily small precision.

Proposition 6.19. *Let Γ_1 and Γ_2 be two LMCs defined over the same alphabet. For all $k \in \mathbb{N}$, it holds that*

$$0 \leq \mathsf{CK}(\Gamma_1, \Gamma_2) - S_k(\Gamma_1, \Gamma_2) \leq m^{-k},$$

where S_k is as defined in (6.13).

Proof. The fact that the difference is non-negative follows trivially from Theorem 6.8 and the fact that $\mathsf{TV}(\mu, \nu) \geq 0$ for any two distributions μ, ν. We now prove the upper bound. By Theorem 6.8, and since $\mathsf{TV}(\mu, \nu) \leq 1$ for any μ, ν, it holds that

$$\begin{aligned}
&\mathsf{CK}(\Gamma_1, \Gamma_2) - \sum_{i=1}^{k} \left(\frac{m-1}{m^i} \right) \mathsf{TV}(\mu_1^i, \mu_2^i) \\
&\quad = \sum_{i=1}^{\infty} \left(\frac{m-1}{m^i} \right) \mathsf{TV}(\mu_1^i, \mu_2^i) - \sum_{i=1}^{k} \left(\frac{m-1}{m^i} \right) \mathsf{TV}(\mu_1^i, \mu_2^i) \\
&\quad = \sum_{i=k+1}^{\infty} \left(\frac{m-1}{m^i} \right) \mathsf{TV}(\mu_1^i, \mu_2^i) \\
&\quad \leq (m-1) \sum_{i=k+1}^{\infty} m^{-i} \\
&\quad = (m-1) \frac{m^{-(k+1)}}{1 - m^{-1}} \\
&\quad = m^{-k},
\end{aligned}$$

which concludes the proof. □

Proposition 6.19 implies that, in order to reach an error of at most $\varepsilon \in$

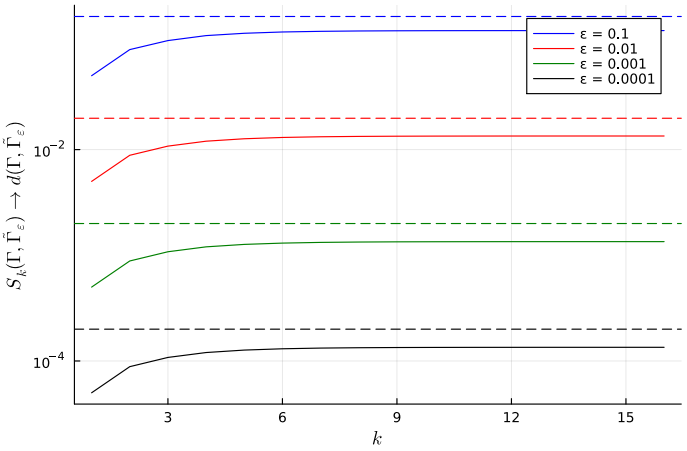


Fig. 6.5 Approximation of the CK distance $\mathsf{CK}(\Gamma, \tilde{\Gamma}_\varepsilon)$ for $\varepsilon \in \{10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}\}$, and where Γ and $\tilde{\Gamma}_\varepsilon$ are the LMCs from Example 6.1. The plain line is $S_k(\Gamma, \tilde{\Gamma}_\varepsilon)$, and the dashed line is the upper bound (6.11). As expected from (6.13), $S_k(\Gamma, \tilde{\Gamma}_\varepsilon)$ converges.

$(0, 1)$, it suffices to choose

$$k \geq \left\lceil \frac{\log(\varepsilon^{-1})}{\log(m)} \right\rceil .$$

Example 6.20. Let Γ be the unbiased Markov chain from Example 6.1, and consider biased Markov chains $\tilde{\Gamma}_\varepsilon$ with $\varepsilon \in \{10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}\}$. In Figure 6.5, we approximated $\mathsf{CK}(\Gamma, \tilde{\Gamma}_\varepsilon)$ with $S_k(\Gamma_1, \Gamma_2)$ for $k = 1, \ldots, 15$, thereby achieving a precision of $m^{-k} = 2^{-15} \approx 3 \cdot 10^{-5}$.

6.4 Computational complexity

In the following, we will show that computing the CK distance is #P-hard. This result relies on computational complexity concepts including *binary encodings* $e : I \to \{0,1\}^\star$, *counting problems* $P : \{0,1\}^\star \to \mathbb{N}$, the *#P complexity class*, *hardness*, and *polynomial reduction*. For details, see Appendix E.

To prove hardness, we reduce a known #P-complete function to the computation of the CK distance. This reduction is based on the notion of *product distribution*, defined hereinafter.

Definition 6.21 (Product distribution)**.** For any $k \in \mathbb{N}$, a *product probability*

measure π over $(\{0,1\}^k, 2^{\{0,1\}^k})$ is described by k parameters $p_i \in [0,1]$ for $i = 1, \ldots, k$, and is defined as

$$\pi(w) = \prod_{i=1}^{k} p_i^{a_i}(1-p_i)^{1-a_i}$$

for all $w := (a_1, \ldots, a_k) \in \{0,1\}^k$.

The problem that we will reduce is taken from [BGM+23] and consists in computing $\mathsf{TV}(\pi_1, \pi_2)$ between any two product distributions. More precisely, and in the framework of Appendix E, the concrete counting problem $f_{\mathsf{TV,prod}} : \{0,1\}^\star \to \mathbb{N}$ is defined as follows.

Definition 6.22. The function $f_{\mathsf{TV,prod}} : \{0,1\}^\star \to \mathbb{N}$ is the counting function that computes the TV distance between the binary encoding of any two product distributions. For any two product distributions π_1 and π_2 defined by $2k$ rational parameters $p_i, q_i \in [0,1] \cap \mathbb{Q}$, it is defined as

$$f_{\mathsf{TV,prod}}(e(\pi_1), e(\pi_2)) = 2M\mathsf{TV}(\pi_1, \pi_2),$$

where e is the binary encoding (as defined in Appendix E), and M is the product of the denominators of $p_1, \ldots, p_k, q_1, \ldots, q_k$ and the denominators of $1-p_1, \ldots, 1-p_k, 1-q_1, \ldots, 1-q_k$.

Note that, in order to frame the computation of the TV distance to a counting problem, the problem that we reduce is not exactly the one of computing $\mathsf{TV}(\pi_1, \pi_2)$ but is equivalent.

Proposition 6.23 ([BGM+23])**.** *The function* $f_{\mathsf{TV},prod} : \{0,1\}^\star \to \mathbb{N}$ *such as defined in Definition 6.22 is #P-complete.*

Definition 6.24. Let $f_{\mathsf{CK}} : \{0,1\}^\star \to \mathbb{R}$ be the function that computes the CK distance between two LMCs. For any two LMCs Γ_1 and Γ_2 with rational parameters, it is defined as

$$f_{\mathsf{CK}}(e(\Gamma_1), e(\Gamma_2)) := \mathsf{CK}(\Gamma_1, \Gamma_2),$$

where e denotes the binary encoding of the parameters of the LMCs (as defined in Appendix E).

Remark 6.25. It is an open question whether there exists an algorithm to compute this function exactly. However, we proved in Proposition 6.19

that the CK distance can be computed up to an arbitrarily small precision with the truncated sum $S_k(\Gamma_1, \Gamma_2)$, for which an algorithm exists[2], making the CK distance *computable up to arbitrary precision.*

We now prove that the CK distance is #P-hard. The proof relies on the reduction of $f_{\mathsf{TV},\mathrm{prod}}$ to the computation of f_{CK} (see Appendix E for a formal definition).

Theorem 6.26. *The function $f_{\mathsf{CK}} : \{0,1\}^\star \to \mathbb{R}$ such as defined in Definition 6.24 is #P-hard.*

Proof. We will prove that there exists a polynomial-time function $r : \{0,1\}^\star \to \{0,1\}^\star$ such that $f_{\mathsf{TV},\mathrm{prod}}/2M = f_{\mathsf{CK}}(r(x), r(y))$ for x, y the binary encodings of two product measures, where M is as defined in Definition 6.22.

First, given a product measure π^k described by k parameters $p_1, \dots, p_k$, we introduce a LMC Γ_{π^k} that *encodes* π^k. Let $\Gamma_{\pi^k} = (S, \{0,1\}, \iota, \tau, l)$ be such that

- S is a set of $2k$ states $\mathbf{0}_1, \dots, \mathbf{0}_k$ and $\mathbf{1}_1, \dots, \mathbf{1}_k$,
- ι is such that $\iota(\mathbf{1}_1) = p_1$, $\iota(\mathbf{0}_1) = 1 - p_1$, and $\iota(\mathbf{1}_i) = \iota(\mathbf{0}_i) = 0$ for all $i = 2, \dots, k$,
- τ is such that, for all $i = 2, \dots, k$, $\tau(\mathbf{1}_i|\mathbf{1}_{i-1}) = p_i$, $\tau(\mathbf{0}_i|\mathbf{1}_{i-1}) = 1 - p_i$, $\tau(\mathbf{1}_i|\mathbf{0}_{i-1}) = p_i$, $\tau(\mathbf{0}_i|\mathbf{0}_{i-1}) = 1 - p_i$, and all other entries are null. Also, we require that $\tau(\mathbf{0}_k|\mathbf{0}_k) = \tau(\mathbf{0}_k|\mathbf{1}_k) = 1$.
- l is such that $l(\mathbf{1}_i) = 1$ and $l(\mathbf{0}_i) = 0$ for all $i = 1, \dots, k$.

An illustration of Γ_π is provided in Figure 6.6 for a product measure π. The LMC Γ_{π^k} is as follows:

- all distributions generated by the LMC Γ_{π^k} up to horizon $i \leq k$, noted $\mu^i_{\pi^k}$, are the product distributions described by the first i parameters $p_1, \dots, p_i$, that is $\mu^i_{\pi^k} \equiv \pi^i$,
- and all distributions generated by the LMC Γ_{π^k} up to horizon $i > k$, are such that, for $w = (a_1, \dots, a_k, a_{k+1}, \dots, a_i) \in A^i$,

$$\mu^i_{\pi^k}(w) = \begin{cases} \pi^k(a_1, \dots, a_k) & \text{if } (a_{k+1}, \dots, a_i) = (0, \dots, 0) \\ 0 & \text{otherwise.} \end{cases}$$

[2]Indeed, for a finite horizon k, computing the TV distance is straightforward with (D.1).

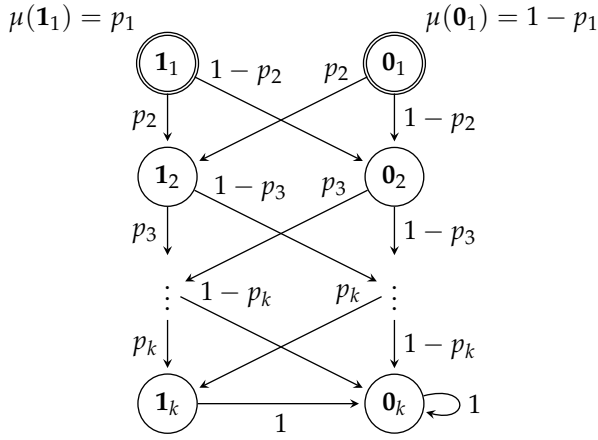


Fig. 6.6 The LMC Γ_π that encodes the product distribution π described by parameters $p_1, \ldots, p_k$.

Now, given two product distributions π_1^k and π_2^k respectively described by $p_1, \ldots, p_k$ and $q_1, \ldots, q_k$, let $\Gamma_{\pi_1^k}$ and $\Gamma_{\pi_2^k}$ be the two LMCs that encode π_1^k and π_2^k, respectively. A first observation is that, for all $i > k$, it holds that

$$\begin{aligned}
\mathsf{TV}(\mu^i_{\pi_1^k}, \mu^i_{\pi_2^k}) &= \frac{1}{2} \sum_{w \in \{0,1\}^i} |\mu^i_{\pi_1^k}(w) - \mu^i_{\pi_2^k}(w)| \\
&= \frac{1}{2} \sum_{w' \in \{0,1\}^k} |\mu^i_{\pi_1^k}(w'0\ldots0) - \mu^i_{\pi_2^k}(w'0\ldots0)| \\
&= \frac{1}{2} \sum_{w' \in \{0,1\}^k} |\pi_1^k(w') - \pi_2^k(w')| \\
&= \mathsf{TV}(\pi_1^k, \pi_2^k).
\end{aligned}$$

The latter yields

$$\begin{aligned}
\mathsf{CK}(\Gamma_{\pi_1^k}, \Gamma_{\pi_2^k}) &= \sum_{i=1}^{\infty} 2^{-i} \mathsf{TV}(\mu^i_{\pi_1^k}, \mu^i_{\pi_2^k}) \\
&= \sum_{i=1}^{\infty} 2^{-i} \mathsf{TV}(\pi_1^i, \pi_2^i) \\
&= \sum_{i=1}^{k-1} 2^{-i} \mathsf{TV}(\pi_1^i, \pi_2^i) + \mathsf{TV}(\pi_1^k, \pi_2^k) \left(\sum_{i=k}^{\infty} 2^{-i} \right) \\
&= \sum_{i=1}^{k-1} 2^{-i} \mathsf{TV}(\pi_1^i, \pi_2^i) + 2^{1-k} \mathsf{TV}(\pi_1^k, \pi_2^k).
\end{aligned} \tag{6.14}$$

In order to lighten the notation, for all $i \leq k$, let $\mathsf{CK}_i := \mathsf{CK}(\Gamma_{\pi_1^k}, \Gamma_{\pi_2^k})$ and $\mathsf{TV}_i = \mathsf{TV}(\pi_1^i, \pi_2^i)$, it holds from (6.14) that

$$\begin{pmatrix} \mathsf{CK}_1 \\ \mathsf{CK}_2 \\ \mathsf{CK}_3 \\ \vdots \\ \mathsf{CK}_{k-2} \\ \mathsf{CK}_{k-1} \\ \mathsf{CK}_k \end{pmatrix} = \begin{pmatrix} 1 & & & & & & \\ 2^{-1} & 2^{-1} & & & & & \\ 2^{-1} & 2^{-2} & 2^{-2} & & & & \\ \vdots & \vdots & \vdots & \ddots & & & \\ 2^{-1} & 2^{-3} & 2^{-2} & \dots & 2^{3-k} & & \\ 2^{-1} & 2^{-3} & 2^{-2} & \dots & 2^{2-k} & 2^{2-k} & \\ 2^{-1} & 2^{-3} & 2^{-2} & \dots & 2^{2-k} & 2^{1-k} & 2^{1-k} \end{pmatrix} \begin{pmatrix} \mathsf{TV}_1 \\ \mathsf{TV}_2 \\ \mathsf{TV}_3 \\ \vdots \\ \mathsf{TV}_{k-2} \\ \mathsf{TV}_{k-1} \\ \mathsf{TV}_k \end{pmatrix}$$

The matrix in the expression above is invertible since it is a triangular matrix whose diagonal entries are all strictly positive. Therefore, one can compute $\mathsf{TV}_k = \mathsf{TV}(\pi_1^k, \pi_2^k)$ in polynomial time by computing k times the CK distance, and solving the linear system of equations above. Therefore, by Proposition E.12, there exists a polynomial-time function $r : \{0,1\}^\star \to \{0,1\}^\star$ such that $f_{\mathsf{CK}} \circ r = f_{\mathsf{TV},\mathrm{prod}}/2M$, where M is the constant defined in Definition 6.22. □

Theorem 6.26 implies that, unless $\mathtt{P} = \mathtt{NP}$, there is no algorithm that computes exactly the CK distance in polynomial time.

We now prove a similar result for the approximation scheme presented in Section 6.3.

Definition 6.27. For any $k \in \mathbb{N}$, let $f_{S_k} : \{0,1\}^\star \to \mathbb{R}$ be the function that computes S_k. For any two LMCs Γ_1 and Γ_2 with rational parameters, it is

defined as

$$f_{S_k}(e(\Gamma_1), e(\Gamma_2)) := S_k(\Gamma_1, \Gamma_2),$$

where e denotes the binary encoding of the parameters of the LMCs (as defined in Appendix E).

Proposition 6.28. *For all $k \in \mathbb{N}$, the function $f_{S_k} : \{0,1\}^\star \to \mathbb{R}$ such as defined in Definition 6.27 is #P-hard.*

Proof. We prove this result in the same way as for the exact computation of f_{CK} (see Theorem 6.26): for any two product distributions π_1^k and π_2^k over $\{0,1\}^k$, we show that the computation of $\mathsf{TV}(\pi_1^k, \pi_2^k)$ can be reduced to the computation of $S_k(\Gamma_1, \Gamma_2)$ in polynomial time.

Given two product distributions π_1^k, π_2^k, let $\Gamma_{\pi_1^k}$ and $\Gamma_{\pi_2^k}$ be the two LMCs that encode π_1^k and π_2^k respectively, in the same fashion as in the proof of Theorem 6.26. Let S_{k-1} and S_k be the truncated sums with the distributions generated by $\Gamma_{\pi_1^k}$ and $\Gamma_{\pi_2^k}$, respectively up to horizons $k-1$ and k, it holds that

$$\mathsf{TV}(\pi_1^k, \pi_2^k) = m^k \left(S_k \left(\Gamma_{\pi_1^k}, \Gamma_{\pi_2^k} \right) - S_{k-1} \left(\Gamma_{\pi_1^k}, \Gamma_{\pi_2^k} \right) \right).$$

Therefore the computation of $\mathsf{TV}(p^k, q^k)$ is reduced to the computation of $S_k(\Gamma_{\pi_1^k}, \Gamma_{\pi_2^k})$ in polynomial time, and the proof is completed in the same fashion as for Theorem 6.26. □

Again Proposition 6.28 shows that, unless $\mathtt{P} = \mathtt{NP}$, there exists no polynomial algorithm to compute $S_k(\Gamma_1, \Gamma_2)$.

6.5 Summary and further research directions

Chapter 6 – Summary

In this chapter, in the context of the adaptive memory algorithm introduced in Chapter 4, we introduced – to the best of our knowledge – the first Cantor-Kantorovich notion of distance between LMCs.

Section 6.1 – Definition and link with Total Variation

We introduced the CK distance. We proved the central result that the CK distance is a discounted sum of TV distances, implying that it is a well-defined trace-based distance.

Section 6.2 – Continuity properties

We showed that the CK distance is both parameter- and (B-LTL)-continuous. We showed that these results give both practical and theoretical insight about CK distance.

Section 6.3 – Approximation

We proved that the CK distance can be approximated with arbitrarily small precision with a truncated discounted sum of TV distances, making the CK distance *computable* in that sense.

Section 6.4 – Computational complexity

We studied the computational complexity of both the exact computation and the truncated sum approximation of the CK distance. We showed that both are #P-hard, making the existence of a polynomial algorithm to compute them possible only if $\mathtt{P} = \mathtt{NP}$.

Chapter 6 – Further research directions

In general we believe that the Cantor-Kantorovich metric, with its topological and continuity properties, offers a powerful tool for both theoretical analysis and practical applications. This opens up several promising avenues for future research.

Efficient approximation scheme

A key open question is whether a polynomial approximation scheme exists for the CK distance. Recent advances have introduced both deterministic and randomized polynomial approximation schemes for the TV distance, even for distributions generated by Markov chains [BGM+24, FLL24]. Building on these results, a promising direction is to combine such schemes with Theorem 6.8 to develop an efficient approximation for the CK distance.

Framework generalization

While the CK distance was introduced specifically for LMCs (in the context of Chapter 4), its definition and properties extend naturally to any pair of probability measures on the measurable space

$(A^\omega, \sigma(\mathcal{T}^\omega))$. Beyond the theoretical interest, this generalization opens the door to applications beyond LMCs (e.g. to more general stochastic processes). Two natural extensions of the setting also deserve attention. First, one could consider a *continuous state-space* setting; however, this direction is arguably less interesting, as the Cantor metric on a continuous alphabet would essentially recover the classical Wasserstein distance with a p-norm on $\mathbb{R}^n$, losing the specificity of the discrete-alphabet construction. Second, a *continuous-time* setting appears more promising: the CK distance extends naturally to continuous-time Markov chains, and studying its properties in that context is an interesting direction for future work.

Applications

In Reinforcement Learning (see e.g. [SB] for an introduction), it is typical to transfer a learned policy from one Markov Decision Process to another. These approaches are referred to as *Transfer Learning* (*TL*) algorithms (see [ZLJZ23] for a review). Recent numerical experiments in [BRJ24] suggest that the CK distance could serve as an effective proxy for TL performance, making it a promising avenue for further investigation in TL algorithms. Other applications can be considered, e.g. the CK distance can be measured for providing sufficient conditions of safety.

Continuity with respect to discounted LTL

In Section 6.2, continuity was studied with respect to B-LTL properties. It could be interesting to investigate continuity with respect to discounted LTL formulae, where the "eventually" operators of LTL are relaxed into discounting operators, so that the satisfaction value – in $[0, 1]$ – decreases the longer it takes to fulfill eventuality requirements (see [ABK14]).

Conclusions and perspectives

THROUGHOUT this thesis, we have investigated the role of *memory* as an alternative to classical state space refinement in abstraction-based methods for the analysis of complex dynamical systems. Motivated by the challenges of epistemic uncertainty and, potentially, aleatoric uncertainty, we have developed a family of memory-dependent abstractions that encode sequences of observations rather than individual outputs. Crucially, these abstractions are *data-driven*: they can be constructed directly from output samples, without requiring an explicit model. In particular, we have shown that our method allows increasing the approximation quality of the abstraction in two major use cases: (1) for black-box systems, where the output is not chosen by the engineer, and (2) for complex systems, where it is sometimes smarter to use memory from a coarser partition than naïvely refining the state space discretisation.

As a transversal contribution, the thesis has introduced the so-called *Cantor-Kantorovich distance*, a novel metric between labelled Markov chains. This distance emerged naturally from the refinement algorithm described in Chapter 4, and its theoretical properties make it a tool of independent interest.

Brief summary of the contributions

In what follows, we summarize the contributions of this thesis.

Part II – Abstractions of deterministic systems

This part studies memory-dependent abstractions of deterministic dynamical systems affected by epistemic uncertainty. In particular, it studies the case where the abstraction is constructed from output sequences of equal length. Our study progresses from uniform memory schemes, where all abstract states encode sequences of equal length and asymptotic correctness is proven, to adaptive schemes, where memory depth is locally adjusted to the dynamics. The adaptive memory framework introduces timed sequences that encode both past and future information of varying lengths, establishes soundness guarantees (behavioural inclusion), and provides a greedy refinement algorithm that iteratively improves the abstraction without requiring specification knowledge. This refinement procedure motivated the development of a metric between labelled Markov chains, leading naturally to the Cantor-Kantorovich distance studied in Part IV.

Part III – Abstractions of stochastic systems

This part extends the memory-dependent abstraction framework to systems subject to *aleatoric uncertainty*, modelled through stochasticity. The central challenge is that partial observability of stochastic systems induces a fundamentally non-Markovian discrete observation process, causing abstractions to suffer from approximation errors. Memory-dependent abstractions address this issue by increasing the memory length in the approximating Markov process. The theoretical analysis departs significantly from the deterministic setting: it relies on *transfer operator theory* and interprets the abstraction construction as a *Galerkin approximation* of a lifted system whose state space encodes ℓ-length trajectories. This functional-analytic perspective provides rigorous approximation guarantees in the form of total variation bounds between the state distributions of the original system and its abstraction. The results demonstrate that memory may be a good alternative to classical state space refinement, even in the presence of intrinsic stochasticity.

Part IV – The Cantor-Kantorovich distance

This part develops the theoretical foundations of a novel distance between labelled Markov chains, the Cantor-Kantorovich distance. While initially motivated by the needs of the adaptive refinement algorithm presented in Part II, this distance constitutes a transversal contribution with potential applications beyond the abstraction framework. The CK distance is constructed by equipping the space of infinite output sequences with the Cantor distance – a distance generating the *natural topology* – and then by computing the Kantorovich distance between the distributions generated by the two labelled Markov chains. This makes the CK distance – or equivalently the discounted TV – a *canonical* choice, directly inherited from the natural topology of the output language. The part formalises this distance, and establishes continuity, approximation and complexity properties. It is shown that the CK distance is a discounted infinite sum of TV distances, enabling an approximation scheme by computing a truncated sum. On the continuity side, parameter-continuity and bounded specification errors are proven. On the complexity side, the exact computation and truncated sum approximation are both shown to be #P-hard.

Perspectives

This thesis opens several paths for future research. At the end of each chapter, we have already added several perspectives for future work. In the following, we gather, in our view, the most important and general ideas.

Smart abstraction landscape It is clear from our results that memory-dependent abstractions belong to the family of *smart abstractions*. Indeed, it was shown from numerical experiments that, for the same abstraction size, a memory-dependent abstraction can achieve better approximation results than a uniform gridding abstraction. However, it is still not clear how memory-based techniques compare to other smart abstraction methods or, in other words, which methods best beat the curse of dimensionality. In particular, given a dynamical system (and potentially a control problem), we are still not able to choose *a priori* the best smart abstraction method. Concerning memory, identifying the family of systems for which memory-dependent abstractions are the best choice would be a crucial breakthrough.

We believe that the solution to this problem may be linked to information theoretic concepts. In the recent preprint [DG25], Delimpaltadakis and Gleizer formalise a fundamental abstraction size-accuracy trade-off limit by formalising abstractions as encoder-decoder pairs, and size and trade-off as *rate* and *distortion* respectively – concepts taken from Shannon's rate-distortion theory [Sha93]. In [DG25], the authors identify examples for which memory reaches this limit, meaning that memory is the best choice in the sense of the accuracy described in the paper. More broadly, we also believe that *mixing* properties of dynamical systems play a key role in determining when memory-dependent abstractions are well-posed. A mixing system is one whose trajectories eventually explore the entire invariant set, guaranteeing that the invariant measure is non-degenerate – a condition that our methods implicitly rely on (see Assumption 3.11 in Chapter 3). Formalising this connection would clarify the scope of applicability of our methods.

Application to large-scale systems While our memory-dependent abstractions may reduce the number of states required to achieve a given approximation accuracy – thereby suffering less from the curse of dimensionality than classical uniform gridding methods – they remain limited in practice by the state space dimension and the number of outputs. Therefore, scaling to very high-dimensional systems (e.g., hundreds of state variables) or to systems with many outputs remains out of reach. A promising direction to address this limitation is to combine our abstraction methods with *model reduction* approaches within the *compositional methods* framework (see e.g. [LSZ21]). Applied to our framework, one could first reduce the system dimension via model reduction techniques and then apply our memory-dependent abstraction to the reduced model. The compositional guarantees would then propagate the approximation error from the reduced abstraction back to the original high-dimensional system, potentially enabling scalable verification for systems that are currently beyond reach.

Fully data-driven setting Our methods are data-driven in the sense that they only require samples to construct abstractions. However, their guarantees overlook two challenges: in practice, abstractions are built from a finite number of observations, and real-world data is often noisy.

To address these issues, several mathematical tools could be explored. The *scenario approach* [CG19] is widely used in control theory, especially

for abstraction-based methods. Such analysis has already been done in [CPM23] for the uniform memory and deterministic framework, focusing on the data-driven side, while our qualitative guarantees on abstraction performance complement these results. Alternative techniques include *conformal prediction* [CPLM24] and classical *Clopper-Pearson intervals* [NBSA25, NBS+25], the latter of which is shown to be less conservative than the scenario approach in the context of [NBSA25]. Additionally, the *Baum-Welch algorithm* – a method for learning labelled Markov chains – has yet to be applied in this context. Some works investigate the statistical guarantees of the Baum-Welch algorithm (see e.g. [YBW15]), and could therefore be used to derive finite-sample guarantees in the context of abstraction learning.

Controller synthesis and specification-driven verification Throughout this thesis, our approximation guarantees are based on the difference between behaviours in the deterministic case and between state distributions in the stochastic case, making them focused on *specification-free verification*. Two natural and crucial extensions would be to develop (1) specification-driven construction of the abstractions for a more efficient verification, and (2) controller synthesis methods that leverage these abstractions, where the control action depends on a history of previous observations.

To develop specification-driven verification, a promising direction is the use of a specification-driven distance in the REFINE algorithm from Chapter 4. To develop controller synthesis for deterministic systems, the soundness guarantees established in Chapter 4 suggest that solving games (as in [BYAG17]) on the adaptive memory abstraction would yield controllers that are correct-by-construction for the original system. For stochastic systems, the bounded approximation errors derived in Chapter 5 could enable probabilistic controller synthesis with quantifiable guarantees, potentially using tools such as PRISM [KNP11].

Field of applications for the CK distance While the Cantor-Kantorovich distance was developed within the abstraction framework, its construction as a natural distance between labelled Markov chains suggests broader applicability. Distances between probabilistic models are central to numerous fields beyond control theory. In *reinforcement learning*, comparing policies or value functions across different Markov Decision Processes is essential for transfer learning [GVF22]. The CK distance's natural construction makes it a promising candidate for addressing these comparison

challenges across diverse application domains.

To achieve more applicability, multiple theoretical extensions can be considered. First, the CK distance can easily be extended to more general probabilistic models on infinite languages. Second, generalizing the CK distance to *Markov Decision Processes* (*MDPs*) by incorporating action spaces could enable comparisons of control-augmented systems. In [BRJ24], a first step was made by computing the CK distance between (fixed policy) MDPs. We showed that the CK distance was correlated to the transfer learning efficiency, and could therefore serve as a good proxy for it.

Appendices

A

Topological and metric spaces

In this appendix we introduce the abstract notion of *topological space*. We then introduce the concept of both *metric* and *ultrametric* spaces, two particular cases of topological spaces. All the concepts and results presented in this chapter are well-established in the literature (see, for example, [Mun00]).

A.1 Topological spaces

Definition A.1 (Topology)**.** Let X be a set. A collection $\mathcal{T}$ of subsets of X is called a *topology* for X if

1. $X \in \mathcal{T}$ and $\emptyset \in \mathcal{T}$,
2. for any collection $\{T_i\}_{i \in I} \subseteq \mathcal{T}$, with I an arbitrary index set, $\bigcup_{i \in I} T_i \in \mathcal{T}$,
3. for any finite collection $T_1, \ldots, T_n \in \mathcal{T}$ with $n \in \mathbb{N}$ an arbitrary number, $\bigcap_{i=1}^{n} T_i \in \mathcal{T}$.

If $\mathcal{T}$ is a topology, its members are called *open sets*.

Intuitively, a topology defines a *structure* for a set.

Definition A.2 (Topological space)**.** Let X be a set and $\mathcal{T}$ be a topology for X, then the tuple $(X, \mathcal{T})$ is called a topological space.

The notion of *open set* has a counterpart, called a *closed set*, which is the complement of an open set. This is formally defined as follows.

Definition A.3 (Closed subset)**.** Let $(X, \mathcal{T})$ be a topological space. A subset $Y \subseteq X$ is said to be *closed* in X if $X \setminus Y \in \mathcal{T}$.

It is sometimes convenient to define a topology by a *basis* generating the topology. It is defined as follows.

Definition A.4 (Topology basis)**.** Let $(X, \mathcal{T})$ be a topological space. A collection $\mathcal{B}$ is called a basis for $(X, \mathcal{T})$ if any open set in $\mathcal{T}$ is the union of sets in $\mathcal{B}$.

Intuitively, a topology generated by a basis is the smallest topology containing all the sets in this basis.

We now give examples of topologies. First, we introduce the useful *discrete* and *product* topologies, defined as follows.

Proposition A.5 (Discrete topological space)**.** *Let X be a set. The* discrete topology $\mathcal{T}$ *on X is defined as the collection of all subsets of X. The topological space $(X, \mathcal{T})$ is then called the* discrete topological space*.*

Definition A.6 (Product topological space)**.** Let I be any non-empty index set and, for all $i \in I$, let $(X_i, \mathcal{T}_i)$ be a topological space. Let X be the *Cartesian product* of the sets X_i, denoted by $X := \prod_{i \in I} X_i$. For all $i \in I$, let

$$p_i := X \to X_i : (x_j)_{j \in I} \mapsto x_i.$$

The collection $\mathcal{T} := \bigotimes_{i \in I} \mathcal{T}_i$, defined as the smallest collection for which all the projections p_i are *continuous* is called the *product topology*. The topological space $(X, \mathcal{T})$ is then called the *product topological space*.

A.2 (Ultra-)metric spaces

We will now provide another famous class of topologies, namely *metrizable* topologies. These are generated by *(ultra-)metric spaces*, defined as follows.

Definition A.7 (Distance)**.** Let X be a set. A function $\mathsf{D} : X \times X \to \mathbb{R}_{\geq 0}$ is called a *distance* if

1. for all $x \in X$, $\mathsf{D}(x,x) = 0$,
2. for all $x, y \in X$, if $x \neq y$, then $\mathsf{D}(x,y) > 0$,
3. for all $x, y \in X$, $\mathsf{D}(x,y) = \mathsf{D}(y,x)$,
4. for all $x, y, z \in X$, $\mathsf{D}(x,z) \leq \mathsf{D}(x,y) + \mathsf{D}(y,z)$.

Definition A.8 (Ultradistance)**.** Let X be a set. A function $\mathsf{D} : X \times X \to \mathbb{R}_{\geq 0}$ is called an *ultradistance* if it satisfies points 1, 2 and 3 of the definition above, and

4′. for all $x, y, z \in X$, $\mathsf{D}(x,z) \leq \max\{\mathsf{D}(x,y), \mathsf{D}(y,z)\}$.

It follows from the definition of an ultradistance that it is also a distance.

Definition A.9 (Metric and ultrametric spaces)**.** Let X be a set, and D be a distance (resp. ultradistance) on X, then the tuple (X, D) is called a *metric space* (resp. *ultrametric space*).

Note that any (ultra-)metric space is also a topological space, although the converse is not true.

Definition A.10 (Topology induced by a metric)**.** Let (X, D) be a metric space. A collection $\mathcal{T}$ of subsets of X is called the *topology induced by* D if

1. for all $x \in X$ and $\varepsilon > 0$, the open ball of radius ε and centred at x, defined as
$$B(x;\varepsilon) := \{y \in X : \mathsf{D}(x,y) < \varepsilon\},$$
is in $\mathcal{T}$,
2. any subset $T \subseteq X$ is in $\mathcal{T}$ if, for all $x \in T$, there exists $\varepsilon > 0$ such that $B(x;\varepsilon) \subseteq T$.

Such topologies are known as metrizable topologies.
We conclude this section by defining the notion of *complete* metric space.

Definition A.11 (Cauchy sequence)**.** Let (X, D) be a metric space. A sequence $(x_1, x_2, \dots)$ of elements in X is said to be *Cauchy* if, for every $\varepsilon > 0$, there exists $K \in \mathbb{N}$ such that for all $k, l > K$, $\mathsf{D}(x_k, x_l) < \varepsilon$.

Definition A.12 (Complete space)**.** A metric space (X, D) is said to be complete if every Cauchy sequence $(x_1, x_2, \dots)$ converges inside X.

B

Measure theory

THIS appendix presents the concepts of *measurable space* and *measure*, which together form the core of *measure theory*. Measure theory is fundamental to probability theory (covered in Appendix D) and will be particularly relevant for this purpose. All concepts and results discussed in this appendix are standard and can be found in the existing literature (see e.g. [Hal50]).

We start by defining the more general notions of *algebra* and *pre-measure*.

Definition B.1 (Algebra)**.** Let X be a set. A collection $\mathcal{A}$ of subsets of X is called an algebra of X if

1. $X \in \mathcal{A}$,
2. if $A \in \mathcal{A}$, then $X \setminus A \in \mathcal{A}$,
3. if $A_1, A_2, \ldots, A_k \in \mathcal{A}$, then $\bigcup_{i=1,2,\ldots,k} A_i \in \mathcal{A}$.

Definition B.2 (Pre-measure)**.** Let X be a set, and $\mathcal{A}$ an algebra of subsets of X. A function $\mu : \mathcal{A} \to \mathbb{R}_{\geq 0} \cup \{+\infty\}$ is called a pre-measure if

1. $\mu(\emptyset) = 0$,
2. for every countable collection of pairwise disjoint sets $A_1, A_2, \cdots \in \mathcal{A}$, $\mu\left(\bigcup_{i=1,2,\ldots} A_i\right) = \sum_{i=1,2,\ldots} \mu(A_i)$.

We now provide the definition of a σ-algebra, a particular case of an algebra.

Definition B.3 (σ-algebra)**.** Let X be a set. A collection $\mathcal{A}$ of subsets of X is called a *σ-algebra* of X if

1. it is an algebra,
2. if a countable collection $A_1, A_2, \cdots \in \mathcal{A}$, then $\bigcup_{i=1,2,\ldots} A_i \in \mathcal{A}$.

Note that it also follows from the definition of a σ-algebra that it is closed under countable intersection as well.

Definition B.4 (Measurable space)**.** Let X be a set and $\mathcal{A}$ be a σ-algebra of X, then the tuple $(X, \mathcal{A})$ is called a measurable space.

The notion of measurable space resembles the notion of topological space, although both concepts are fundamentally different. A topology defines a family of *open sets* closed under arbitrary unions and finite intersections, while a σ-algebra defines a family of *measurable* sets closed under countable unions, countable intersections, and complements [Tao10]. Intuitively, a topology defines a *structure*, whereas a σ-algebra defines the notion of *size*.

Similarly to topologies, it is sometimes convenient to define a σ-algebra from a basis of subsets. The definition of *basis* extends to σ-algebras as follows.

Definition B.5 (σ-algebra basis)**.** Let $(X, \mathcal{A})$ be a measurable space. A collection $\mathcal{B}$ is called a basis for $(X, \mathcal{A})$ if $\mathcal{A}$ is the smallest σ-algebra containing all sets in $\mathcal{B}$.

A topology can generate a σ-algebra, in which case it is called a *Borel σ-algebra*. Note however that not all topologies are σ-algebras, nor are all σ-algebras generated by topologies.

Definition B.6 (Borel σ-algebra)**.** Let $(X, \mathcal{T})$ be a topological space. The Borel σ-algebra is the σ-algebra whose basis is $\mathcal{T}$. It is denoted $\sigma(\mathcal{T})$.

In the following, we define the notion of *power set*.

Definition B.7 (Power set)**.** Let X be a set. The collection composed of all subsets of X is called the power set of X, and is denoted 2^X.

While power sets are defined for arbitrary sets, they take on a particularly simple form when X is finite. Specifically, the power set is the only σ-algebra on a finite set.

Proposition B.8. *If X is finite, the unique σ-algebra of X is 2^X.*

We now define the notion of measure, a notion that will be useful for defining *probability measures* in Appendix D. The notion of measure is a particular case of a pre-measure, defined above.

Definition B.9 (Measurable function)**.** Let $(X, \mathcal{A})$ and $(Y, \mathcal{B})$ be two measurable spaces. A function $f : X \to Y$ is said to be measurable if, for every $B \in \mathcal{B}$, $f^{-1}(B) \in \mathcal{A}$, where $f^{-1}(B) := \{x \in X : f(x) \in B\}$ is called the *pre-image* of B.

Definition B.10 (Measure)**.** Let $(X, \mathcal{A})$ be a measurable space. A function $\mu : \mathcal{A} \to \mathbb{R}_{\geq 0} \cup \{+\infty\}$ is called a measure if

1. for all $A \in \mathcal{A}$, $\mu(A) \geq 0$,
2. for every countable collection of pairwise disjoint sets $A_1, A_2, \cdots \in \mathcal{A}$, $\mu\left(\bigcup_{i=1,2,\ldots} A_i\right) = \sum_{i=1,2,\ldots} \mu(A_i)$.

Note that countable additivity (point 2 above) implies that $\mu(\emptyset) = 0$ as, for any $A \in \mathcal{A}$ such that $\mu(A) < +\infty$, $\mu(A) = \mu(A \cup \emptyset) = \mu(A) + \mu(\emptyset)$, which implies $\mu(\emptyset) = 0$. A (pre-)measure is said to be *finite* if there is no subset $A \in \mathcal{A}$ such that $\mu(A) = +\infty$.

Sometimes, it is convenient to define measures *by extension*, i.e. we would like to extend a notion of pre-measure defined on some algebra to a notion of measure defined on the σ-algebra generated by this algebra[1] *Carathéodory's extension theorem* allows for such a construction.

Theorem B.11 (Carathéodory's extension theorem)**.** *Let X be a set, and let $\mathcal{A}_0$ be an algebra of X, and $\mathcal{A}$ be a σ-algebra generated by $\mathcal{A}_0$, that is the σ-algebra whose basis is $\mathcal{A}_0$. Let $\mu_0 : \mathcal{A}_0 \to \mathbb{R}_{\geq 0} \cup \{+\infty\}$ be a pre-measure, and let the function $\mu_\star : 2^X \to \mathbb{R}_{\geq 0} \cup \{+\infty\}$ be defined as*

$$\mu_\star(A) := \inf\left\{\sum_{i=1}^{\infty} \mu_0(A_i) : A_i \in \mathcal{A}_0, A \subseteq \bigcup_{i=1}^{\infty} A_i\right\}.$$

Then the function defined as $\mu : \mathcal{A} \to \mathbb{R} \cup \{+\infty\} : A \mapsto \mu_\star(A)$ is a measure on $\mathcal{A}$. Moreover, if μ_0 is finite, then so is μ.

Carathéodory's extension theorem is, for instance, useful for defining measures on finite sets $(X, 2^X)$. Indeed, a corollary of this theorem is that any measure on $(X, 2^X)$ is uniquely defined by its values on the singletons

[1] In the sense that the algebra forms a basis of the σ-algebra.

$\{x\}$, for $x \in X$, which form the algebra in Theorem B.11. It holds that, for all $A \in 2^X$,

$$\mu(A) := \sum_{x \in A} \mu(\{x\}).$$

From now on, by abuse of notation, we will use $\mu(x) := \mu(\{x\})$ for any $x \in X$ in the case of finite sets.

Next, we introduce the *Radon-Nikodym theorem*, which defines a notion of *density* for measures.

Definition B.12 (Absolute continuity)**.** Let $(X, \mathcal{A})$ be a measurable space, and let μ and ν be two finite measures on $(X, \mathcal{A})$. ν is said to be *absolutely continuous* with respect to μ, denoted $\nu \ll \mu$, if, for all $A \in \mathcal{A}$, if $\mu(A) = 0$, then $\nu(A) = 0$.

Theorem B.13 (Radon-Nikodym theorem)**.** *Let $(X, \mathcal{A})$ be a measurable space, and let μ and ν be two measures on $(X, \mathcal{A})$. If $\nu \ll \mu$, then there exists a measurable function $f : X \to \mathbb{R}_{\geq 0}$ such that, for all $A \in \mathcal{A}$,*

$$\nu(A) = \int_{x \in A} f(x) \mu(dx). \tag{B.1}$$

The function f above is uniquely defined μ-almost everywhere.[2] It is denoted $\mathrm{d}\nu/\mathrm{d}\mu$, and is called the *Radon-Nikodym derivative*. The Radon-Nikodym derivative can be interpreted as a density of ν with respect to μ. This notion will be useful to define *probability density functions* in Appendix D.

Measures can be pushed forward by functions, a notion that is particularly useful in the context of dynamical systems (see Chapter 2).

Definition B.14 (Pushforward measure by a function)**.** Let $(X, \mathcal{A})$ and $(Y, \mathcal{B})$ be two measurable spaces, and let $f : X \to Y$ be a measurable function. Given a measure $\mu : \mathcal{A} \to \mathbb{R}_{\geq 0}$, the *pushforward measure* of μ by f is the measure $f\#\mu : \mathcal{B} \to \mathbb{R}_{\geq 0}$ given by $(f\#\mu)(B) = \mu(f^{-1}(B))$ for all $B \in \mathcal{B}$.

Successive pushforward measures are denoted by $f^k\#\mu$, for $k \in \mathbb{N}$. It is formally defined recursively as $f^{k+1}\#\mu := f\#(f^k\#\mu)$, with $f^1\#\mu := f\#\mu$.

We conclude this section by defining a particular collection of measures, called *transition kernels*. This notion will be useful to define *Markov kernels* in Appendix D.

[2] That is, if there is another g satisfying (B.1), then $f(x) = g(x)$ for μ-almost every $x \in X$.

Definition B.15 (Transition kernel)**.** Let $(X, \mathcal{A})$ and $(Y, \mathcal{B})$ be two measurable spaces. A function $\tau : \mathcal{B} \times X \to \mathbb{R}_{\geq 0}$ is called a *transition kernel* from $(X, \mathcal{A})$ to $(Y, \mathcal{B})$ if

1. for all $B \in \mathcal{B}$, the map $x \mapsto \tau(B|x)$ is measurable,
2. for all $x \in X$, the map $B \mapsto \tau(B|x)$ is a measure on $(Y, \mathcal{B})$.

Similarly to functions, measures can be pushed forward by transition kernels.

Definition B.16 (Pushforward measure by a kernel)**.** Let $(X, \mathcal{A})$ and $(Y, \mathcal{B})$ be two measurable spaces, and let $\tau : \mathcal{B} \times X \to \mathbb{R}_{\geq 0}$ be a transition kernel. Given a measure $\mu : \mathcal{A} \to \mathbb{R}_{\geq 0}$, the pushforward measure of μ by τ is the measure $\tau \# \mu : \mathcal{B} \to \mathbb{R}_{\geq 0}$ given by

$$(\tau \# \mu)(B) = \int_{x \in X} \tau(B|x) \mu(\mathrm{d}x)$$

for all $B \in \mathcal{B}$.

Again, similarly to functions, successive pushforwards by kernels are denoted $\tau^k \# \mu$ for $k \in \mathbb{N}$.

C

Linear Algebra

In this appendix, we present concepts from *linear algebra*. Throughout this appendix, we assume that X is a real *vector space*, whose definition is skipped for the sake of brevity (see e.g. [HK71] for a formal definition). More precisely, we will define the notions of *normed* and *Banach spaces*, as well as *inner products* and *Hilbert spaces*. We will conclude by introducing $L^p(\mu)$ spaces of functions. All definitions and results introduced in this appendix are standard in the literature (see e.g. [Wil22]).

C.1 Norms and Banach spaces

We start by introducing *normed spaces*.

Definition C.1 (Norm)**.** Let X be a vector space. A function $\|\cdot\| : X \to \mathbb{R}$ is called a *norm* on X if

1. for all $x, y \in X$, $\|x + y\| \leq \|x\| + \|y\|$,
2. for all $x \in X$ and $\alpha \in \mathbb{R}$, $\|\alpha x\| = |\alpha|\|x\|$, and
3. for all $x \in X$, $\|x\| = 0$ implies $x = 0$.

Note that the definition above implies that any norm is non-negative, that is, for all $x \in X$, $\|x\| \geq 0$.

Definition C.2 (Normed space)**.** Let X be a vector space, and $\|\cdot\| : X \to \mathbb{R}_{\geq 0}$ be a norm for X, then the tuple $(X, \|\cdot\|)$ is called a *normed space*.

Any norm induces a distance of the form $\mathsf{D}(x, y) = \|x - y\|$ for any $x, y \in X$. Therefore, one can say that a normed space $(X, \|\cdot\|)$ is also a metric space (X, D), with D as defined above.

We conclude by introducing Banach spaces, i.e. complete normed spaces (see Definition A.12 for a formal definition of completeness).

Definition C.3 (Banach space)**.** Let $(X, \|\cdot\|)$ be a normed space. If the metric space (X, D) induced by the norm is complete, then it is called a *Banach space*.

C.2 Inner products and Hilbert spaces

We now introduce *inner product spaces*. Note that, in this thesis, we restrict to real-valued inner products.

Definition C.4 (Inner product)**.** Let X be a vector space. A function $\langle\cdot,\cdot\rangle : X \times X \to \mathbb{R}$ is an inner product if

1. for all $x, y \in X$, $\langle x, y\rangle = \langle y, x\rangle$,
2. for all $x, y, z \in X$ and $\alpha, \beta \in \mathbb{R}$, $\langle \alpha x + \beta y, z\rangle = \alpha\langle x, z\rangle + \beta\langle y, z\rangle$,
3. for all $x \neq 0$, $\langle x, x\rangle > 0$.

Definition C.5 (Inner product space)**.** Let X be a vector space, and $\langle\cdot,\cdot\rangle : X \times X \to \mathbb{R}$ be an inner product on X, then the tuple $(X, \langle\cdot,\cdot\rangle)$ is called an *inner product space*.

Inner products define *orthogonality*. Two vectors $x, y \in X$ are said to be *orthogonal* with each other if

$$\langle x, y\rangle = 0.$$

Also, any inner product induces a norm of the form $\|x\| = \sqrt{\langle x, x\rangle}$ for any $x \in X$. Therefore, one can say that an inner product space $(X, \langle\cdot,\cdot\rangle)$ is also a normed space $(X, \|\cdot\|)$, with $\|\cdot\|$ as defined above. One can ultimately say that it is therefore also a metric space (X, D), with

$$\mathsf{D}(x, y) = \|x - y\| = \sqrt{\langle x - y, x - y\rangle} \tag{C.1}$$

for all $x, y \in X$.

We now introduce Hilbert spaces, i.e. complete inner product spaces.

Definition C.6 (Hilbert space)**.** Let $(X, \langle \cdot, \cdot \rangle)$ be an inner product space. If the normed space $(X, \| \cdot \|)$ induced by the inner product is also a Banach space, then it is called a *Hilbert space*.

Definition C.7 (Orthogonal complement)**.** Let $(X, \langle \cdot, \cdot \rangle)$ be a Hilbert space, and let $Y \subseteq X$ be a closed subset of X.[1] The *orthogonal complement* of Y is defined as

$$Y^\perp := \{x \in X : \langle x, y \rangle = 0, \forall y \in Y\}.$$

Proposition C.8. *Let $(X, \langle \cdot, \cdot \rangle)$ be a Hilbert space, and let $Y \subseteq X$ be a closed subset of X. It holds that*

1. *$Y \cup Y^\perp = X$ and $Y \cap Y^\perp = \emptyset$,*
2. *$(Y^\perp)^\perp = Y$, and*
3. *$Y^\perp$ is closed in X.*

C.3 $L^p(\mu)$ spaces

We conclude this chapter by introducing spaces of integrable functions, named $L^p(\mu)$ spaces. These spaces and their properties (especially for *square integrable functions*, i.e. $p = 2$) will be useful to manipulate probability density functions in Chapter D and are useful in Chapter 5.

Definition C.9 ($L^p(\mu)$ space)**.** Let $(X, \mathcal{A})$ be a measurable space endowed with the measure μ. For any $f : X \to \mathbb{R}$, and for any $p \in \mathbb{N} \cup \{+\infty\}$, let $\| \cdot \|_p$ be defined as

$$\|f\|_p := \left(\int_{x \in X} |f(x)|^p \mu(\mathrm{d}x) \right)^{1/p}. \tag{C.2}$$

for finite p and

$$\|f\|_\infty := \inf\{C \in \mathbb{R}_{\geq 0} : |f(x)| \leq C \ \mu\text{-almost everywhere}\}.$$

for $p = \infty$. For any $p \in \mathbb{N} \cup \{+\infty\}$, the set $L^p(\mu)$ denotes the set of all functions $f : X \to \mathbb{R}$ such that $\|f\|_p < +\infty$.

[1] In the sense that Y is closed in X.

Proposition C.10. *Let $(X, \mathcal{A})$ be a measurable space endowed with a measure μ. Let $1 \leq p < q < +\infty$. If μ is a finite measure, then $L^q(\mu) \subseteq L^p(\mu)$.*

The functions $\|\cdot\|_p$ as defined in (C.2) are norms, and the resulting normed space is complete. This is formalised in the following proposition.

Proposition C.11. *For all $p \in \mathbb{N} \cup \{+\infty\}$, the tuple $(L^p(\mu), \|\cdot\|_p)$ is a Banach space.*

However, $L^2(\mu)$, called the set of *square-integrable functions*, is the only Hilbert space among $L^p(\mu)$ spaces.

Proposition C.12. *For $f, g \in L^2(\mu)$, define*

$$\langle f, g \rangle := \int_{x \in X} f(x) g(x) \mu(dx).$$

Then $(L^2(\mu), \langle \cdot, \cdot \rangle)$ is a Hilbert space.

Note that the norm induced by the inner product $\langle \cdot, \cdot \rangle$ is the norm $\|\cdot\|_2$ as defined in (C.2).

In Chapter 5, we upper bound the $\|\cdot\|_1$ norm of functions in the subset $L^2(\mu) \subseteq L^1(\mu)$. The following fundamental result from Hölder is useful for this purpose.

Theorem C.13 (Hölder's inequality)**.** *Let $(X, \mathcal{A})$ be a measurable space endowed with a measure μ. Let $p, q, r \in \mathbb{N} \cup \{+\infty\}$ that satisfy $1/p + 1/q = 1/r$. If $f \in L^p(\mu)$ and $g \in L^q(\mu)$, then $fg \in L^r(\mu)$ and*

$$\|fg\|_r \leq \|f\|_p \|g\|_q.$$

The following directly follows from Hölder's inequality with $p = q = 2$ and $r = 1$, and from Proposition C.10.

Corollary C.14. *Let $(X, \mathcal{A})$ be a measurable space endowed with a finite measure μ. Then, for all functions $f \in L^2(\mu) \subseteq L^1(\mu)$, it holds that*

$$\|f\|_1 \leq \|f\|_2.$$

D

Probability theory

In this appendix we introduce *probability theory*, a cornerstone of this thesis' contributions. Building on the foundations established in Appendix B, we begin by defining *probability measures* and *density functions*, concepts that are useful in all chapters of this thesis. To conclude, we develop notions of distances between probability measures, which are essential for the results presented in Chapter 6.

D.1 Probability measures and densities

In this section, we introduce the foundational concepts of probability theory, namely probability measures, *probability density functions*, and *Markov kernels*. We also delve into the notion of invariant measures, which plays a central role in this thesis. All the results and definitions presented here are standard and can be found in the existing literature (see e.g. [Bil12]).

We start by defining the notion of *probability measure*, a particular family of measures (see Definition B.10).

Definition D.1 (Probability measure)**.** Let $(X, \mathcal{A})$ be a measurable space. A measure $\mu : \mathcal{A} \to \mathbb{R}_{\geq 0}$ is a probability measure if

1. for all $A \in \mathcal{A}$, $\mu(A) \in [0, 1]$,
2. $\mu(X) = 1$.

If μ is a probability measure on $(X, \mathcal{A})$, the tuple $(X, \mathcal{A}, \mu)$ is often referred to as the *probability space* in the literature, and we call X the *sample space* and $\mathcal{A}$ the *event space*.

It is straightforward from point 1 in the definition above that a probability measure is always finite. Also, note that, given a measurable space $(X, \mathcal{A})$, for any measurable function $f : X \to X$ and probability measure $\mu : \mathcal{A} \to [0, 1]$, the pushforward measure $f\#\mu$ is also a probability measure.

Remark D.2. Let us focus on the special case when $X = \{x_1, \ldots, x_n\}$ is finite, and therefore $\mathcal{A} = 2^X$. In that case, in order to define a probability measure, it suffices to define a function $\mu : 2^X \to [0, 1]$ such that

1. for all $x \in X$, $\mu(x) \in [0, 1]$, and
2. $\sum_{x \in X} \mu(x) = 1$,

where $\mu(x) := \mu(\{x\})$ by abuse of notation. And, for any $A \in 2^X$, the definition is extended via

$$\mu(A) = \sum_{x \in A} \mu(x).$$

It is then straightforward that this function is a probability measure on $(X, 2^X)$.

The notion of *random variable* formalises a quantity that depends on a random event.

Definition D.3 (Random variable)**.** Let $(X, \mathcal{A})$ be a measurable space, together with a probability measure $\mu : \mathcal{A} \to [0, 1]$. A random variable (associated with μ) is a function $\mathsf{x} : X \to Y$, where x is a measurable function, and $(Y, \mathcal{B})$ a measurable space. An element of the image of x, that is an element $y := \mathsf{x}(x)$ for some $x \in X$, is called a *realisation* of the random variable. The *probability* of a random variable associated with measure μ is noted

$$\mathbb{P}_\mu(\mathsf{x} \in B) := \mu\left(\{x \in X : \mathsf{x}(x) \in B\}\right)$$

for any $B \in \mathcal{B}$.

In the definition above, when clear from the context, the probability $\mathbb{P}_\mu(\mathsf{x} \in B)$ is simply denoted $\mathbb{P}(\mathsf{x} \in B)$. Intuitively, a random variable formalises a quantity that depends on a random event, and a realisation is one observation of this quantity.

We will now define the notions of *joint* and *conditional* probabilities. We first introduce the notion of *coupling*. In the following, we will consider the *product σ-algebra*. Let $(X, \mathcal{A})$ and $(Y, \mathcal{B})$ be two measurable spaces. The product σ-algebra of $X \times Y$ is denoted as $\mathcal{A} \otimes \mathcal{B}$, and is defined as the smallest σ-algebra containing all measurable sets $A \times B$, with $A \in \mathcal{A}$ and $B \in \mathcal{B}$.

Definition D.4 (Coupling)**.** Let $(X, \mathcal{A})$ and $(Y, \mathcal{B})$ be two measurable spaces, and $\mu : \mathcal{A} \to [0,1]$ and $\nu : \mathcal{B} \to [0,1]$ be two probability measures, respectively defined on $(X, \mathcal{A})$ and $(Y, \mathcal{B})$. A probability measure $\gamma : \mathcal{A} \otimes \mathcal{B} \to [0,1]$, defined on the product measurable space $(X \times Y, \mathcal{A} \otimes \mathcal{B})$ is called a coupling of μ and ν if

1. for all $A \in \mathcal{A}$, $\gamma(A \times Y) = \mu(A)$, and
2. for all $B \in \mathcal{B}$, $\gamma(X \times B) = \nu(B)$.

The set of all couplings of two probability measures μ and ν is denoted $\mathcal{C}(\mu, \nu)$.

The notion of coupling formalises (among others) the case when two random events happen in the same time. Let $(X, \mathcal{A})$ and $(Y, \mathcal{B})$ be two measurable spaces, together with probability measures μ and ν, respectively. Let $\gamma \in \mathcal{C}(\mu, \nu)$ be a coupling of μ and ν. Let $\mathsf{x} : \mathcal{A} \to \overline{X}$ and $\mathsf{y} : \mathcal{B} \to \overline{Y}$ be two random variables, for any two measurable spaces $(\overline{X}, \overline{\mathcal{A}})$ and $(\overline{Y}, \overline{\mathcal{B}})$. Then the *joint probability* of x and y is noted

$$\mathbb{P}(\mathsf{x} \in \overline{A}, \mathsf{y} \in \overline{B}) := \gamma(\{x \in X : \mathsf{x}(x) \in \overline{A}\} \times \{y \in Y : \mathsf{y}(y) \in \overline{B}\})$$

for any two sets $\overline{A} \in \overline{\mathcal{A}}$ and $\overline{B} \in \overline{\mathcal{B}}$.

We also define the notion of *conditional probability*, i.e. a probability that depends on another random variable. Let $\mu, \nu, \gamma, \mathsf{x}$ and y be such as defined above. The conditional probability of x given y is defined as

$$\mathbb{P}(\mathsf{x} \in \overline{A} \,|\, \mathsf{y} \in \overline{B}) := \frac{\mathbb{P}(\mathsf{x} \in \overline{A}, \mathsf{y} \in \overline{B})}{\mathbb{P}(\mathsf{y} \in \overline{B})}$$

for any two sets $\overline{A} \in \overline{\mathcal{A}}$ and $\overline{B} \in \overline{\mathcal{B}}$.

A classical tool to work with probability measures is the notion of *density* with respect to another probability measure. Let ν and μ be two probability measures on a measurable space $(X, \mathcal{A})$, such that $\nu \ll \mu$ (i.e. ν is

absolutely continuous with respect to μ, see Definition B.12). The Radon-Nikodym derivative $f := \mathrm{d}\nu/\mathrm{d}\mu$ (see Theorem B.13) is called the *probability density function of ν with respect to μ*. When clear from the context, we will simply refer to such functions as *densities*.

Remark D.5. Classically, the notion of *probability density function* is integrated in the following way:

$$\nu(A) = \int_{x \in A} f(x) \mathrm{d}x.$$

However, we draw the attention of the reader to the fact that such probability density functions correspond to the special case where μ is the Lebesgue measure (see e.g. [Roy88] for a formal definition). In this thesis, we will not make use of such functions.

Note that any density f (corresponding to any probability measure ν) satisfies

$$\int_{x \in X} |f(x)| \mu(\mathrm{d}x) = \int_{x \in X} f(x) \mu(\mathrm{d}x) = \nu(X) = 1 < +\infty.$$

Therefore all densities belong to $L^1(\mu)$.

We now define the notion of *invariant measure*. This notion is crucial for many of the core contributions in this thesis (see Chapter 3 and Chapter 5).

Definition D.6 (Invariant measure under function)**.** Let $(X, \mathcal{A})$ be a measurable space, and let $f : X \to X$ be a measurable function. A probability measure $\mu : \mathcal{A} \to [0, 1]$ is said to be *invariant under f* if the pushforward probability measure $f\#\mu$ is equal to the original probability measure μ.

In the context of dynamical systems (see Chapter 2), an invariant measure captures the idea of a *stationary probability measure* of a *deterministic* system (see Definition 2.1): if μ describes the initial probability distribution of states $x \in X$, then this distribution remains unchanged as the system evolves over time. The following fundamental result ensures the existence of such a measure.

Theorem D.7 (Krylov–Bogolyubov theorem)**.** *Let $(X, \mathcal{T})$ be a compact and metrizable topological space, and let $\sigma(\mathcal{T})$ be the Borel σ-algebra for $\mathcal{T}$. Then, if f is continuous, there exists an invariant measure under f.*

We now define Markov kernels, defined as a transition kernel for which the measures $\tau(\cdot|x)$ are probability measures.

Definition D.8 (Markov kernel)**.** Let $(X, \mathcal{A})$ and $(Y, \mathcal{B})$ be two measurable spaces. A transition kernel $\tau : \mathcal{B} \times X \to [0,1]$ is called a *Markov kernel* from $(X, \mathcal{A})$ to $(Y, \mathcal{B})$ if, for all $x \in X$, the map $B \mapsto \tau(B|x)$ is a probability measure.

In the same fashion as in the case of functions, the pushforward measure $\tau\#\mu$ also remains a probability measure.

Definition D.9 (Invariant measure under kernel)**.** Let $(X, \mathcal{A})$ be a measurable space, and let $\tau : \mathcal{A} \times X$ be a Markov kernel from $(X, \mathcal{A})$ to itself. A probability measure $\mu : \mathcal{A} \to [0,1]$ is said to be invariant under τ if the pushforward probability measure $\tau\#\mu$ is equal to the original probability measure μ.

Note that, similarly as with functions, in the context of dynamical systems (see Chapter 2), an invariant measure is the stationary probability measure of a *stochastic* system (see Definition 2.9).

D.2 Distances between probability measures

We conclude this chapter by introducing notions of *distances* between probability measures. In particular, we will define two of them: the *Total Variation* and the *Kantorovich* distances. These notions will be crucial in the development and analysis of the so-called *Cantor-Kantorovich* distance between Markov models in Chapter 6. This section is based on [Vil09].

Remark D.10. Given a measurable space $(X, \mathcal{A})$, let $\mathcal{P}(X, \mathcal{A})$ be the set of all probability measures defined on $(X, \mathcal{A})$. Note that all distances $\mathsf{D} : \mathcal{P}(X, \mathcal{A}) \times \mathcal{P}(X, \mathcal{A}) \to \mathbb{R}_{\geq 0}$ defined in this chapter are proper distances (in the sense of Definition A.7), and therefore define a metric space $(\mathcal{P}(X, \mathcal{A}), \mathsf{D})$.

We start by defining the Total Variation distance between two probability distributions.

Definition D.11 (Total Variation distance)**.** Let $(X, \mathcal{A})$ be a measurable space, and let μ and ν be two probability measures on $(X, \mathcal{A})$. The Total Variation distance (or *TV* distance for short) between μ and ν is given by

$$\mathsf{TV}(\mu, \nu) := \sup_{A \in \mathcal{A}} |\mu(A) - \nu(A)|.$$

In intuitive words, the TV distance measures the maximum difference between the probabilities that μ and ν assign to the same event. The following proposition gives an alternative definition of the TV distance, involving probability density functions.

Proposition D.12. *Let $(X, \mathcal{A})$ be a measurable space, and let ν_1, ν_2 and μ be probability measures on $(X, \mathcal{A})$ such that $\nu_1 \ll \mu$ and $\nu_2 \ll \mu$. Let $f_1 = d\nu_1/d\mu$ and $f_2 = d\nu_2/d\mu$ be the densities of ν_1 and ν_2, respectively. Then it holds that*

$$\mathsf{TV}(\nu_1, \nu_2) = \frac{1}{2} \int_{x \in X} |f_1(x) - f_2(x)| \mu(dx) = \frac{1}{2} \|f_1 - f_2\|_{L^1(\mu)}.$$

If, in addition X is finite, then the TV distance takes the following simpler form.[1]

Corollary D.13. *Let X be a finite set, and let μ and ν be two probability measures on $(X, 2^X)$. Then it holds that*

$$\mathsf{TV}(\mu, \nu) = \frac{1}{2} \sum_{x \in X} |\mu(x) - \nu(x)|. \tag{D.1}$$

We conclude this section by defining the *Kantorovich distance*. As we will see, this definition requires that (X, D) is an underlying metric space. Because it directly depends on D, we call this distance the D-Kantorovich distance.

Definition D.14 (D-Kantorovich distance)**.** Let $(X, \mathcal{A})$ be a measurable space, and $\mathsf{D} : X \times X \to \mathbb{R}_{\geq 0}$ be a distance on X. Let μ and ν be two probability measures on $(X, \mathcal{A})$. The D-*Kantorovich distance* between μ and ν is given by

$$\mathsf{K}_{\mathsf{D}}(\mu, \nu) := \inf_{\gamma \in \mathcal{C}(\mu, \nu)} \int_{x,y \in X} \mathsf{D}(x, y) \gamma(\mathrm{d}x \times \mathrm{d}y).$$

It follows from the definition of the D-Kantorovich metric that, if X is finite, then computing this distance amounts to solving the following linear program

$$\mathsf{K}_{\mathsf{D}}(\mu, \nu) = \min_{c : X \times X \to [0,1]} \sum_{x,y \in X} \mathsf{D}(x, y) c(x, y) \text{ such that} \tag{D.2}$$

$$\forall x, y \in X, \quad c(x, y) \geq 0 \tag{D.3}$$

[1] We recall that, when X is finite, the only measurable space is $(X, 2^X)$ (see Proposition B.8).

$$\begin{aligned} \forall x \in X, \quad & \sum_{y \in X} c(x,y) = \mu(x), \\ \forall y \in X, \quad & \sum_{x \in X} c(x,y) = \nu(y). \end{aligned} \tag{D.4}$$

Remark D.15. As we can see in the formulation given by (D.2), (D.3), (D.4), computing the D-Kantorovich distance amounts to solving an *optimal transport problem* described as follows:

- *locations* are events $x \in X$,
- *supplies* are probabilities $\mu(x)$ for all $x \in X$,
- *demands* are probabilities $\nu(y)$ for all $y \in X$,
- the *marginal cost* of moving mass from x to y is the underlying distance $\mathsf{D}(x,y)$.

The D-Kantorovich distance between μ and ν is the minimal cost to satisfy the demand with the supply. More information and intuition is given in the textbook [Vil09], and an illustration is provided in Figure D.1.

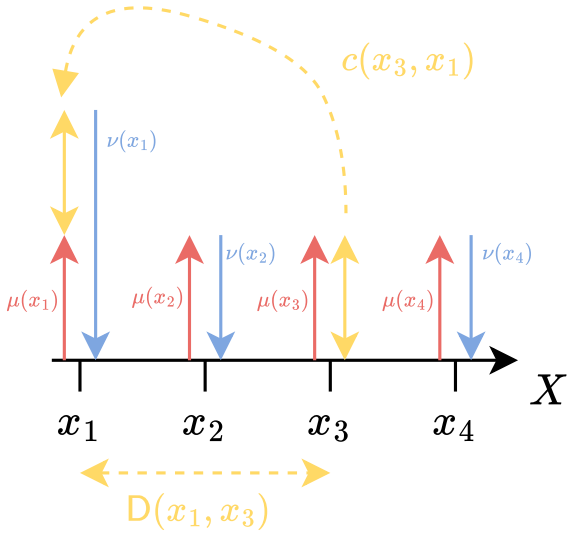


Fig. D.1 Optimal transport problem interpretation of the D-Kantorovich metric. Suppose that the sample space $X = \{x_1, x_2, x_3, x_4\}$. Let μ be such that $\mu(x_1) = \mu(x_2) = \mu(x_3) = \mu(x_4) = 1/4$, and let ν be defined as $\nu(x_1) = 1/2$, $\nu(x_2) = \nu(x_4) = 1/4$, and $\nu(x_3) = 0$. The quantities $\mu(x)$ are the supplies (in red) and $\nu(x)$ are the demands (in blue). The cheapest way of satisfying the demand with the supplies is to move $c(x_3, x_1) = 1/4$ from x_3 to x_1 (in yellow), with marginal cost $\mathsf{D}(x_1, x_3)$. The total cost is the D-Kantorovich distance, i.e. $\mathsf{K}_{\mathsf{D}}(\mu, \nu) = \mathsf{D}(x_1, x_3)/4$.

E

Computational complexity

This appendix introduces the foundational concepts of *computational complexity*, drawing primarily from [CLRS09]. We begin by formalising the notion of *problem*, distinguishing between *decision problems* (binary solutions) and *counting problems* (natural number solutions). Using the language framework introduced in Chapter 1, we define key complexity classes, including P and NP, languages that are respectively *decidable* and *verifiable* in polynomial time. Central to this framework is the notion of *polynomial-time reducibility*, which establishes the relative difficulty of problems and gives rise to NP-hard and NP-complete problems. For counting problems, we introduce the class #P, capturing functions that *count* the number of valid certificates for a given instance. We explore the relationship between NP and #P, highlighting how counting problems generalize decision problems and exhibit greater computational hardness. The chapter concludes with definitions of #P-hardness and #P-completeness, extending the complexity hierarchy to functions beyond binary decisions, and introducing concepts that is be central in Chapter 6.

E.1 Basics

Computational complexity deals with the challenge of characterising the complexity of solving a problem, defined as follows.

Definition E.1 (Problem)**.** An *abstract problem* is a binary relation on a set I of *instances* and a set S of *solutions*. A *decision problem* is an abstract problem with $S = \{0, 1\}$. A *counting problem* is an abstract problem with $S = \mathbb{N}$.

For example, consider the problem of deciding whether a natural number is *even* (associated with 0) or *odd* (associated with 1). Formally, this problem is defined as

$$P = \{(n, n \bmod 2)\}_{n \in \mathbb{N}},$$

with $I = \mathbb{N}$ and $S = \{0, 1\}$.

Now, in order for a computer to be able to solve such a problem, the input should be *encoded*, e.g. in binary or unary. The problem becomes

$$P' = \{e(n), n \bmod 2\}_{n \in \mathbb{N}}$$

where $e : \mathbb{N} \to \{0, 1\}^\star$ is an encoding function.[1] For example, *unary* and *binary* encoding (respectively noted e_1 and e_2) are such that

$$\begin{aligned} e_1(1) &= (1), & e_2(1) &= (1), \\ e_1(2) &= (1, 1), & e_2(2) &= (1, 0), \\ e_1(3) &= (1, 1, 1), & e_2(3) &= (1, 1), \\ e_1(4) &= (1, 1, 1, 1), & e_2(4) &= (1, 0, 0). \end{aligned}$$

Such problems always have the set of *strings* $\{0, 1\}^\star$ as input set and are called *concrete problems*.

Definition E.2 (Concrete problem)**.** A *concrete* problem is an abstract problem whose $I = \{0, 1\}^\star$.

All the complexity results that follow depend on the encoding of an abstract problem as a concrete problem (e.g. in unary or binary). In this thesis, we discard the expensive unary encoding and focus our analysis on binary encoding.[2]

[1] We recall that for any set A, $A^\star$ refers to the *Kleene star* of A, defined in Chapter 1.

[2] Note that any base strictly greater than 1 results in equivalent computational complexity results. However, there exists concrete problems encoded in unary that can be solved in polynomial time, but whose corresponding binary problem necessitates non-polynomial algorithms.

E.2 Decision problems

We start by restricting our analysis to concrete decision problems ($I = \{0,1\}^\star$, $S = \{0,1\}$). One can view any concrete problem as a language in $\{0,1\}^\star$ (in the sense of Chapter 1). Let P be a concrete decision problem. The language of P is defined as

$$L = \{x \in \{0,1\}^\star : 1 \in P(x)\}.$$

With computers, problems are solved with *algorithms*, defined as a function $A : \{0,1\}^\star \to S$. An algorithm A *accepts* an input $x \in \{0,1\}^\star$ if $A(x) = 1$. An algorithm A *rejects* a string if $A(x) = 0$. We now define the central notion of *decided language*.

Definition E.3 (Decided language)**.** Let $A : \{0,1\}^\star \to \{0,1\}$ be an algorithm. A language $L \subseteq \{0,1\}^\star$ is *decided* by A if, for all $x \in \{0,1\}^\star$,

1. if $x \in L$, then $A(x) = 1$,
2. if $x \notin L$, then $A(x) = 0$.

Moreover we say that L is *decided in polynomial time* by A if there exists $c \in \mathbb{N}$ such that, for any $x \in \{0,1\}^\star$ of length $|x| = k$,

1. if $x \in L$, A accepts x in time $\mathcal{O}(k^c)$, and
2. if $x \notin L$, A rejects x in time $\mathcal{O}(k^c)$.

We are now able to define the class of problems that can be decided in polynomial time.

Definition E.4 (P class of problems)**.** The complexity class of *polynomial problems* is defined as the set of all languages $L \subseteq \{0,1\}^\star$ such that there exists an algorithm A that decides L in polynomial time, and is denoted P.

We will now define the class of problems that can be *verified* in polynomial time, called NP problems. A *verification algorithm* is a function $V : \{0,1\}^\star \times \{0,1\}^\star \to \{0,1\}$. The second element of the domain in called the *certificate*. Let $T : \mathbb{N} \to \mathbb{N}$. We say that a verification algorithm V runs in time $\mathcal{O}(T(k))$ if, for any instance $x \in \{0,1\}^\star$ of length $k = |x|$ and any certificate $y \in \{0,1\}^\star$, the computation of $V(x,y)$ terminates within $\mathcal{O}(T(k))$ steps.

Definition E.5 (Verified language)**.** Let $V : \{0,1\}^\star \times \{0,1\}^\star \to \{0,1\}$ be a verification algorithm. A language $L \subseteq \{0,1\}^\star$ is said to be verified by V if

1. if $x \in L$, there exists $y \in \{0,1\}^\star$ such that $V(x,y) = 1$,
2. if $x \notin L$, then for all $y \in \{0,1\}^\star$, $V(x,y) = 0$.

Moreover the language is *verified in polynomial time* by V if V runs in polynomial time and if there exists $c \in \mathbb{N}$ such that

$$L = \big\{x \in \{0,1\}^\star : \text{there exists } y \in \{0,1\}^\star \text{ with } |y| = \mathcal{O}(|x|^c) \\ \text{such that } V(x,y) = 1\big\}.$$

We are now able to define the class of problems that can be verified in polynomial time.

Definition E.6 (NP class of problems)**.** The complexity class of *nondeterministic polynomial problems* is defined as the set of all languages $L \subseteq \{0,1\}^\star$ such that there exists a verification algorithm V that verifies L in polynomial time, and is denoted NP.

It is straightforward that P $\subseteq$ NP. However, the question whether P $\subset$ NP, thereby making NP problems strictly "harder" than P problems, is still open.

It is useful to be able to characterise problems as "as hard as" NP problems. Such problems are known as *NP-hard* problems, formalised below.

Definition E.7 (Polynomial-time reducibility)**.** A language $L_1 \subseteq \{0,1\}^\star$ is *polynomial-time reducible* to a language $L_2 \subseteq \{0,1\}^\star$ if there exists a polynomial-time computable function $r : \{0,1\}^\star \to \{0,1\}^\star$ such that, for all $x \in \{0,1\}^\star$, $x \in L_1$ if and only if $r(x) \in L_2$.

The function $r : \{0,1\}^\star \to \{0,1\}^\star$ is called the *reduction function*, and a polynomial-time algorithm R that computes r is a *reduction algorithm*.

Definition E.8 (NP-hard and NP-complete)**.** A language $L \subseteq \{0,1\}^\star$ is said to be NP-hard if, for all $L' \in$ NP, L' is polynomial-time reducible to L. If, in addition, $L \in$ NP, then L is said to be *NP-complete*.

NP-complete problems are the "hardest problems" within NP. To show that a problem P is NP-hard, we show that an NP-complete problem is polynomial-time reducible to P.

E.3 Counting problems

We now restrict our analysis to *concrete counting problems*, i.e. problems such that $I = \{0,1\}^\star$ and $S = \mathbb{N}$ (see Section E.1). Note that we restrict

to problems with only one answer (i.e. relations that can be formalised as functions).

In the following, we will define the #P class of counting problems.

Definition E.9 (#P class of functions)**.** Let $f : \{0,1\}^\star \to \mathbb{N}$ be a counting problem. This function belongs to #P if there exists a polynomial verification algorithm V and $c \in \mathbb{N}$ such that, for all $x \in \{0,1\}^\star$,

$$f(x) = |\{y \in \{0,1\}^\star : |y| \in \mathcal{O}(|x|^c) \text{ and } V(x,y) = 1\}|\,.$$

We can make a direct link between NP and #P. The latter class can be understood as the counting version of NP. For example, the *subset sum problem* is the decision problem stated as

"Given a list of integers, is there a subset of this list that adds up to zero?"

The corresponding counting problem is

"Given a list of integers, how many subsets of this list add up to zero?"

More generally, every counting problem $f : \{0,1\}^\star \to \mathbb{N} \in$ #P has a corresponding decision problem

$$g : \{0,1\}^\star \to \{0,1\} : x \mapsto \begin{cases} 1 & \text{if } f(x) > 0 \\ 0 & \text{otherwise.} \end{cases}$$

Therefore, any decision problem in NP can be reduced to a counting problem in #P, making #P "harder" than NP.

The concepts of *hardness* and *completeness* (as defined in Definition E.8 for NP decision problems) extend naturally to #P counting problems.

Definition E.10 (Polynomial-time reducibility (for functions))**.** A function $f_1 : \{0,1\}^\star \to \mathbb{N}$ is polynomial-time reducible to a function $f_2 : \{0,1\}^\star \to \mathbb{N}$ if there exists a polynomial-time computable function $r : \{0,1\}^\star \to \{0,1\}^\star$ such that, for all $x \in \{0,1\}^\star$, $f_1(x) = f_2(r(x))$.

Definition E.11 (#P-hard and #P-complete)**.** A function $f : \{0,1\}^\star \to \mathbb{N}$ is said to be *#P-hard* if, for all $f' : \{0,1\}^\star \to \mathbb{N} \in$ #P, f' is polynomial-time reducible to f. A function $f : \{0,1\}^\star \to \mathbb{N}$ is said to be *#P-complete* if

1. $f \in$ #P,
2. f is #P-hard.

The following formally states how to prove that a function is #P-hard.

Proposition E.12. *Let $f : \{0,1\}^\star \to \mathbb{N}$ be a function. If there exists a #P-complete function $g : \{0,1\}^\star \to \mathbb{N}$ such that g is polynomial-time reducible to f, then f is #P-hard.*

Remark E.13. Note that the notion of #P-hardness extends to functions $f : \{0,1\}^\star \to S$ where S is not necessarily the natural numbers $\mathbb{N}$. We say that such a function is #P-hard if there exists a #P-complete function $g : \{0,1\}^\star \to \mathbb{N}$, a polynomial-time computable function $h : \{0,1\}^\star \to S$, and a polynomial-time computable function $r : \{0,1\}^\star \to \{0,1\}^\star$ such that $f \circ r = h \circ g$.

In Chapter 6, we show that, if P $\neq$ NP, one cannot compute the *Cantor-Kantorovich distance* in polynomial time by proving that it is #P-hard. We will also demonstrate a similar result for a particular approximation scheme for this distance.

Bibliography

[ABK14] Shaull Almagor, Udi Boker, and Orna Kupferman. *Discounting in LTL*, pages 424–439. Springer Berlin Heidelberg, 2014. doi:10.1007/978-3-642-54862-8_37.

[AJP24] A. M. Annaswamy, K. H. Johansson, and G. Pappas. Control for societal-scale challenges: Road map 2030. *IEEE Control Systems*, 44(3):30–32, June 2024. doi:10.1109/mcs.2024.3382376.

[AKLP10] Alessandro Abate, Joost-Pieter Katoen, John Lygeros, and Maria Prandini. Approximate model checking of stochastic hybrid systems. *European Journal of Control*, 16(6):624–641, January 2010. doi:10.3166/ejc.16.624-641.

[Ala22] Mazen Alamir. Learning against uncertainty in control engineering. *Annual Reviews in Control*, 53:19–29, 2022. doi:10.1016/j.arcontrol.2022.03.007.

[Alu15] Rajeev Alur. *Principles of Cyber-Physical Systems*. MIT Press, 2015.

[ATH+23] Anil Alan, Andrew J. Taylor, Chaozhe R. He, Aaron D. Ames, and Gábor Orosz. Control barrier functions and input-to-state safety with application to automated vehicles. *IEEE Transactions on Control Systems Technology*, 31(6):2744–2759, November 2023. doi:10.1109/tcst.2023.3286090.

[BA17] Gaoang Bian and Alessandro Abate. *On the Relationship Between Bisimulation and Trace Equivalence in an Approximate Probabilistic Context*, pages 321–337. Springer Berlin Heidelberg, 2017. doi:10.1007/978-3-662-54458-7_19.

[BAJ25] Adrien Banse, Alessandro Abate, and Raphaël M. Jungers. Comparing labeled markov chains: A cantor-kantorovich approach, 2025. doi:10.48550/ARXIV.2511.18103.

[BDL+25] Adrien Banse, Giannis Delimpaltadakis, Luca Laurenti, Manuel Mazo, and Raphaël M. Jungers. Memory-dependent abstractions of stochastic systems through the lens of transfer operators. In *Proceedings of the 28th ACM International Conference on Hybrid Systems: Computation and Control*, HSCC '25, pages 1–12. ACM, May 2025. doi:10.1145/3716863.3718039.

[BDMJJ25] Adrien Banse, Giannis Delimpaltadakis, Manuel Mazo Jr., and Raphaël M. Jungers. Smart abstraction of stochastic systems using memory. In *44th Benelux Meeting on Systems and Control*, 2025.

[BGK21] Philipp Braun, Lars Grüne, and Christopher M. Kellett. *(In-)Stability of Differential Inclusions: Notions, Equivalences, and Lyapunov-like Characterizations*. Springer International Publishing, 2021. doi:10.1007/978-3-030-76317-6.

[BGM+23] Arnab Bhattacharyya, Sutanu Gayen, Kuldeep S. Meel, Dimitrios Myrisiotis, A. Pavan, and N. V. Vinodchandran. On approximating total variation distance. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*, IJCAI-2023. International Joint Conferences on Artificial Intelligence Organization, August 2023. doi:10.24963/ijcai.2023/387.

[BGM+24] Arnab Bhattacharyya, Sutanu Gayen, Kuldeep S. Meel, Dimitrios Myrisiotis, A. Pavan, and N. V. Vinodchandran. Total variation distance meets probabilistic inference. In *Forty-first International Conference on Machine Learning*, 2024. URL: https://openreview.net/forum?id=6OSLjErBhh.

[Bil12] Patrick Billingsley. *Probability and measure*. Wiley, 2012.

[BK08] Christel Baier and Joost-Pieter Katoen. *Principles of Model Checking*. The MIT Press, 2008.

[BLW13] Michael Benedikt, Rastislav Lenhardt, and James Worrell. *LTL Model Checking of Interval Markov Chains*, pages 32–46. Springer Berlin Heidelberg, 2013. doi:10.1007/978-3-642-36742-7_3.

[BRAJ23a] Adrien Banse, Licio Romao, Alessandro Abate, and Raphaël M. Jungers. Data-driven abstractions via adaptive refinements and a kantorovich metric. In *2023 62nd IEEE Conference on Decision and Control (CDC)*, pages 6038–6043. IEEE, December 2023. doi:10.1109/cdc49753.2023.10383513.

[BRAJ23b] Adrien Banse, Licio Romao, Alessandro Abate, and Raphaël M. Jungers. Data-driven memory-dependent abstractions of dynamical systems. In Nikolai Matni, Manfred Morari, and George J. Pappas, editors, *Proceedings of The 5th Annual Learning for Dynamics and Control Conference*, volume 211 of *Proceedings of Machine Learning Research*, pages 891–902. PMLR, 15–16 Jun 2023. URL: https://proceedings.mlr.press/v211/banse23a.html.

[BRAJ24] Adrien Banse, Licio Romao, Alessandro Abate, and Raphaël M. Jungers. Smart abstraction of dynamical systems via a cantor-kantorovich distance between markov chains. In *43rd Benelux Meeting on Systems and Control*, 2024.

[BRAJ25] Adrien Banse, Licio Romao, Alessandro Abate, and Raphaël M. Jungers. Data-driven memory-dependent abstractions of dynamical systems via a cantor–kantorovich metric. *IEEE Transactions on Automatic Control*, 70(12):8092–8103, December 2025. doi:10.1109/tac.2025.3582534.

[BRJ24] Adrien Banse, Venkatraman Renganathan, and Raphaël M. Jungers. A cantor-kantorovich metric between markov decision processes with application to transfer learning. In *26th International Symposium on Mathematical Theory of Networks and Systems (Cambridge, UK).*, 2024.

[BV04] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[BWJ22a] Adrien Banse, Zheming Wang, and Raphaël M. Jungers. Black-box stability analysis of hybrid systems with sample-

based multiple lyapunov functions. In *2022 IEEE 61st Conference on Decision and Control (CDC)*, pages 7284–7289. IEEE, December 2022. doi:10.1109/cdc51059.2022.9993008.

[BWJ22b] Adrien Banse, Zheming Wang, and Raphaël M. Jungers. Learning stability guarantees for data-driven constrained switching linear systems. In *41st Benelux Meeting on Systems and Control*, 2022.

[BWJ22c] Adrien Banse, Zheming Wang, and Raphaël M. Jungers. Learning stability guarantees for data-driven constrained switching linear systems. In *25th International Symposium on Mathematical Theory of Networks and Systems (Bayreuth, Germany)*, 2022.

[BWJ24] Adrien Banse, Zheming Wang, and Raphaël M. Jungers. Learning stability guarantees for constrained switching linear systems from noisy observations. *Nonlinear Analysis: Hybrid Systems*, 51:101425, February 2024. doi:10.1016/j.nahs.2023.101425.

[BYAG17] Calin Belta, Boyan Yordanov, and Ebru Aydin Gol. *Formal Methods for Discrete-Time Dynamical Systems*. Springer International Publishing, 2017. doi:10.1007/978-3-319-50763-7.

[CBLJ24] Julien Calbert, Adrien Banse, Benoît Legat, and Raphaël Jungers. Dionysos.jl: a modular platform for smart symbolic control. *JuliaCon Proceedings*, 6(66):160, December 2024. doi:10.21105/jcon.00160.

[CEJ24] Julien Calbert, Lucas N. Egidio, and Raphaël M. Jungers. Smart abstraction based on iterative cover and non-uniform cells. *IEEE Control Systems Letters*, 8:2301–2306, 2024. doi:10.1109/lcsys.2024.3409106.

[CFH03] Feng-Sheng Chen, Chih-Ming Fu, and Chung-Lin Huang. Hand gesture recognition using a real-time tracking method and hidden markov models. *Image and Vision Computing*, 21(8):745–758, 2003. URL: https://www.sciencedirect.com/science/article/pii/S0262885603000702, doi:10.1016/S0262-8856(03)00070-2.

[CG19] Marco C. Campi and Simone Garatti. *Introduction to the Scenario Approach*. Society for Industrial and Applied Mathematics, 2019.

[CGJ24] Julien Calbert, Antoine Girard, and Raphaël M. Jungers. Classification of simulation relations for symbolic control, 2024. doi:10.48550/ARXIV.2410.06083.

[Chu89] G. Churchill. Stochastic models for heterogeneous dna sequences. *Bulletin of Mathematical Biology*, 51(1):79–94, 1989. doi:10.1016/s0092-8240(89)80049-7.

[CK14] Taolue Chen and Stefan Kiefer. On the total variation distance of labelled markov chains, 2014. doi:10.48550/ARXIV.1405.2852.

[CLEJ21] Julien Calbert, Benoît Legat, Lucas N. Egidio, and Raphaël M. Jungers. Alternating simulation on hierarchical abstractions. In *2021 60th IEEE Conference on Decision and Control (CDC)*, pages 593–598. IEEE, December 2021. doi:10.1109/cdc45484.2021.9683448.

[CLRS09] Thomas H. Cormen, Charles Eric Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms*. The MIT Press, 2009.

[Cop26] R. Coppola. *Abstraction Learning with Guarantees: Data-Driven Approaches to Symbolic Control and Verification*. Dissertation (tu delft), Delft University of Technology, 2026. doi:10.4233/uuid:d5fe16ad-231a-4c62-b717-9f09cb9b2a30.

[CPLM24] Rudi Coppola, Andrea Peruffo, Lars Lindemann, and Manuel Mazo. Scenario approach and conformal prediction for verification of unknown systems via data-driven abstractions. In *2024 European Control Conference (ECC)*, pages 558–563. IEEE, June 2024. doi:10.23919/ecc64448.2024.10590827.

[CPM23] Rudi Coppola, Andrea Peruffo, and Manuel Mazo. Data-driven abstractions for verification of linear systems. *IEEE Control Systems Letters*, pages 1–1, 2023. doi:10.1109/lcsys.2023.3288731.

[CSBD24] Siddharth Chandak, Pratik Shah, Vivek S. Borkar, and Parth Dodhia. Reinforcement learning in non-markovian environments. *Systems & Control Letters*, 185:105751, March 2024. doi:10.1016/j.sysconle.2024.105751.

[CvBW12] Di Chen, Franck van Breugel, and James Worrell. *On the Complexity of Computing Probabilistic Bisimilarity*, pages 437–451. Springer Berlin Heidelberg, 2012. doi:10.1007/978-3-642-28729-9_29.

[DD09] Yuxin Deng and Wenjie Du. The kantorovich metric in computer science: A brief survey. *Electronic Notes in Theoretical Computer Science*, 253(3):73–82, November 2009. doi:10.1016/j.entcs.2009.10.006.

[DEKM98] Richard Durbin, Sean R. Eddy, Anders Krogh, and Graeme Mitchison. *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. Cambridge University Press, April 1998. doi:10.1017/cbo9780511790492.

[DG25] Giannis Delimpaltadakis and Gabriel Gleizer. An information theory of finite abstractions and their fundamental scalability limits, 2025. doi:10.48550/ARXIV.2512.03977.

[DGJP04] Josée Desharnais, Vineet Gupta, Radha Jagadeesan, and Prakash Panangaden. Metrics for labelled markov processes. *Theoretical Computer Science*, 318(3):323–354, June 2004. doi:10.1016/j.tcs.2003.09.013.

[DGV13] Giuseppe De Giacomo and Moshe Y. Vardi. Linear temporal logic and linear dynamic logic on finite traces. In *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*, IJCAI '13, pages 854–860. AAAI Press, 2013.

[DHC22] Maxence Dutreix, Jeongmin Huh, and Samuel Coogan. Abstraction-based synthesis for stochastic systems with omega-regular objectives. *Nonlinear Analysis: Hybrid Systems*, 45:101204, August 2022. doi:10.1016/j.nahs.2022.101204.

[DHKP16] Przemysław Daca, Thomas A. Henzinger, Jan Křetínský, and Tatjana Petrov. Linear distances between markov chains, 2016. doi:10.48550/ARXIV.1605.00186.

[DLM24] Giannis Delimpaltadakis, Luca Laurenti, and Manuel Mazo. Formal analysis of the sampling behavior of stochastic event-triggered control. *IEEE Transactions on Automatic Control*, 69(7):4491–4505, July 2024. doi:10.1109/tac.2023.3333748.

[DLMJL23] Giannis Delimpaltadakis, Morteza Lahijanian, Manuel Mazo Jr., and Luca Laurenti. Interval markov decision processes with continuous action-spaces. In *Proceedings of the 26th ACM International Conference on Hybrid Systems: Computation and Control*, HSCC '23, pages 1–10. ACM, May 2023. doi:10.1145/3575870.3587117.

[ELJ22] Lucas N. Egidio, Thiago Alves Lima, and Raphaël M. Jungers. State-feedback abstractions for optimal control of piecewise-affine systems. In *2022 IEEE 61st Conference on Decision and Control (CDC)*, pages 7455–7460. IEEE, December 2022. doi:10.1109/cdc51059.2022.9992495.

[EZSA13] Sadegh Esmaeil Zadeh Soudjani and Alessandro Abate. Adaptive and sequential gridding procedures for the abstraction and verification of stochastic processes. *SIAM Journal on Applied Dynamical Systems*, 12(2):921–956, January 2013. doi:10.1137/120871456.

[FLL24] Weiming Feng, Liqiang Liu, and Tianren Liu. *On Deterministically Approximating Total Variation Distance*, pages 1766–1791. Society for Industrial and Applied Mathematics, January 2024. arXiv:https://epubs.siam.org/doi/pdf/10.1137/1.9781611977912.70, doi:10.1137/1.9781611977912.70.

[Fog02] N. Pytheas Fogg. *Substitutions in Dynamics, Arithmetics, and Combinatorics*. Springer, 2002. doi:10.1007/b13861.

[FPLL24] Eduardo Figueiredo, Andrea Patane, Morteza Lahijanian, and Luca Laurenti. Uncertainty propagation in stochastic systems via mixture models with error quantification. In *2024 IEEE 63rd Conference on Decision and Control (CDC)*, pages 328–335. IEEE, December 2024. doi:10.1109/cdc56724.2024.10886416.

[GCW+23] Jiacheng Guo, Minshuo Chen, Huan Wang, Caiming Xiong, Mengdi Wang, and Yu Bai. Sample-efficient learning of

pomdps with multiple observations in hindsight, 2023. doi:10.48550/ARXIV.2307.02884.

[GCW+24] Jiacheng Guo, Minshuo Chen, Huan Wang, Caiming Xiong, Mengdi Wang, and Yu Bai. Sample-efficient learning of POMDPs with multiple observations in hindsight. In *The Twelfth International Conference on Learning Representations*, 2024. URL: https://openreview.net/forum?id=1hsVvgW0rU.

[GLD00] Robert Givan, Sonia Leach, and Thomas Dean. Bounded-parameter markov decision processes. *Artificial Intelligence*, 122(1–2):71–109, September 2000. doi:10.1016/s0004-3702(00)00047-3.

[GM23] Gabriel de Albuquerque Gleizer and Manuel Mazo. Chaos and order in event-triggered control. *IEEE Transactions on Automatic Control*, 68(11):6541–6556, November 2023. doi:10.1109/tac.2023.3242334.

[GP13] I. S. Grant and W. R. Phillips. *Electromagnetism*. Manchester Physics Series. Wiley, 2013. URL: https://books.google.be/books?id=Wi073n5G-8oC.

[GPT10] A. Girard, G. Pola, and P. Tabuada. Approximately bisimilar symbolic models for incrementally stable switched systems. *IEEE Transactions on Automatic Control*, 55(1):116–126, January 2010. doi:10.1109/tac.2009.2034922.

[GVF22] Javier García, Álvaro Visús, and Fernando Fernández. A taxonomy for similarity metrics between markov decision processes. *Machine Learning*, 111(11):4217–4247, October 2022. doi:10.1007/s10994-022-06242-4.

[Hai06] Martin Hairer. Ergodic properties of markov processes. 2006. URL: https://api.semanticscholar.org/CorpusID:51990013.

[Hal50] Paul R. Halmos. *Measure Theory*. Springer New York, 1950. doi:10.1007/978-1-4684-9440-2.

[HJMS02] Thomas A. Henzinger, Ranjit Jhala, Rupak Majumdar, and Grégoire Sutre. Lazy abstraction. In *Proceedings of the 29th*

ACM SIGPLAN-SIGACT symposium on Principles of programming languages, POPL02, pages 58–70. ACM, January 2002. doi:10.1145/503272.503279.

[HJT12] W. P. M. H. Heemels, K. H. Johansson, and P. Tabuada. An introduction to event-triggered and self-triggered control. In *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*, pages 3270–3285. IEEE, December 2012. doi:10.1109/cdc.2012.6425820.

[HK71] Kenneth Hoffman and Ray Kunze. *Linear Algebra*. Prentice-Hall mathematics series. Prentice-Hall, Englewood Cliffs, N.J., 1971.

[HK14] Stanislav Hencl and Pekka Koskela. *Lectures on Mappings of Finite Distortion*. Springer International Publishing, 2014. doi:10.1007/978-3-319-03173-6.

[HMMS19] Kyle Hsu, Rupak Majumdar, Kaushik Mallik, and Anne-Kathrin Schmuck. *Lazy Abstraction-Based Controller Synthesis*, pages 23–47. Springer International Publishing, 2019. doi:10.1007/978-3-030-31784-3_2.

[JBCJ25a] Adrien Janssens, Adrien Banse, Julien Calbert, and Raphaël M. Jungers. Abstractions with adaptive time-step for reachability analysis. In *44th Benelux Meeting on Systems and Control*, 2025.

[JBCJ25b] Adrien Janssens, Adrien Banse, Julien Calbert, and Raphaël M. Jungers. Agile temporal discretization for symbolic optimal control. In *2025 IEEE 64th Conference on Decision and Control (CDC)*, pages 6649–6654. IEEE, December 2025. doi:10.1109/cdc57313.2025.11312201.

[JMGLM14] Manfred Jaeger, Hua Mao, Kim Guldstrand Larsen, and Radu Mardare. *Continuity Properties of Distances for Markov Processes*, pages 297–312. Springer International Publishing, 2014. doi:10.1007/978-3-319-10696-0_24.

[Jun09] Raphaël Jungers. *The Joint Spectral Radius*. Springer Berlin Heidelberg, 2009. doi:10.1007/978-3-540-95980-9.

[KD09] Armen Der Kiureghian and Ove Ditlevsen. Aleatory or epistemic? does it matter? *Structural Safety*, 31(2):105–112, March 2009. doi:10.1016/j.strusafe.2008.06.020.

[KENSA19] Daniel Kuhn, Peyman Mohajerin Esfahani, Viet Anh Nguyen, and Soroosh Shafieezadeh-Abadeh. *Wasserstein Distributionally Robust Optimization: Theory and Applications in Machine Learning*, pages 130–166. INFORMS, October 2019. doi:10.1287/educ.2019.0198.

[Kie18] Stefan Kiefer. On Computing the Total Variation Distance of Hidden Markov Models. In Ioannis Chatzigiannakis, Christos Kaklamanis, Dániel Marx, and Donald Sannella, editors, *45th International Colloquium on Automata, Languages, and Programming (ICALP 2018)*, volume 107 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 130:1–130:13, Dagstuhl, Germany, 2018. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.ICALP.2018.130.

[KK12] Kyoung-Dae Kim and P. R. Kumar. Cyber–physical systems: A perspective at the centennial. *Proceedings of the IEEE*, 100(Special Centennial Issue):1287–1308, May 2012. doi:10.1109/jproc.2012.2189792.

[KLvHS01] Anders Krogh, Bjorn Larsson, Gunnar von Heijne, and Erik L. L. Sonnhammer. Predicting transmembrane protein topology with a hidden markov model: application to complete genomes. *Journal of Molecular Biology*, 305(3):567–580, January 2001. doi:10.1006/jmbi.2000.4315.

[KM11] I. Kontoyiannis and S. P. Meyn. Geometric ergodicity and the spectral gap of non-reversible markov chains. *Probability Theory and Related Fields*, 154(1–2):327–339, June 2011. doi:10.1007/s00440-011-0373-4.

[KNP11] Marta Kwiatkowska, Gethin Norman, and David Parker. *PRISM 4.0: Verification of Probabilistic Real-Time Systems*, pages 585–591. Springer Berlin Heidelberg, 2011. doi:10.1007/978-3-642-22110-1_47.

[LAB15] Morteza Lahijanian, Sean B. Andersson, and Calin Belta. Formal verification and synthesis for discrete-time stochastic systems. *IEEE Transactions on Automatic Control*, 60(8):2031–2045, August 2015. doi:10.1109/tac.2015.2398883.

[Lee08] Ji-Woong Lee. *Separation in Stability Analysis of Piecewise Linear Systems in Discrete Time*, pages 626–629. Springer Berlin Heidelberg, 2008. doi:10.1007/978-3-540-78929-1_50.

[Lee15] Edward Lee. The past, present and future of cyber-physical systems: A focus on models. *Sensors*, 15(3):4837–4869, February 2015. doi:10.3390/s150304837.

[LLA+21] Luca Laurenti, Morteza Lahijanian, Alessandro Abate, Luca Cardelli, and Marta Kwiatkowska. Formal and efficient synthesis for continuous-time linear stochastic hybrid processes. *IEEE Transactions on Automatic Control*, 66(1):17–32, January 2021. doi:10.1109/tac.2020.2975028.

[LM95] Douglas Lind and Brian Marcus. *An Introduction to Symbolic Dynamics and Coding*. Cambridge University Press, November 1995. doi:10.1017/cbo9780511626302.

[LPN99] Rune B. Lyngsø, Christian N. S. Pedersen, and Henrik Nielsen. Metrics and similarity measures for hidden markov models. In *Proceedings of the Seventh International Conference on Intelligent Systems for Molecular Biology*, pages 178–186. AAAI Press, 1999.

[LS91] Kim G. Larsen and Arne Skou. Bisimulation through probabilistic testing. *Information and Computation*, 94(1):1–28, September 1991. doi:10.1016/0890-5401(91)90030-6.

[LS16] Edward A. Lee and Sanjit A. Seshia. *Introduction to Embedded Systems A Cyber-Physical Systems Approach*. MIT Press, 2016.

[LSAZ22] Abolfazl Lavaei, Sadegh Soudjani, Alessandro Abate, and Majid Zamani. Automated verification and synthesis of stochastic hybrid systems: A survey. *Automatica*, 146:110617, 2022. URL: https://www.sciencedirect.com/science/article/pii/S0005109822004794, doi:10.1016/j.automatica.2022.110617.

[LSZ20] Abolfazl Lavaei, Sadegh Soudjani, and Majid Zamani. Compositional abstraction of large-scale stochastic systems: A relaxed dissipativity approach. *Nonlinear Analysis: Hybrid Systems*, 36:100880, May 2020. doi:10.1016/j.nahs.2020.100880.

[LSZ21] Abolfazl Lavaei, Sadegh Soudjani, and Majid Zamani. Compositional abstraction-based synthesis of general mdps via approximate probabilistic relations. *Nonlinear Analysis: Hybrid Systems*, 39:100991, February 2021. doi:10.1016/j.nahs.2020.100991.

[Lue64] David G. Luenberger. Observing the state of a linear system. *IEEE Transactions on Military Electronics*, 8(2):74–80, 1964. doi:10.1109/tme.1964.4323124.

[LVS12] Frank L. Lewis, Draguna L. Vrabie, and Vassilis L. Syrmos. *Optimal Control*. Wiley, January 2012. doi:10.1002/9781118122631.

[LWY+23] Boqian Li, Shiping Wen, Zheng Yan, Guanghui Wen, and Tingwen Huang. A survey on the control lyapunov function and control barrier function for nonlinear-affine control systems. *IEEE/CAA Journal of Automatica Sinica*, 10(3):584–602, March 2023. doi:10.1109/jas.2023.123075.

[Mar13] A. A. Markov. An example of statistical analysis of the text of eugene onegin illustrating the association of trials into a chain. *Bulletin de l, Acadamie Imperiale des Sciences de St. Petersburg, ser*, 6(7):153–162, 1913.

[Mat07] Hermann G. Matthies. *QUANTIFYING UNCERTAINTY: MODERN COMPUTATIONAL REPRESENTATION OF PROBABILITY AND APPLICATIONS*, pages 105–135. Springer Netherlands, 2007. doi:10.1007/978-1-4020-5656-7_4.

[May03] I. D. Mayergoyz. *Mathematical models of hysteresis and their applications*. Elsevier series in electromagnetism. Elsevier, Amsterdam ;, 1st ed. edition, 2003.

[MGW18] Pierre-Jean Meyer, Antoine Girard, and Emmanuel Witrant. Compositional abstraction and safety synthesis using overlapping symbolic models. *IEEE Transactions on Automatic Control*, 63(6):1835–1841, June 2018. doi:10.1109/tac.2017.2753039.

[MH89] E. A. Misawa and J. K. Hedrick. Nonlinear observers—a state-of-the-art survey. *Journal of Dynamic Systems, Measurement, and Control*, 111(3):344–352, September 1989. doi:10.1115/1.3153059.

[MKS+13] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning, 2013. doi:10.48550/ARXIV.1312.5602.

[ML23] Yiming Meng and Jun Liu. Robustly complete finite-state abstractions for control synthesis of stochastic systems. *IEEE Open Journal of Control Systems*, 2:235–248, 2023. doi:10.1109/ojcsys.2023.3294829.

[MLL24] Frederik Baymler Mathiesen, Morteza Lahijanian, and Luca Laurenti. Intervalmdp.jl: Accelerated value iteration for interval markov decision processes. *IFAC-PapersOnLine*, 58(11):1–6, 2024. doi:10.1016/j.ifacol.2024.07.416.

[MPL15] Kostas Margellos, Maria Prandini, and John Lygeros. On the connection between compression learning and scenario based single-stage and cascading optimization problems. *IEEE Transactions on Automatic Control*, 60(10):2716–2721, October 2015. doi:10.1109/tac.2015.2394874.

[MR99] T. Moor and J. Raisch. Supervisory control of hybrid systems within a behavioural framework. *Systems & Control Letters*, 38(3):157–166, October 1999. doi:10.1016/s0167-6911(99)00061-4.

[MRO02] Thomas Moor, Jörg Raisch, and Siu O'Young. Discrete supervisory control of hybrid systems based on l-complete approximations. *Discrete Event Dynamic Systems*, 12(1):83–107, January 2002. doi:10.1023/a:1013339920783.

[MS10] Neal Madras and Deniz Sezer. Quantitative bounds for Markov chain convergence: Wasserstein and total variation distances. *Bernoulli*, 16(3), August 2010. doi:10.3150/09-bej238.

[Mun00] James R. Munkres. *Topology*. Prentice Hall, Inc., 2000.

[NBBJ26] Léa Ninite, Adrien Banse, Guillaume Berger, and Raphaël M. Jungers. A path-complete approach for optimal control of switched systems, 2026.

[NBS+25] Mahdi Nazeri, Thom Badings, Anne-Kathrin Schmuck, Sadegh Soudjani, and Alessandro Abate. Data-driven abstraction and synthesis for stochastic systems with unknown dynamics, 2025. doi:10.48550/ARXIV.2508.15543.

[NBSA25] Mahdi Nazeri, Thom Badings, Sadegh Soudjani, and Alessandro Abate. Data-driven yet formal policy synthesis for stochastic nonlinear dynamical systems, 2025. doi:10.48550/ARXIV.2501.01191.

[NEG05] Arnab Nilim and Laurent El Ghaoui. Robust control of markov decision processes with uncertain transition matrices. *Operations Research*, 53(5):780–798, October 2005. doi:10.1287/opre.1050.0216.

[NFW13] Adam Nielsen, Konstantin Fackeldey, and Marcus Weber. On a generalized transfer operator. Technical Report 13-74, ZIB, Takustr. 7, 14195 Berlin, 2013.

[Nor97] J. R. Norris. *Markov Chains*. Cambridge University Press, February 1997. doi:10.1017/cbo9780511810633.

[PHC+22] Eric Pairet, Juan David Hernandez, Marc Carreras, Yvan Petillot, and Morteza Lahijanian. Online mapping and motion planning under uncertainty for safe navigation in unknown environments. *IEEE Transactions on Automation Science and Engineering*, 19(4):3356–3378, October 2022. doi:10.1109/tase.2021.3118737.

[PM23] Andrea Peruffo and Manuel Mazo. Data-driven abstractions with probabilistic guarantees for linear petc systems. *IEEE*

Control Systems Letters, 7:115–120, 2023. doi:10.1109/lcsys.2022.3186187.

[PP04] Dominique Perrin and Jean-Eric Pin. *Infinite Words Automata, Semigroups, Logic and Games*. Elsevier Science & Technology Books, 2004.

[PW97] Jan Willem Polderman and Jan C. Willems. *Introduction to Mathematical Systems Theory*. Springer, 1997.

[PWS+11] Jan-Hendrik Prinz, Hao Wu, Marco Sarich, Bettina Keller, Martin Senne, Martin Held, John D. Chodera, Christof Schütte, and Frank Noé. Markov models of molecular kinetics: Generation and validation. *The Journal of Chemical Physics*, 134(17), May 2011. doi:10.1063/1.3565032.

[Rab89] L. R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989. doi:10.1109/5.18626.

[RAC04] Z. Rached, F. Alajaji, and L. L. Campbell. The kullback–leibler divergence rate between markov sources. *IEEE Transactions on Information Theory*, 50(5):917–921, May 2004. doi:10.1109/tit.2004.826687.

[RB25] Samuel Rincé and Adrien Banse. Ecologits: Evaluating the environmental impacts of generative ai. *Journal of Open Source Software*, 10(111):7471, July 2025. doi:10.21105/joss.07471.

[RCJ21] Wei Ren, Julien Calbert, and Raphaël M. Jungers. Zonotope-based controller synthesis for ltl specifications. In *2021 60th IEEE Conference on Decision and Control (CDC)*, pages 580–585. IEEE, December 2021. doi:10.1109/cdc45484.2021.9683150.

[RMD17] J. Rawlings, D. Q. Mayne, and Moritz Diehl. *Model Predictive Control: Theory, Computation, and Design*. January 2017.

[Roy88] H. L. Royden. *Real Analysis*. Macmillan Publishing Company, 1988.

[RS18] Daniel Rudolf and Nikolaus Schweizer. Perturbation theory for Markov chains via Wasserstein distance. *Bernoulli*, 24(4A), November 2018. doi:10.3150/17-bej938.

[RW87] P. J. Ramadge and W. M. Wonham. Supervisory control of a class of discrete event processes. *SIAM Journal on Control and Optimization*, 25(1):206–230, January 1987. doi:10.1137/0325013.

[RW89] P. J. G. Ramadge and W. M. Wonham. The control of discrete event systems. *Proceedings of the IEEE*, 77(1):81–98, 1989. doi:10.1109/5.21072.

[SB] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning An Introduction*. A Bradford Book.

[SBB+24] Marnix Suilen, Thom Badings, Eline M. Bovy, David Parker, and Nils Jansen. *Robust Markov Decision Processes: A Place Where AI and Formal Methods Meet*, pages 126–154. Springer Nature Switzerland, November 2024. doi:10.1007/978-3-031-75778-5_7.

[Sha93] Claude E. Shannon. *Coding Theorems for a Discrete Source With a Fidelity CriterionInstitute of Radio Engineers, International Convention Record, vol. 7, 1959.*, pages 325–350. 1993. doi:10.1109/9780470544242.ch21.

[SHD01] Ch. Schütte, W. Huisinga, and P. Deuflhard. *Transfer Operator Approach to Conformational Dynamics in Biomolecular Systems*, pages 191–223. Springer Berlin Heidelberg, 2001. doi:10.1007/978-3-642-56589-2_9.

[Shi16] Albert N. Shiryaev. *Probability-1: Volume 1*. Springer New York, 2016. doi:10.1007/978-0-387-72206-1.

[SLJ+03] Satinder Singh, Michael L. Littman, Nicholas K. Jong, David Pardoe, and Peter Stone. Learning predictive state representations. In *Proceedings of the Twentieth International Conference on Machine Learning (ICML)*, pages 712–719, 2003.

[SMM24] Dániel Szekeres, Kristóf Marussy, and István Majzik. *A Lazy Abstraction Algorithm for Markov Decision Processes: Theory and Initial Evaluation*, pages 81–96. Springer Nature Switzerland, September 2024. doi:10.1007/978-3-031-70753-7_6.

[SN90] N. Sri Namachchivaya. Stochastic bifurcation. *Applied Mathematics and Computation*, 38(2):101–159, 1990. doi:10.1016/0096-3003(90)90051-4.

[SNS10] Marco Sarich, Frank Noé, and Christof Schütte. On the approximation quality of markov state models. *Multiscale Modeling and Simulation*, 8(4):1154–1177, January 2010. doi:10.1137/090764049.

[SS10] Sibylle Schwarz and Ludwig Staiger. *Topologies Refining the Cantor Topology on Xω*, pages 271–285. Springer Berlin Heidelberg, 2010. doi:10.1007/978-3-642-15240-5.

[STR15] Anne-Kathrin Schmuck, Paulo Tabuada, and Jörg Raisch. Comparing asynchronous l-complete approximations and quotient based abstractions. In *2015 54th IEEE Conference on Decision and Control (CDC)*, pages 6823–6829. IEEE, December 2015. doi:10.1109/cdc.2015.7403294.

[Tab09] Paulo Tabuada. *Verification and Control of Hybrid Systems: A Symbolic Approach*. Springer US, 2009. doi:10.1007/978-1-4419-0224-5.

[Tao10] Terence Tao. *An epsilon of room, II pages from year three of a mathematical blog*. American Mathematical Society, 2010.

[vBSW07] Franck van Breugel, Babita Sharma, and James Worrell. *Approximating a Behavioural Pseudometric Without Discount for Probabilistic Systems*, pages 123–137. Springer Berlin Heidelberg, 2007. doi:10.1007/978-3-540-71389-0_10.

[vBW05] Franck van Breugel and James Worrell. A behavioural pseudometric for probabilistic transition systems. *Theoretical Computer Science*, 331(1):115–142, 2005. Automata, Languages and Programming. URL: https://www.sciencedirect.com/science/article/pii/S0304397504006711, doi:10.1016/j.tcs.2004.09.035.

[Vil09] Cédric Villani. *Optimal Transport*. Springer Berlin Heidelberg, 2009. doi:10.1007/978-3-540-71050-9.

[WC17] Sheng-Jhih Wu and Moody T. Chu. Markov chains with memory, tensor formulation, and the dynamics of power iteration.

Applied Mathematics and Computation, 303:226–239, June 2017. doi:10.1016/j.amc.2017.01.030.

[Wil22] Michel Willem. *Functional Analysis: Fundamentals and Applications*. Springer International Publishing, 2022. doi:10.1007/978-3-031-09149-0.

[YBW15] Fanny Yang, Sivaraman Balakrishnan, and Martin J. Wainwright. Statistical and computational guarantees for the baum-welch algorithm. In *2015 53rd Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 658–665. IEEE, September 2015. doi:10.1109/allerton.2015.7447067.

[YMR20] Jung-Min Yang, Thomas Moor, and Jörg Raisch. Refinements of behavioural abstractions for the supervisory control of hybrid systems. *Discrete Event Dynamic Systems*, 30(3):533–560, April 2020. doi:10.1007/s10626-020-00315-z.

[ZFM10] Enlu Zhou, Michael C. Fu, and Steven I. Marcus. Solving continuous-state pomdps via density projection. *IEEE Transactions on Automatic Control*, 55(5):1101–1116, May 2010. doi:10.1109/tac.2010.2042005.

[ZLJZ23] Zhuangdi Zhu, Kaixiang Lin, Anil K. Jain, and Jiayu Zhou. Transfer learning in deep reinforcement learning: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(11):13344–13362, November 2023. doi:10.1109/tpami.2023.3292075.

[ZTA16] Majid Zamani, Ilya Tkachev, and Alessandro Abate. Towards scalable synthesis of stochastic control systems. *Discrete Event Dynamic Systems*, 27(2):341–369, November 2016. doi:10.1007/s10626-016-0233-6.