Abstraction Learning with Guarantees
Data-Driven Approaches to Symbolic Control and Verification

Coppola, R.

DOI
10.4233/uuid:d5fe16ad-231a-4c62-b717-9f09cb9b2a30

Publication date
2026

Document Version
Final published version

Citation (APA)
Coppola, R. (2026). *Abstraction Learning with Guarantees: Data-Driven Approaches to Symbolic Control and Verification*. [Dissertation (TU Delft), Delft University of Technology]. https://doi.org/10.4233/uuid:d5fe16ad-231a-4c62-b717-9f09cb9b2a30

Abstraction Learning with Guarantees:

Data-Driven Approaches to Symbolic Control and Verification

Abstraction Learning with Guarantees:

Data-Driven Approaches to Symbolic Control and Verification

Dissertation

for the purpose of obtaining the degree of doctor
at Delft University of Technology
by the authority of the Rector Magnificus, Prof. dr. ir. H. Bijl,
chair of the Board for Doctorates
to be defended publicly on
Tuesday 3 February 2026 at 17:30

by

Rudi COPPOLA

This dissertation has been approved by the promotors.

Composition of the doctoral committee:

Rector Magnificus,	chairperson
Dr. M. Mazo Espinosa,	Delft University of Technology, *promotor*
Dr. L. Laurenti,	Delft University of Technology, *copromotor*
Independent members:	
Prof. dr. ir. B. De Schutter	Delft University of Technology
Prof. dr. F.A. Oliehoek	Delft University of Technology
Prof. dr. R. Jungers	University of Louvain, Belgium
Dr. A. Girard	University of Paris-Saclay, France
Dr. ir. S. Haesart,	Eindhoven University of Technology
Prof. dr. T. Keviczky	Delft University of Technology, reserve member

Keywords: Abstraction, formal methods, data-driven, statistical learning, scenario approach, symbolic control and verification, discrete-time, deterministic systems, stochastic systems.

Cover by: Rudi Coppola, original sketch with AI-assisted enhancement.

An electronic copy of this dissertation is available at https://repository.tudelft.nl/.

To my loving family,
for always supporting me throughout this long journey far from home.

Contents

Summary **xi**

Samenvatting **xiii**

1. Introduction **1**
- 1.1. Motivation 2
- 1.2. Related Work 4
  - 1.2.1. Data-driven Abstractions of Deterministic Systems 4
  - 1.2.2. Data-driven Abstractions of Stochastic Systems 4
  - 1.2.3. Multi-resolution Abstractions 5
- 1.3. Contributions and Outline of this Dissertation 6

2. Preliminaries and Notation **9**
- 2.1. Notation 10
- 2.2. Transition Systems 10
- 2.3. System Relations 11
- 2.4. Strongest Asynchronous l-complete Abstractions 13
- 2.5. Markov Models 16
- 2.6. Scenario Theory Background 17

I. Deterministic Systems with Unknown Dynamics **19**

3. Data-driven Abstractions for Verification of Dynamical Systems **21**
- 3.1. Introduction 22
- 3.2. Notation and Modelling Framework 23
- 3.3. Sampling and Abstractions 25
  - 3.3.1. Trajectory Sampling 25
  - 3.3.2. Data-driven Abstraction: Finite-time Guarantees 27
- 3.4. Experimental Evaluation 31
  - 3.4.1. Linear Stable System 31
  - 3.4.2. Path Planning 31
  - 3.4.3. Concluding Example 32
- 3.5. Conclusions 34

4. Data-Driven Abstractions for Control Systems via Random Exploration **35**
- 4.1. Introduction 36

4.2. Notation and Modelling Framework 36
4.2.1. System Relations and SAlCA 36
4.3. Data-driven Abstractions 38
4.3.1. Constructing the Data-driven Abstraction 40
4.4. On Modelling Initial Conditions as Random 44
4.5. Beyond the Sampling Horizon 45
4.5.1. Contracting Systems 47
4.5.2. Autonomous Systems 47
4.6. Experimental Evaluation 48
4.7. Discussion and Concluding Remarks 50
4.8. Appendix 51
4.8.1. Temporal Logic Characterization 51
4.8.2. Proofs 51

5. Reinforcement Learning for Robust Ageing-Aware Control of Li-Ion Battery Systems with Data-Driven Formal Verification 55
5.1. Introduction 56
5.2. Preliminaries 58
5.2.1. Models and abstractions 58
5.2.2. Counterexample Guided Inductive Synthesis 59
5.3. Battery Management 59
5.3.1. Physics-based Cell Modeling 60
5.3.2. Desired Performance Specification 63
5.4. Proposed Approach 64
5.4.1. Reinforcement Learning 64
5.4.2. Data-driven Synthesis and Verification 65
5.5. Results 67
5.5.1. Benchmarking 67
5.5.2. Counterexample-Guided Inductive Synthesis - Results 71
5.6. Conclusion 76
5.7. Appendix 76
5.7.1. Parameters, cell quality and SOH 76

6. On Training-Conditional Conformal Prediction and Binomial Proportion Confidence Intervals 79
6.1. Introduction 80
6.2. Binomial Proportion Confidence Intervals 80
6.3. Training-conditional Conformal Prediction 81
6.4. A Special Case of Interest 83
6.4.1. A Note on Safety Verification for Dynamical Systems 87
6.5. Conclusion 88
6.6. Appendix 88
6.6.1. Numerical Validation 88

II. Stochastic Systems with Partially Known Dynamics 91

7. Enhancing Data-Driven Stochastic Control via Bundled Interval MDP 93
7.1. Introduction . . . 94
7.2. Notation and Modelling Framework . . . 94
7.3. Finite-State Abstraction . . . 95
7.3.1. State Space Discretization . . . 96
7.3.2. Actions . . . 96
7.3.3. Transition Probabilities . . . 97
7.3.4. Shortcomings and Motivating Example . . . 99
7.4. Uncertain Transition Probabilities . . . 100
7.5. PAC Probability Intervals via Sampling . . . 100
7.6. Experimental Evaluation . . . 103
7.7. Discussion and Conclusions . . . 104

III. Deterministic Systems with Known Dynamics 107

8. Multi-resolution Approximate Bisimulation Learning 109
8.1. Introduction . . . 110
8.2. Preliminaries . . . 111
8.3. Multi-resolution Bisimulation . . . 112
8.3.1. Incrementally Globally Asymptotically Stable Systems . . . 114
8.3.2. Non-Incrementally Globally Asymptotically Stable Systems . 114
8.4. Compact Abstractions . . . 116
8.5. Counterexample-Guided Multi-resolution Abstraction Synthesis . . . 117
8.5.1. Proposed Algorithm - Overview . . . 118
8.5.2. Counterexample Feedback . . . 120
8.5.3. Proposed Algorithm - In Detail . . . 120
8.5.4. Relation Templates . . . 122
8.5.5. Low Level Loop - One Shot . . . 124
8.5.6. Low Level Loop - Parallelised . . . 126
8.6. Incrementally Uniformly Bounded Systems . . . 129
8.7. Numerical Examples . . . 130
8.7.1. Linear System . . . 130
8.7.2. Compactness Comparison . . . 130
8.7.3. Non-δ-GAS System . . . 131
8.7.4. Non-Differentiable System . . . 132
8.7.5. Comparison with Tazaki et al. . . . 132
8.8. Conclusions and Discussion . . . 133
8.9. Appendix . . . 135
8.9.1. Proofs . . . 135
8.9.2. Safety and Reachability . . . 138

9. Conclusions and Research Outlook **139**
9.1. General Conclusions 140
9.2. Part I - Deterministic Systems with Unknown Dynamics 140
9.3. Part II - Stochastic Systems with Partially Known Dynamics 142
9.4. Part III - Deterministic Systems with Known Dynamics 143

Bibliography **145**

Acknowledgements **157**

Curriculum Vitæ **159**

List of Publications **161**

Summary

Modern engineering systems, ranging from autonomous vehicles to energy storage devices, are required to operate reliably under uncertainty while satisfying increasingly complex performance and safety requirements. Ensuring that such systems behave as intended is the domain of verification, while the even more ambitious goal of designing controllers that guarantee correct behaviour by construction is known as controller synthesis. Achieving these objectives is especially difficult when systems are nonlinear or only partially known.

A central paradigm to address this challenge is the use of symbolic abstractions: simplified models that preserve the essential behaviours of the underlying system while improving analytical tractability. Abstractions enable the use of automated methods for verification and controller synthesis, making it possible to reason about safety, reachability, or performance in a mathematically rigorous way. In particular, symbolic control leverages finite-state abstractions to enable automated algorithmic synthesis of controllers that come with formal correctness guarantees. Yet, traditional abstraction techniques require complete system knowledge, limiting their applicability in practical scenarios where model knowledge is scarce, while data is abundant.

This thesis investigates how to overcome this limitation by **learning abstractions directly from data** and **learning abstractions in combination with data** when partial knowledge of the dyamics is available; further, we demonstrate how such abstractions can be used for verification and control under uncertainty. The contributions are organised into three parts, addressing complementary scenarios:

- Part I considers **deterministic systems with unknown dynamics and a finite output alphabet**. By sampling initial conditions and collecting finite-length output sequences, we show how to construct finite-state abstractions suitable for verifying temporal properties and synthesising controllers. The resulting methods provide probabilistic guarantees of correctness and can extend beyond the sampling horizon when limited prior knowledge of the system is available.

- Part II addresses **systems combining known deterministic dynamics with stochastic disturbances of unknown distribution**. We propose data-driven abstractions in the form of Markov models, built from samples of the stochastic dynamics, and demonstrate how they can be used to synthesise controllers that satisfy probabilistic specifications.

- Part III focuses on **deterministic systems with known dynamics where standard abstraction techniques are computationally inefficient** or intractable. Here, we introduce an efficient synthesis algorithm that blends data and model knowledge, relying on novel approximate system relations that allow for multi-resolution abstractions with quantifiable error bounds.

Across these settings, we establish how data can be systematically leveraged to produce abstractions that are both accurate and equipped with formal, quantifiable guarantees. To place these contributions in a broader statistical learning perspective, we also investigate the role of probabilistic uncertainty quantification, comparing frameworks such as scenario theory and conformal prediction, and clarifying the limitations of recently proposed approaches.

Beyond theory, the thesis illustrates the practical impact of data-driven abstractions through a case study in energy storage systems. Using a realistic electrochemical model of a lithium-ion battery, we integrate reinforcement learning with data-driven abstractions to design an ageing-aware charging protocol. The resulting strategy balances fast charging with long-term durability and remains robust to uncertainties in battery parameters, highlighting the practical relevance of the proposed framework.

Samenvatting

Moderne technische systemen, variërend van autonome voertuigen tot energieopslagsystemen, moeten betrouwbaar functioneren onder onzekerheid en tegelijkertijd voldoen aan steeds complexere prestatie- en veiligheidseisen. Het waarborgen dat dergelijke systemen zich gedragen zoals bedoeld, valt onder het domein van verificatie, terwijl het nog ambitieuzere doel om regelaars te ontwerpen die correct gedrag van meet af aan garanderen bekendstaat als controller-synthese. Het behalen van deze doelstellingen is bijzonder uitdagend wanneer systemen niet-lineair zijn of slechts gedeeltelijk bekend.

Een centraal paradigma om deze uitdaging aan te pakken is het gebruik van symbolische abstracties: vereenvoudigde modellen die het essentiële gedrag van het onderliggende systeem behouden, terwijl de analytische hanteerbaarheid wordt vergroot. Abstracties maken het mogelijk om geautomatiseerde methoden voor verificatie en controller-synthese toe te passen, waardoor over veiligheid, bereikbaarheid of prestaties op een wiskundig rigoureuze manier kan worden geredeneerd. In het bijzonder maakt symbolische besturing gebruik van eindige-toestandsabstracties om geautomatiseerde, algoritmische synthese van regelaars mogelijk te maken die voorzien zijn van formele correctheidsgaranties. Traditionele abstractietechnieken vereisen echter volledige systeemkennis, wat hun toepasbaarheid beperkt in praktische situaties waarin modelkennis schaars is, terwijl data overvloedig aanwezig is.

Dit proefschrift onderzoekt hoe deze beperking kan worden overwonnen door **abstracties rechtstreeks uit data te leren** en **abstracties te leren in combinatie met data** wanneer gedeeltelijke kennis van de dynamica beschikbaar is; bovendien laten we zien hoe dergelijke abstracties kunnen worden toegepast voor verificatie en besturing onder onzekerheid. De bijdragen zijn georganiseerd in drie delen, die complementaire scenario's behandelen:

- Deel I beschouwt **deterministische systemen met onbekende dynamica en een eindig outputalfabet**. Door begintoestanden te bemonsteren en eindige outputreeksen te verzamelen, laten we zien hoe eindige-toestandsabstracties kunnen worden geconstrueerd die geschikt zijn voor het verifiëren van temporele eigenschappen en het synthetiseren van regelaars. De resulterende methoden bieden probabilistische correctheidsgaranties en kunnen, mits beperkte voorkennis van het systeem beschikbaar is, verder reiken dan de bemonsteringshorizon.

- Deel II richt zich op **systemen die bekende deterministische dynamica combineren met stochastische verstoringen van onbekende verdeling**. We stellen data-gedreven abstracties voor in de vorm van Markov-modellen, opgebouwd uit steekproeven van de stochastische dynamica, en tonen aan hoe

deze kunnen worden gebruikt om regelaars te synthetiseren die voldoen aan probabilistische specificaties.

- Deel III concentreert zich op **deterministische systemen met bekende dynamica waarvoor standaardabstractietechnieken computationeel inefficiënt of onhanteerbaar zijn**. Hier introduceren we een efficiënte synthese-algoritme dat data en modelkennis combineert en gebruikmaakt van nieuwe benaderende systeemrelaties die multi-resolutieabstracties met kwantificeerbare foutmarges mogelijk maken.

In al deze contexten tonen we aan hoe data systematisch kan worden benut om abstracties te produceren die zowel nauwkeurig zijn als voorzien van formele, kwantificeerbare garanties. Om deze bijdragen in een bredere statistisch-lerende context te plaatsen, onderzoeken we bovendien de rol van probabilistische onzekerheidskwantificatie, waarbij we kaders zoals scenario-theorie en conformal prediction vergelijken en de beperkingen van recent voorgestelde benaderingen verduidelijken.

Naast de theorie illustreert het proefschrift de praktische impact van data-gedreven abstracties aan de hand van een casestudy in energieopslagsystemen. Met behulp van een realistisch elektrochemisch model van een lithium-ionbatterij combineren we reinforcement learning met data-gedreven abstracties om een verouderingsbewust laadprotocol te ontwerpen. De resulterende strategie balanceert snel laden met duurzaamheid op de lange termijn en blijft robuust ten aanzien van onzekerheden in batterijparameters, wat de praktische relevantie van het voorgestelde raamwerk onderstreept.

1
Introduction

Modern engineered systems require reliable autonomy under increasingly demanding performance requirements while interacting with uncertain environments. From autonomous vehicles navigating busy urban roads, to robotic assistants collaborating with humans, to energy management systems coordinating renewable sources and storage devices, these systems must operate reliably under a wide range of conditions. A single failure may have catastrophic consequences: damage to expensive hardware, aircraft accidents, or large-scale power outages [1–5]. Ensuring safety and correctness in such contexts is therefore not only a desirable property but a fundamental requirement.

The challenge of guaranteeing correct behaviours in complex systems has been studied from different angles. In control theory, a rich set of mathematical tools exists to model dynamical systems, analyse their behaviour, and design controllers. Techniques such as Lyapunov stability analysis, robust control, and optimal control have been enormously successful in many industrial domains. Yet, these approaches typically require accurate modelling of the system and, possibly, of the inherent uncertainty. In many situations, this can be a costly requirement, either because the physics are too complex, because the parameters are uncertain, or because the environment itself is unpredictable. In addition, designing systems that satisfy complex specifications, beyond safety and reachability, with traditional approaches is often cumbersome.

In parallel, the field of formal methods emerged within computer science as a rigorous approach to specify, analyse, and verify the behaviour of software processes. Temporal logic is a common mathematical language that allows for the rigorous expression of properties that the software must abide by, while model checking is the research field that studies and develops automated methods to establish whether a program satisfies certain properties [5]. Initially applied to purely digital systems, formal methods later began to address the challenges arising in cyber-physical systems, where software interacts with physical processes. This naturally motivated an intersection with control theory: verifying and designing systems that span both continuous dynamics and discrete decision-making requires ideas from both disciplines [6].

In recent years, a third ingredient has entered this picture: data-driven approaches. Advances in sensing, computation, and machine learning have made it possible to work directly with data rather than relying solely on explicit models. Data-driven methods can capture behaviours of complex, nonlinear, or uncertain systems that are otherwise difficult to model. However, they typically lack the rigorous guarantees that are the hallmark of control theory and formal methods.

The convergence between control theory, formal methods, and data-driven approaches has generated significant interest in the research community. The common denominator is to develop methodologies that exploit data to build tractable, finite abstractions of complex systems, and then to apply formal reasoning and control design techniques on these abstractions. In this way, one aims to combine the generality of data-driven methods with the structure and rigour of control theory and formal methods.

This thesis is situated precisely in this intersection. Its overarching objective is to develop data-driven abstraction techniques that come with rigorous (probabilistic) guarantees. These techniques leverage a finite set of data samples to generate an abstraction that i) is a faithful representation of the underlying system, ii) the approximation is rigorously quantifiable, iii) controllers or verification results computed on the abstraction carry over to the true system. By uniting ideas from control theory, formal methods, and learning, this work seeks to contribute to the foundations of reliable data-driven control.

1.1. Motivation

While the combination of control theory, formal methods, and data-driven approaches is conceptually appealing, it raises a number of concrete challenges. This thesis is motivated by three such challenges:

- *From finite-horizon trajectories to long-horizon guarantees.* In a purely data-driven setting, where the explicit *dynamics are unknown* but deterministic, information about a system's behaviour typically comes from finite trajectories of length H. Standard statistical model checking techniques can then be used to test whether these sampled behaviours satisfy a specific property [7]. By contrast, our approach is to construct an abstraction that, up to a controlled approximation, captures all possible behaviours of the system simultaneously. This provides a compact description from which many properties can be verified at once, rather than checking them individually. Such data-driven abstractions naturally support reasoning about properties restricted to the sampled horizon H. However, this raises a further fundamental question: how can one extrapolate guarantees from data collected over horizon H to properties defined over larger horizons? The first line of work in this thesis addresses both issues, developing a framework that combines finite trajectory sampling with Probably Approximately Correct (PAC) guarantees.

- *Abstractions of stochastic systems from data.* In many practical situations, system dynamics are subject to non-negligible randomness, arising, for instance, from sensor and actuator noise or uncertain environmental conditions. When dynamics are inherently stochastic, purely non-probabilistic abstractions are either too conservative or fail to capture the relevant behaviour. The predominant framework adopted by the research community for these situations is that of Markov processes, a formalism developed to analyse stochastic systems. Yet, explicitly modelling the underlying random processes is often infeasible or inaccurate, while data of their realisations is easily accessible. The central question in this setting is: how can one abstract a Markov process directly from *noise realisations*, while ensuring they faithfully represent the true process? The second line of work in this thesis addresses this question, building on a recent related result [8], by developing methods for building data-driven abstractions of stochastic processes and establishing their formal relationship with the original system. PAC-style guarantees are again pivotal, but with a different interpretation: they provide quantitative bounds on how likely it is that the abstraction accurately captures the stochastic dynamics of the underlying Markov process.

- *Multi-resolution abstractions.* Abstractions inevitably involve a trade-off between fidelity and tractability: coarse abstractions are easier to construct and analyse but may miss important behavioural nuances, while fine abstractions provide higher accuracy at the cost of greater complexity. In practice, however, it is often desirable to vary the abstraction's granularity across the state space, depending on the local dynamics, the property of interest, or available computational resources. This motivates the development of multi-resolution abstractions, where different parts of the system can be represented at different levels of precision within a single abstraction. The third line of work in this thesis leverages the notion of approximate bisimulations [6, 9] and introduces the theory and algorithms necessary for constructing such multi-resolution approximate abstractions [10] from a combination of *data and model knowledge*. The framework ensures that the relationship between the concrete system and its abstraction is formally characterised, while allowing the user to specify a *resolution function* that dictates, locally in the state set, what degree of coarseness in the dynamics is acceptable. The proposed algorithm then synthesizes an abstraction and a corresponding relation that respect this specification, thus providing computational scalability while preserving higher fidelity exactly where it is required.

Taken together, these three lines of work investigate how data can be systematically leveraged to construct approximate models that are tractable. They show how information from finite trajectories, stochastic noise realisations, and model knowledge, when available, can be integrated into principled abstraction frameworks. Across all cases, the unifying theme is to balance rigour with practicality: data-driven abstractions are approximate yet come with formal guarantees, enabling reliable reasoning about system behaviour beyond the raw data. In this way, the thesis contributes to bridging the gap between data-driven approaches and traditional model-based analysis, providing tools that are flexible enough for uncertain, stochastic, and complex systems, yet grounded in the formal guarantees required by control and formal methods.

1.2. Related Work

1.2.1. Data-driven Abstractions of Deterministic Systems

Recent work on data-driven control has focused on using collected data from a system to directly (i.e., without an explicit model) construct barrier functions certifying invariance [11–15], or to build finite abstractions for verification and controller synthesis [16–20]. Within this area, a recurring case studied in many works [16, 18, 21–23] is that of deterministic systems for which a model is unavailable. Here, the only source of uncertainty lies in the initialization: the initial state is drawn from some probability distribution, while the control policy is usually selected from a finite set of possible actions. The available data then consists of individual transitions or, more generally, finite-length trajectories. In this setting, scenario theory optimisation [24, 25] provides a powerful framework for estimating uncertainty and deriving Probably Approximately Correct (PAC) guarantees for the performance metric of interest. Scenario optimization requires independent samples generated from the distribution governing the system's uncertainty. Such independence is naturally achieved by independently sampling initial states. Two main approaches can be distinguished in the literature. The first relies on sampling multi-step transitions (finite-length trajectories) to measure the closeness between the system and a given abstraction [21, 26]. However, the resulting PAC guarantees are necessarily tied to the horizon of the sampled trajectories; without further information on the system, no direct conclusions can be drawn for longer horizons. The second approach instead samples one-step transitions (pairs of an initial condition and its successor) to construct a candidate abstraction from data [16, 19, 20]. In this case, generalizing information obtained from such samples to arbitrary horizons typically requires partial knowledge of the dynamics. Indeed, since the transition function modifies the distribution of states over time, closeness between trajectories of the abstraction and the concrete system is usually inferred under assumptions such as known Lipschitz constants [19, 20].

Our work in Chapters 3 and 4 is positioned between these two extremes. While we rely on multi-step trajectories, we develop a detailed analysis of the interplay between the sampling horizon and the horizon of the PAC guarantees of interest. Specifically, we focus on abstraction synthesis for verification and control from sampled input-output sequences, with satisfaction of specifications over finite horizons ensured by PAC bounds. At the same time, we also consider a fundamentally different setting: in contrast with the aforementioned works, we study dynamical systems where the observable output belongs to a finite alphabet. In this case, establishing non-trivial metrics between outputs or trajectories is not possible, which makes the direct application of scenario theory significantly more challenging.

1.2.2. Data-driven Abstractions of Stochastic Systems

A common modelling framework in the stochastic setting is that of *Markov Decision Processes* (MDPs) [5], which capture both the probabilistic nature of the dynamics and the presence of control inputs. A large body of work has focused on constructing finite-state abstractions of such systems, still in the form of Markov models, but exploiting model knowledge to establish formal relationships between the abstraction and the original system [27–30]. Typically, this process partitions the state space into a finite set of regions, each representing an abstract state, and computes transition probabilities amongst them, using an explicit mathematical representation of the underlying dynamics. Inevitably, such abstractions approximate the original system, motivating richer frameworks such as Robust MDPs (RMDPs) [31] and Interval MDPs (IMDPs) [32, 33], which can simultaneously capture both probabilistic and nondeterministic behaviours. However, computing the transition probabilities required for these abstractions depends critically on having an exact model of the system dynamics. Even when the model is available, abstracting the system to a finite-state representation often entails evaluating integrals of the stochastic kernel over regions of the state space, an operation that is computationally expensive and, for high-dimensional systems, often impractical. For these reasons, recent work has shifted toward leveraging data to derive guarantees directly from samples of the dynamics [12, 26, 34, 35], or to construct data-driven abstractions [8, 19, 36–41], grounded in rigorous statistical learning frameworks such as scenario theory and conformal prediction[42].

Chapter 7 contributes to the latter line of research. We focus on infinite-state MDPs whose transition function can be expressed as deterministic dynamics with additive unknown noise. In the seminal works [8, 37], it was shown that, under certain controllability assumptions, stochastic dynamics can be learnt from a finite set of noise realizations, enabling the abstraction of the MDP into an IMDP, up to a confidence probability. We expand the scope of these results by proposing a more flexible abstraction scheme that approximates the MDP with a special instance of an RMDP, thereby enabling policy synthesis in a wider variety of settings.

1.2.3. Multi-resolution Abstractions

A central theme in abstraction-based control is *scalability.* Early work on abstractions for infinite-state systems introduced ϵ-approximate bisimulations and bisimulation functions, where system behaviours are matched within a uniform error margin [6, 9, 43]. These ideas were highly influential in bridging control theory with formal methods, as they enabled infinite-state systems to be represented as finite-state abstractions. Such approaches, however, rely on *uniform-resolution* abstractions, typically obtained by gridding the state space. While these methods provide rigorous guarantees, they suffer from poor scalability due to the curse of dimensionality: under uniform gridding, the number of abstract states grows exponentially with the system's dimension. Moreover, many of these techniques require the existence of certificates, such as Lyapunov-like functions [44, 45], which are often difficult to construct in practice. This motivates the search for abstraction techniques that are both more flexible and more broadly applicable. To address scalability, several works have proposed *multi-resolution* approaches, where abstractions are built using grids of varying coarseness [46–49]. These methods yield more compact abstractions by allocating abstract states where it matters most. In parallel, the notion of *multi-resolution approximate bisimulation* was introduced [10, 50], allowing for heterogeneous error bounds across the state space, enabling

finer approximations in regions of interest while keeping coarser ones elsewhere. When no certificates are available, abstraction construction becomes even more challenging. In this context, recent data-driven and learning-based methods [51–53] exploit datasets of system transitions to train parameterised functions that elicit a formal relation between the original system and a candidate abstraction.
Building on these developments, Chapter 8 adopts the foundational setting of [10, 50] and considers deterministic systems with known dynamics. By combining sampled transitions with model knowledge, we develop a flexible algorithm that extends the applicability of abstraction methods to previously unaddressed dynamical systems. Numerical examples demonstrate that our scheme not only reduces the number of abstract states but also achieves higher-than-standard resolution where needed.

1.3. Contributions and Outline of this Dissertation

In Chapter 2 we introduce the mathematical notation and formalisms that recur throughout the dissertation. Specifically, we recall transition systems, system relations, abstraction models, and scenario theory. Since the thesis spans different settings, we also introduce additional notation at the beginning of each chapter whenever required.

Chapter 3 presents the first technical contribution: a method to construct data-driven abstractions of black-box deterministic systems with random initialization, based on a finite number of symbolic output trajectories, for the purpose of verifying temporal logic specifications. A key insight here is that a special class of abstractions, Strongest Asynchronous l-complete Abstractions (SAlCAs) [54], can be leveraged to cast the abstraction-construction problem into a scenario theory optimization. The resulting abstraction behaviorally includes the concrete system and enables the verification of formulas in the universal fragment of bounded-time Computation Tree Logic (CTL), with guarantees expressed in PAC bounds. In addition, we provide a detailed analysis of how the choice of sampling horizon affects these guarantees, thereby clarifying limitations of existing approaches in the literature [55].

Building on these results, Chapter 4 develops data-driven SAlCAs for abstraction-based control synthesis. Here, we introduce a new notion of probabilistic alternating simulation relation, which allows us to exploit powerful algorithms originally designed for finite-state systems. These algorithms are used to synthesize controllers satisfying bounded-time Alternating-time Temporal Logic (ATL) specifications on the abstraction, and the solution can then be transferred to the concrete system via refinement. Moreover, we establish sufficient conditions on the system dynamics under which the validity of the abstraction can be extended beyond the sampling horizon, thereby reducing the gap between finite-horizon data and long-term guarantees [56].

In Chapter 5 we shift focus to a case study in the domain of energy storage systems. Specifically, we consider ageing-aware fast charging of lithium-ion battery cells, modeled through a realistic electrochemical description. We propose an automated procedure that interlocks reinforcement learning, for data-driven policy synthesis, with data-driven abstractions, for verifying performance and safety. The resulting charging protocol is thoroughly documented and shown to outperform industrial standards, highlighting the practical impact of combining formal methods with learning techniques [57].

Chapter 6 investigates an alternative statistical framework to scenario theory, namely Conformal Prediction (CP). Although CP has recently gained significant attention in the control community and has been successfully applied in various contexts, we demonstrate that a recent extension of CP is not suitable for the safety verification of dynamical systems. This result clarifies the scope and limitations of CP in formal verification and emphasizes the continuing relevance of scenario-based approaches [58].

In Chapter 7 we return to theoretical developments on data-driven abstractions, this time focusing on stochastic control systems with unknown additive noise. Leveraging scenario theory, we present a methodology to characterize the noise distribution and to construct abstractions that incorporate a higher degree of nondeterminism. This added nondeterminism increases the flexibility of the abstraction by enlarging the search space of admissible policies, extending existing methods in the literature [59].

Finally, Chapter 8 addresses the scalability challenge by considering deterministic autonomous systems with known dynamics and developing multi-resolution abstractions. Our approach combines data with model knowledge to construct abstractions that use sparser abstract state sets than state-of-the-art methods, while allowing variable resolution across the state space. This flexibility enables us to tackle a previously unaddressed class of systems, namely incrementally uniformly bounded systems. We provide original theoretical results for the existence of multi-resolution abstractions and propose a parallelizable algorithm, based on Satisfiability Modulo Theories (SMT) solvers, to efficiently compute them. The effectiveness of the approach is demonstrated through several numerical case studies [60].

2
Preliminaries and Notation

2.1. Notation

The set of natural numbers (including 0) is denoted by $\mathbb{N}$ ($\mathbb{N}_0$). The set of (positive) real numbers is denoted by $\mathbb{R}$ ($\mathbb{R}_{>0}$). The set of integers is denoted by $\mathbb{Z}$. Given a set $\mathcal{X}_a$ we denote its cardinality as $|\mathcal{X}_a|$. For any non negative integer n we denote the n-th cartesian product of $\mathcal{X}_a$ as $\mathcal{X}_a^n$, with $\mathcal{X}_a^0 = \emptyset$, and its power set by $\wp(\mathcal{X}_a)$. A *cover* of a set $\mathcal{X}_a$ is a finite collection of sets $\mathcal{T} = \{T_i\}_{i=1}^M$ such that each element of the collection is a subset of $\mathcal{X}_a$ and such that the union of the elements in the collection contains $\mathcal{X}$. A *partition* of a set $\mathcal{X}_a$ is a cover $\mathcal{Q} = \{Q_i\}_{i=1}^N$ such that the elements of the collection are pairwise disjoint. Given the sets $\mathcal{X}_a$, $\mathcal{X}_b$, and a binary relation $R \subseteq \mathcal{X}_a \times \mathcal{X}_b$, we define $R(x_b) \doteq \{x_a \in \mathcal{X}_a : (x_a, x_b) \in R\}$. We indicate the inverse relation of R by R^{-1}, i.e. $(x_b, x_a) \in R^{-1}$ if and only if $(x_a, x_b) \in R$.

2.2. Transition Systems

We adopt the framework of [6] and we model dynamical systems as Transition Systems (TS).

Definition 1 (Transition System [6]). *A transition system S is a tuple $(\mathcal{X}, \mathcal{X}_0, \mathcal{U}, \delta, \mathcal{Y}, \mathcal{H})$ where:*

- *$\mathcal{X}$ is the (possibly infinite) set of states,*
- *$\mathcal{X}_0 \subseteq \mathcal{X}$ is the set of initial states,*
- *$\mathcal{U}$ is the input set,*
- *$\delta \subseteq \mathcal{X} \times \mathcal{U} \times \mathcal{X}$ is the set of edges, or transitions,*
- *$\mathcal{Y}$ is the set of outputs, and*
- *$\mathcal{H} : \mathcal{X} \to \mathcal{Y}$ is the output map.*

We define the set of u-successor states of a state x as $\mathrm{Post}_u(x) \doteq \{x' \in \mathcal{X} : (x, u, x') \in \delta\}$ and the set of admissible inputs at x as $U_\delta(x) \doteq \{u \in \mathcal{U} : \mathrm{Post}_u(x) \neq \emptyset\}$. If $U_\delta(x) \neq \emptyset$ for all x, the system is *non-blocking*. An *internal behavior of horizon H* of the TS is a sequence $\mathtt{r} = x_0 u_0 x_1 \ldots u_{H-1} x_H$ that satisfies $x_0 \in \mathcal{X}_0$ and $(x_{i-1}, u_{i-1}, x_i) \in \delta$ for all $i = 1, \ldots, H$. A sequence $\mathtt{b} = y_0 u_1 y_1 \ldots u_{H-1} y_H$ is an *external behavior of horizon H* if there exists an internal behavior $\mathtt{r} = x_0 u_0 x_1 \ldots u_{H-1} x_H$ such that $y_i = \mathcal{H}(x_i)$ for all $i = 0, \ldots, H$. To indicate that $\mathtt{b}$ is the external behavior corresponding to $\mathtt{r}$ we overload the notation and occasionally write $\mathtt{b} = \mathcal{H}(\mathtt{r})$. By $\mathtt{r}[i]$ we indicate x_i; by $\mathtt{r}[i, i+j]$ with $j \geq 0$ we denote the subsequence $x_i u_i \ldots u_{i+j-1} x_{i+j}$, and similarly for $\mathtt{b}[i]$ and $\mathtt{b}[i, i+j]$. By $\mathtt{r}|^x$ ($\mathtt{r}|^u$) we denote the subsequence $x_0 x_1 \ldots x_H$ obtained by removing the inputs (states), and similarly for $\mathtt{b}|^y$ ($\mathtt{b}|^u$). The sets $\mathcal{I}_H(S)$ and $\mathcal{B}_H(S)$ contain all H-long internal and external behaviors of the TS, respectively. Given $x \in \mathcal{X}_0$ and an H-long input sequence $\mathbf{u}_H \in \mathcal{U}^H$ we define respectively the set of internal and external H-behavior of a TS S starting in x_0 under input sequence $\mathbf{u}_H$ as

$$\mathcal{I}_H(S, x_0, \mathbf{u}_H) \doteq \{\mathtt{r} \in \mathcal{I}_H(S) : \mathtt{r}(0) = x_0 \ \wedge \ \mathtt{r}|^u = \mathbf{u}_H\}, \tag{2.1}$$

$$\mathcal{B}_H(S, x_0, \mathbf{u}_H) \doteq \{\mathtt{b} \in \mathcal{B}_H(S) : \exists \mathtt{r} \in \mathcal{I}_H(S, x_0, \mathbf{u}_H) \ . \ \mathtt{b} = \mathcal{H}(\mathtt{r})\}. \tag{2.2}$$

Given two sequences $s = x_0 u_0 ... u_{i-1} x_i$ and $s' = x'_0 u'_0 ... u'_{j-1} x'_j$ with $x_i = x'_0$, we denote their concatenation by $s \cdot s' \doteq x_0 u_0 ... u_{i-1} x'_0 u'_0 ... u'_{j-1} x'_j$.

A TS is *autonomous* if $|\mathcal{U}| = 1$: in this case, for notational convenience, we denote the TS as $S = (\mathcal{X}, \mathcal{X}_0, \delta, \mathcal{Y}, \mathcal{H})$, where $\delta \subseteq \mathcal{X} \times \mathcal{X}$. To streamline the exposition for autonomous systems, we apply the following simplifications; an *internal behavior of horizon* H of the autonomous TS S is a sequence $\mathtt{r} = x_0 x_1 \dots x_H$ that satisfies $x_0 \in \mathcal{X}_0$ and $(x_{i-1}, x_i) \in \delta$ for all $i = 1, \dots, H$. Similarly, a sequence $\mathtt{b} = y_0 y_1 \dots y_H$ is an *external behavior of horizon* H if there exists an internal behavior $\mathtt{r} = x_0 x_1 \dots x_H$ such that $y_i = \mathcal{H}(x_i)$ for all $i = 0, \dots, H$. By $\mathtt{r}[i, i+j]$ with $j \geq 0$ we denote the subsequence $x_i \dots x_{i+j}$, and similarly for $\mathtt{b}[i, i+j]$. The sets $\mathcal{I}_H(S)$ and $\mathcal{B}_H(S)$ contain all H-long internal and external behaviors of the autonomous TS, respectively. Given $x \in \mathcal{X}_0$ we define respectively the set of internal and external H-behavior of the autonmous TS S starting in x_0 as

$$\mathcal{I}_H(S, x_0) \doteq \{\mathtt{r} \in \mathcal{I}_H(S) : \mathtt{r}(0) = x_0\}, \tag{2.3}$$

$$\mathcal{B}_H(S, x_0) \doteq \{\mathtt{b} \in \mathcal{B}_H(S) : \exists \mathtt{r} \in \mathcal{I}_H(S, x_0) \ . \ \mathtt{b} = \mathcal{H}(\mathtt{r})\}. \tag{2.4}$$

The notation is used unambiguously: Chapter 3 deals exclusively with abstractions for autonomous TS, while Chapter 4 treats the general case for TSs with nontrivial input spaces, therefore, the semantics are clear from the context.

2.3. System Relations

The central idea of abstraction-based verification and control is to substitute the original system S with a simpler model, the abstraction, where solving the problem of interest is easier. Transferring results from the abstraction back to the original system requires a formal relation establishing how similar the two models are and which properties are transferable. One of the weakest notions of similarity between models is that of *behavioral inclusion*.

Definition 2 (Behavioral Inclusion [6])**.** *Given two transition systems S_a and S_b, S_b behaviorally includes S_a for horizon H, or equivalently S_a is behaviorally included in S_b for horizon H, if $\mathcal{B}_H(S_b) \supseteq \mathcal{B}_H(S_a)$. Equivalently, for all $x_0^a \in \mathcal{X}_0^a$ and $\mathbf{u}_H \in \mathcal{U}^H$, $\mathcal{B}_H(S_b) \supseteq \mathcal{B}_H(S_a, x_0, \mathbf{u}_H)$. We denote this with the notation $S_a \preceq_{\mathcal{B}_H} S_b$. If $S_a \preceq_{\mathcal{B}_H} S_b$ holds for all $H > 0$, we say S_a is behaviorally included in S_b.*

Behavioral Inclusion allows for reasoning on properties like *safety*: if all the behaviors of S_b are deemed safe, we can conclude that all the behaviors of S_a must be safe as well. Clearly, for this argumentation to be appealing it is necessary for S_b to be much simpler to analyze than S_a.

Simulation is a stronger notion than behavioral inclusion: it requires the existence of a binary relation $R \subseteq \mathcal{X}_a \times \mathcal{X}_b$ describing which states of S_a are simulated by which states of S_b.

Definition 3 (Simulation Relation (SR) [6]). *Consider two systems S_a and S_b with $\mathcal{Y}_a = \mathcal{Y}_b$. A relation $R \subseteq \mathcal{X}_a \times \mathcal{X}_b$ is a simulation relation from S_a to S_b, written $S_a \preceq_S^R S_b$, if the following three conditions are satisfied:*

- *For all $x_{a0} \in \mathcal{X}_{a0}$ there exists $x_{b0} \in \mathcal{X}_{b0}$ with $(x_{a0}, x_{b0}) \in R$,*
- *For all $(x_a, x_b) \in R$ it holds $\mathcal{H}_a(x_a) = \mathcal{H}_b(x_b)$,*
- *For all $(x_a, x_b) \in R$ for all $u_a \in U_{\delta_a}(x_a)$ there exists u_b in $U_{\delta_b}(x_b)$ such that for all $x'_a \in \mathrm{Post}_{u_a}(x_a)$ there exists $x'_b \in \mathrm{Post}_{u_b}(x_b)$ with $(x'_a, x'_b) \in R$.*

Simulation and behavioral inclusion are central for verification problems. It is easy to see from the definition of simulation relation that, if $S_a \preceq_S^R S_b$ holds, then for any H it holds that $S_a \preceq_{\mathcal{B}_H} S_b$. The notion of simulation relation is stronger than behavioral inclusion because it requires *relating states* of S_a with those of S_b, so as to guarantee that S_b can mimic step-by-step S_a, see Figure 2.1.

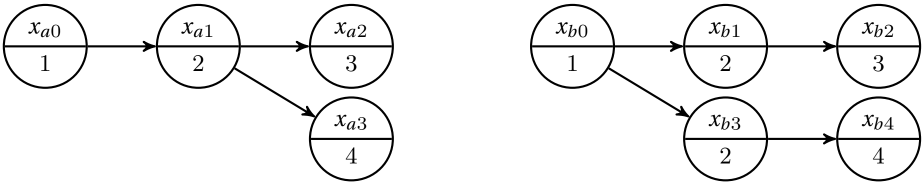


Figure 2.1.: Behavioral inclusion does not guarantee the existence of a simulation relation. The top part of each node indicates the state, the bottom indicates the output. Both S_a (left) and S_b (right) are autonomous. For any H it holds that $S_a \preceq_{\mathcal{B}_H} S_b$, but x_{a1} can not be simulated by either of x_{b1} and x_{b3}, hence $S_a \npreceq_S^R S_b$.

For control synthesis problems the central notion that allows to refine a controller designed by means of an abstraction to the original system is that of *alternating simulation relation*, a notion of similarity that accounts for how different input choices affect transitions in the presence of non-determinism.

Definition 4 (Alternating Simulation Relation (ASR) [6]). *Consider two systems S_a and S_b with $\mathcal{Y}_a = \mathcal{Y}_b$. A relation $R \subseteq \mathcal{X}_a \times \mathcal{X}_b$ is an alternating simulation relation from S_a to S_b, written $S_a \preceq_{AS}^R S_b$, if the following three conditions are satisfied:*

- *For all $x_{a0} \in \mathcal{X}_{a0}$ there exists $x_{b0} \in \mathcal{X}_{b0}$ with $(x_{a0}, x_{b0}) \in R$,*
- *For all $(x_a, x_b) \in R$ it holds $\mathcal{H}_a(x_a) = \mathcal{H}_b(x_b)$,*
- *For all $(x_a, x_b) \in R$ for all $u_a \in U_{\delta_a}(x_a)$ there exists u_b in $U_{\delta_b}(x_b)$ such that for all $x'_b \in \mathrm{Post}_{u_b}(x_b)$ there exists $x'_a \in \mathrm{Post}_{u_a}(x_a)$ with $(x'_a, x'_b) \in R)$.*

When there exists an (alternating) simulation relation from S_a to S_b we say that S_b (alternatingly) simulates S_a.

2.4. Strongest Asynchronous l-complete Abstractions

SAlCA abstractions, introduced in [54, 61] and reformulated here as transition systems, rely solely on external behaviors—making them well-suited for data-driven methods that bypass internal model details. While their construction can be generalized under a single set of rules, below we distinguish the case where the system to be abstracted is or not autonomous. This differentiation is a consequence of the fact that for non-autonomous TSs the external behaviors include inputs and outputs, wheras for autonomous TSs external behaviors comprise exclusively outputs.

SAlCA for Non-autonomous Transition Systems

For initial states $x \in \mathcal{X}_0$, following [54], we extend behaviors to negative time indices using the symbol $\diamond$, so $\mathtt{r}|^x[k] = \mathtt{r}|^u[k] = \mathtt{b}|^y[k] = \mathtt{b}|^u[k] = \diamond$ for $k \leq -1$.
Consider the set of external behaviors of length H and an integer l with $0 \leq l < H$. Each $\mathtt{b} \in \mathcal{B}_H(S)$ is divided into l-long subsequences (or l-sequences). The set of all l-sequences from these behaviors is denoted by

$$\Pi_{l,H} \doteq \bigcup_{\mathtt{b} \in \mathcal{B}_H(S)} \bigcup_{k \in [0,H]} \mathtt{b}[k-l,k]. \tag{2.5}$$

Each $\mathtt{q} \in \Pi_{l,H}$ contains $l+1$ outputs and l inputs. We omit the dependence of $\Pi_{l,H}$ on S since it always refers to the concrete system.

A state x of the transition system S may be put in relation with the set of l-sequences based on the outputs generated immediately before, immediately after, or while reaching x. A detailed discussion of the effects of different choices is available in [54]. For non-autonomous systems we consider past inputs, effectively representing the recent memory in terms of inputs and outputs of a state. The set of *past* corresponding external strings (CESs) of length l for a state $x \in \mathcal{X}$ is defined as

$$\mathcal{E}_{l,H}(x) \doteq \{\mathtt{q} \in \Pi_{l,H} \ : \exists \mathtt{r} \in \mathcal{I}_H(S), \exists j \in \mathbb{N}_0 \ . \ \mathtt{q} = \mathcal{H}(\mathtt{r}[j-l,j]) \wedge \mathtt{r}[j] = x\}. \tag{2.6}$$

$\mathcal{E}_{l,H}(x)$ represents all subsequences of external behaviors with $l+1$ outputs that the system can generate before reaching x in at most H steps. If x is reached in fewer than l transitions from an initial state, some CESs will include the symbol $\diamond$. For instance, if $l = 3$ and x' is reached in one transition from the initial state x by choosing control input u, then $\diamond\diamond\mathcal{H}(x)u\mathcal{H}(x') \in \mathcal{E}_{3,H}(x')$. The equivalence class of an l-sequence $\mathtt{q} \in \Pi_{l,H}$ is

$$[\mathtt{q}] \doteq \{x : \mathtt{q} \in \mathcal{E}_{l,H}(x)\}. \tag{2.7}$$

Example 1. Let S be the TS depicted in Fig. 2.2, where $\mathcal{X} = \{x_1, x_2, x_3, x_4\}$, $\mathcal{X}_0 = \{x_1\}$, $\mathcal{U} = \{u_a, u_b\}$, and $\mathcal{Y} = \{y_1, y_2\}$. For brevity, let us fix the time horizon $H = 4$ and input sequence $\mathbf{u}_3 = u_a u_b u_a$ – a similar reasoning can be carried out for all the different input sequences. The internal and external behaviors initialized from x_1, are given by $\mathcal{I}_3(S, x_1, \mathbf{u}_3) = \{x_1 u_a x_2 u_b x_4 u_a x_2\}$ and $\mathcal{B}_3(S, x_1, \mathbf{u}_3) = \{y_1 u_a y_2 u_b y_2 u_a y_2\}$. To obtain $\Pi_{0,3}$, following equation (2.5) we split $\mathcal{B}_3(S, x_1, \mathbf{u}_3) \in \mathcal{B}_H(S)$ in with $l = 0$ and conclude that $\{y_1, y_2\} = \mathcal{Y} = \Pi_{0,3}$. Moreover, from equation (2.6) we have $y_1 \in \mathcal{E}_{0,3}(x_1)$, $y_2 = \mathcal{E}_{0,3}(x_2)$, and $y_2 \in \mathcal{E}_{0,3}(x_4)$.

Similarly, to obtain $\Pi_{1,3}$, we have $\{\diamond\diamond y_1, y_1 u_a y_2, y_2 u_b y_2, y_2 u_a y_2\} \subset \Pi_{1,3}$. Moreover, $\diamond\diamond y_1 \in \mathcal{E}_{1,3}(x_1)$, $y_1 u_a y_2 \in \mathcal{E}_{1,3}(x_2)$, $y_2 u_b y_2 \in \mathcal{E}_{1,3}(x_4)$, and again $y_2 u_a y_2 \in \mathcal{E}_{1,3}(x_2)$.

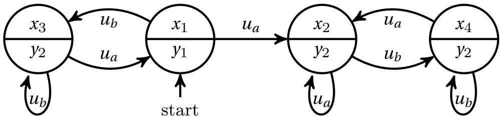


Figure 2.2.: Illustration of the TS of Example 1.

Remark 1. When $\mathcal{X}_0 = \mathcal{X}$ the horizon H does not affect the definition of equation (2.5) and equation (2.6), since every l-sequence that the system can generate at any point in time, can also be generated as an initial sequence. Formally, it holds that $\mathcal{E}_{l,l+1}(x) = \mathcal{E}_{l,H}(x)$ for every $H > l$. Moreover, $\bigcup_x \mathcal{E}_{l,H}(x) = \Pi_{l,H}$. From now on, we drop the second subscript and denote the past CESs strings of a state x simply as $\mathcal{E}_l(x)$ and the set of all l-sequences of all external behaviors as Π_l. The case with $\mathcal{X}_0 \neq \mathcal{X}$ follows *mutatis mutandis* with no conceptual modifications.

Definition 5. (SAlCA (nonautonomous) [54]) *Let* $S \doteq (\mathcal{X}, \mathcal{X}, \mathcal{U}, \delta, \mathcal{Y}, \mathcal{H})$ *be a TS, and consider* Π_l *and* Π_{l+1}. *The TS* $S_l \doteq (\mathcal{X}_l, \mathcal{X}_{l0}, \mathcal{U}, \delta_l, \mathcal{Y}, \mathcal{H}_l)$ *is the SAlCA of* S*, where* $\mathcal{X}_l \doteq \Pi_l$ *is the state set,* $\mathcal{X}_{l0} \doteq \{\mathsf{q} \in \Pi_l : \exists \mathsf{b} \in \mathcal{B}_H(S) \;.\; \mathsf{b}[-l, 0] = \mathsf{q}\}$ *is the set of initial states,* $H_l(\mathsf{q}) \doteq \mathsf{q}|^y(l)$ *is the output map, and the transition relation is given by*

$$\delta_l \doteq \{(\mathsf{q}, u, \mathsf{q}') \in \mathcal{X}_l \times \mathcal{U} \times \mathcal{X}_l \;:\; \exists \mathsf{q}'' \in \Pi_{l+1} \;.\; \mathsf{q}''[0, l] = \mathsf{q} \wedge \mathsf{q}''[1, l+1] = \mathsf{q}' \wedge \mathsf{q}''|^u(l) = u\}.$$

The SAlCA state space comprises all l-long sequences from system S, with transitions defined by the *domino rule*: a transition occurs when the suffix of one sequence matches the prefix of the next, inputs align, and the combined sequence belongs to Π_{l+1}, a set of valid external behaviors. As SAlCA (over-)approximates S, increasing l improves precision, with $\mathcal{B}_H(S) \subseteq \mathcal{B}_H(S_{l+1}) \subseteq \mathcal{B}_H(S_l)$ for all H; moreover, Π_l can be derived from Π_{l+1}. Referring to Example 1, the corresponding SAlCAs with $l = 0$ and $l = 1$ is shown in Figure 2.3.

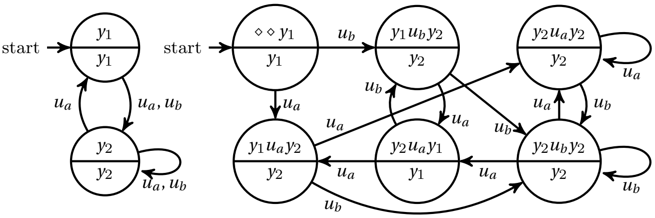


Figure 2.3.: Illustration of the SAlCA for the system described in Example 1 for $l = 0$, S_0 (left) and $l = 1$, S_1, (right), derived using the set $\Pi_{1,4}$ and $\Pi_{2,4}$ respectively.

SAlCA for Autonomous Transition Systems

Let us divide each external behavior $\mathtt{b} \in \mathcal{B}_H(S)$ of the autonomous TS S into l-long subsequences (or l-sequences), with $l \in \mathbb{N}$ and $H \geq l$. The set of all such subsequences is

$$\Upsilon_{l,H} \doteq \bigcup_{\mathtt{b} \in \mathcal{B}_H(S)} \bigcup_{k \in [l-1,H]} \mathtt{b}[k-l+1,k]. \tag{2.8}$$

Each $\mathtt{q} \in \Upsilon_{l,H}$ contains l outputs. For autonomous TSs, a state x is related to l-sequences based on the outputs generated immediately after reaching x. The set of *future* CESs of length l for a state $x \in \mathcal{X}$ is defined as

$$\mathcal{L}_{l,H}(x) \doteq \{\mathtt{q} \in \Upsilon_{l,H} \;:\exists \mathtt{r} \in \mathcal{I}_H(S), \exists j \in \mathbb{N}_0 \;.\; \mathtt{q} = \mathcal{H}(\mathtt{r}[j, j+l-1]) \wedge \mathtt{r}[j] = x\}. \tag{2.9}$$

$\mathcal{L}_{l,H}(x)$ represents all subsequences of external behaviors with l outputs that the system can generate immediately after reaching x in at most H steps. The equivalence class of an l-sequence $\mathtt{q} \in \Upsilon_{l,H}$ is

$$[\mathtt{q}] \doteq \{x : \mathtt{q} \in \mathcal{L}_{l,H}(x)\}. \tag{2.10}$$

Following Remark 1, we drop the H index in the following discussion. Under these simplifications, the SAlCA for autonomous systems may be presented in a simplified form.

Definition 6 ((SAlCA (autonomous) [54, 62]). *Let $S := (\mathcal{X}, \mathcal{X}, \delta, \mathcal{Y}, \mathcal{H})$ be an autonomous TS, and let $\mathcal{X}_l = \Upsilon_l \subseteq \mathcal{Y}^l$ Then, the system $S_l = (\mathcal{X}_l, \mathcal{X}_{l,0}, \delta_l, \mathcal{Y}^l, \mathcal{H})$ is called the Strongest Asynchronous l-complete Abstraction (SAlCA) of S, where $\mathcal{X}_{l0} \doteq \{\mathtt{q} \in \Upsilon_l : \exists \mathtt{b} \in \mathcal{B}_H(S) \;.\; \mathtt{b}[0, l-1] = \mathtt{q}\}$*

- $\delta_l = \{(k\sigma, \sigma k') \,|\, k, k' \in \mathcal{Y},\ \sigma \in \mathcal{Y}^{l-1},\ k\sigma,\ \sigma k' \in \mathcal{X}_l\}$,
- $\mathcal{H}(k\sigma) = k$.

The SAlCA encodes each state as an l-long subsequence of an external behavior. As before, the transitions of the SAlCA obey the domino rule. For instance, an abstract state $y_0y_1y_2$, transitions to another abstract state if and only if the successor abstract state begins with y_1y_2; e.g. state $y_0y_1y_2$ can transition to $y_1y_2y_0$, $y_1y_2y_1$, and $y_1y_2y_2$. The output of a state is its first element, so $\mathcal{H}(y_0y_1y_2) = y_0$. An example of SA$l$CA is given in Figure 2.4.

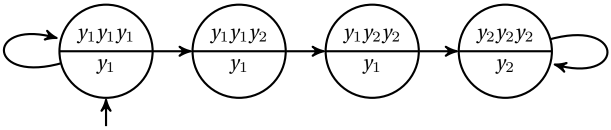


Figure 2.4.: Example of SAlCA, with $l = 3$.

Analogously to the non-autonomous case, the SAlCA of an autonomous TS (over-)approximates S and increasing l improves precision, with $\mathcal{B}_H(S) \subseteq \mathcal{B}_H(S_{l+1}) \subseteq \mathcal{B}_H(S_l)$ for all H.

Remark 2. The main difference between the SAlCA of an autonomous TS and the SAlCA of a non-autonomous TS lies in how the states of the concrete TS relate to those of its abstraction. For an autonomous TS, its states are related to the l-sequences representing its *future*. For a non-autonomous TS its states are related to the l-sequences representing its *past*. We discuss the consequences of this choice in Chapter 3 and Chapter 4.

2.5. Markov Models

We recall the notions required for our discussion on abstractions of stochastic dynamical systems.

Definition 7 ([32]). *A* Markov Decision Process *(MDP) is a tuple* $M = (\mathcal{S}, \mathcal{A}, P, r)$ *where* $\mathcal{S}$ *is a finite* set of states, $\mathcal{A}$ *is a set of* actions *where* $\mathcal{A}(s)$ *indicates the enabled actions in* $s \in \mathcal{S}$, $P : \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{P}(\mathcal{S})$ *is a* transition probability function *and* $R : \mathcal{S} \rightarrow \mathbb{R}$ *is a* reward function. $P(s, a)(s')$ *denotes the probability of transitioning from state* s *under action* a *to the state* s'.

Definition 8 ([32]). *An* Interval Markov Decision Process *(IMDP) is a tuple* $M_{\updownarrow} = (\mathcal{S}, \mathcal{A}, P_{\updownarrow}, R)$ *where* $\mathcal{S}$, $\mathcal{A}$, *and* r *are defined as in Definition 7,* $P_{\updownarrow} : \mathcal{S} \times \mathcal{A} \rightrightarrows \mathcal{P}(\mathcal{S})$ *is an* uncertain transition probability function *such that for all* s, s' *and* a *there exists* $0 \leq \underline{p} \leq \overline{p} \leq 1$ *such that* $P_{\updownarrow}(s, a)(s') = [\underline{p}, \overline{p}]$.

IMDPs are instances of Robust MDPs where the *ambiguity set* has a special structure, see [31]. A deterministic time-varying policy for an IMDP is a function $\pi : \mathcal{S} \times \mathbb{N}_0 \rightarrow \mathcal{A}$, with $\pi \in \Pi_{M_{\updownarrow}}$ being the admissible policy space [5]. Given a goal set $\mathcal{S}_G \subset \mathcal{S}$ and unsafe set $\mathcal{S}_U \subset \mathcal{S}$ a reach-avoid specification for an MDP state $\mathcal{S}$ [5] is denoted as

$$\begin{aligned} \varphi_s'^H \doteq \{s_0 s_1 \ldots s_{H-1} : & s_0 = s \wedge \exists a \in \mathcal{A} \,.\, P(s_i, a)(s_{i+1}) > 0, i < H - 1 \wedge \\ & \exists i < H \,.\, s_i \in \mathcal{S}_G \wedge \forall j \leq H \,.\, s_j \notin \mathcal{S}_U\}. \end{aligned} \tag{2.11}$$

We denote the *probability of satisfying a reach-avoid specification* given a policy π and a fixed transition probability function $P \in P_{\updownarrow}$ as $\mathbb{P}_{\pi,P}\{\varphi_s'^H\}$. In this dissertation, the optimal policy $\overline{\pi} \in \mathbf{\Pi}_{M_{\updownarrow}}$ for the IMDP maximises the worst-case probability of satisfying the specification with respect to all the possible transition probability functions coherent with the IMDP (see [32] for other notions of optimal policies). Formally,

$$\overline{\pi} \in \arg\max_{\pi \in \mathbf{\Pi}_{M_{\updownarrow}}} \min_{P \in P_{\updownarrow}} \mathbb{P}_{\pi,P}\{\varphi_s'^H\}. \tag{2.12}$$

Remark 3. Provably, the probability of satisfaction of a reach-avoid specification on an MDP can be equivalently expressed by computing the value function for a reward function defined as $r(s) = 1$ for all $s \in \mathcal{S}_G$ and $r(s) = 0$ elsewhere, and by making all states $s \in \mathcal{S}_U$ absorbing, i.e. $P(s, a)(s) = 1$, see [5].

2.6. Scenario Theory Background

Let $(\Delta, \mathcal{F}, \mathbb{P})$ be a probability space, where Δ is the sample space, endowed with a σ-algebra $\mathcal{F}$ and a probability measure $\mathbb{P}$; further, denote by Δ^N the N-Cartesian product of the sample space and with $\mathbb{P}^N$ its product measure. A point in $(\Delta^N, \mathcal{F}^N, \mathbb{P}^N)$ is thus a sample $(\delta_1, \dots, \delta_N)$ of N elements drawn independently from Δ according to the same probability $\mathbb{P}$. Each δ_i is regarded as an observation, or *scenario* [63, 64][1]. A set Θ, the decision space, contains the decisions, i.e. the optimization space – no particular structure is assumed for this set. To every $\delta \in \Delta$ there is associated a constraint set $\Theta_\delta \subseteq \Theta$ which identifies the decisions that are admissible for the situation represented by δ.

Typically, the scenario theory refers to an optimisation program, which computes θ_N^*, the solution of the optimisation program based on N samples. Once θ_N^* is computed, we are interested in assessing how it generalises to unseen scenarios $\delta \in \Delta$, or, rather, the probability of extracting a sample that violates the constraints defined by θ_N^*. We define:

Definition 9 (Violation [63]). *The violation probability of a given $\theta \in \Theta$ is defined as*

$$V(\theta) = \mathbb{P}[\, \delta \in \Delta \mid \theta \notin \Theta_\delta \,]. \tag{2.13}$$

$V(\theta)$ quantifies the probability with which a new randomly selected constraint Θ_δ is violated by θ. If $V(\theta) \leq \epsilon$, we say that θ is ϵ-robust against constraint violation. □

Notice that in general $V(\theta)$ is not directly computable since $\mathbb{P}$ is not known. From [64], under mild assumptions, a confidence bound can be derived as follows:

Theorem 1 (PAC bounds [64, Theorem 1]). *Let θ_N^* denote the solution to the scenario program. Given a confidence parameter $\beta \in (0,1)$, consider the polynomial equation in the v variable*

$$\binom{N}{k}(1-v)^{N-k} - \frac{\beta}{N}\sum_{m=k}^{N-1}\binom{m}{k}(1-v)^{m-k} = 0, \tag{2.14}$$

and let $\epsilon(k)$ be the unique solution over the interval $(0,1)$. Also, define $\epsilon(N) = 1$. For any $\mathbb{P}$, it holds

$$\mathbb{P}^N[V(\theta_N^*) \leq \epsilon(s_N^*)] \geq 1 - \beta, \tag{2.15}$$

where s_N^* is the so-called complexity of the solution – it represents the minimum number of constraints ($m \leq N$) that yield the same solution θ_N^*. □

Remark 4. In this work, the event space Δ is discrete; therefore, we refer to scenario theory for degenerate problems, as per [63, 64].

[1] As indicated in footnote 1 on [63] one could equivalently consider δ_i as independent random elements of the probability space $(\Delta, \mathcal{F}, \mathbb{P})$.

I

Deterministic Systems with Unknown Dynamics

3

Data-driven Abstractions for Verification of Dynamical Systems

In this chapter, we develop an approach for the construction of symbolic abstractions to verify temporal logic specifications for autonomous systems with unknown dynamics. Typically, abstractions require an exhaustive knowledge of the concrete model, which can be difficult to obtain in real-world applications, and perform computationally expensive reachability analysis. To overcome this, we propose to sample finite-length output trajectories of the system under investigation and build a data-driven model based on Strongest Asynchronous l-complete Abstractions. To this end, we introduce the notion of probabilistic behavioural inclusion. We provide probably approximately correct (PAC) guarantees that such an abstraction, constructed from experimental trajectories of finite length external behaviours, includes all behaviours of the concrete system. Finally, our approach is displayed with numerical examples.

This chapter is based on the publication [55].

3.1. Introduction

Here, we provide an overview of the related literature studying data-driven abstraction for deterministic systems. We categorise these works based on the data used: single transitions versus multi-step trajectories.
One-step: among the works in the first class, [16] proposes gridding the state space of a concrete system to obtain the state set of an abstraction, compute transitions between abstract states by sampling one-step transitions, and define a PAC alternating simulation relationship between the abstraction and an underlying deterministic system. In [18], approximations of monotone systems are computed, which are then used to build models for unknown monotone systems; their approach requires knowing partial derivatives of the transition map, which is estimated with scenario theory based on one-step transitions. In [19], the authors compute the growth bound of a system with additive disturbance from a data set of one-step transitions, which is then used to construct a model abstraction and synthesise a controller. The overarching role that data plays in these works is to substitute one-step conditions involving a universal quantifier "$\forall x$" over infinite sets, like the state space of a dynamical system or the domain of a disturbance, with a universal quantifier over a finite set of data points, namely the sampled data set. An example of such conditions is those for Lyapunov functions, where a certain decay rate must be guaranteed "for all the states" of a system, or Barrier function conditions [11], or even for the Lipschitzness of a transition function [18]. Typically, these conditions can be encoded as a robust optimisation program (ROP), which in turn can be substituted by a scenario optimisation program involving a finite set of samples. This last step requires some care: the dynamics of the system transform at each time step the distribution used to sample the data, hindering the connection between the solution of the scenario program and the original ROP. In Section 3.3.1 and 3.4.3 we illustrate this issue and why the approach presented in [16] fails to generate the PAC alternating simulation relation defined therein. However, a connection between the ROP and the scenario program can be safely established [65] when some knowledge of the dynamics is available, typically involving Lipschitz constants [19, 66, 67]. With the exception of [18], the literature cited above generates the abstractions by gridding the concrete system's state set and defines the abstraction's state set using reference points in each cell of the gridding. Instead, our construction does not require state measurements, nor does it require a metric on the abstraction's state set or any form of reachability analysis to define the abstract transitions.

Multi-step: in [26], a metric is defined between trajectories of a concrete system and its abstraction, and a scenario optimisation program is used to probabilistically bound the deviation between the two under the same random input. This work considers autonomous systems, addresses only safety verification, and does not treat control synthesis. In [21], the authors use sampled trajectories over a finite interval to fit a templated model approximating the concrete system over a metric output space, up to PAC guarantees, to solve safety problems, similarly to [26]. In [68] the authods build on [26] with a specification-centric approach for control synthesis, but relies on infinite-state abstractions, making control synthesis non-trivial [68, Section 3]. Both works assume a pre-existing abstraction and use state trajectories, where p-norm metrics are natural. In contrast, we construct specification-agnostic, finite-state abstractions directly from samples of output (and inputs when the TS is non-autonomous) sequences, without assuming a metric or topology on the output labels, and compatible with existing temporal-logic synthesis tools. This setting requires a fundamentally different treatment, leading us to propose, in this chapter, a new notion of Probabilistic Behavioral Inclusion, and, in Chapter 4, of Probabilistic (Alternating) Simulation Relation.

We focus on Strongest Asynchronous l-complete Abstractions (SAlCAs) [54, 61], which, unlike state-based abstractions, can be constructed from complete knowledge of a system's input-output trajectories, and offer properties well-suited to our purposes. With only partial knowledge, as when relying on sampled trajectories, we will show that additional challenges arise. SAlCAs have been applied in event-triggered control [69, 70]. Related work on data-driven memory-based Markov models [40, 41] underscores the importance of memory, which is also central to our approach.

In this chapter, we consider autonomous deterministic systems with unknown dynamics and random initialisation. We present a data-driven construction of finite abstractions, relying on SAlCAs and a notion of probabilistic behavioural inclusion, which captures the relation between a randomly sampled deterministic model and a transition system based on the collected system's behaviours. We demonstrate that obtaining a probabilistic behavioural inclusion can be written as a scenario optimisation program. We leverage scenario theory for degenerate problems to provide PAC guarantees for the inclusion of the concrete system's finite behaviours in those of the abstractions.

3.2. Notation and Modelling Framework

In this chapter, we address abstractions for autonomous deterministic systems described by

$$\Sigma := \begin{cases} x_{k+1} = f(x_k), \\ y_k = h(x_k), \end{cases} \tag{3.1}$$

where $x_k \in \mathcal{X} \subseteq \mathbb{R}^n$ is the plant's state at time $k \in \mathbb{N}_0$, n is the state-space dimension, $y_k \in \mathcal{Y}$ is the system output with $|\mathcal{Y}| < \infty$, and $x_0 \in \mathcal{X}$. We may think of the map $h(\cdot)$ as a *partitioning* map that returns the unique label (or index) corresponding to a cell of the induced partition. It is immediate to notice that the dynamical system in equation (3.1) can be equivalently framed as a deterministic, non-blocking, autonomous TS $S = (\mathcal{X}, \mathcal{X}, \delta, \mathcal{Y}, \mathcal{H})$ where $(x, x') \in \delta$ if and only if $x' = f(x)$ and $\mathcal{H} = h$. Following the notiation introduced in Chapter 2 for autonomous TSs, we recall; an *internal behavior of horizon* H of the autonomous TS S is a sequence $\mathtt{r} = x_0 x_1 \dots x_H$ that satisfies $x_0 \in \mathcal{X}$ and $(x_{i-1}, x_i) \in \delta$ for all $i = 1, \dots, H$. Similarly, a sequence $\mathtt{b} = y_0 y_1 \dots y_H$ is an *external behavior of horizon* H if there exists an internal behavior $\mathtt{r} = x_0 x_1 \dots x_H$ such that $y_i = \mathcal{H}(x_i)$ for all $i = 0, \dots, H$. Let $\mathtt{q} \in \mathcal{Y}^l$ be an output sequence and let $\mathtt{b}$ be an external behavior of horizon H: we say that $\mathtt{b}$ exhibits $\mathtt{q}$ if there exists $k \in [0, H - l + 1]$ such that $\mathtt{b}[k, k + l - 1] = \mathtt{q}$, denoted $\mathtt{b} \models F\mathtt{q}$. Given a set $B \subseteq \mathcal{B}_H(S)$ we write $B \models F\mathtt{q}$ if there exists $\mathtt{b} \in B$ such that

$\mathtt{b} \models F\mathtt{q}$. Our goal in this chapter is to construct an abstraction that behaviorally includes S. When the system dynamics are known, a common approach to obtain an abstraction that simulates, and, therefore, behaviorally includes S is to apply the *bisimulation algorithm* to compute the quotient transition system [71, 72]; we refer to such abstractions as Quotient-Based-Abstractions (QBAs). An alternative approach is to construct the SAlCA, which is guaranteed to behaviorally include S (but not necessarily simulate it). Let S_l denote the SAlCA of S constructed according to Definition 6. Since S is autonomous and deterministic, the relation defined by $\mathcal{L}_l \doteq \{(x, x') \in \mathcal{X} \times \mathcal{X} : \mathcal{L}_l(x) = \mathcal{L}_l(x')\}$ forms an equivalence relation on $\mathcal{X}$[1]. This allows for an intuitive graphical interpretation of the SAlCA, illustrated by the following example.

Example 2. Consider the autonomous deterministic bi-dimensional system defined by

$$x_{k+1} = \frac{1}{3}\begin{bmatrix} 1 & 2 \\ 1 & -1.8 \end{bmatrix} x_k$$

with $\mathcal{X} = [-1, 1]^2$, output set $\mathcal{Y} = \{1, 2, 3\}$, and output map h as shown in Figure 3.1. The central figure shows the equivalence relation $\mathcal{L}_1$ induced by the future CES defined in equation (2.10). The first symbol of each label indicates the current output, whereas the second symbol indicates the output generated in the next time step. The resulting abstraction S_1, shown on the right, illustrates the transitions based on the domino rule.

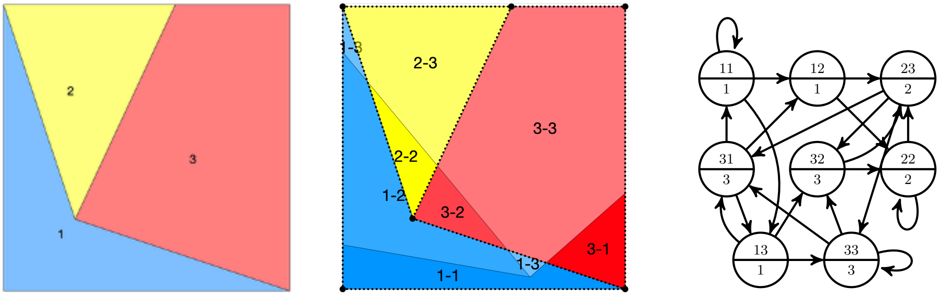


Figure 3.1.: Partition of the domain induced by the output map h (left), partition of the domain based on the equivalence classes of the l-sequences in set Υ_2 sequences (center), and the resulting SAlCA for $l = 2$ (right).

In contrast with QBAs, which require computing operations involving the dynamics (e.g. computing preimages of sets), constructing the SAlCA requires knowing the set Υ_l defined in equation (2.8), which in turn can be constructed solely from the external behaviours (i.e. the output sequences) of the concrete system. The SAlCA generates all possible behaviours of an underlying system, as long as its state space (i.e. the l-sequences) contains all the l-sequences that the system may

[1] When S is autonomous and deterministic, its SAlCA is bisimilar to the l-th QBA, see [54] for details.

exhibit. As such, we only need to collect the possible l-behaviours of a system to construct an l-complete model. This motivates our interest in SAlCAs: can we generate a data-driven SAlCA by collecting, through a sampling scheme, *all* the possible l-sequences? We formalise this in the following problem statement

Problem Statement 1. Given an **unknown deterministic autonomous system**, build an **abstraction from sampled external behaviors** such that, with high confidence, the probability of witnessing a behavior that is not included in the abstraction's behaviors is below a threshold value.

3.3. Sampling and Abstractions

Let us reframe the model in equation (3.1) as a dynamical system with deterministic dynamics and random initialisation. Formally, let $(\mathcal{X}, \mathcal{G}, \mu_x)$ be a probability measure space. We consider a random variable (RV) x_0 supported on $\mathcal{X}$. We summarise this by saying that x_0 is μ_x-distributed, or $x_0 \sim \mu_x$. We have

$$\Sigma := \begin{cases} x_{k+1} = f(x_k) \\ y_k = h(x_k), \\ x_0 \sim \mu_x. \end{cases} \tag{3.2}$$

From here on, we assume that f and h are measurable maps. This ensures that external behaviors of horizon H are measurable, that is $\mathcal{B}_H(S, x_0)$ is measurable with respect to μ_x.

In view of the system description in equation (3.2), we adapt the notion of behavioral inclusion introduced in Definition 2.

Definition 10 (Probabilistic Behavioural inclusion). *Consider two systems S_a and S_b with $\mathcal{Y}_a = \mathcal{Y}_b$. Let $x_0 \sim \mu_x$ be an RV denoting the random initial condition of S_a. We say that S_a is behaviourally included in S_b, or equivalently S_b probabilistically behaviorally includes S_a, with probability greater or equal than $1 - \epsilon$, denoted by if for $x_0 \sim \mathcal{P}$ it holds that:*

$$\mu_x \left[\mathcal{B}_H(S_a, x_0) \subseteq \mathcal{B}_H(S_b)\right] \geq 1 - \epsilon, \tag{3.3}$$

We denote this by $\mu_x[S_a \preceq_{\mathcal{B}_H} S_b] \geq 1 - \epsilon$.

Our objective is to abstract the system described in equation (3.2) by sampling external behaviors of horizon H from it: by leveraging scenario theory we will show how to construct a data-driven SAlCA guaranteed to probabilistically behaviorally include the original system.

3.3.1. Trajectory Sampling

Before proceeding, we discuss some implications of the choice of the sampling horizon H. The following example illustrates the shortcomings of using one-step transitions to infer longer horizon properties [16] (or considering transitions extracted from the same multi-step trajectory as independent [11]). In [16] the authors pursue a problem statement similar to ours, and can be summarized as follows: let S be an autonomous deterministic TS, grid its state space with a uniform grid and pick a reference point for each cell to construct the state set of the abstraction. Next, to compute transitions in the abstraction stemming from an abstract state $\hat{x}$ sample according to μ_x concrete states x_0 in the cell corresponding to $\hat{x}$, compute the image $f(x_0)$ of those points using the dynamics of S, and add to the list of transitions from $\hat{x}$ all the abstract states $\hat{x}'$ containing at least one of the images of the sampled points: this construction provides a guarantee that *when sampling according to* μ_x the abstraction will be able to simulate from $\hat{x}$ any transition of a related state in the concrete system, up to some probability. Next, repeat the same construction for $\hat{x}'$, and so on. While at first glance it may seem possible to concatenate guarantees on μ_x relative to $\hat{x}$ with guarantees on μ_x relative to $\hat{x}'$, there is a fundamental flaw: while x_0 is μ_x-distributed, $f(x_0)$ is not in general. This can be easily illustrated by the following example.

Example 3. Consider a partition of the state space as depicted in Fig. 3.2, and let us sample initial conditions from P_0, the partition in the bottom left corner. All trajectories starting from P_0 reach the blue region of P_4, where the blue region is strictly smaller than P_4. Let us further assume that every trajectory starting from the blue region reaches an unsafe region (P_6) in one step, whereas the trajectories starting from the white portion reach a safe set (P_7) in one step. If we sample transitions uniformly from the partition P_4, we see that *some* of these reach the safe set (the ones starting from the white portion) and others reach the unsafe set (starting from the blue portion). Whilst the probability of reaching the unsafe set from P_0 is 1, using the sampled one-step transitions yields a probability of reaching the unsafe set strictly smaller than 1 and actually suggests the safe set can be reached. □

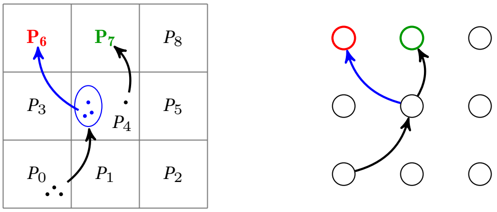


Figure 3.2.: Abstraction based on one-step transitions.

In other words, sampling one-step transitions can only provide probabilistic guarantees for a single time step of a trajectory, without further assumptions. In this chapter, we sample trajectories for a selected horizon H and obtain guarantees relative to the same horizon H. In Chapter 4 we show sufficient conditions that allow the extension of guarantees obtained from a data set of external behaviors of horizon H to guarantees valid for a longer horizon $H' > H$.

3.3.2. Data-driven Abstraction: Finite-time Guarantees

We collect N external behaviors of horizon H by sampling N initial conditions according to the distribution μ_x. We obtain the data set

$$D \doteq \bigcup_{i=1}^{N} \mathcal{B}_H(S, x_0^i). \tag{3.4}$$

Note since S is deterministic, $\mathcal{B}_H(S, x_0)$ is necessarily a singleton. For $l \leq H$ we denote by $\hat{\Upsilon}_l$ the set of all witnessed subsequences of length l, that is

$$\hat{\Upsilon}_l \doteq \bigcup_{\mathtt{b}\in D} \bigcup_{k\in[l-1,H]} \mathtt{b}[k-l+1,k]. \tag{3.5}$$

From now on we use the symbol $\hat{}$ as shown above to denote the quantities depending on the N samples drawn according to μ_x. We are now ready to define the data-driven SAlCA for an autonomous TS.

Definition 11 (Data-driven SAlCA)**.** *The TS $\hat{S}_l = (\hat{\mathcal{X}}, \hat{\mathcal{X}}_{l0}, \hat{\delta}_l, \mathcal{Y}, \mathcal{H}_l)$ is called the data-driven SAlCA of the autonmous TS S, where $\hat{\mathcal{X}}_l = \hat{\Upsilon}_l$, $\mathcal{X}_{l0} \doteq \{\mathtt{q} \in \hat{\Upsilon}_l : \exists \mathtt{b} \in D \,.\, \mathtt{b}[0, l-1] = \mathtt{q}\}$,*

$$\hat{\delta}_l = \{(k\sigma, \sigma k') \mid k, k' \in \mathcal{Y}, \sigma \in \mathcal{Y}^{l-1}, k\sigma, \sigma k', \in \hat{\mathcal{X}}_l\},$$

and the output map follows Definition 6. □

Remark 5 (Domino Completion). A *blocking* data-driven SAlCA may arise after collecting N samples, as depicted in Fig.3.3, where the state corresponding to $y_1y_2y_1$ has no outgoing transitions. The existence of $y_1y_2y_1$ *implies* the existence of at least one sequence starting with y_2y_1. Thus, we may add artificially *all* states corresponding to sequences y_2y_1*. We repeat the procedure until we obtain a non-blocking transition system. □

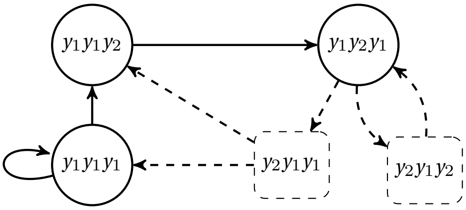


Figure 3.3.: Construction of a non-blocking automaton. Dashed lines indicate artificial states and transitions, added by the domino completion.

From here on, we assume that the data-driven SAlCA is non-blocking. Once we collect N trajectories from equation (3.2) and construct the corresponding data-driven SAlCA, we leverage the scenario theory to provide PAC bounds for $\hat{S}_l$ probabilistically behaviorally including S. In practice, we upper bound the probability of drawing an initial condition x_0 in the concrete system that produces an H-long behaviour which cannot be generated by the abstraction. As a first step towards the definition of the scenario problem, let us outline a chance-constrained program, where we assume to know both the probability distribution μ_x and the concrete system S. Let $K = |\mathcal{Y}^l|$ and denote by q_i as the i-th l-sequence of the set $\mathcal{Y}^l$, for $i \in \{1, ..., K\}$. For the sake of simplicity, let us formulate the program for $l = H$ (if $H > l$ the same reasoning applies), and $H \geq 1$. Let $\mathbb{B} = \{0, 1\}$, and define the random vector $\delta : \mathcal{X} \to \mathbb{B}^K$ as:

$$\delta = [X_1, X_2, ..., X_K]^T, \text{ where } X_i \sim \text{Bern}_{p_i}, \quad p_i = \mu_x[\, x_0 : \mathcal{B}_H(S, x_0) \models F\mathsf{q}_i \,], \tag{3.6}$$

for $i \in \{1, ..., K\}$ and $x_0 \sim \mu_x$. Note that, since the flow $f(\cdot)$ is deterministic, the vector δ depends entirely on x_0, hence the X_i's are mutually dependent. The vector δ has one unique entry equal to 1 when $l = H$, hence δ follows a categorical distribution; otherwise, it may have multiple ones.

Example 4. Let us assume a model has output space $\mathcal{Y} = \{a, b, c, d, e\}$, and let us consider $l = H = 2$, thus giving $K = 25$. Let us assume that the underlying system produces solely the 2-sequences $\Upsilon_2 = \{aa, bc, cd, de, eb\}$. The corresponding equivalence classes partition the domain into 5 regions, as depicted in Figure 3.4. The vector δ is composed of 25 elements, and p_1, i.e. the probability of sampling aa, is equal to 0.04. The probabilities of bc, ..., eb are respectively $p_2, \dots, p_5$, which are equal to 0.24. The remaining p_i for $i > 5$ are zero. At each initial condition x_0, randomly sampled within the domain, corresponds one 2-sequence and one vector δ. For instance, sampling the 2-sequence aa has probability 0.04 and is encoded as the vector $\delta = [1, 0, \dots, 0]$. □

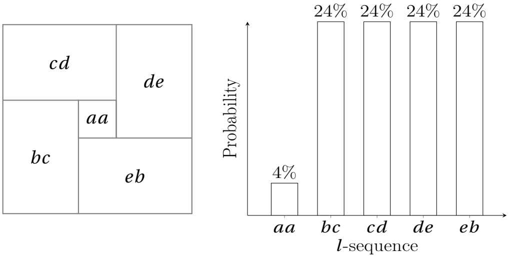


Figure 3.4.: Domain partition and probability of sampling the 2-sequences.

Remark 6. Given a horizon H, it would be possible to construct a data-driven SAlCA of S from the knowledge of for which $i \in \{1, \dots, K\}$ it holds that $p_i > 0$. That is for $\hat{S}_l$ constructed from the set $\hat{\mathcal{X}}_l = \{\mathsf{q} \in \Upsilon_l : \mu_x[\mathcal{B}_H(S, x_0) \models F\mathsf{q}] > 0\}$ it holds that $\mu_x[S \preceq_{\mathcal{B}_H} \hat{S}_l] = 1$. □

As $\mathbb{B}^K$ is a discrete set with cardinality not greater than K, we can rewrite the i-th l-sequence q_i as a one-hot vector δ_i of length K, as outlined in equation (3.6), and let $\Theta = \mathbb{R}^K$.

We are now ready to state the chance-constrained problem:

$$\begin{aligned} \min_{\theta\in\Theta,\mathbb{B}^K_\epsilon\subseteq\mathbb{B}^K} \quad & \mathbf{1}_K^{\mathrm{T}}\cdot\theta, \\ \text{s.t.} \quad & \sum_{i=1}^{K} p_i\cdot\mathbb{1}_{\mathbb{B}^K_\epsilon}(\delta_i)\geq 1-\epsilon, \\ & (\theta-\delta_i)\geq 0, \quad \text{for } \delta_i\in\mathbb{B}^K_\epsilon, \end{aligned} \tag{3.7}$$

where $\Theta = \mathbb{R}^K$, $\mathbf{1}_K$ is a column vector of ones with length K, and $\mathbb{1}_{\mathbb{B}^K_\epsilon}(\cdot)$ is the indicator function. The first constraint in the above optimisation ensures that $\mu_x(\mathbb{B}^K_\epsilon)\geq 1-\epsilon$. The program can be interpreted as follows. Given a threshold ϵ, find the maximum number of δ_i such that the sum of their probabilities is smaller than ϵ. Recalling Example 4, given $\epsilon = 0.05$, we shall find a set of events $\mathbb{B}^K_\epsilon$ such that its probability mass is not smaller than $1-\epsilon$. We then discard the vector in $\mathbb{B}^K$ encoding aa, since its probability is 0.04. Next, we build an abstraction composed of the four remaining l-sequences, which accounts for a cumulative probability $\mu_x(\mathbb{B}^K_\epsilon)\geq 1-\epsilon$.

The optimal solution of equation (3.7) is denoted $(\theta^*, \mathbb{B}^{K*}_\epsilon)$, where θ^* represents a vector encoding the l-sequences that satisfy the probabilistic constraint $\mu_x(\mathbb{B}^K_\epsilon)\geq 1-\epsilon$. We can thus construct an approximate SAlCA, where the states derive from θ^*, and the transitions are governed by the domino rule. In fact, if the i-th row of θ^* is non-zero ($\theta^*\in\{0,1\}^K$), then $p_i > 0$, and q_i is part of the state set of S^ϵ_l. Note that there could be more than one optimal solution, since $\mathbb{B}^K$ is a discrete (and finite) set - in that case, we choose the smallest θ^* in lexicographic order. The following result follows trivially:

Proposition 1. *Given ϵ, for any optimal solution $(\theta^*, \mathbb{B}^{K*}_\epsilon)$ to (3.7) and the corresponding approximate SAlCA, S^ϵ_l, for a new initial condition x_0 sampled from μ_x it holds that*

$$\mu_x\left[\mathcal{B}_H(S,x_0)\in\mathcal{B}_H(S^\epsilon_l)\right]\geq 1-\epsilon. \tag{3.8}$$

Proposition 1 states that the set of H-long behaviours which belong to S but do not belong to S^ϵ_l have a probability measure not bigger than ϵ. As outlined in Definition 10, we can rewrite equation (3.8) as

$$\mu_x[\mathcal{S}\preceq_{\mathcal{B}_H}\mathcal{S}^\epsilon_l]\geq 1-\epsilon.$$

Since the flow $f(\cdot)$ is assumed to be completely unknown, it is not possible to solve (3.7) exactly. We resort to scenario theory to find an approximate solution to (3.7): as both θ and δ take values over a discrete set, we refer to the general scenario theory [64].

Let us sample N i.i.d. initial conditions $\{x_0^i\}_{i=1}^N$ in the dynamical system, and consider the resulting H-long behaviours displayed by S, denoted by $\{\mathcal{B}_H(S,x_0^i)\}_{i=1}^N$. We directly obtain N i.i.d scenarios $\{\delta_i\}_{i=1}^N$ where

$$\delta_i(j) = \begin{cases} 1 \text{ if } \mathcal{B}_H(S,x_0^i)\models F\mathsf{q}_j, \\ 0 \text{ else }, \end{cases}$$

for $j \in \{1, ..., K\}$ and $\mathsf{q}_j \in \mathcal{Y}^l$. We formally define the scenario program as

$$\begin{aligned} \min_{\theta \in \Theta} \quad & \mathbf{1}_K^{\mathrm{T}} \cdot \theta \\ s.t. \quad & (\theta - \delta_i) \geq 0, \quad i = 1, \dots, N. \end{aligned} \tag{3.9}$$

The solution θ_N^* is trivially unique and in practice indicates which l-sequences $\hat{\Upsilon}_l$ were witnessed in the samples collected; the solution changes solely when we collect a new value for δ_i, previously unseen. Then, if $l = H$, the complexity s_N^* is equal to the number of 1's in the vector θ_N^*, or in other words, s_N^* is equal to the number of different l-sequences exhibited by the N H-sequences. If $H > l$, the complexity is equal to the cardinality of the smallest subset of the N H-sequences collected that yield the same solution to θ_N^*.

Remark 7. The scenario theory provides a bound on the probability of collecting a new, unseen, label from an unknown probability mass function [63, Section V.C.]. In this sense, we interpret program equation (3.9) as upper bounding the cumulative probability mass of the unseen support of a finite probability mass function from a set of i.i.d. realisations, in our setting being the l-sequences. As a second step, we employ the collected labels to construct a data-driven abstraction, which inherits the scenario probability guarantees. □

The scenario theory provides the following guarantees for the data-driven SAlCA.

Proposition 2. *Consider a confidence β, and N trajectories of length H collected from equation (3.2) and the corresponding data-driven SAlCA S_l^N based on the observed l-sequences. For a new initial condition $x_0 \sim \mu_x$ it holds that*

$$\mu_x^N\left[\mu_x[\mathcal{B}_H(S, x_0) \in \mathcal{B}_H(\hat{S}_l)] \geq 1 - \epsilon(s_{N,l}^*)\right] \geq 1 - \beta. \tag{3.10}$$

Proof. Referring to the scenario program in equation (3.9), the scenario theory ensures that the probability of sampling an x_0 that generates an unseen l-sequence, over time horizon H, is bounded by ϵ, with confidence $1 - \beta$. In other words, let $V(\theta_N^*) = \{x_0 \in \mathcal{X} : \mathcal{B}(S, x_0) \models F\mathsf{q}, \mathsf{q} \notin \hat{\Upsilon}_l\}$ be the violation set of the scenario program. The behaviors of data-driven SAlCA are obtained by the domino rule applied on the set $\hat{\Upsilon}_l$. Consequently, $x_0 \in V(\theta_N^*)$ if and only if $x_0 \in \{x_0 \in \mathcal{X} : \mathcal{B}(S, x_0) \notin \mathcal{B}(\hat{S}_l)\}$. Since $\mu_x^N[\mu_x[V(\theta_N^*)] \leq \epsilon] \geq 1 - \beta$ the thesis follows. □

According to notation introduced in Proposition 1, we denote this property by

$$\mu_x^N\left[\mu_x[\mathcal{S} \preceq_{\mathcal{B}_H} \mathcal{S}_l^N] \geq 1 - \epsilon(s_{N,l}^*)\right] \geq 1 - \beta. \tag{3.11}$$

Remark 8. Constructing the SAlCA using the set of sampled behaviors D entails solving a scenario program. Its complexity s_N^* is the cardinality of the smallest subset of D equation (3.4) which would result in the same set of all witnessed l-sequences $\hat{\Upsilon}_l$. To highlight the dependency of the complexity on the parameter l, from here on we denote it by $s_{N,l}^*$. One may use Theorem 1 using any upper bound of $s_{N,l}^*$; a close estimate of its value can be obtained using a greedy set cover algorithm, which runs in $O(NHl)$. We further discuss the computational complexity and effect of the parameter l in Chapter 4.

3.4. Experimental Evaluation

3.4.1. Linear Stable System

Let us consider the linear stable system

$$x_{k+1} = \frac{1}{3}\begin{bmatrix} 1 & 2 \\ -1.8 & 1 \end{bmatrix} x_k. \tag{3.12}$$

The state space $\mathcal{D} = [-1, 1]^2$ is partitioned into 81 regions by a uniform grid. We sample $N = 10^5$ initial conditions x_0 from the uniform distribution $\mathcal{U}_{\mathcal{D}}$, we collect trajectories of length $H = 9$, and we consider $l = 3$. We collect 454 l-sequences and construct the corresponding abstraction. Setting $\beta = 10^{-12}$, we compute the scenario bounds equation (3.10),

$$\overline{\epsilon} = \epsilon(s^*_{N,l}) = 3.54 \cdot 10^{-3}.$$

To verify these bounds empirically, we sample in addition $M = 10^6$ initial conditions, and get an empirical violation probability $\hat{V} \simeq 6 \cdot 10^{-6}$, a value well below the bounds.

Finer Partitioning

Consider again system equation (3.12), where $\mathcal{D}$ is uniformly partitioned into 81^2 regions. We sample $N = 10^6$ initial conditions $x_0 \sim \mathcal{U}_{\mathcal{D}}$, we collect trajectories of length $H = 9$, and we consider $l_1 = 3$ and $l_2 = H$, whose results are reported in Table 3.1, with the latter needing a domino completion adding 4857 sequences. In the first case we collect a total of 33541 l_1-sequences, resulting in a complexity $s^*_{N,l_1} = 17221$, in the latter case we collect 67099 l_2-sequences, resulting in a complexity $s^*_{N,l_2} = 67099$. We highlight the trade-off between l and $\overline{\epsilon}$: a smaller l provides a tighter $\overline{\epsilon}$, but it may entail a coarser over-approximation of the original system's external behaviors.

l	β	# sequences	s^*_N	$\overline{\epsilon}$	$\hat{V}$
3	10^{-12}	33541	17221	0.019	$1.4 \cdot 10^{-3}$
9	10^{-12}	67099	67099	0.069	$5.2 \cdot 10^{-3}$

Table 3.1.: Results with 81^2 partitions, for two values of l.

3.4.2. Path Planning

We consider a path planning problem, as depicted in Figure 3.5, where an agent lies within a 10 × 10 grid state space. It is tasked to reach the green target area – with coordinates $[7, 8] \times [7, 9]$ – whilst avoiding the obstacles (shown in red) and remaining within the borders of the state space. The agent's initial state is chosen uniformly at random within the white area of the state space, and it can choose among four actions (up, down, left, right) at every time step, in order to reach the target area. First, we run a standard Q-learning algorithm [73] to train the agent, with time horizon $H = 40$.

After the training, we use the newly synthesised control policy for a continuous-space experiment, where the agent can take positions over the continuous $[0, 10]^2$ domain, and its actions are obtained as a weighted average of the actions corresponding to the closest grid points. Formally, the action $a(x)$ results $a(x) = \sum_{d(x,g)<1} w_d \cdot a(g)$, where x, g are the locations in the continuous and grid space, respectively, $d(x, g)$ is the distance between points x and g, and w_d is a coefficient depending on the $d(x, g)$ – the weights w_d sum up to 1. We sample the system and collect the agent's position in terms of W, R, G labels (white, red, and green, respectively). We obtain $N = 10^4$ trajectories with horizon $H = 40$, and consider subsequences of length $l = 27$. We collect 27 different l-sequences, which can be obtained from a total of $s^*_N = 3$ trajectories. By setting the confidence to $\beta = 10^{-12}$, the scenario bound evaluates at

$$\epsilon(s^*_{N,l}) = 4.06 \cdot 10^{-3}.$$

We construct the data-driven SAlCA with the 27 l-sequences, and we verify a safety property: the system always reaches (and remains within) the target set, avoiding the obstacles. Hence, with confidence $1 - \beta$, the concrete model always reaches the target set with probability greater than or equal to $1 - \epsilon$ (for a horizon $H = 40$).

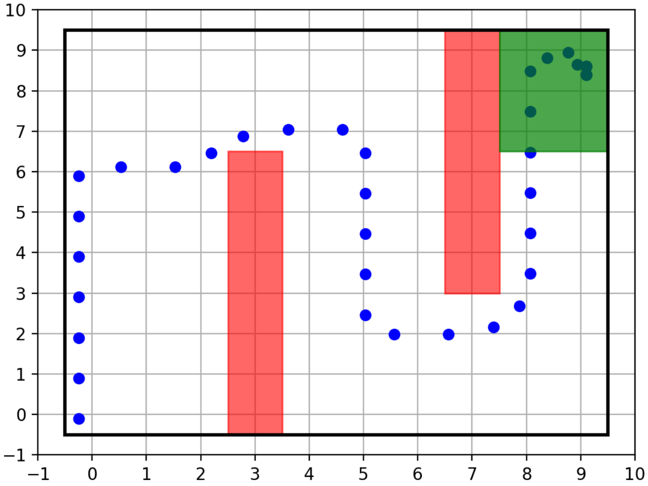


Figure 3.5.: Example trajectory of the path planning example (blue) with obstacles (red) and target area (green).

3.4.3. Concluding Example

We conclude this chapter by presenting a simple example that shows the role of the horizon H in Proposition 2, and why the extension of equation (3.11) to infinite horizons is challenging. The scenario theory bounds the probability of sampling a new, unseen l-sequence within horizon H. Let us consider a one-dimensional system,

$$x_{k+1} = \begin{cases} \frac{1}{2}x_k, & \text{if } x_k \in (\lambda, 1] \\ \frac{1}{2}x_k + \frac{1}{2} & \text{if } x_k \in [0, \lambda] \end{cases} \tag{3.13}$$

where $x_0 \sim \mu_x$ is uniformly distributed on $[0, 1]$, and $0 < \lambda < 2^{-4}$.

The state space $[0, 1]$ is partitioned into five regions according to $P_i = (2^{-i}, 2^{-i+1}]$ for $i = 1, \ldots, 4$, and $P_5 = [0, 2^{-4}]$, as shown in Fig. 3.6. Let us denote by y_i the output of the system if the state belongs to P_i. It is easy to see that this system visits infinitely many times all 5 partitions, no matter what the initial condition is. For instance, if $x_0 \in P_1$, the system generates the repeating sequence $(y_1 y_2 y_3 y_4 y_5^t)^\omega$, where t denotes the number of repetitions of the output y_5 and depends on λ, i.e. the width of the window that makes the system jump back to P_1.

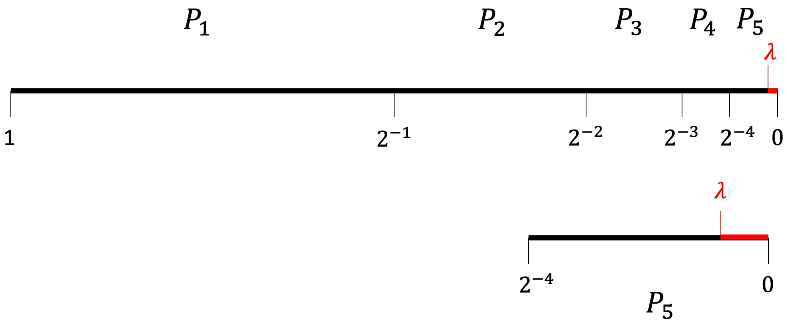


Figure 3.6.: Partition for the state space of the dynamical system in Example 3.4.3.

Let us now assume we sample uniformly $[0, 1]$ and take $l = H = 2$: this is equivalent to considering one-step transitions. It is easy to see that the probability of witnessing any $y_i y_{i+1}$ is equal to 2^{-i} for $i = 1, \ldots, 4$. The probability of sampling $y_5 y_5$ is $2^{-4} - \lambda$, and we witness the sequence $y_5 y_1$ only if $x_0 \in (0, \lambda] \subset P_5$, an event that occurs with probability λ, as summarised below:

$$\begin{aligned} &\mu_x[\mathcal{B}_2(x_0) \models F y_1 y_2] = 2^{-1}, && \mu_x[\mathcal{B}_2(x_0) \models F y_2 y_3] = 2^{-2}, \\ &\mu_x[\mathcal{B}_2(x_0) \models F y_3 y_4] = 2^{-3}, && \mu_x[\mathcal{B}_2(x_0) \models F y_4 y_5] = 2^{-4}, \\ &\mu_x[\mathcal{B}_2(x_0) \models F y_5 y_5] = 2^{-4} - \lambda, && \mu_x[\mathcal{B}_2(x_0) \models F y_5 y_1] = \lambda, \\ &\mu_x[\mathcal{B}_2(x_0) \models F y_5 y_2] = 0, \end{aligned} \tag{3.14}$$

Note that λ is a parameter of the system and can be arbitrarily small.

Let us now consider a longer horizon $H' > 2$ and focus on the sequence $y_5 y_1$. It is easy to see that the subsequence $y_5 y_1$ will *eventually* be generated by *every* trajectory, if the horizon H' is long enough[2]. Formally, for $H' \geq 1 - \lceil \log_2(\lambda) \rceil$ it

[2] The maximum number of steps to observe $y_5 y_1$ is $1 + \lceil -\log_2(\lambda) \rceil$.

holds that

$$\mu_x[\mathcal{B}_{H'}(x_0) \models F y_5 y_1] = 1.$$

This represents a challenge for our approach: whilst the probability of witnessing $y_5 y_1$ *as an initial sequence* is arbitrarily small, the probability of seeing $y_5 y_1$ over a sufficiently long horizon is actually 1. In general, we cannot use the scenario bounds *generated from samples over time horizon H* to infer properties over longer time horizons. Finally, observe that the system exhibits the behaviour $y_5 y_2$ if and only if it is initialised exactly at $x_0 = 0$. Such an event has a zero probability measure w.r.t. μ_x, hence, any data-driven SAlCA will almost surely not include such behaviour. In Chapter 4, we provide sufficient conditions that allow for the extension of the guarantees beyond the sampling horizon.

3.5. Conclusions

We have presented a method to construct a finite, data-driven abstraction of an unknown affine deterministic system under uniform random sampling of a set of initial conditions. Note that, with little effort, this can be generalised to other classes of distributions, e.g. piecewise constant. We introduce the notion of probabilistic behavioural inclusion, and use it to bound the probability of unseen behaviours of the concrete system. We then build an l-complete automaton that generates behaviours of the concrete system, based on trajectories up to time H.

4

Data-Driven Abstractions for Control Systems via Random Exploration

In this chapter, we present a data-driven framework for constructing finite-state abstractions of deterministic control systems with unknown dynamics, using only randomly sampled finite-horizon input-output trajectories and without requiring state measurements. Our approach builds Strongest Asynchronous l-Complete Abstractions from data and introduces a novel notion of probabilistic alternating simulation over finite horizons. This enables abstraction-based correct-by-construction synthesis of controllers directly from data, which, when refined to the original system, yields Probably Approximately Correct (PAC) guarantees for satisfying temporal logic specifications. The time horizon of validity of the guarantees can be extended beyond the sampling horizon when coarse information on the dynamics of the system is available. The method is specification-agnostic, supports tunable abstraction granularity, and avoids costly reachability analysis. We demonstrate its effectiveness on linear and nonlinear benchmarks, showing that meaningful guarantees can be achieved from finite data, making the framework attractive for safety-critical, black-box control applications where models are expensive or impossible to obtain.

This chapter is mainly based on the publication [56], and partially on [55].

4.1. Introduction

In this chapter, we consider deterministic control systems with unknown dynamics and a random initialisation. Our approach provides a construction of data-driven finite abstractions, built on the notions of alternating simulations and a particular class of abstractions known as Strongest Asynchronous l-complete Abstractions (or Approximations) (SAlCAs) [54]. The data-driven abstraction presented in this chapter enables the synthesis of a controller for the unknown system to solve Alternating-time Temporal Logics (ATL), including common reachability and reach-avoid specifications, over finite time horizons. We introduce a notion of probabilistic alternating simulation, instrumental in describing the relation between a deterministic (but randomly sampled) model and a transition system constructed upon the collected system's behaviors. Leveraging the scenario theory, we establish PAC guarantees for the inclusion of the concrete system's finite behaviors in those of the abstraction. We clarify the role of the trajectory length used for constructing the abstraction; successively, when coarse information of the dynamics is available, we characterise conditions (see Assumptions 1 and 2) to extend PAC guarantees over *longer* trajectory lengths. Together with Chapter 3, the present results enable the verification of properties, and synthesis of control policies, over an arbitrarily long finite horizon while preserving the PAC guarantees for several classes of nonlinear systems.

4.2. Notation and Modelling Framework

We consider a deterministic non-autonomous dynamical system

$$\Sigma(x) \doteq \begin{cases} x_{k+1} = f(x_k, u_k), \\ y_k = h(x_k), \end{cases} \tag{4.1}$$

where $x_k \in \mathcal{X} \subset \mathbb{R}^n$ is the system's state at time $k \in \mathbb{N}_0$ (natural numbers including zero), n is the state-space dimension, $x_0 \in \mathcal{X}$ is the initial state, $y_k \in \mathcal{Y}$ is the system output where $\mathcal{Y}$ is an arbitrary output set with cardinality $|\mathcal{Y}| < \infty$, $u_k \in \mathcal{U}$ is the system input at time k, $\mathcal{U}$ is a finite input set, i.e. $|\mathcal{U}| < \infty$. We denote as $\mathbf{u}_H \in \mathcal{U}^H$ a sequence of control inputs of length H. We assume that $f(\cdot, u)$ is measurable on the standard Borel space associated with $\mathbb{R}^n$ for all u. The system described in equation (4.1) can be equivalently described the non-blocking non-autonomous TS $S = (\mathcal{X}, \mathcal{X}, \mathcal{U}, \delta, \mathcal{Y}, \mathcal{H})$. Note that S has *free input*, i.e. $U_\delta(x) = \mathcal{U}$ for all x; consequently $|\mathrm{Post}_u(x)| = 1$ for every $x \in \mathcal{X}$ and $u \in \mathcal{U}$.

4.2.1. System Relations and SAlCA

Before presenting our methodology, first, we revisit and adapt the notions of simulation relation and alternating simulation relation. In Chapter 2 we have recalled a standard notion of simulation relation: the state of a concrete system is simulated by an abstract state if for every transition in the concrete system there exists a transition in the abstract system matching the observed output label while remaining within the relation; in other words, the relation is output preserving.

Below, we introduce a stronger notion of simulation relation, such that related transitions must, in addition, share the same input, as proposed in [54]. Similarly, we adapt the notion of alternating simulation relation.

Definition 12. (Simulation relation (SR) [54]) *Consider two systems S_a and S_b with $\mathcal{Y}_a = \mathcal{Y}_b$, and $\mathcal{U}_a = \mathcal{U}_b$. A relation $R \subseteq \mathcal{X}_a \times \mathcal{X}_b$ is a simulation relation from S_a to S_b w.r.t. $\mathcal{U} \times \mathcal{Y}$, written $S_a \preceq_S^R S_b$, if the following three conditions are satisfied:*

- $\forall x_{a0} \in \mathcal{X}_{a0}\ .\ \exists x_{b0} \in \mathcal{X}_{b0}$ *with* $(x_{a0}, x_{b0}) \in R$,
- $(x_a, x_b) \in R \implies \mathcal{H}_a(x_a) = \mathcal{H}_b(x_b)$,
- $(x_a, x_b) \in R \implies (U_{\delta_a}(x_a) \subseteq U_{\delta_b}(x_b) \wedge \forall u \in U_{\delta_a}(x_a)\ .\ (x_a, u, x'_a) \in \delta_a \implies \exists x'_b \in \mathcal{X}_b\ .\ (x_b, u, x'_b) \in \delta_b \wedge (x'_a, x'_b) \in R)$.

Definition 13. (Alternating simulation relation (ASR) [6]) *Consider two systems S_a and S_b with $\mathcal{Y}_a = \mathcal{Y}_b$, and $\mathcal{U}_a = \mathcal{U}_b$. A relation $R \subseteq \mathcal{X}_b \times \mathcal{X}_a$ is an alternating simulation relation from S_b to S_a w.r.t. $\mathcal{U} \times \mathcal{Y}$, written $S_b \preceq_{AS}^R S_a$, if the following three conditions are satisfied:*

- $\forall x_{b0} \in \mathcal{X}_{b0}\ .\ \exists x_{a0} \in \mathcal{X}_{a0}$ *with* $(x_{b0}, x_{a0}) \in R$,
- $(x_b, x_a) \in R \implies \mathcal{H}_a(x_b) = \mathcal{H}_b(x_a)$,
- $(x_b, x_a) \in R \implies (U_{\delta_b}(x_b) \subseteq U_{\delta_a}(x_a) \wedge \forall u \in U_{\delta_b}(x_b).\ (x_a, u, x'_a) \in \delta_a \implies \exists x'_b \in \mathcal{X}_b\ .\ (x_b, u, x'_b) \in \delta_b \wedge (x'_b, x'_a) \in R)$.

Observe that both definitions require that at every step, the inputs in the two systems must match. We rely on ASRs to synthesise controllers for specifications expressed in ATL [74] over finite horizons. In the numerical examples, we demonstrate our approach for reachability specifications.

In Chapter 3 we have motivated our focus on SAlCA for the fact that these models are guaranteed to behaviorally include the original system, simply by knowing the set of l-sequences that the system can exhibit. For the purpose of verification, constructing an explicit relation between concrete and abstract states was not necessary. For abstraction-based controller synthesis, however, relating abstract and concrete states is necessary in order to formally guarantee that an action selected in the abstraction can be refined to an action for the concrete system at a given state. The choice of abstracting the system as a SAlCA is motivated by the following properties: $(i.)$ Knowledge of Π_{l+1} equation (2.5), derived from all external behaviors of S, is sufficient to construct the SAlCA. $(ii.)$ The set of CES can be defined based on either the *past* or the *future* of a state. (iii) When CES is defined based on the *past*, as in equation (2.6), the relation

$$R = \{(x, \mathsf{q}) \in \mathcal{X} \times \mathcal{X}_l :\ \mathsf{q} \in \mathcal{E}_l(x)\} \tag{4.2}$$

is a SR (w.r.t. $\mathcal{U} \times \mathcal{Y}$) from S to S_l and the inverse relation R^{-1} is an ASR from S_l to S provided S has free input. This last claim is suggested in [54, Sec. V.D] but not formally proven. For completeness, we formalise the claim in the next proposition, proved in Appendix 4.8.2.

Proposition 3. *Consider a system S, let S_l be its SAlCA as per Definition 5 and let R be the relation defined in equation (4.2). Then R is SR from S to S_l w.r.t. $\mathcal{U} \times \mathcal{Y}$. Further, if S has free input, R^{-1} is an ASR from S_l to S w.r.t. $\mathcal{U} \times \mathcal{Y}$.*

In Section 4.3, we demonstrate how to construct a data-driven version of the relation described by equation (4.2) and establish a similar connection with the inverse relation. We conclude by noting that defining CES based on the past means that R does not need to be known explicitly. Once the SAlCA is constructed, if $(x, \mathsf{q}) \in R$ and applying input u causes the system to transition from x to x' with observation $\mathcal{H}(x')$, it automatically follows that there is a transition in S_l from q to q' where $\mathsf{q}'[l-1, l] \doteq \mathsf{q}|^{y}(l)u\mathcal{H}(x')$ and $(x', \mathsf{q}') \in R$.

Similarly to Chapter 3, we are interested in abstracting the unknown system in equation (4.1) from a set of sampled external behaviors of horizon H.

Problem Statement 2. Given a data set of input-output sequences with time horizon H randomly sampled from a deterministic system with random initial conditions and unknown dynamics, provide a suitable notion of alternating simulation relation enabling the construction of a specification-agnostic data-driven abstraction suitable for control synthesis on horizon H.

In addition, we characterise a set of sufficient conditions that allow for extending the guarantees to longer-than-sampling horizons.

Problem Statement 3. If the system satisfies suitable properties, extend the validity of the abstraction to horizons longer than H while relying on the same data set.

Section 4.3 addresses Problem 2 by introducing a data-driven approach to constructing abstractions for *unknown* (black-box) systems, using scenario theory to establish PAC-type guarantees for the horizon used for sampling; Section 4.5 addresses Problem 3 by extending these guarantees to longer time horizons by making specific assumptions about the dynamics.

4.3. Data-driven Abstractions

Constructing a SAlCA typically requires knowledge of all possible external behaviors of the system, which can be costly in practice. To address this, we aim to build an abstraction using only sampled external behaviors.
Consider a TS S_a as in equation (4.1) and the probability measure space $(\mathcal{X}_a, \mathcal{G}, \mu_x)$; we define the random variable $i^a_{\mathbf{u}_H} : \mathcal{X}_a \to \wp(\mathcal{I}_H(S_a))$

$$i^a_{\mathbf{u}_H} \doteq \mathcal{I}_H(S_a, x_{a0}, \mathbf{u}_H). \tag{4.3}$$

Equation equation (4.3) defines the set of full behaviors of S_a for a given $\mathbf{u}_H$, starting from the randomly chosen

initial condition x^a_0. For deterministic systems, $i^a_{\mathbf{u}_H}$ is a singleton, while for nondeterministic systems, it may be a set of behaviors. Given a system S_b and a relation $R \subseteq \mathcal{X}_a \times \mathcal{X}_b$ we define the random variable $i^b_{\mathbf{u}_H} : \mathcal{X}_a \to \wp(\mathcal{I}_H(S_b))$

$$i^b_{\mathbf{u}_H} \doteq \bigcup_{x_{b0} \in R^{-1}(x_{a0}) \cap \mathcal{X}_{b0}} \mathcal{I}_H(S_b, x_{b0}, \mathbf{u}_H), \tag{4.4}$$

which describes the set of full behaviors of S_b, for a given $\mathbf{u}_H$, starting from all initial conditions that are related to x_0^a through R.

In Section 4.3.1 we construct a data-driven approximation of the SAlCA by sampling randomly initial conditions *and* input sequences. By the scenario theory, we obtain statistical guarantees with respect to the sampling process of both, see for instance Lemma 2. As will be shown, decoupling the random sampling of initial conditions from the random sampling of input sequences is necessary for control synthesis. The following definition of *probabilistic* (alternating) simulation relations is formulated with this goal, in particular, quantifying the probability of drawing an initial condition for which there exists *any* input sequence such that, when applied to the concrete system and its abstraction, it generates behaviors not in the relation.

Definition 14. (Probabilistic simulation relation (PSR)) *Consider two non-blocking systems S_a and S_b with $\mathcal{Y}_a = \mathcal{Y}_b$ and $\mathcal{U}_a = \mathcal{U}_b = \mathcal{U}$, and a relation $R \subseteq \mathcal{X}_a \times \mathcal{X}_b$. Let $x_{a0} \sim \mu_x$ be the random initial condition of S_a; R is a probabilistic simulation relation from S_a to S_b with respect to $\mathcal{U} \times \mathcal{Y}$ until horizon H with probability not less than $1 - \epsilon$ if*

$$\mu_x[x_{a0} \in \mathcal{V}(S_b, R, H)] \leq \epsilon, \tag{4.5}$$

where the violation set $\mathcal{V}(S_b, R, H)$ is defined

$$\begin{aligned}\mathcal{V}(S_b, R, H) \doteq \{x_{a0} : \exists \mathbf{u}_H \in \mathcal{U}^H \ .\ i^a_{\mathbf{u}_H} \neq \emptyset \ \wedge \exists \mathtt{r}_a \in i^a_{\mathbf{u}_H} \ . \\ (\nexists \mathtt{r}_b \in i^b_{\mathbf{u}_H} \ .\ \mathcal{H}_a(\mathtt{r}_a) = \mathcal{H}_b(\mathtt{r}_b) \wedge \forall k \geq 0 \ .\ (\mathtt{r}_a(k), \mathtt{r}_b(k)) \in R)\},\end{aligned} \tag{4.6}$$

with $i^a_{\mathbf{u}_H}$ and $i^b_{\mathbf{u}_H}$ defined as in equation (4.3) and equation (4.4). More compactly, we write $\mu_{x_a}(S_a \preceq^R_{S_H} S_b) > 1 - \epsilon$, where $\preceq^R_{S_H}$ highlights that the probabilistic simulation relation is referring to the relation R and to a time horizon H.

Definition 15. (Probabilistic alternating simulation relation (PASR)) *Under the same conditions of Definition 14, we say that $Z \subseteq \mathcal{X}_b \times \mathcal{X}_a$ is a probabilistic alternating simulation relation from S_b to S_a with respect to $\mathcal{U} \times \mathcal{Y}$ until horizon H with probability greater than $1 - \epsilon$ if*

$$\mu_x[x_{a0} \in \mathcal{Q}(S_b, Z, H)] \leq \epsilon, \tag{4.7}$$

where the violation set $\mathcal{Q}(S_b, Z, H)$ is defined

$$\begin{aligned}\mathcal{Q}(S_b, Z, H) \doteq \{x_{a0} : \exists \mathbf{u}_H \in \mathcal{U}^H \ .\ i^b_{\mathbf{u}_H} \neq \emptyset \wedge (i^a_{\mathbf{u}_H} = \emptyset \vee \exists \mathtt{r}_a \in i^a_{\mathbf{u}_H} \ . \\ (\nexists \mathtt{r}_b \in i^b_{\mathbf{u}_H} \ .\ \mathcal{H}_a(\mathtt{r}_a) = \mathcal{H}_b(\mathtt{r}_b) \wedge \forall k \geq 0 \ .\ (\mathtt{r}_a(k), \mathtt{r}_b(k)) \in Z)\}.\end{aligned} \tag{4.8}$$

where $i^b_{\mathbf{u}_H}$ is defined as per equation (4.4) considering the relation $Z^{-1} \subseteq \mathcal{X}_a \times \mathcal{X}_b$. More compactly, we write $\mu_{x_a}(S_b \preceq^Z_{AS_H} S_a) > 1 - \epsilon$, where $\preceq^Z_{AS_H}$ highlights that the probabilistic alternating simulation relation is referring to the relation Z and to a time horizon H.

Expressions (4.5-4.6) bound the probability of drawing an initial condition x_{a0} such that there exists an input sequence $\mathbf{u}_H$, *admissible for* S_a, which generates at least one full H-behavior $\mathtt{r}_a$ in S_a that cannot be related to any $\mathtt{r}_b$ in S_b by R. Expressions equation (4.7)-equation (4.8) bound the probability of drawing an initial condition x_{a0} such that there exists an input sequence $\mathbf{u}_H$, *admissible for* S_b, which either generates at least one full H-behavior $\mathtt{r}_a$ that can't be related to any $\mathtt{r}_b$ by Z, or no full H-behavior at all, if $\mathbf{u}_H$ is inadmissible for S_a. The key difference is in the validity of the input sequence: in the first case, $\mathbf{u}_H(k)$ must belong to the set $U_a(\mathtt{r}_a(k))$, ensuring $i^a_{\mathbf{u}_H} \neq \emptyset$, in the second case it belongs to $U_b(\mathtt{r}_b(k))$, ensuring $i^b_{\mathbf{u}_H} \neq \emptyset$. The condition $i^a_{\mathbf{u}_H} = \emptyset$ in equation (4.8) captures initial conditions where $\mathbf{u}_H$ generates behaviors in $i^b_{\mathbf{u}_H}$ but not in $i^a_{\mathbf{u}_H}$, that is $\mathbf{u}_H$ is inadmissible for S_a starting at x_{a0}.

Remark 9. The definition of PSR and PASR is a natural extension of SR and ASR to a setting where the initialisation of a deterministic (concrete) system is random and the horizon over which its internal behaviors must remain related to its abstraction's internal behaviors is finite. If the violation sets in equation (4.6) and equation (4.8) are empty for every choice of H, then $\epsilon = 0$, and Definitions 12 and 13 are equivalent to Definitions 14 and 15 respectively, modulo zero measure behaviors. In this scenario, the difference between the two pairs of definitions is purely notational: the first two use transition-based requirements, while the latter two employ trajectory-based requirements, as in [74, Lem. 1], which are often easier to handle in finite horizon settings.

4.3.1. Constructing the Data-driven Abstraction

Consider the probability spaces $(\mathcal{X}, \mathcal{G}, \mu_x)$, $(\mathcal{U}^H, \mathcal{F}, \mu_{\mathbf{u}_H})$ and denote by $(\mathcal{P}, \mathcal{W}, \mu_p)$ the product space, $\mathcal{W}$ is the product σ-algebra of $\mathcal{G}$ and $\mathcal{F}$, and μ_p is the product measure of μ_x and $\mu_{\mathbf{u}_H}$: when sampling the system S for $x \in \mathcal{X}$ and $\mathbf{u}_H \in \mathcal{U}^H$ we assume to have access only to the external H-behavior of the system, that is $\mathcal{B}_H(S, x, \mathbf{u}_H)$: note since S is deterministic, the latter is necessarily a singleton. In order to construct a data-driven SAlCA of S we pursue a *random exploration* of the system's external behaviors, using random initial conditions and input sequences. In other words, we draw N i.i.d. pairs $(x^i, \mathbf{u}^i_H)$ according to the product probability measure μ_p, and we obtain the set of sampled external behaviors

$$D \doteq \{\mathcal{B}_H(S, x^i, \mathbf{u}^i_H) \ : \ i = 1, 2, ...N\}. \tag{4.9}$$

For $l < H$ we denote by $\hat{\Pi}_l$ the set of all witnessed subsequences of length l, that is

$$\hat{\Pi}_l \doteq \bigcup_{\mathtt{b} \in D} \bigcup_{k \in [0,H]} \mathtt{b}[k-l, k]. \tag{4.10}$$

From now on we use the symbol $\hat{\cdot}$ as shown above to denote the quantities depending on the N samples drawn according to μ_p. We are now ready to define the data-driven SAlCA.

Definition 16. (Data-driven SAlCA) *Given $\hat{\Pi}_{l+1}$, the TS $\hat{S}_l = (\hat{\mathcal{X}}_l, \hat{\mathcal{X}}_{l0}, \mathcal{U}, \hat{\delta}_l, \mathcal{Y}, \mathcal{H}_l)$ is called the data-driven (strongest asynchronous) l-complete abstraction (SAlCA) of S, where $\hat{\mathcal{X}}_l \doteq \hat{\Pi}_l$ is the state set, $\hat{\mathcal{X}}_{l,0} \doteq \{\mathsf{q} \in \hat{\Pi}_l : \exists \mathsf{b} \in D \ . \ \mathsf{b}[-l, 0] = \mathsf{q}\}$ is the initial set, and the transition relation is given by*

$$\begin{aligned}\hat{\delta}_l \doteq \{(\mathsf{q}, u, \mathsf{q}') \in \hat{\mathcal{X}}_l \times \mathcal{U} \times \hat{\mathcal{X}}_l \ : \ \exists \mathsf{q}'' \in \hat{\Pi}_{l+1} \ . \\ \mathsf{q}''[0, l] = \mathsf{q} \wedge \mathsf{q}''[1, l+1] = \mathsf{q}' \wedge \mathsf{q}''|^u(l) = u\}.\end{aligned}$$

Trivially, $\hat{\Pi}_{l+1} \subseteq \Pi_{l+1}$. In the following, we show how to derive a PASR from $\hat{S}_l$ to S until horizon H with probability not lower than $1 - \epsilon$ up to some confidence β.

Lemma 1. *Consider S, its data-driven SAlCA $\hat{S}_l$ constructed from the set $\hat{\Pi}_{l+1}$ and the relation*

$$\hat{R} \doteq \{(x, \mathsf{q}) \in \mathcal{X} \times \hat{\mathcal{X}}_l : \ \mathsf{q} \in \mathcal{E}_l(x)\}. \tag{4.11}$$

If $(x, \mathsf{q}) \in \hat{R}$ and $(x, u, x') \in \delta$ with $\mathcal{H}(x') = y$ then there exists $(\mathsf{q}, u, \mathsf{q}') \in \hat{\delta}_l$ with $\mathcal{H}_l(\mathsf{q}') = y$ and $(x', \mathsf{q}') \in \hat{R}$ if and only if $\mathsf{q} \cdot \mathsf{q}'[l-1, l] \in \hat{\Pi}_{l+1}$ with $\mathsf{q}'|^u(l-1) = u$.

Lemma 1 is a simple consequence of the Definition 16, and its proof is omitted. It states that if a state trajectory of S results in the same external behavior of a state trajectory of the SAlCA at every time step, the pair given by the state of the former and the state of the latter are in $\hat{R}$.

Lemma 2. *Consider S, a confidence β and the data-driven SAlCA constructed from $\hat{\Pi}_{l+1}$. It holds that*

$$\mu_p^N[\mu_p[\mathcal{B}_H(S, x, \mathbf{u}_H) \notin \mathcal{B}_H(\hat{S}_l)] \leq \epsilon] \geq 1 - \beta, \tag{4.12}$$

where $\epsilon \doteq \epsilon(s^*_{N,l})$ is defined as in Theorem 1.

Lemma 2, detailed in Appendix 4.8.2, provides an upper bound on the probability that a sampled external behavior from the concrete system does not belong to the set of external behaviors of the data-driven SAlCA. This is a consequence of the probabilistic behavioral inclusion notion presented in Chapter 3.
Regarding the sampling complexity of equation (4.12), while the scenario approach for convex optimization with non-degenerate constraints has a sample complexity of $O(\epsilon^{-1} \ln(\beta^{-1}))$ [75], we consider a finite sample space ($\mathcal{B}_H(S)$) and use recent results on degenerate problems [64]. Note that, due to the degeneracy of the problem at hand (the distribution is *discrete*) we must use the *a-posteriori* scenario results. For fixed β and N, the violation probability is higher when all N drawn symbols are distinct ($s^*_{N,l} = N$) than when they are identical ($s^*_{N,l} = 1$) [63, Section V.C.].
In our case, two factors influence the complexity $s^*_{N,l}$. The first factor is l: reducing l decreases the alphabet size and $s^*_{N,l}$, tightening the scenario bounds: referring to Figure 2.3, note how the state set (and the number of transitions) of S_0 is smaller than that of S_1. However, a smaller l may reduce the SAlCA's precision, as it introduces potential spurious behaviors not present in the original system. For example, in Figure 2.3, the behavior $y_1 u_a y_2 u_b y_2 u_a y_1$ is possible in S_1 and S_0 but

not in the original system, while $y_1 u_a y_2 u_a y_1$ is possible in S_0 but not in S_1 or the original system. Larger l values reduce such spurious behaviors. The second, is the "intrinsic richness" of a system's external behaviors: in a hypothetical situation where two systems S and S' have $|\mathcal{B}_H(S)| > |\mathcal{B}_H(S')|$ and the behaviors of each system are equally likely to be sampled, we can expect that the data-driven SAlCA of S will result in a higher complexity than that of S'. The behavioral richness of a system depends both on the state dynamics and on the output map. When the latter can be selected, a reasonable choice is to define it based on the specifications of interest, as we show in Section 4.6.
Finally, we show that Lemmas 1 and 2 allow us to bound the probability measure of pairs $(x, \mathbf{u}_H)$ resulting in an internal behavior of S that cannot be related by $\hat{R}$ to one of the abstraction $\hat{S}_l$.

Proposition 4. *Consider S , the product probability space $(\mathcal{X} \times \mathcal{U}^H \mathcal{W}, \mu_p)$, and the data-driven SA$l$CA $\hat{S}_l$ obtained from the set $\hat{\Pi}_{l+1}$. Given a confidence parameter β, it holds that*

$$\mu_p^N [\mu_p [(x, \mathbf{u}_H) \in \overline{\mathcal{V}}(\hat{S}_l, \hat{R}, H)] \leq \epsilon] \geq 1 - \beta \tag{4.13}$$

where $\epsilon \doteq \epsilon(s^*_{N,l})$ as defined in equation (2.15), and

$$\overline{\mathcal{V}}(\hat{S}_l, \hat{R}, H) \doteq \{(x, \mathbf{u}_H) : \exists \mathtt{r} \in i_{\mathbf{u}_H} \ . \ (\nexists \mathtt{r}_l \in i^l_{\mathbf{u}_H} \ . \tag{4.14}$$

$$\mathcal{H}(\mathtt{r}) = \mathcal{H}_l(\mathtt{r}_l) \wedge \forall k \geq 0 \ . \ (\mathtt{r}(k), \mathtt{r}_l(k)) \in \hat{R})\} \tag{4.15}$$

Proof. First, we show that the probability of drawing a pair $(x, \mathbf{u}_H)$ resulting in an external H-behavior not contained in the set of all external H-behaviors of $\hat{S}_l$ is bounded by ϵ with confidence $1 - \beta$. Consider the random variable $w_{x,\mathbf{u}_H} : \mathcal{X} \times \mathcal{U}^H \to \mathcal{B}_H(S)$ defined as $w(x, \mathbf{u}_H) \doteq \mathcal{B}_H(S, x, \mathbf{u}_H)$. From Lemma 2, $\mu_p^N [\mu_p [(x, \mathbf{u}_H) : w(x, \mathbf{u}_H) \notin \mathcal{B}_H(\hat{S}_l)] \leq \epsilon] \geq 1 - \beta$. Suppose that $(x, \mathbf{u}_H) \in \overline{\mathcal{V}}(\hat{S}_l, \hat{R}, H)$: by Lemma 1, this implies that $w(x, \mathbf{u}_H) \notin \mathcal{B}_H(\hat{S}_l)$. Since the negation of the latter holds with probability greater than $1 - \epsilon$ then the negation of the former holds with at least the same probability, up to a confidence of at least $1 - \beta$. □

The distinction between the sets $\overline{\mathcal{V}}(\hat{S}_l, \hat{R}, H)$ and $\mathcal{V}(\hat{S}_l, \hat{R}, H)$, as defined in equation (4.6) (considering $S_b = \hat{S}_l$) is subtle but important. The first set includes *pairs* $(x, \mathbf{u}_H)$ that lead to an external behavior in the concrete system absent in the data-driven SAlCA. The second set includes *states* x for which there *exists* an input sequence $\mathbf{u}_H$ causing an external behavior in the concrete system not present in the data-driven SAlCA. However, for the guarantees in Proposition 4 to hold, pairs $(x, \mathbf{u}_H)$ must be drawn according to the product measure μ_p, meaning both the initial condition x and the input sequence $\mathbf{u}_H$ must be randomly sampled. Since our goal is to create an abstraction suitable for control, we need the flexibility to select inputs arbitrarily after constructing the data-driven SAlCA, rather than being constrained by the probability distribution $\mu_{\mathbf{u}_H}$. We expand the result of Proposition 4 to cover arbitrarily chosen control sequences.

Proposition 5. *Consider S, the product probability space $(\mathcal{X} \times \mathcal{U}^H \mathcal{W}, \mu_p)$ of $(\mathcal{X}, \mathcal{G}, \mu_x)$ and $(\mathcal{U}^H, \mathcal{F}, \mu_{\mathbf{u}_H})$, and the data-driven SAlCA $\hat{S}_l$ obtained from the set $\hat{\Pi}_{l+1}$. If $\mu_{\mathbf{u}_H}$ is uniformly distributed, given a confidence β, with $\overline{\epsilon} = \min(1, \epsilon|\mathcal{U}^H|)$ it holds that*

$$\mu_p^N[\mu_x[S \preceq^{\hat{R}}_{S_H} \hat{S}_l] > 1 - \overline{\epsilon}] \geq 1 - \beta. \tag{4.16}$$

Proof. For brevity, let $\mathcal{V}$ and $\overline{\mathcal{V}}$ represent the sets $\mathcal{V}(\hat{S}_l, \hat{R}, H)$, as defined in equation (4.6), and $\overline{\mathcal{V}}(\hat{S}_l, \hat{R}, H)$, as defined in equation (4.14). In Proposition 4 we have shown that $\mu_p^N[\mu_p[(x, \mathbf{u}_H) \in \overline{\mathcal{V}}] \leq \epsilon] \geq 1 - \beta$. We define the set $J(x) \doteq \{\mathbf{u}_H \in \mathcal{U}^H : (x, \mathbf{u}_H) \in \overline{\mathcal{V}}\}$. Let c and z be the densities of μ_x and $\mu_{\mathbf{u}_H}$ respectively. Then,

$$\mu_p[(x, \mathbf{u}_H) \in \overline{\mathcal{V}}] = \int_{\mathcal{V}} c(x) \int_{J(x)} z(\mathbf{u}_H) d\mathbf{u}_H dx \tag{4.17}$$

$$= \int_{\mathcal{V}} c(x) \frac{|J(x)|}{|\mathcal{U}^H|} dx \geq \int_{\mathcal{V}} \frac{c(x)}{|\mathcal{U}^H|} dx = \frac{\mu_x[x \in \mathcal{V}]}{|\mathcal{U}^H|}, \tag{4.18}$$

from which follows the thesis $\mu_x[x \in \mathcal{V}] \leq \epsilon|\mathcal{U}^H|$, holding with a probability of at least $1 - \beta$. Note that inequality equation (4.18) is tight: it becomes an equality if each violating initial condition has exactly one input sequence $\mathbf{u}_H \in \mathcal{U}^H$ generating a new behavior. □

Proposition 5 establishes a PAC bound on the probability that a sampled initial condition lies in $\mathcal{V}$, where there exists an input $\mathbf{u}_H$ yielding an external behavior not related by $\hat{R}$ to $\hat{S}_l$. For $x \notin \mathcal{V}$, arbitrary input sequences generate behaviors captured by the abstraction, ensuring a PSR from the concrete system to $\hat{S}_l$ with high confidence. This proposition is the main result of this section and is crucial because it *i*) decouples the sampling procedure from the control design: once the abstraction is built, control inputs in the abstraction can be picked arbitrarily; *ii*) enables control synthesis with guarantees, bounding the probability that an initial condition leads to a behavior not captured by the abstraction; *iii*) enables PASR-based controller refinement.

Remark 10. Proposition 5 assumes that $\mu_{\mathbf{u}_H}$ is uniform. The same reasoning applies to other measures, provided the smallest non-zero probability assigned to an element $\mathcal{U}^H$ is known. However, if $\mu_{\mathbf{u}_H}$ is selectable, the bound in equation (4.18) is the tightest with the uniform measure.

Next, we claim that $\hat{Z} \doteq (\hat{R})^{-1}$ defines a PASR.

Corollary 1. *Under the assumptions of Proposition 5, $\hat{Z}$ defines a PASR from $\hat{S}_l$ to S with respect to $\mathcal{U} \times \mathcal{Y}$ until horizon H with probability not less than $1 - \overline{\epsilon}$, with $\overline{\epsilon} = \min(1, \epsilon|\mathcal{U}^H|)$, with confidence greater than $1 - \beta$, that is*

$$\mu_p^N[\mu_x[\hat{S}_l \preceq^{\hat{Z}}_{AS_H} S] > 1 - \overline{\epsilon}] \geq 1 - \beta. \tag{4.19}$$

Proof. From Proposition 5 we know with confidence $1-\beta$ that $\mu_x[x \in \mathcal{V}(\hat{S}_l, \hat{R}, H)] \leq \epsilon|\mathcal{U}^H|$. Hence, it is sufficient to show that $\mathcal{V}(\hat{S}_l, \hat{R}, H) \supseteq \mathcal{Q}(\hat{S}_l, \hat{Z}, H)$. Suppose that $x \in \mathcal{Q}(\hat{S}_l, \hat{Z}, H)$: since S has free input, any $\mathbf{u}_H$ satisfying the first line of equation (4.8) is an admissible sequence of inputs for S, hence $i_{\mathbf{u}_H} \neq \emptyset$, which implies that $x \in \mathcal{V}(\hat{S}_l, \hat{R}, H)$. □

Once we have established the PASR between the data-driven SAlCA and the concrete system, we can adopt classical synthesis methods to design a controller enforcing a desired specification, see e.g. [6]. Refining a controller from the data-driven SAlCA to the original system is trivial: according to Definition 15, the input dictated by the abstraction coincides with the one applied to the original system, details in Appendix 4.8.1.

Remark 11. The overall procedure of generating the abstractions entails four steps: I) sampling N trajectories of length H ($O(NH)$), II) generating $\hat{\Pi}_{l+1}$ and constructing $\hat{S}_l$ ($O(NHl)$) , III) scenario complexity computation by the greedy set cover algorithm ($O(NHl)$) [76], and IV) scenario bounds computation (this can be cheaply tabulated in advance). The overall time complexity is dominated by $O(NHl)$.

4.4. On Modelling Initial Conditions as Random

We briefly discuss the relevance of modelling equation (4.1) as a deterministic system with random initialisation. Representing initial conditions by probability distributions is natural in many contexts. A common example is found in robotics, particularly in Simultaneous Localisation and Mapping (SLAM), where the initial position of the robot is typically modelled as random. Another example arises in the study of chaotic systems, where randomising initial conditions provides a tractable way to model their trajectories, in contrast with the highly sensitive and unpredictable behavior generated by fixed initial conditions [77]. Similarly, in System Identification, random initialisation is a standard approach to sample a variety of system behaviours.

In our setting, the motivation for random initialisation is the following. Throughout Chapter 3 and up to Section 4.3.1, we assume no prior knowledge of the system to be abstracted. Given an initial condition (and input sequence, if the system is non-autonomous), the system produces an external behaviour in the form of sequences of output (and input) labels. While the state space is typically a compact subset of Euclidean space, we do not observe the state trajectory directly, and the outputs need not lie in a metric space. Suppose instead that we were to fix the initial condition deterministically and observe a single output sequence. In the absence of assumptions such as Lipschitz continuity of the dynamics, we would have no way of estimating how the output would vary under perturbations of the initial state. Modelling the initial condition as a distribution therefore provides a meaningful and, in our view, appealing way to capture the system's range of behaviours.

In the next section, we introduce Lipschitz-like assumptions that allow us to reason about how the guarantees extend when increasing the horizon of validity.

When coarse information about the dynamics is available and the state is observable, an alternative approach is to grid the set of initial conditions (deterministic initialization) and exploit, for instance, Lipschitz constants and growth bounds to approximate reachable sets [20].

4.5. Beyond the Sampling Horizon

So far, the only assumption made on the system's dynamics is that it has free input and that it is deterministic; this is general enough to be applied to most black-box or *unknown* systems. We have shown that, with a dataset of trajectories of length H, we can derive a PAC bound for a PASR between the abstraction and the concrete system for horizon H. In this section, we extend these guarantees to horizons beyond H. As demonstrated in [78], without further assumptions, nontrivial bounds for larger horizons are unattainable. We begin by providing the following definition used later in this section We provide the following definition, later used in the

Definition 17. *Let* $(\mathcal{X}, d_X)$ *and* $(\mathcal{U}, d_U)$ *be complete metric spaces. Let there exist* $m_X > 0$ *such that for all* $x, x' \in \mathcal{X}$ *and* $u \in \mathcal{U}$ $m_X d_X(x, x') \leq d_X(f(x,u), f(x',u))$. *Then,* f *is* Lipschitz invertible.
The map f *in equation* (3.1) *is* uniformly contracting *w.r.t.* x *if there exists* $0 < l_X < 1$ *such that* $d_X(f(x,u), f(x',u)) \leq l_X d_X(x,x')$, *for all* $x, x' \in \mathcal{X}$, *and* $u \in \mathcal{U}$. *Similarly,* f *is uniformly Lipschitz w.r.t.* u *if there exists* $l_U > 0$ *such that* $d_X(f(x,u), f(x,v)) \leq l_U d_U(u,v)$, $\forall x \in \mathcal{X}$, *and* $u, v \in \mathcal{U}$.

Hereafter, we present a set of sufficient conditions for extending the horizon of validity of the PAC guarantees. These conditions are presented in Assumptions 1 and 2, which still capture a fairly broad class [1] of systems.
Consider the implications of $\mathcal{V}(\hat{S}_l, \hat{R}, H) \neq \emptyset$. Recall that $\hat{R} \subseteq \mathcal{X} \times \hat{\mathcal{X}}_l \subseteq \mathcal{X} \times \mathcal{X}_l$. If $\hat{\Pi}_{l+1} = \Pi_{l+1}$, then, as stated in Section 4.2.1, $\hat{\mathcal{X}}_l = \mathcal{X}_l$, $\hat{R} = R$ and $\mathcal{V} = \emptyset$, since the data-driven SAlCA coincides with the (complete) SAlCA. The converse also holds. The set $\mathcal{V}$ can be alternatively expressed as the union of equivalence classes of the *missing* $l+1$-sequences, i.e., those in $\Pi_{l+1} \setminus \hat{\Pi}_{l+1}$:

$$\mathcal{V}(\hat{S}_l, \hat{R}, H) = \{x_0 \in \mathcal{X} : \exists \mathbf{u}_H \in \mathcal{U}^H, \exists \mathrm{r} \in \mathcal{I}_H(S, x, \mathbf{u}_H), \exists k \geq 0 \,.\, \mathrm{r}'(k) \in \mathcal{K}\}, \tag{4.20}$$

$$\mathcal{K} \doteq \bigcup_{\mathrm{q} \in \Pi_{l+1} \setminus \hat{\Pi}_{l+1}} [\mathrm{q}]. \tag{4.21}$$

We have shown that the probability measure μ_x assigns to the set of initial conditions that can visit an equivalence class $[w_{l+1}] \notin \hat{\Pi}_{l+1}$—those not captured during sampling—is bounded by $\epsilon|\mathcal{U}^H|$. Given that the data-driven SAlCA was constructed using a set of sampled H-long external behaviors D, we now examine how this bound changes when considering the probability of visiting an l-sequence not acquired during sampling over a horizon $H + T$, with $T \in \mathbb{N}$. To do so, we analyse the change in measure under the system's time-reversed dynamics, leading to the following lemma, detailed in Appendix 4.8.2.

[1] Besides affine systems, this class includes, among others, contractive systems [79], nonlinear systems with smooth derivatives, and invertible neural networks.

Lemma 3. *Given a compact set $\mathcal{X} \subset \mathbb{R}^n$, the p-norm $||\cdot||_p : \mathcal{X} \to \mathbb{R}$, and a function $g : \mathcal{X} \to \mathcal{X}$ such that $||g(x) - g(x')||_p \le L||x - x'||_p$, if for constants $c, q > 0$ it holds that $q^{-1}||x - x'||_p \le ||x - x'||_2 \le c||x - x'||_p$ for all $x, x' \in \mathcal{X}$, then for any Riemann integrable set $P \subseteq \mathcal{X}$ it holds that $\int_{g(P)} dV \le (cLq)^n \int_P dV$.*

Assumption 1. *The map f in equation (3.1) is Lipschitz invertible (see Definition 17), where the distance is induced by the p-norm on $\mathcal{X}$.*

Proposition 6. *Let μ_x and $\mu_{\mathbf{u}_H}$ be the uniform probability measures on $(\mathcal{X}, \mathcal{G})$ and $(\mathcal{U}^H, \mathcal{F})$ respectively. Consider S with Σ satisfying Assumption 1, and the data-driven SAlCA $\hat{S}_l$ obtained from $\hat{\Pi}_{l+1}$. For any positive integer T it holds that*

$$\mu_x[S \preceq^{\hat{R}}_{S_{T+H}} \hat{S}_l] \le \nu(\lambda)\mu_x[S \preceq^{\hat{R}}_{S_H} \hat{S}_l] \tag{4.22}$$

where

$$\nu(\lambda) \doteq \begin{cases} 1 + \lambda^T \sum_{i=0}^{\tau-1} \lambda^{-i(H+1)} & \textit{for } \lambda \ge 1, \\ \lambda^T + \sum_{i=0}^{\tau-1} \lambda^{i(H+1)} & \textit{for } 0 < \lambda < 1, \end{cases}$$

$$\tau = \lceil (H + T + 1)/(H + 1) \rceil - 1, \quad \lambda = |\mathcal{U}| \left(\frac{cq}{m_X} \right)^n,$$

and c, q and $m_X^{-1} = L$ are defined as in Lemma 3.

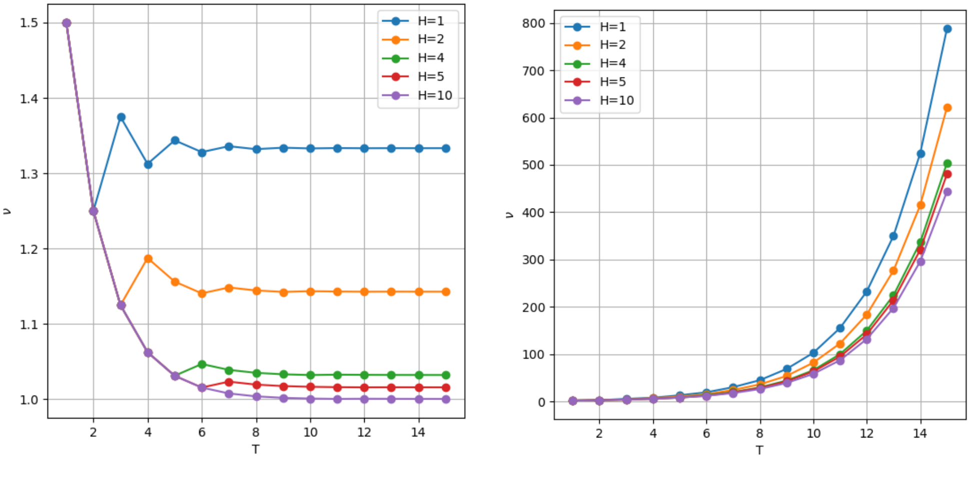


Figure 4.1.: Plot of ν with $\lambda = 0.5$ (left) and $\lambda = 1.5$ (right) for different values of T and H.

Proposition 6, proved in Appendix 4.8.2, connects the probability of selecting an initial condition that leads to a new behavior, absent in $\hat{S}_l$, within the time horizon H used to construct the data-driven SAlCA, with the probability of this occurring over a horizon *greater* than H. Note that Assumption 1, and in particular the value m_X, is not linked to the stability of Σ and it is satisfied by a broad set of nonlinear stable/unstable systems. Figure 4.1 shows the function ν for different parameters.

4.5.1. Contracting Systems

For the class of contracting systems, it is possible to show that there exists a time H after which they always produce the same output.

Assumption 2. *The map f of the Σ described in equation (3.1), is uniformly contracting w.r.t. x with constant l_X, uniformly Lipschitz w.r.t. u with constant l_U, and there exist $x^* \in \mathcal{X}$ and $u^* \in \mathcal{U}$ s.t. $f(x^*, u^*) = x^*$.*

For instance, contractive control-affine systems satisfy Assumption 2 ($f(x, u)$ depends linearly on u for a fixed x). Note that the uniformly contracting assumption is slightly stronger than the necessary condition to develop our technique: our method requires a condition on the measure of the preimage of sets related to the dynamical flow $f(x, u)$ (see the proof of Proposition 6 in the Appendix for more details).
The following lemma can be easily derived using the triangular inequality for distances.

Lemma 4. *Under Assumption 2, with $\bar{l} = \frac{l_U}{1-l_X}$*

$$d(x_k, x^*) \leq l_X^k d(x_0, x^*) + \bar{l} \max_{0 \leq i < k} \{d(u_i, u^*)\}. \tag{4.23}$$

Proposition 7. *Let Σ satisfying Assumption 2. If there exist a $y \in \mathcal{Y}$ and $r > \rho$ s.t. for all $x \in B(x^*, r)$ it holds that $h(x_k) = y^*$ then for any $H \geq \bar{k}$ the data-driven SAlCA $\hat{S}_l$ of S satisfies*

$$\mu_x[S \preceq^{\hat{R}}_{S_H} \hat{S}_l] = \mu_x[S \preceq^{\hat{R}}_{S_{\bar{k}}} \hat{S}_l] \tag{4.24}$$

where $\bar{k} = \lceil \log_{l_X}(r - \rho) - \log_{l_X} \psi \rceil$, $\psi = \sup_{x \in \mathcal{X}} d(x, x^*)$, and $\rho = \bar{l} \sup_{u \in \mathcal{U}} d(u, 0)$.

Proof. After $\bar{k}$ time steps the distance of any trajectory's state x_k from x^* is smaller than r, by Lemma 4 we know that x_i will remain within that distance for all $u \in \mathcal{U}$. Moreover, by assumption of the proposition we have $h(x_k) = y^*$ for all such x's. Thus, the next input-output pairs are (u, y^*) for all $u \in \mathcal{U}$. □

The proposition assures that, if there exists a sufficiently large ball around the fixed point of f, where all points have the same output, the contractivity of the flow f allows us to determine when all trajectories enter the ball. Hence, if an initial condition does not belong to the violation set defined in the right-hand side of equation (4.24), then neither will it belong to the violation set defined in the left-hand side.

Remark 12. If the initial conditions of the system can be selected arbitrarily and not following a distribution, if the Lipschitz constant of the system is known, other methodologies exist to construct an abstraction, e.g. [20].

4.5.2. Autonomous Systems

In Chapter 3 we provide a framework for the construction of the data-driven SAlCA of an autonomous system, which, for example, can be utilized for verifying whether a linear temporal logic formula holds. As a special case, equation (3.1) is autonomous if $|\mathcal{U}| = 1$. Through Proposition 4 we obtain a guarantee that $\hat{R}$ is a PSR from S to $\hat{S}_l$ with respect to $\mathcal{U} \times \mathcal{Y}$ until horizon H with probability not less than $1 - \epsilon$ (with confidence β). Lemma 2 implies that $\hat{S}_l$ behaviorally includes S until horizon H with the same probability. In summary, we recover the guarantees provided by Proposition 2.

4.6. Experimental Evaluation

A Linear System Let us consider the linear system $x_{k+1} = Ax_k + Bu_k$, where $\mathcal{U} = \{-0.3, 0, 0.3\}$,

$$A = \frac{1}{4}\begin{bmatrix} 1 & 2 \\ -1.8 & 1 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}. \tag{4.25}$$

The state space $\mathcal{X} = [-3, 3]^2$ is partitioned into 9 regions by a uniform grid, each uniquely labelled and defining the output set $\mathcal{Y}$.

We sample $N = 2 \cdot 10^6$ initial conditions $(x_0, \mathbf{u}_H)$ uniformly, with $H = 4$ and $l = 2$. The sampling process returns $\hat{\Pi}_{l+1}$, containing 342 sequences, and we construct the corresponding abstraction. Setting $\beta = 10^{-6}$, we compute the scenario bounds according to Proposition 4, $\epsilon(s^*_{N,l}) = 1.51 \cdot 10^{-4}$. Additionally, by Corollary 1, with confidence at least $1 - \beta$, $\hat{Z}$ defines a PASR from the abstraction to the concrete system with respect to $\mathcal{U} \times \mathcal{Y}$ until horizon H with probability not less than $1 - \overline{\epsilon}$, where $\overline{\epsilon} \doteq \epsilon(s^*_{N,l})|\mathcal{U}^H| = 1.23 \cdot 10^{-2}$. The computational time is reported in Table 4.1. In order to extend the guarantees from horizon $H = 4$ to any finite horizon we employ Proposition 7. Using the parameters $l_X \simeq 0.56$, $\psi \simeq 4.24$, $l_U = 1$, $\rho \simeq 0.68$, $r = 1$, we obtained $\overline{k} = 5$. We then construct a new abstraction after collecting $N = 10^6$ trajectories with horizon $H' = 5$. In line with the discussion in Section 4.5, we conclude that $\hat{Z}$ defines a PASR from the abstraction to the concrete system until horizon H', and hence any horizon, with probability not less than $1 - \overline{\epsilon}'$, where $\overline{\epsilon}' = 6.18 \cdot 10^{-2}$, and confidence $1 - \beta$. Alternatively, instead of resampling, we could have applied Proposition 6; we can extend the guarantee of PASR from the abstraction to the concrete system until horizon H to horizon H' with probability not less than $1 - \nu\overline{\epsilon} = 1 - 1.41 \cdot 10^{-1}$, where we have used equation (4.22) with $T = 1$ to compute the correcting factor $\nu = 11.4$. Proposition 7 provides tighter bounds using half of the samples compared to Proposition 6.

Mountain Car We adapt the mountain car benchmark [80]. The domain $\mathcal{X} = [-1.2, 0.6] \times [-0.07, 0.07]$, uniformly sampled, accounts for position x and velocity v. The goal of the car is to reach any point with $x \geq 0.5$ (the top of a hill) as fast as possible, in at most 250 time steps. We compare two schemes to derive a controller with guaranteed performance, for a fixed budget of samples $N = 10^6$, and confidence $\beta = 10^{-3}$: $(i.)$ a single-stage approach where we construct the abstraction from the uncontrolled system as per Section 4.3 and derive a controller by solving a reachability game [6]; $(ii.)$ a two-stage approach where first we design a controller using standard model-free Q-learning from reinforcement learning (RL) and then we provide performance guarantees on the controlled system, as per Section 4.5.2. The final results are shown in Figure 4.2. $(i.)$ We partition the domain in 6 regions, solely across the position axis, that is $[0.5, 0.6]$ is labeled G, and $[-1.2, 0.5)$ is divided by 5 intervals of equal length, labeled $R_1, ..., R_5$. We impose a zero-order hold control input over $T = 50$ time steps, and observe the system's output accordingly every T steps: this allows us to shorten the effective control horizon and improve our guarantees, at the cost of a more restrictive controller design. We sample $N = 10^6$ pairs of initial conditions and input sequences from a uniform distribution, we collect trajectories of length $H = 5$ and set $l = 2$. Note that the system runs for $H \cdot T$ time steps in total. We obtain the set $\hat{\Pi}_{l+1}$, containing 1283 sequences, and a complexity of $s^*_{N,l} = 633$. This means that out of 10^6 trajectories, 633 contain all the $l + 1$ sequences that constitute the SAlCA. Proposition 4 returns a bound on the violation probability for a randomly extracted pair of initial conditions and input sequences of $\epsilon \doteq \epsilon(s^*_{N,l}) = 7.49 \cdot 10^{-4}$, see Table 4.1 for details on the computational time. By Proposition 5 and Corollary 1, with confidence at least $1 - \beta = 1 - 10^{-3}$, we can establish a PASR from the abstraction to the concrete system until horizon H with probability not less than $1 - \overline{\epsilon}$ where, $\overline{\epsilon} \doteq \epsilon|\mathcal{U}^H| = 2.40 \cdot 10^{-2}$. We frame the synthesis of the controller as a reachability game on the data-driven abstraction. We define as goal states all the l-sequences where the last symbol is G, i.e. the car is in the goal set. The solution of the reachability game returns a set of abstract states and actions that are guaranteed to drive the car to the goal set. Among the returned abstract states, there are all five l-sequences of the form $\diamond \diamond \diamond \diamond\, y$, where y is $R_1, ..., R_5$. Hence, we can refine the controller to drive the car to the goal set from every initial state in at most 250 time steps, with the above guarantees. $(ii.)$ We use a 32×32 uniform grid for the domain $\mathcal{X} = [-1.2, 0.5] \times [-0.07, 0.07]$, labeled $R_1, ..., R_{1024}$ (G as before), and define the Q-table accordingly. The learning agent receives a reward of -1 for every time step until the car reaches the goal set. We allocate $N_{\text{RL}} = 5 \cdot 10^4$ episodes for training (exploration rate of 0.01, learning rate of 0.1) and $M = N - N_{\text{RL}}$ episodes for verifying the closed-loop SAlCA, for which we choose $l = 100$, and obtain $\overline{\epsilon} = 1.43 \cdot 10^{-2}$.

Parameter study. We test our approach with several values of N, l, while maintaining a fixed time horizon of $H = 5$, see Figure 4.3. We observe that, while for $l = 1, 2$ the growth of $(l + 1)$-sequences (or transitions) rapidly tapers off with N, this is not the case for $l = 3, 4$; nevertheless, for $N > 10^6$ we can derive nontrivial bounds.

Table 4.1.: Time in seconds for computing: the set $\hat{\Pi}_{l+1}$, the data-driven SAlCA, the complexity $s^*_{N,l}$, the controller.

	$\hat{\Pi}_{l+1}$	$\hat{S}_l$	$s^*_{N,l}$	Control Synthesis
Linear System	4.8	<0.1	4.75	-
Mountain Car	39.7	0.2	5.8	<0.1

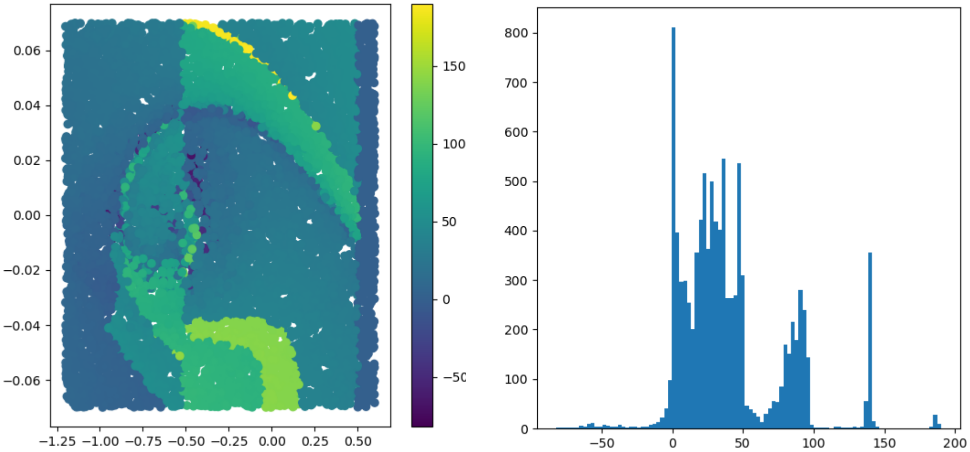


Figure 4.2.: The difference in time steps required to reach the goal set between controllers $(i.)$ and $(ii.)$ is shown on the domain $\mathcal{X}$ (left) and its histogram (right), tested on 10^4 new initial conditions. On average the controller obtained in $(i.)$ requires 39.3 more time-steps than the one in $(ii.)$.

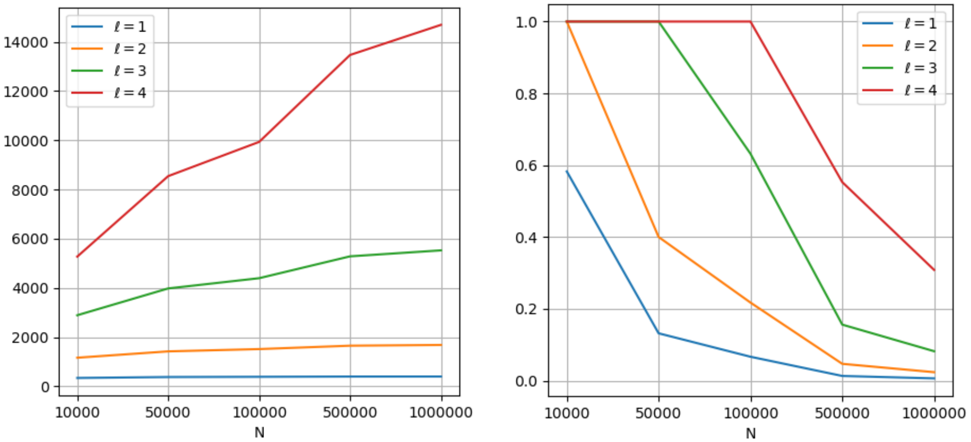


Figure 4.3.: Number of transitions in the abstraction (left) and relative $\overline{\epsilon}$ bound (right), for $l = 1, \dots, 4$, and $\beta = 10^{-3}$.

4.7. Discussion and Concluding Remarks

This chapter presented a novel data-driven approach for constructing finite abstractions of control systems with formal probabilistic guarantees and minimal assumptions, by sampling input-output behaviors, thereby avoiding costly reachability analysis. We derive PAC bounds that quantify the abstraction's fidelity with respect to the true system dynamics. The framework is designed to be specification-agnostic, allowing the same abstraction to be reused across multiple control tasks without retraining. It also accommodates varying levels of abstraction granularity thanks to the tunable l-long sequences, offering a trade-off between precision and computational complexity. The main limitations of our approach are two: first, that the sample complexity cannot be determined in advance, as it relies on scenario theory for degenerate problems; second, the PAC guarantees deteriorate rapidly as the size of the input space and the time horizon increase. Our experiments demonstrate that meaningful guarantees can be obtained from finite data, even in the absence of explicit system models. If the control objective is fixed, synthesising a controller with a mature technique for black-box models, such as RL, and successively verifying the design might result in tighter bounds, as argued in Section 4.5.2. However, when the specification is complex, the application of RL is not straightforward. If the control objective is not fixed, abstraction and control synthesis by means of the data-driven SAlCA may be more sample efficient.

4.8. Appendix

4.8.1. Temporal Logic Characterization

Below, we focus for simplicity on bounded-time reach-avoid specifications. The reasoning can be generalised to $\langle\langle sys \rangle\rangle$-ATL* [74], restricted to finite horizons. An output-feedback controller $C : \bigcup_{k=0}^{H-1} \mathcal{B}_H(S) \to \mathcal{U}$ is a mapping from finite external behaviors to inputs. A bounded-time reach-avoid specification $\varphi = (R, A, H)$ is a tuple where R and A are disjoint subsets of $\mathcal{Y}$ and H is a positive integer. We denote by $\models_{S,C,\varphi}$ the set of initial conditions x_0 of S such that for all the internal behaviors $\mathtt{r} = x_0 u_0 x_1, \dots x_H$ and corresponding external $\mathtt{b} = y_0 u_0 y_1, \dots y_H$ with $u_i = C(\mathtt{b}[0, i])$, it holds that there exists $k \leq H$ such that $\mathtt{b}|^x(k) \in R$ and for all $j \leq k$ it holds that $\mathtt{b}|^x(j) \notin A$.

Proposition 8. *If* $\mu_x[\hat{S}_l \preceq^{\hat{Z}}_{AS_H} S] > 1-\epsilon$ *and* $\models_{\hat{S}_l, C, \varphi^H_{R,A}} = \hat{\mathcal{X}}_{l,0}$ *then* $\mu_x[\models_{S,C,\varphi^H_{R,A}}] > 1-\epsilon$.

Proof. The probability of drawing an initial condition x_0 in S such that for all admissible input sequences from $\hat{Z}(x_0)$ for all internal behaviors $\mathtt{r} \in i_{\mathbf{u}_H}$ there exists an internal behavior $\mathtt{r}_l \in i^l_{\mathbf{u}_H}$ such that $\mathtt{r}$ and $\mathtt{r}_l$ are step-wise related with matching external behaviors is at least $1 - \epsilon$. Since every external behavior stemming from every initial condition of $\hat{S}_l$ satisfies the specification the thesis follows.

4.8.2. Proofs

Proof of Proposition 3.

We begin by showing that R defines a SR from S to S_l w.r.t. $\mathcal{U} \times \mathcal{Y}$ and use this result to prove the claim of the proposition. Pick $(x, \mathtt{q}) \in R$, i.e. $\mathtt{q} \in \mathcal{E}(x)$, and observe that by Definition 5 we have that $\mathcal{H}_l(\mathtt{q}) = \mathtt{q}|^y(l)$, and by equation (2.6) there exists an internal behavior $\mathtt{r}$ of S such that $\mathtt{q} = \mathcal{H}(\mathtt{r}[j-l, j])$ and $\mathtt{r}|^x(j) = x$ for some non negative integer j: hence $\mathcal{H}_l(\mathtt{q}) = \mathcal{H}(\mathtt{r}|^x(j)) = \mathcal{H}(x)$ which proves the second requirement of a SR. Next, we prove that the third requirement holds. If $(x, u, x') \in \delta$ is a transition in S, then there exists an internal behavior $\mathtt{r}$ of S

such that $\mathtt{q} = \mathcal{H}(\mathtt{r}[j-l, j])$ (since $(x, \mathtt{q}) \in R$) and $\mathtt{r}[j, j+1] = xux'$. Let $\mathtt{b}$ be the corresponding external behavior of $\mathtt{r}$, i.e. $\mathtt{b} = \mathcal{H}(\mathtt{r})$: by equation (2.5) we obtain that $\mathtt{q}' \doteq \mathcal{H}(\mathtt{r}[j-l+2, j+1]) \in \Pi_l$ belongs to the state set of the SAlCA, moreover, since $\mathtt{q} \cdot \mathtt{q}'[l-1, l] = \mathcal{H}(\mathtt{r}[j-l, j+1]) \in \Pi_{l+1}$, there exists a transition $(\mathtt{q}, u, \mathtt{q}') \in \delta_l$ in the SAlCA. Since $\mathtt{q}' \in \mathcal{E}(x')$ we conclude that the third requirement holds. The first requirement is proved analogously. We established that $S \preceq_S^R S_l$. To prove that $S_l \preceq_{AS}^{R^{-1}} S$ it is sufficient to prove the third requirement of for ASRs. To see this, observe that if $(\mathtt{q}, x) \in R^{-1}$ then $(x, \mathtt{q}) \in R$, and the third requirement of SRs implies that for every admissible input $u \in U_\delta(x)$, if x' is a u-successor of x there exists a u-successor $\mathtt{q}'$ of $\mathtt{q}$ such that $(x', \mathtt{q}') \in R$. If S has free input, from the above discussion, we obtain that S_l has free input too. To conclude, $U_\delta(x) = \mathcal{U} = U_{\delta_l}(x)$, hence the third requirement of ASRs holds.

Proof of Lemma 2.

In Chapter 3 it was shown that, for a deterministic *autonomous* system $S_{\Sigma'}$, and its data-driven SAlCA $\hat{S}_l$ constructed from N i.i.d. trajectories of length H according to a probability measure μ, the probability of sampling a new external behavior not existent in $\hat{S}_l$ can be bounded by ϵ, with confidence $1-\beta$. Formally, $\mu_x^N[\mu_x[\{x' : \mathcal{B}_H(S_{\Sigma'}, x') \notin \mathcal{B}_H(\hat{S}_l)\}] < \epsilon] \geq 1-\beta$, where $\mathcal{B}_H(S_{\Sigma'}, x')$ is *the* external behavior generated by the autonomous $S_{\Sigma'}$ when initialized in x', $\epsilon \doteq \epsilon(s^*_{N,l}, \beta, N)$ as defined in Theorem 1, and $s^*_{N,l}$ is as per Remark 8. Since $\mathbf{u}_H$ is available at time 0, it is sufficient to define Σ' on the domain $\mathcal{X} \times \mathcal{U}^H$ as an augmentation of Σ, where $\mathbf{u}_H$ is part of the initial conditions.

Proof of Lemma 3.

The volume of a parallelotope $Q(x_1, ..., x_k) = \left\{\sum_{i=1}^k r_i x_i \ : \ r_i \in [0,1]\right\}$ is recursively computed as $\mathrm{vol}(Q(x_1, ..., x_k)) \doteq \mathrm{vol}(Q(x_1, ..., x_{k-1}))h$ where h is the Euclidean distance of x_k from $\mathrm{span}(x_1, ..., x_{k-1})$. Let de_i be the vector of value dx_i at the i-th component and 0 elsewhere. The infinitesimal element of volume shifted by x is given by $x + Q(de_1, ..., de_n)$ and $\mathrm{vol}(x + Q(de_1, ..., de_n)) = \mathrm{vol}(Q(de_1, ..., de_n)) = \prod_{i=1}^n dx_i$. In first approximation, the shifted hypercube $x + Q(de_1, ..., de_n)$ is transformed into the shifted parallelotope given by $g(x) + Q_g^n$, where $Q_g^n \doteq Q(g(x+de_1) - g(x), ..., g(x+de_n) - g(x))$. We can upper bound its volume as $\mathrm{vol}(Q_g^n) \leq \mathrm{vol}(Q_g^{n-1})||g(x+de_n) - g(x))||_2 \leq \prod_{i=1}^n ||g(x+de_i) - g(x))||_2$. By assumption $\prod_{i=1}^n ||g(x+de_i) - g(x))||_2 \leq \prod_{i=1}^n c||g(x+de_i) - g(x))||_p \leq \prod_{i=1}^n cLq||de_i||_2 = (cLq)^n \prod_{i=1}^n dx_i$ To conclude, for the infinitesimal volume it holds that $\mathrm{vol}(Q(g(x+de_1) - g(x), ..., g(x+de_n) - g(x))) \leq (cLq)^n \mathrm{vol}(Q(de_1, ..., de_n))$.

Proof of Proposition 6.

For any set in $Q \subseteq \mathcal{X}$ we define the following operations: $\mathrm{Pre}_u(Q) \doteq \{x \in \mathcal{X} : f(x,u) \in Q\}$, $\mathrm{Pre}_*^0(Q) \doteq Q$, and

$$\begin{aligned} \mathrm{Pre}_*^k(Q) \doteq \{x \in \mathcal{X} : \exists \mathbf{u}_k \in \mathcal{U}^k, \\ \mathtt{r} \in \mathcal{I}_H(S, x, \mathbf{u}_k) \ . \ \mathtt{r}(k) \in Q\}, \end{aligned} \tag{4.26}$$

$$\mu_x^{0,H}(Q) \doteq \mu_x \left[\bigcup_{i=0}^{H} \mathrm{Pre}_*^k(Q) \right]. \tag{4.27}$$

By Assumption 1 and Lemma 3, the pre-image of a set Q is bounded, specifically $\mu_x[\mathrm{Pre}_u(Q)] \leq \lambda \mu_x[Q]$,[2] with $\lambda = \left(\frac{cu}{m_X}\right)^n$. By the union bound, $\mu_x^{0,1}(Q) \leq \bigcup_{u \in \mathcal{U}} \mu_x[\mathrm{Pre}_u(Q)] + \mu_x[Q] \leq (1+\lambda)\mu_x[Q]$ where $\lambda = |\mathcal{U}|\eta$. Note that $\mathrm{Pre}_*^{k+1}(Q) = \mathrm{Pre}_*^1(\mathrm{Pre}_*^k(Q))$. Let $E_p^q := \bigcup_{i=p}^q \mathrm{Pre}_*^i(Q)$. Then, for $\tau = \lceil (H+T+1)/(H+1) \rceil - 1$, we can express $E_{i=0}^{H+T}$ in two equivalent forms

$$E_0^{H+T} = E_{j=0}^{H} \cup \bigcup_{i=0}^{\tau-1} E_{j=T-i(H+1)}^{H+T-i(H+1)}, \tag{4.28}$$

$$E_{i=0}^{H+T} = E_{j=T}^{H+T} \cup \bigcup_{i=0}^{\tau-1} E_{j=i(H+1)}^{H+i(H+1)}. \tag{4.29}$$

For $\lambda \geq 1$, Using equation (4.28) and equation (4.29), we derive the following bounds for $\lambda \geq 1$ and $0 < \lambda < 1$ respectively

$$\mu_x^{0,H+T}(Q) \leq \mu_x^{0,H}(Q) \left(1 + \lambda^T \sum_{i=0}^{\tau-1} \lambda^{-i(H+1)} \right), \tag{4.30}$$

$$\mu_x^{0,H+T}(Q) \leq \mu_x^{0,H}(Q) \left(\lambda^T + \sum_{i=0}^{\tau-1} \lambda^{i(H+1)} \right). \tag{4.31}$$

Recall equation (4.21) and set $Q = \mathcal{K}$. By combining equation (4.20) with equation (4.26) and equation (4.27) we obtain that $\mu_x[\mathcal{V}(\hat{S}_l, \hat{R}, H)] = \mu_x^{0,H}(\mathcal{K})$.

[2] In general, Proposition 6 applies to any system satisfying for all Q the inequality $\mu_x[\mathrm{Pre}_u(Q)] \leq \lambda \mu_x[Q]$.

5

Reinforcement Learning for Robust Ageing-Aware Control of Li-Ion Battery Systems with Data-Driven Formal Verification

Rechargeable Lithium(Li)-Ion Batteries are a ubiquitous element of modern technology. In the last decades, the production and design of such batteries and their adjacent embedded charging and safety protocols, denoted by Battery Management Systems (BMS), have taken central stage. A fundamental challenge to be addressed is the trade-off between the speed of charging and the ageing behavior, resulting in the loss of capacity in the battery cell. In this chapter, we rely on a high-fidelity physics-based battery model and propose an approach to data-driven charging and safety protocol design by employing an existing Reinforcement Learning (RL) pipeline and extending the work with the developments of the previous chapters. This is done by constructing data-driven abstractions capable of verifying desired formal system specifications under probabilistic guarantees. Furthermore, a Counterexample-Guided Inductive Synthesis (CEGIS) scheme is proposed to additionally guide the training of the charging policy based on information from the abstraction construction.

This chapter is based on [57]. Minor changes have been made to streamline the presentation.

5.1. Introduction

The development of advanced battery control strategies is a critical enabler for the widespread electrification of both energy systems and transportation. These strategies must ensure the safe and reliable operation of batteries under a wide range of conditions throughout their service life. At the same time, growing economic pressures and sustainability goals demand more efficient use of battery systems—minimizing degradation while maximizing performance [81]. Whether in grid-scale storage or electric vehicles, improving how batteries are charged and managed is central to achieving these objectives. To this end, significant progress in Battery Management Systems (BMS) is required. Conventional BMS implementations rely heavily on Equivalent Circuit Models (ECMs), consisting of circuit models describing the battery as a set of simplified electrical components. Although ECMs offer fast computation and are widely used in practice, they fail to capture many internal electrochemical processes of the battery. As a result, they are limited in their ability to estimate internal states or inform control strategies that account for degradation mechanisms.

In contrast, physics-based electrochemical models, typically framed as Doyle-Fuller-Newman (DFN) models, offer a more detailed description of the internal dynamics governing battery behavior. These models can, in principle, enable estimation and control of internal variables that are otherwise unobservable, potentially leading to smarter, degradation-aware charging protocols. However, the high computational complexity of DFN models hinders their application in online optimisation-based approaches, e.g. nonlinear Model Predictive Control (MPC), or to develop a formal feedback control design. To bridge this gap, we explore a Reinforcement Learning (RL) approach that leverages the accuracy of physics-based models to construct a light-weight controller by interacting offline with a DFN model simulator. This chapter presents an automatic model-free RL framework that learns optimal charging strategies through interaction with a simulated environment governed by the full electrochemical model. While training relies on DFN simulations, the final controller relies solely on measurable quantities, such as voltage and temperature, rendering our scheme easily applicable in practice. In parallel, we apply a modern statistical tool to provide safety and performance guarantees of the closed-loop system. This approach offers a promising path toward advanced battery control that is both physically grounded and practically implementable.

A lithium-ion (Li-Ion) battery consists of a negative electrode (anode), a separator, and a positive electrode (cathode). The electrodes are porous, composed of microscopic particles, and immersed in an ion-conducting liquid electrolyte. Charging occurs as the applied external current forces the internal movement of the ions from the cathode to the anode. During discharging, the process is reversed. While this mechanism is highly reversible, enabling hundreds to thousands of charge–discharge cycles, the long-term performance of a battery is constrained by irreversible ageing mechanisms that accumulate over time [82]. One of the most prominent degradation processes is the growth of the solid–electrolyte interphase (SEI) layer on the surface of the negative electrode. The SEI layer forms as a result of electrolyte decomposition during the initial cycles and continues to grow during subsequent operation due to parasitic side reactions. As the SEI thickens, it increases interfacial resistance and consumes active lithium, effectively reducing the battery's usable capacity and power capability [83]. The kinetics of SEI growth have been modelled as an electron-limited process driven by side reactions at the anode–electrolyte interface, with foundational work provided by Darling and Newman [84]. Importantly, SEI growth is strongly correlated with charging rates: faster charging typically leads to increased SEI formation due to elevated reaction rates and higher overpotentials. This creates an inherent trade-off in charging strategies between minimising charge time and limiting long-term degradation.

Recent studies have begun to address this by developing ageing-aware charging protocols that balance performance and longevity using physics-based models in a control context, as done in [82]. Ageing-Aware MPC schemes are implemented on the DFN model, such as in [85, 86], where an approximate model is used to describe the internal battery dynamics. This work is expanded in [87], where a nonlinear MPC framework is implemented using nonlinear optimisation methods. However, the control scheme developed is implemented in full-state feedback, as no state estimator concept is developed. As a result, the corresponding experimental runs are performed in an open-loop environment, which is detrimental to the charging performance and safety. Additionally, the computational difficulty of MPC based on the DFN model is illustrated in [88], where several problem formulation strategies are considered for the DFN model, each resulting in substantially large time required for a single optimisation call (45-65 seconds), which renders real-time MPC on such a complex model infeasible in most practical applications.
Reinforcement Learning (RL) is a model-free approach to policy design. Such a policy, or controller, is designed in [89], where the DFN Model is used to emulate a cylindrical cell Li-Ion Battery System, and a controller is trained for the case of both state feedback and output feedback, i.e., the controller is assumed to only know the cell voltage, the average temperature of the cell, and its State of Charge (SOC). The resulting protocol is shown to perform better than standard rule-based charging methods. The authors model ageing as lithium plating [90, 91]; accordingly, the cost function does not aim at extending the life cycle of the battery, rather it aims at fast charging while avoiding destructive phenomena. In our work, we model ageing as the SEI growth causing the decay of the State of Health (SOH) of the cell, a measure of its capacity; we explicitly incorporate in the cost function of the RL problem a metric for capacity loss. In [92], the authors apply RL to the problem of safe fast charging and augment it with a safety layer that projects unsafe actions into a feasible region defined by a Gaussian Process (GP) surrogate model. In contrast with our work, the authors do not model ageing constraints. Moreover, GP-based approaches rely on the samples used to construct the surrogate model to be Gaussian-distributed, which is often difficult to verify.

Different from the works cited above, we focus on providing distribution-free probabilistic performance guarantees, while iteratively constructing and refining (a set of) controllers. We adopt the theoretical framework developed in Chapters 3 and 4, which allow for the construction of a finite abstraction of the battery that conservatively approximates the behavior of the battery under different initial conditions and manufacturing parameters: this makes the abstraction suitable to describe the properties that the battery satisfies. We demonstrate our approach by synthesising a safe controller for a time-bounded Reach-While-Avoid specification. In particular, we follow the architecture of Counterexample Guided Inductive Synthesis (CEGIS) [93–95], where the synthesis of a suitable controller is reached by means of an iterative interaction between a learner and a verifier, detailed in Section 5.2. From a practical standpoint, we develop controllers trained on a high-fidelity DFN model of the battery cell. Our framework accounts for the battery's manufacturing parameter uncertainty and State of Health decay, rendering the obtained controller applicable to batteries throughout their entire life span and robust to parameter variation. We obtain a charging protocol with formal guarantees on the maximum rate of ageing of the battery, while ensuring fast and safe charging. The charging protocol is implemented in output feedback, relying on realistic output measurements. We compare our solution against the standard Constant-Current-Constant-Voltage (CC-CV) approach as well as against an approach based solely on RL, and we illustrate an improved tradeoff between fast charging and ageing.

5.2. Preliminaries

5.2.1. Models and abstractions

We consider control systems of the form:

$$\Sigma_{\mathrm{c}} = \begin{cases} x_{k+1} = f_{\mathrm{c}}(x_k, u_k), \\ z_k = g(x_k), \\ y_k = \phi(z_k), \\ x_0 = x, \end{cases} \tag{5.1}$$

where $x_k \in \mathcal{X} \subseteq \mathbb{R}^n$ is the n-dimensional state of the cell at time $k \in \mathbb{N}$ contained in a domain of interest $\mathcal{X}$, $g : \mathcal{X}^n \to \mathcal{Z}$ is the measurement map with $\mathcal{Z} \subseteq \mathbb{R}^m$ and $m < n$, z_k is the (real-valued) *output measurement*, and $y_k \in \mathcal{Y}$ is the *output label* with $\mathrm{card}(\mathcal{Y}) < \infty$, where $\mathrm{card}(\cdot)$ denotes for the cardinality of a set. We employ output measurements for controller synthesis and output labels for verification. The mapping $\phi : \mathcal{Z} \to \mathcal{Y}$ is a partitioning map that returns a label, i.e., an element of the finite set $\mathcal{Y}$, corresponding to the output measurement z_k. We seek to synthesize output feedback controllers $\mathcal{K}$, i.e., $u_k = \mathcal{K}(g(x_k))$, which when applied to equation (5.1) results in autonomous systems of the form:

$$\Sigma \doteq \begin{cases} x_{k+1} = f(x_k) \doteq f_{\mathrm{c}}(x_k, \mathcal{K}(g(x_k))) \\ y_k = h(x_k) \doteq \phi(g(x_k)), \\ x_0 = x, \end{cases} \tag{5.2}$$

where we have defined $h \doteq \phi \circ g$.

Remark 13 (Parameter Uncertainty). First-principle models, as the ones we employ to model batteries later on, often depend on a number of parameters that, while constant, may be difficult to measure and uncertain. Let $p \in \Delta_p$, be a vector of uncertain parameters in a known parametric uncertainty set. Assuming the uncertainties are static, by augmenting the state vector x with p, the uncertain system can be reformulated in the same form of equation (5.2) as:

$$\bar{\Sigma} \doteq \begin{cases} \begin{bmatrix} x_{k+1} \\ p_{k+1} \end{bmatrix} = \begin{bmatrix} f(x_k, p_k) \\ p_k \end{bmatrix}, \\ y_k = h(x_k), \\ x_0 = x, p_0 = p, \end{cases} \tag{5.3}$$

In this chapter, we approximate the concrete system equation (5.2), representing the battery cell in closed loop with a controller, by constructing its data-driven SAlCA as illustrated in Chapter 3, to make it amenable to computer-based verification algorithms [6].

5.2.2. Counterexample Guided Inductive Synthesis

Counterexample Guided Inductive Synthesis (CEGIS) is an automated procedure which can be used to iteratively construct a controller enforcing a desired specification on a system by learning from counterexamples generated (formally) in the process. CEGIS can be intuitively explained by describing the interaction between its two main blocks, a *learner* and a *verifier*, see Fig. 5.1:

1. **Learning (Inductive synthesis) step:** the learner proposes a candidate solution for a specification of interest using any *synthesis tool*. As we detail in Section 5.4.1, in our case, this step is achieved employing Reinforcement-Learning (RL) to synthesise controllers.

2. **Verification step:** the verifier performs a check of the validity of the candidate solution using a *verification tool*. We propose to employ for this step a data-driven SAlCA, detailed in Section 5.4.2. If the closed-loop satisfies the RWA specification (with the desired probability and confidence), the synthesis loop is terminated. Otherwise, one must produce counterexamples, in our case, initial conditions resulting in H-long behaviors violating the specification.

3. **Refinement step:** As long as the property is not satisfied, the learner is retrained employing the generated counterexamples, and refined solutions (controllers) are proposed.

5.3. Battery Management

In this section, we describe the model of the battery cell's dynamics and the desired specification.

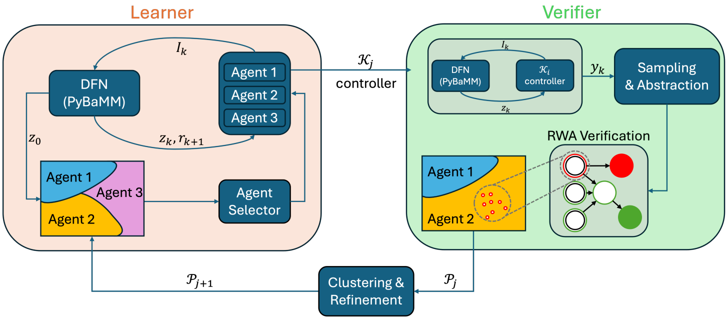


Figure 5.1.: Schematic of the CEGIS architecture. The learner (orange block) proposes a controller $\mathcal{K}_j$ based on a number of trained agents, each responsible for a given set of initial conditions. The verifier (green block) uses the closed-loop system with the proposed controller, samples a set of behaviors given by H-long output label sequences, and constructs the data-driven SAlCA. If the resulting abstraction satisfies the RWA specification, the loop terminates with probabilistic guarantees. Otherwise, the domain of initial conditions of the battery cell is partitioned according to the counterexamples.

5.3.1. Physics-based Cell Modeling

The electrochemical behavior of a Li-Ion cell can be described using the physics-based model developed by Doyle, Fuller, and Newman (DFN) [96]. The DFN Model consists of Partial Differential Equations (PDE), describing the spatio-temporal evolution of solid and electrolyte phase concentrations and potentials. Additionally, algebraic constraints are employed to enforce charge and mass conservation. A schematic representation of the DFN Model is shown in Figure 5.2.

Output Dynamics

The model adopts a *pseudo-two-dimensional* (P2D) approximation based on the DFN model [96, 97]. It assumes homogeneous behavior along the lateral and circumferential directions of the cylindrical cell, thereby reducing the problem to a two-scale, one-dimensional model in space. Spatial variation is considered along the through-thickness direction of the cell, denoted by $x \in [0, L]$, while radial variations in concentration within representative spherical particles are captured using local radial coordinates $r_n \in [0, R_n]$ and $r_p \in [0, R_p]$ for the negative and positive electrodes, respectively.

The domain $[0, L]$ is divided into three contiguous regions: I) negative electrode: $x \in [0, L_n]$; II) separator: $x \in [L_n, L_n + L_s]$; III) positive electrode: $x \in [L_n + L_s, L]$.

Along the electrode thickness, a set of representative spherical particles is placed, used to model lithium diffusion in the solid phase. The decoupling of the x- and r-domains is achieved through the interfacial reaction kinetics, which couple the local surface concentration in each particle to the macroscopic current distribution

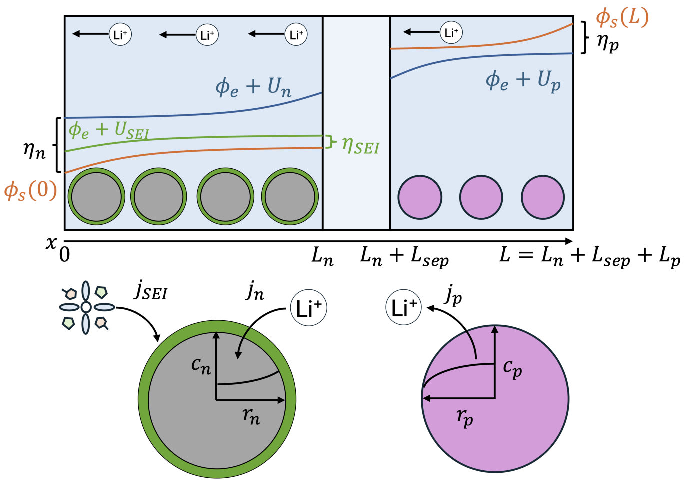


Figure 5.2.: Schematic of the DFN Model. Upper: electrode-scale transport regulates the electrolyte and electric potential drops (ϕ_e, ϕ_s), as well as the electrolyte concentration. Consequently affecting the reaction overpotentials (η_n, $\eta_S EI$, η_p). Lower: solid diffusion regulates the concentration drop within the particles (c_n, c_p)

across the cell thickness.

The model evolves over time t and is governed by the coupled behavior of:

- Negative electrode lithium concentration $c_n(r_n, x, t)$
- Positive electrode lithium concentration $c_p(r_p, x, t)$
- Electrolyte concentration $c_e(x, t)$
- Solid-phase potential $\phi_s(x, t)$
- Electrolyte-phase potential $\phi_e(x, t)$

The local reactions evolve according to the reaction overpotentials of positive (η_p) and negative (η_n) electrodes. These are defined based on ϕ_s, ϕ_e and the surface potentials of the electrodes, $U_n(c_{n,surf})$ and $U_p(c_{p,surf})$. In addition, the model accounts for temperature-dependent properties (e.g., diffusion coefficients, reaction rates, and conductivities), enabling a more realistic description of the electrochemical and thermal dynamics under various operating conditions. The PDEs describing the DFN dynamics can be observed in detail in [98, 99]. Hereafter, expressions related to this work are presented.

The cell voltage is defined by the solid-phase potential difference at the two ends of the cell, i.e.,

$$V(t) = \phi_s(L, t) - \phi_s(0, t). \tag{5.4}$$

The **State of Charge (SOC)** is defined based on the average lithium concentration in the solid phase of the negative electrode. Defining $c_{n,\text{min}}$ and $c_{n,\text{max}}$ as the stoichiometric concentrations corresponding to the voltage cutoffs at 0% and 100% SOC, respectively, the SOC is expressed as:

$$SOC(t) = \frac{\bar{c}_n(t) - c_{n,\text{min}}}{c_{n,\text{max}} - c_{n,\text{min}}},$$

where $\bar{c}_n(t)$ is the spatially averaged lithium concentration in the solid particles of the negative electrode at time t. The temperature variations are assumed to occur homogeneously throughout the cell, resulting in a lumped thermal model. The cell temperature $T(t)$ evolves according to the energy balance:

$$mc_p \frac{dT}{dt}(t) = \frac{T_{\text{amb}} - T(t)}{R_{\text{th}}} + q(I_{\text{cell}}, SOC), \tag{5.5}$$

where m is the mass of the cell, c_p is the effective heat capacity, R_{th} is the convective thermal resistance to the environment, and $q(I_{\text{cell}}, SOC)$ represents the total heat generation rate due to electrochemical processes. In general, the function $q(I_{\text{cell}}, SOC)$ increases with the magnitude of the applied current and is influenced by the total internal resistance of the cell. However, it also accounts for nonlinear electrochemical effects and thermodynamic heat generation or absorption associated with the electrode reactions. These include ohmic heating, reaction overpotentials, and entropic contributions. A detailed formulation of these heat sources can be found in [100].

Ageing Dynamics

The model couples ageing dynamics to the lithium intercalation process via a reaction-limited model for the formation of the solid electrolyte interphase (SEI). For the exact mathematical expressions and model parameters, we refer the reader to [86, 101]; here, we report the qualitative dependence of the relevant variables for simplicity.

The ageing-related **side reaction current** is denoted by $j_{\text{SEI}}(x, t)$ and is modeled as

$$j_{\text{SEI}}(x, t) \propto -j_0(T(t)) \exp\left(-\frac{\eta_{\text{SEI}}(x, t)}{T(t)}\right), \tag{5.6}$$

where $j_0(T)$ is the empirical exponential pre-factor that depends on temperature (T) following an Arrhenius relation, and $\eta_{\text{SEI}}(x, t)$ is the **side reaction overpotential** in the negative electrode. It is defined in terms of the electrode potentials and SEI resistance (R_{SEI}) as

$$\eta_{\text{SEI}} = \phi_s - \phi_e - U_{\text{SEI}} - (j_n + j_{\text{SEI}})R_{\text{SEI}}. \tag{5.7}$$

Battery capacity is lost due to the irreversible trapping of lithium ions within the SEI layer, so that the capacity loss Q_l, evolves as

$$\frac{dQ_l}{dt} \propto \int_0^{L_n} j_{\mathrm{SEI}}(x,t)\,dx. \tag{5.8}$$

In parallel to the side reaction, the main lithium intercalation reaction in the negative electrode j_n is also governed by an exponential dependence on its own overpotential, defined as

$$\eta_n = \phi_s - \phi_e - U_n(c_{n,\mathrm{surf}}) - (j_n + j_{\mathrm{SEI}})R_{\mathrm{SEI}}. \tag{5.9}$$

The evolution of the applied voltage is thus influencing both the intercalation current and the SEI formation dynamics. Mitigating ageing while maintaining fast charge rates requires careful dynamic balancing of the competing intercalation and side reactions, along with effective control of the cell temperature T.

Discretization

The PDE dynamics describing the cell behavior are discretised in both spatial and temporal domains, and are solved numerically to result in a discrete-time model. Specifically, for the implementation of the model and its numerical solution, we rely on PyBaMM, an open-source battery simulation package [102]. According to 5.1, the discretised representation of the DFN model is given by

$$\Sigma_{\mathrm{c}} \doteq \begin{cases} x_{k+1} = f_{\mathrm{c}}(x_k, u_k), \\ z_k = g(x_k), \\ x_0 = x, \end{cases} \tag{5.10}$$

where $x_k \in \mathcal{X} \subseteq \mathbb{R}^n$ is the n-dimensional state of the cell at time $k \in \mathbb{N}$ contained in a domain of interest $\mathcal{X}$, $u_k \in \mathcal{I}$ is the control input at time k representing the charging current, $g : \mathcal{X} \rightarrow \mathcal{Z}$ returns the cell's output measurements. In this work, we assume that z_k comprises the **time-step, SOC, Voltage, Temperature, Past Input Current**, indicated respectively by k, SOC_k, V_k, T_k, I_{k-1}, which are typically measurable by the electronics embedded within a BMS. Hereafter, the cell is assumed to be modelled by Equation equation (5.10). Note that we have omitted y_k, as it is only used for verification purposes (Section 5.4.2), and it is not used for training the controller.

5.3.2. Desired Performance Specification

With mathematical expressions describing the electrochemical behavior of a Li-Ion cell, the control goals of a charging protocol can be formalised in a mathematical framework. The control goal is to minimise the loss of capacity Q_l over one charging cycle, while minimising the charging time t_{final}.

Concretely, we define a reward function that aims at minimising Q_l and a maximum charging time $t_{\max}$, specifying that the battery needs to reach 90% SOC with $t_{\mathrm{final}} \leq\ t_{\max}$.

In addition to the ageing-awareness goal, safety specifications are to be met by maintaining a cell voltage below $V_{\max}$ and a temperature below $T_{\max}$. By maintaining those conditions, the charging protocol can reduce the chances of thermal runaway, or irreversible degradation of the cell components. The above goals can be formalised as a Reach-While-Avoid (RWA) specification.

In the case of a Li-Ion Battery cell, the RWA specification can be defined as follows:

- **Initial Set $\mathcal{X}_0$:** The set of states where $(V, T) \in [\underline{V}, \overline{V}] \times [\underline{T}, \overline{T}]$,
- **Goal Set $\mathcal{X}_G$:** The set of states where $SOC \in [\underline{SOC}, 1]$,
- **Safe Set $\mathcal{X}_S$:** The set of states where:
    1. Cell Voltage $V \leq V_{\max}$,
    2. Cell Temperature $T \leq T_{\max}$.
- **Time bound:** The goal must be reached in at most t_{max} steps.

The proposed performance must be achieved for a range of batteries having varying manufacturing parameters and state of health (SOH). In order to include cell-to-cell manufacturing variations, the particles' diffusion coefficients, the electrodes' tortuosities and the heat transfer coefficients are varied within a 10% range following a (clipped) Gaussian distribution. Moreover, an SOH between 100 % and 85 % is defined as the ratio between the nominal and the aged capacity. The loss of capacity is assumed to be solely due to the loss of Li in the SEI layer, so that a corresponding SEI thickness is initialised based on the SOH. Finally, the cation transference number is also scaled with the SOH to include ageing-related loss of electrolyte properties. Details are available in 5.7.1.

5.4. Proposed Approach

As introduced in Section 5.2, we follow the CEGIS approach summarised in Figure 5.1. In what follows, we describe first our *Learning* stage, and our *Verification* (and counterexample generation) stage.

5.4.1. Reinforcement Learning

The learning approach proposed in this work consists of informing the controller design with observation data from a large number of battery simulations (or experiments). We propose to use Reinforcement Learning (RL) to design a charging protocol in output feedback. In RL, the goal is to find a policy that maximises rewards from an environment, in our case, the controlled plant: a Li-Ion cell. At every time-step $k \in \mathbb{Z}^+$, the environment is described by its state x_k and its output z_k, whereas the control policy $\pi(z_k)$ (possibly stochastic) produces an action u_k, resulting in the state x_{k+1}, following a transition probability $p(x_{k+1}|x_k, u_k)$, and the scalar reward $r_{k+1} = r(x_k, u_k)$. The goal of RL is to learn the optimal policy π^* that is associated with the maximum expected reward, i.e., the policy that maximises the value function

$$V^{\pi}(x) = \mathbb{E}_{\pi}[R_k | x_k = x], \tag{5.11}$$

where R_k denotes the total reward from time k onward, obtained following a policy π. In our case, the state and action spaces are continuous, motivating the use of the Actor-Critic framework, which is a policy gradient approach that employs function approximators, e.g., neural networks. The parameters describing the actor and critic are updated using the Soft Actor-Critic algorithm [103]. This update is done by performing (approximate) gradient descent updates repeatedly, employing simulation (or experimental) data from tuples $(z_k, u_k, r_{k+1}, z_{k+1})$, improving the current best policy (actor) and the resulting estimate of equation (5.11).

The controller operates in Output Feedback, i.e. it measures only SOC, Voltage, Temperature, Past Input Current, and time-step (SOC_k, V_k, T_k, I_{k-1}, k), while the reward function is designed to balance ageing, charging time, and safety:

$$r_{k+1} = \lambda_1 r_{SOC} + \lambda_2 r_{fast} + \lambda_3 r_{cap} + r_{final} \tag{5.12}$$

where,

$$r_{SOC} = SOC_{k+1} - SOC_k, \tag{5.13}$$

$$r_{fast} = k + 1, \tag{5.14}$$

$$r_{cap} = -(Q_{l,k+1} - Q_{l,k}), \tag{5.15}$$

$$r_{final} = \begin{cases} R_{fail} \text{ if unsafe,} \\ R_{succ} \text{ if goal,} \\ 0 \text{ otherwise.} \end{cases} \tag{5.16}$$

In words, r_{SOC} represents the increase in the SOC between consecutive time-steps, r_{fast} is a penalization of elapsed time, encouraging fast charging, r_{cap} is the capacity loss between consecutive time-steps, as per equation (5.8), and r_{final} is the terminal reward of the episode, receiving a large positive value R_{succ} if the SOC safely reaches the goal set $\mathcal{X}_G$, a large negative one R_{fail} if it violates the safety constraints $\mathcal{X}_S$, or 0 otherwise. Note that reaching the goal set or exiting the safe set immediately terminates the learning episode, and the environment is reset.

5.4.2. Data-driven Synthesis and Verification

After extracting a policy $\mathcal{K}_0 : \mathcal{Z} \to \mathcal{I}$, following the RL scheme in Section 5.4.1, we formally verify the performance post-training. As the reward function equation (5.12) used to train the RL agent is merely a proxy of the RWA specification of interest, we use a data-driven abstraction to provide safety guarantees.

To this end, we define a finite set of symbols, the output labels, corresponding to different regions of the battery's state; we do this by defining a partition as follows:

- The $[0, 1]$ interval describing the SOC is finely partitioned in 19 identical sections plus 1 for the Goal set. This is done to avoid self-loops in the abstraction and is pivotal to verifying the reachability of the Goal set, which consists of the region $[\underline{SOC}, 1]$. The cells or the partition are assigned a unique label, i.e. $a, b, c, ..., s, t$, where t labels the Goal set.

- The rest of the output variables are partitioned based on their atomic proposition, i.e., the symbol a for safety, such as $V \leq V_{\max}$, $T \leq T_{\max}$. The violation of those constraints results in a symbol b.

Accordingly, at every time-step k the cell outputs an output label $y_k \in \mathcal{Y}$, where $\mathcal{Y} = [a, b, \ldots, t] \times [a, b] \times [a, b]$.

Let j be an integer denoting the current iteration of the CEGIS algorithm, initialised as $j = 0$.

Step 1 : Data-Driven Abstraction

For a candidate charging protocol $\mathcal{K}_i$ we analyze the resulting autonomous system, and, according to equation equation (5.2), its dynamics are described by

$$\Sigma = \begin{cases} x_{k+1} = f(x) \doteq f_{\mathrm{na}}(x_k, \mathcal{K}_i(g(x_k))), \\ y_k = h(x_k) \doteq \phi(g(x_k)), \\ x_0 = x^{(i)}. \end{cases}$$

Let $C = \bigcup_i^N x_0^i$ be a set of i.i.d. sampled initial conditions, including random manufacturing parameters and SOHs [1], and according to equation (3.4), let $D \doteq \{\mathcal{B}_H(S, x_0^i) \ : \ i = 1, 2, ...N\}$ be the set of resulting external behaviors. For a selected l, construct the TS data-driven SAlCA $\hat{S}_l \doteq (\hat{\mathcal{X}}_l, \hat{\mathcal{X}}_{l0}, \hat{\delta}_l, \mathcal{Y}, \mathcal{H}_l)$.

Step 2 : Verification of RWA Specification

Next, we employ the resulting finite-state data-driven SAlCA $\hat{S}_l$ to verify the given RWA specification. Solving a Safety Game followed by a Reachability Game [6], one can find the set of counterexamples of initial states in the abstraction $\hat{\mathcal{X}}_l^{\mathrm{c}} \in \hat{\mathcal{X}}_{l0}$ leading to a violation of the RWA specification. If this set is empty, the desired guarantees are derived according to Proposition 2, and the CEGIS iterations are terminated. If $\hat{\mathcal{X}}_l^{\mathrm{c}}$ is not empty, we need to extract the associated counterexamples to refine the controller. Note first that the elements $\mathsf{q} \in \hat{\mathcal{X}}_l^{\mathrm{c}}$ are sequences of output labels of length l. Let us denote by $h^{-1}(\mathsf{q})$ the set of states that can produce the sequence of symbols $\mathsf{q} = \sigma_0, \ldots, \sigma_{l-1}$, i.e.

$$h^{-1}(\boldsymbol{\sigma}) := \{x \in \mathcal{X} \mid \mathsf{q} \in \mathcal{B}_l(S)\}, \tag{5.17}$$

and, abusing notation, $h^{-1}(\hat{\mathcal{X}}_l^{\mathrm{c}}) = \bigcup_{\mathsf{q} \in \hat{\mathcal{X}}_l^{\mathrm{c}}} h^{-1}(\mathsf{q})$. While computing $h^{-1}(\hat{\mathcal{X}}_l^{\mathrm{c}})$ is in general not possible, we can nonetheless extract a set of counterexamples in the domain of the original system S from said set using the sampled initial conditions as:

$$\hat{\mathcal{X}}^{\mathrm{c}} = h^{-1}(\hat{\mathcal{X}}_l^{\mathrm{c}}) \cap C, \tag{5.18}$$

[1] We select the initial voltage and temperature following a uniform distribution on the initial set. In practice, a manufacturer or client may have a data set describing the empirical distribution of the initial conditions: in that case we can easily adapt our sampling to a realistic setting

where $\hat{\mathcal{X}}^{\mathrm{c}} \subset \mathcal{X}$. Increment the counter $j = j + 1$.

Step 3: Clustering

Next, based on the counterexamples $\hat{\mathcal{X}}^{\mathrm{c}}$, we generate a partition of the set of initial conditions of the battery cell $\bigcup\{\mathcal{R}^{(m)}\}_{m=1}^{M_j} = \mathcal{X}_0$ using $M_j > 1$ sets, an arbitrary hyper-parameter of our procedure satisfying $M_j > M_{j-1}$. The increase of clusters per iteration is to ensure that at every new clustering iteration, the set of initial conditions $\mathcal{X}_0$ is divided into increasingly finer partitions. The clustering is performed based on the associated outputs of the counterexamples, i.e. $g(\hat{x}_k)$ for $\hat{x}_k \in \hat{\mathcal{X}}^{\mathrm{c}}$, to retain the output dependence of the switched controller. While there exist several algorithms for partitioning [104], in our proposal, for simplicity, the partition is generated by a uniform grid of rectangular hyper-intervals.

Step 4: Refinement

Finally, for each region $\mathcal{R}^{(m)}$ we re-train a new agent, for a total of M_j distinct RL agents from which we extract distinct controllers $\mathcal{K}_j^{(m)} : \mathcal{Z} \to \mathcal{I}$. The resulting controller is defined as

$$\mathcal{K}_j \doteq \begin{cases} \mathcal{K}_j^{(1)} \text{ if } g(x_0) \in \mathcal{R}^{(1)}, \\ \dots \\ \mathcal{K}_j^{(M_j)} \text{ if } g(x_0) \in \mathcal{R}^{(M_j)}. \end{cases} \tag{5.19}$$

The resulting $\mathcal{K}_j$ can be thought of as a switched controller where the switching depends solely on the measured output at the initial condition, i.e. the region $\mathcal{R}^{(m)}$ containing the initial condition x_0 determines the corresponding controller $\mathcal{K}_j^{(m)}$ to be applied to the battery cell. Given this construction, return to Step 1.

5.5. Results

We select for demonstration of our approach a battery model based on the LGM50LT Lithium-Ion cell [98, 100], which is a commonly used cylindrical cell composed by a Graphite anode and an NMC cathode. The selected battery has a capacity of roughly 5000mAh and a maximum charging voltage of 4.2 V.

We compare our CEGIS resulting protocol with the industry standard Constant-Current-Constant-Voltage (CC-CV) protocol.

5.5.1. Benchmarking

The rule-based Constant-Current-Constant-Voltage (CC-CV) protocol comprises two phases, the Constant-Current (CC) and the Constant-Voltage (CV) phases. During the CC phase, a constant charging I_{CC} is supplied to the cell, until the cell voltage reaches a maximum voltage V_{CV}. Next, the charging protocol transitions to the CV phase, during which the cell's voltage is held constant at V_{CV}, while the current rapidly decays until either the battery is fully charged or a minimum charging current is reached.

Notably, the CV phase of the protocol is not trivial to maintain, as the current profile that maintains a constant voltage is not directly known and may vary among different batteries. Some cases ensure that the voltage remains at the maximum value by embedding voltage regulators into the circuitry of the Battery Management System, which is prone to high temperature and energy loss.

Given the defined parameter ranges, we identified the cells based on "quality" and SOH. The quality refers to the manufacturing parameters, with the lowest quality corresponding to the lower bound for the particles' diffusion coefficients and the upper bound for the electrodes' tortuosity. The standard quality corresponds instead to the average parameters of the defined Gaussian distribution.

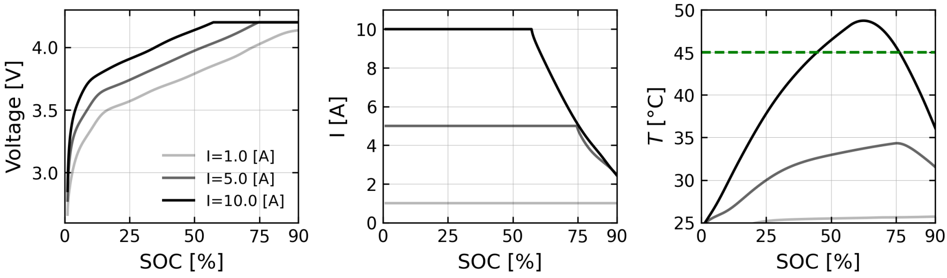


Figure 5.3.: Example of the effects of various input currents on the voltage, current and temperature evolution during a CC-CV protocol.

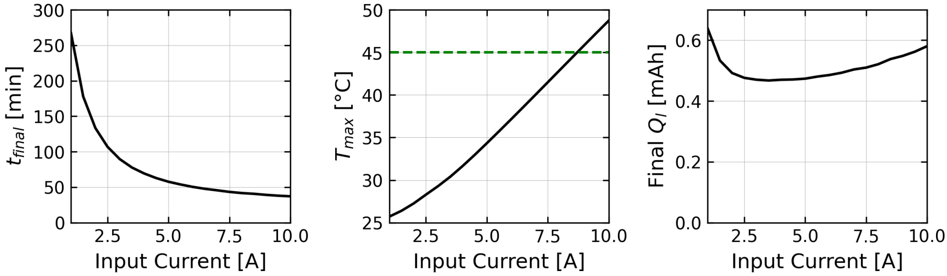


Figure 5.4.: Benchmarking of the CC-CV protocol. Relations between input current and charging times, maximum temperatures and capacity losses. The CC-CV protocol was ended at SOC $=$ 90 %.

To elucidate the effect of different initial input currents on the performance of the CC-CV protocol, we show in Figure 5.3 the relation between three different values of current during the CC phase and the relevant variables, for the case of a pristine (SOH $=$ 100 %) standard quality cell. Figure 5.3 shows the voltage, current and temperature evolution as a function of the SOC for a CC-CV protocol with the CC phase at 1.0, 5.0 and 10.0 Ampere. The CV phase begins when the voltage of the battery reaches a value of 4.2 Volts. We can observe that the larger the charging current for the CC phase is, the earlier the CV phase is triggered, and the higher the maximum temperature reached. Notably, the temperature trajectory corresponding to a 10.0 Ampere CC phase violates the safety constraints provided by the battery manufacturer. In Figure 5.4, we show the total charging time (from 1% to 100%), the maximum temperature reached and the loss of capacity as a function of the input current selected for the CC phase. The charging time decreases with the input current, reaching a lower limit due to the internal resistances within the battery. The maximum temperature reached during the protocol (T_{max}) increases roughly linearly with the current input exceeding the safety threshold above 9 A, suggesting the battery should not be operated at this current. The capacity loss is minimal for a current of approximately 3.5 A, coinciding with the recommendation of the producer; accordingly, we select it as the benchmark CC-CV protocol, resulting in ≈ 77 minutes of charging time, 0.47 mAh of capacity loss and a maximum temperature of 30°C.

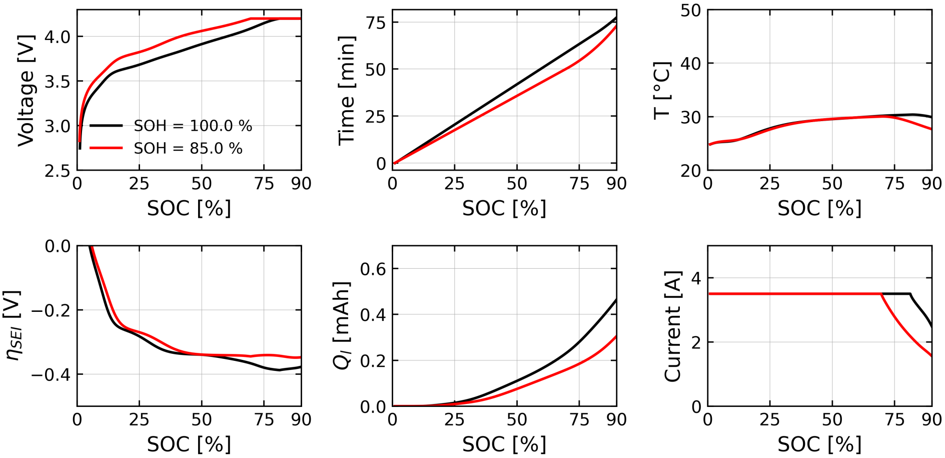


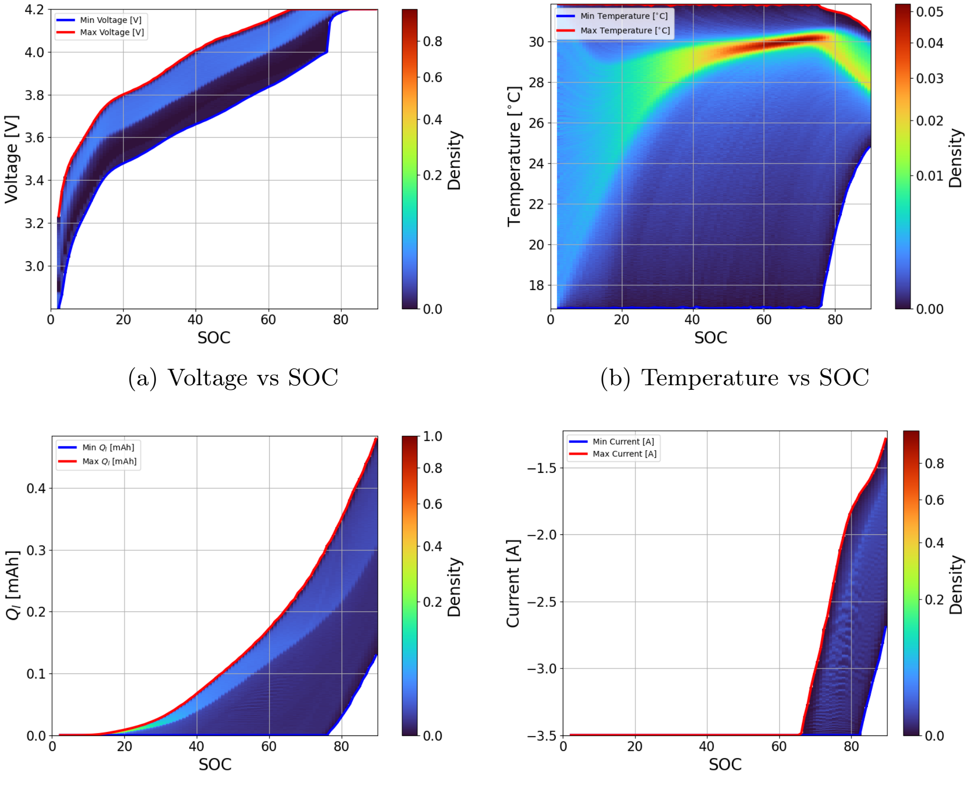
Figure 5.5.: Effect of the CC-CV protocol on pristine and aged cells with average manufacturing parameters.

To further understand the role of cell SOH, we tested the selected CC-CV protocol (I = 3.5 A) for the case of pristine and aged cells corresponding to an SOH of 100% and 85%, respectively.

Figure 5.5 shows the evolution of the variables described in Section 5.3 during the CC-CV protocol. The rate of capacity loss is initially modest, since at low SOCs the negative electrode operates at relatively high voltages. However, once the SOC reaches ≈ 15%, the low (x-averaged) η_{SEI} induces a fast degradation of the electrolyte, accelerating the formation of the SEI layer and the rate of capacity loss. Regulating the negative electrode potential is thus critical to minimise Q_l.

Moreover, applying the same CC-CV protocol to both aged and pristine batteries, we observe marked differences in voltage profiles and degradation kinetics. Notably, while the increased SEI layer thickness leads to higher overpotential, the additional resistance (eq. 5.7) is also preventing further SEI growth, reducing the additional degradation of the aged battery.

In addition, we consider the set of parameters describing a variation on built quality and $SOH \in [0.85, 1.0]$. See 5.7.1 for the exact parameter ranges. In general, the voltage and temperature profile can vary significantly depending on the SOH and the parameters characterising the cell's production quality. To illustrate this, we sample $N_{\text{CC-CV}}$ random initial conditions, the parameters characterizing the manufacturing quality and SOH. In Figure 5.6, we show the trajectories obtained from the sampled conditions and display them as a distribution for different SOCs; specifically, the SOC is binned, and each vertical slice shows the normalised histogram of the measurement within that bin.



(a) Voltage vs SOC

(b) Temperature vs SOC

(c) Capacity loss vs SOC

(d) Current vs SOC

Figure 5.6.: Trajectory distribution for the CC-CV Protocol

Based on the simulations of the CC-CV protocol and the safety constraints provided by the cell manufacturer, we define the RWA specification used for CEGIS:

- **Initial Set $\mathcal{X}_0$:** The set of states where $(V, T) \in [2.8, 4.0] \times [17, 32]$,
- **Goal Set $\mathcal{X}_G$:** The set of states where $SOC \in [0.9, 1]$,

- **Safe Set $\mathcal{X}_S$:** The set of states where:
  1. Cell Voltage $V \leq 4.2V$,
  2. Cell Temperature $T \leq 45°C$.
- **Time bound:** The goal must be reached in at most 80 minutes of charging time.

5.5.2. Counterexample-Guided Inductive Synthesis - Results

Below, we present the results for the scheme described in Section 5.4. We begin with the first RL run, using a single actor-critic network to control the cell, starting from all the points in the initial set $\mathcal{X}_0$.

Reinforcement Learning Results

The learning algorithm, along with all subsequent ones, is run using the DelftBlue supercomputer [105]. Each training episode is initialised by sampling random initial conditions, manufacturing parameters and SOH.

Category	**Parameter**	**Value**
Learning (RL agent)	Actor network	256x256
	Critic network	256x256
	Control frequency	$6.67 \cdot 10^{-2}$ Hz (1/15s)
	Training steps	$1.6 \cdot 10^{7}$
	Reward weights	$\lambda_1, \lambda_2, \lambda_3 = 10^2,\ 10^5,\ 2.5 \cdot 10^{-2}$
	Number of agents (refined)	8
Verification (SAlCA)	Number of trajectories N	10^5
	Memory length l	6
	Horizon H	320 (80 min)
	Confidence β	10^{-6}
	Scenario complexity s_N^*	13
	SAlCA states	166

Table 5.1.: Summary of training and verification parameters for CEGIS.

Referring to Figure 5.1, the first iteration of our CEGIS scheme starts from the Learner stage, where a single RL agent is trained by interacting with the discretised DFN model described by Equation equation (5.10). The agent selects a new action I_k every 15 seconds of the charging protocol: the DFN model implemented in PyBaMM is integrated using the selected current over the duration and returns the output measurements and reward to the agent. We select a set of parameters for the final agent, resulting in a low-frequency, lightweight controller that can be easily deployed in real-world BMSs. See Table 5.1 for a summary of the main parameters used for training. When the training is completed, the first candidate controller $\mathcal{K}_0$ is ready for testing, and it is parsed to the Verifier stage. We close the loop using the obtained output feedback controller $\mathcal{K}_0$, resulting in the autonomous system described by Equation equation (5.2). We sample 10^5 random initial conditions, manufacturing parameters, and SOH, simulate the closed-loop system for a horizon $H = 320$ discrete time-steps (corresponding to 80 minutes of maximum charging time), and collect the resulting set of H-long behaviors $\widehat{\mathcal{B}^H(S)}$. We select $l = 6$, construct the data-driven SAlCA, and check whether it satisfies the RWA specification. We observe that the abstraction does not satisfy the desired properties, and according to Section 5.4.2, we extract a set of counterexamples.

For the second iteration, as the counterexamples spread rather uniformly on the set of initial conditions, we define a rectangular partition of the set of initial conditions by dividing the initial voltage interval $[2.8, 4.0]$ into 4 identical segments and the initial temperature interval $[17, 32]$ into 2 identical segments: accordingly, we define 8 independent agents, each of which is trained, in parallel, by drawing initial conditions corresponding to the respective cell, using the same parameters shown in Table 5.1. The resulting controller $\mathcal{K}_1$ defined by the 8 agents according to Equation equation (5.19) is used to close the loop, and proceed to the Verifier stage. We construct the data-driven SAlCA using the parameters in Table 5.1, and verify that it satisfies the RWA specification. No counterexamples are found, concluding the CEGIS loop with a successful design.

Remark 14. In our experiments, a single instance of clustering and refinement is sufficient to achieve the desired specifications. If counterexamples were found in one of the refined rectangular cells, it is sufficient to apply the clustering and refinement steps to that specific region, leaving the remaining cells untouched.

In Figure 5.7, we show the performance calculated as the cumulative reward along all the sampled trajectories: each bin of the heatmap averages the performance across the sampled manufacturing parameters and SOH. We observe that the synthesised controller $\mathcal{K}_1$ outperforms the benchmark CC-CV for every initial condition, irrespective of the manufacturing parameters and SOH, when evaluated on the reward function defined in Equation 5.12, where the weights are selected as in Table 5.1. In Figure 5.8, we further analyse the performance: in particular, we

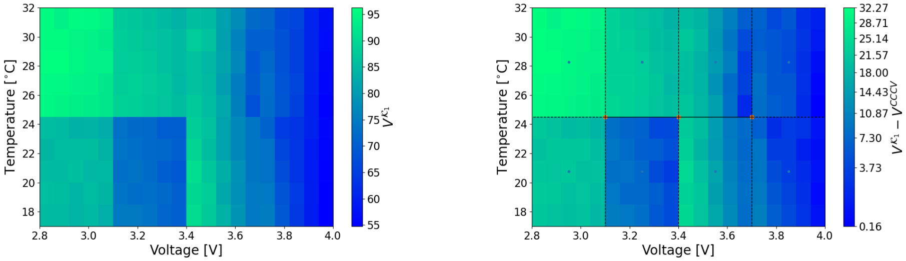


Figure 5.7.: (Left) Average cumulative reward obtained by the battery controlled with $\mathcal{K}_1$. (Right) Average cumulative reward difference between the battery controlled with $\mathcal{K}_1$ and a battery controlled by the benchmark CC-CV protocol.

plot the capacity loss measured along charging trajectories as a function of the initial condition in voltage and temperature. In Figure 5.9, we plot the charging

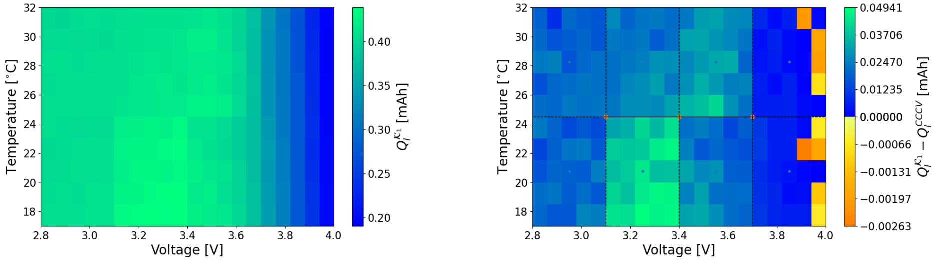


Figure 5.8.: (Left) Average capacity loss obtained by the battery controlled with $\mathcal{K}_1$. (Right) Average capacity loss difference between the battery controlled with $\mathcal{K}_1$ and a battery controlled by the benchmark CC-CV protocol.

time measured along charging trajectories as a function of the initial condition in voltage and temperature. Comparing Figures 5.8 and 5.9, we notice that in the three cells forming the top-left corner, $\mathcal{K}_1$ provides essentially the same capacity loss as the benchmark CC-CV protocol, but in a much shorter time, saving up to 20 minutes in total charging time. In the remaining cells, while the reward function is overall better optimised by $\mathcal{K}_1$, the incurred cost for the capacity loss is comparable to the incurred saving in charging time. In Figure 5.10 we plot the

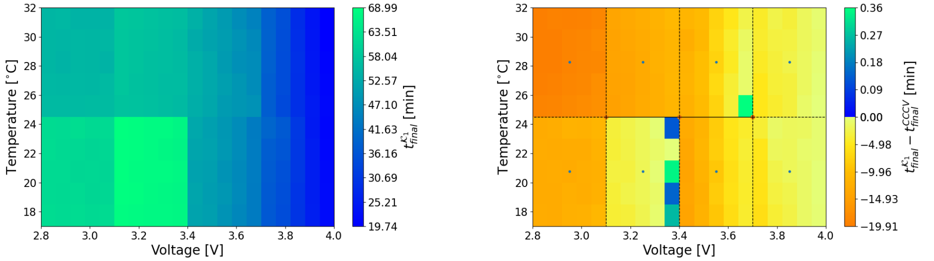


Figure 5.9.: (Left) Average charging time obtained by the battery controlled with $\mathcal{K}_1$. (Right) Average charging time difference between the battery controlled with $\mathcal{K}_1$ and a battery controlled by the benchmark CC-CV protocol.

evolution of the variables described in Section 5.3 under the charging protocol implemented by $\mathcal{K}_1$ with nominal manufacturing parameters and varying SOHs. We observe that, in contrast with the CC-CV trajectories shown in Figure 5.5, $\mathcal{K}_1$ applies larger (in magnitude) input currents for lower SOCs, especially on pristine cells. This allows the controller to better utilise the initial section of the charging protocol where η_{SEI} is less negative, obtaining a reduction of charging times without additional capacity losses. Accordingly, the temperature rises significantly more, while remaining within the safety constraints. This is also part of the protocol optimisation: higher temperatures reduce the internal resistance of the battery by allowing faster ionic reactions and transport. Notably, the current profile is considerably less smooth, as the reward function does not penalise sudden changes in input current. Finally, in Figure 5.11, we show the trajectory distribution for voltage, temperature, capacity loss, and input current as a function of the SOC. In contrast with what we observe in Figure 5.6, the controller $\mathcal{K}_1$ displays more variance across the trajectory distribution, utilising a broader spectrum of current values, particularly for low SOCs. We also observe that $\mathcal{K}_1$ spans a larger range of temperature values, always within the safety constraints. The data-driven SAlCA constructed in the final step of the CEGIS loop comprises a total of 166 unique l-sequences, corresponding to the state set of the SAlCA. The scenario complexity s_N^* equals 13, implying that out of the 10^5 collected H-long behaviors, 13 of them contain all the 166 unique l-sequences. With this information, we conclude that

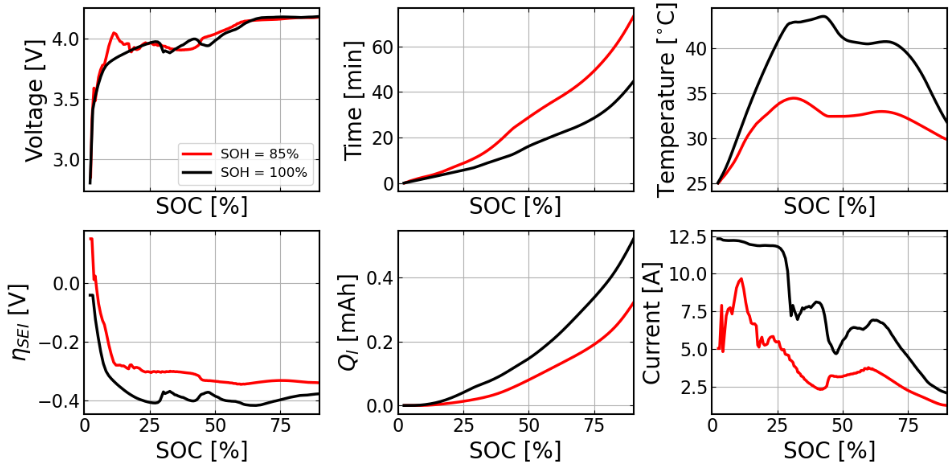
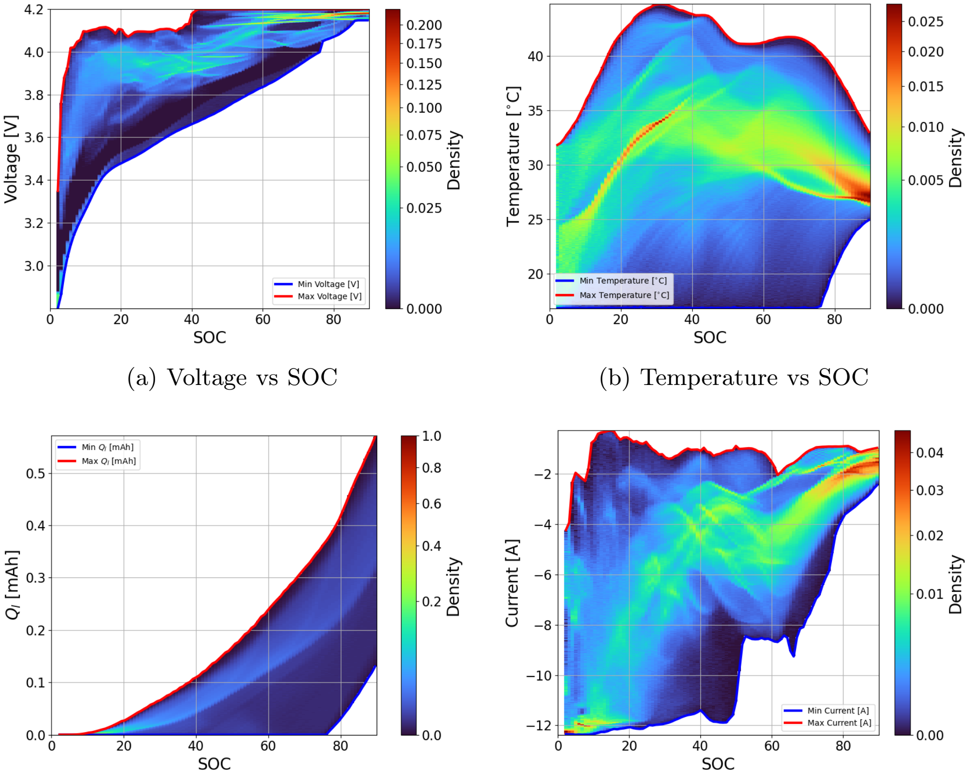


Figure 5.10.: Effect of the CC-CV protocol on pristine and aged cell with average manufacturing parameters.

$$\mathbb{P}^N\left[\mathbb{P}[\mathcal{B}_H(S(x_0)) \in \mathcal{B}_H(\hat{S}_l)] \geq 1 - 4.44 \cdot 10^{-4}\right] \geq 1 - 10^{-6}. \tag{5.20}$$

In other words, the probability of sampling a new random initial condition (of voltage, temperature, manufacturing parameter and SOH) such that the sequence of output labels forms an H-long behavior that does not exist in the set H-long behaviors of the data-driven SAlCA is smaller than $4.44 \cdot 10^{-4}$, with confidence greater or equal than 99.9999%. Since all of the behaviors of the SAlCA satisfy the RWA specification, we conclude that with probability at least 99.956% and confidence of at least 99.9999% drawing a new initial condition in the battery will return a behavior that also satisfies the RWA specification.



(a) Voltage vs SOC

(b) Temperature vs SOC

(c) Capacity loss vs SOC

(d) Current vs SOC

Figure 5.11.: Trajectory distribution for the final synthesised controller $\mathcal{K}_1$.

In contrast with established methods for obtaining statistical guarantees using concentration inequalities [7], generating a full abstraction of the original system has an important advantage: while we have focused on a single specification, the RWA, $\hat{S}_l$ provides a rich behavioral description of the original system S, up to the PAC bounds shown above, in the compact form of a finite state abstraction. Consequently, now that we have obtained a successful design, we can reuse $\hat{S}_l$ to check for other properties of interest; for instance, recalling the partition defined in Section 5.4.2, labels from $\boldsymbol{g}$ to $\boldsymbol{t}$ and from $\boldsymbol{o}$ to $\boldsymbol{t}$ indicate SOCs greater or equal to 50% and 70%. From the abstraction, we can verify that every abstract state reaches labels $\boldsymbol{g}$ and $\boldsymbol{o}$ in at most 135 steps and 197 steps, indicating that, with the same probability described in Equation equation (5.20), the battery is guaranteed to reach 50% and 70% of its SOC within at most 33.75 minutes and 49.25 minutes respectively, accounting for random initial conditions, manufacturing parameters and SOH.

5.6. Conclusion

In this chapter, we presented a novel framework that integrates RL, data-driven formal verification, and CEGIS for the design of ageing-aware charging protocols in Li-Ion batteries. By leveraging high-fidelity physics-based models, our approach enables the synthesis of output-feedback controllers that explicitly trade off charging speed against long-term degradation while guaranteeing safety with respect to voltage and temperature limits.

A key contribution of this work lies in the use of data-driven abstractions to provide probabilistic, distribution-free guarantees on closed-loop behavior, robust to manufacturing parameter variations and to the dynamics' variations due to ageing. The incorporation of CEGIS further allowed us to iteratively refine candidate controllers in response to counterexamples, ultimately yielding a protocol that significantly improves upon the standard CC–CV method. Our results demonstrate not only faster charging times but also reduced ageing, validated with statistical confidence and robustness to manufacturing variability and state-of-health differences.

While we have relied on a conventional DFN model with SEI-driven capacity fade, our framework is not limited to this specific choice. More advanced electrochemical and ageing models, incorporating, for example, concentration-dependent diffusivities [106], coupled ion–electron transfer kinetics [107], or phase-field representations of phase-separating materials [91, 108–110], could be employed to more accurately capture cell dynamics across different chemistries and operating regimes [109, 110]. From a control perspective, such models would primarily serve as richer training environments, challenging the synthesis pipeline with additional nonlinearities, coupled degradation mechanisms, e.g., lithium plating [90, 91] or mechanical stress [111], and nontrivial observability issues. Although this would increase the computational cost of training and verification, it would provide an even more stringent validation of the proposed design methodology and ultimately improve its applicability to real-world cells and chemistries.

Extensions of this work may include such advanced models and validating the resulting controllers experimentally, thereby further bridging the gap between formal, model-driven control design and practical battery management systems.

5.7. Appendix

5.7.1. Parameters, cell quality and SOH

We use the `Chen2020` PyBaMM [112] parameter set as the base[98]. Thermal properties (heat capacities, thermal conductivities, densities, and OCP entropic changes) and the Arrhenius activation energies for solid diffusivities are taken from `O'Regan2022`[100]. Solid diffusivities follow $D_{\mathrm{s},k}(T) = D_{\mathrm{s},k}^{\mathrm{ref}}\ d_k\ \exp\left[\frac{E_{D,k}}{R}\left(\frac{1}{298.15} - \frac{1}{T}\right)\right]$, with $k \in \{\mathrm{p}, \mathrm{n}\}$ and $E_{D,\mathrm{p}} = 12084\ Jmol^{-1}K$, $E_{D,\mathrm{n}} = 17447\ Jmol^{-1}K$.

Cell quality

The nominal parameters were modified within a defined range to account for cell-to-cell heterogeneity.

Quantity	**Variation (relative to nominal)**	**Bounds**
Total heat transfer coefficient	$h_{\mathrm{tot}}/h_{\mathrm{tot},0} \sim \mathcal{N}(1, 0.03)$	[0.9, 1.1]
Solid diffusivity	$D_{\mathrm{s},k}/D_{\mathrm{s},k,0} \sim \mathcal{N}(1, 0.03)$	[0.9, 1.1]
Bruggeman coefficient (electrolyte)	$b_k/b_{k,0} \sim \mathcal{N}(1, 0.03)$	[0.9, 1.1]

Table 5.2.: Uncertainty factors applied to selected parameters, expressed as ratios to their nominal values. Here $\mathcal{N}(1, 0.03)$ denotes a normal distribution with mean 1 and standard deviation 0.03. The subscript $k \in \{\mathrm{p}, \mathrm{n}\}$ indicates positive and negative electrodes.

SOH adjustments

The variation in SOH were accounted by assuming only SEI formation and electrolyte degradation. We map degradation to transport and capacity as:

$$t_+ = \mathrm{SOH} \cdot t_{+,0}, \qquad c_{\mathrm{s,n}}(x, 0) = \mathrm{SOH} \cdot c_{\mathrm{s,n},0}, \qquad Q_{\mathrm{nom}} = \mathrm{SOH} \cdot Q_{\mathrm{nom},0}.$$

Uniform SEI thickening links capacity loss to added SEI volume:

$$\delta_{\mathrm{SEI},0} = \delta_{\mathrm{SEI,init}} + Q_{\mathrm{nom}}\,(1 - \mathrm{SOH})\,\frac{\bar{v}_{\mathrm{SEI}}}{a}\,\frac{3600}{F}\,\frac{1}{z\,V_{\mathrm{electrode}}},$$

where $\bar{v}_{\mathrm{SEI}}$ is the partial molar volume of the SEI, $a = 3\varepsilon_{\mathrm{act,n}}/R_{\mathrm{n}}$ and $V_{\mathrm{electrode}} = h\,w\,L_{\mathrm{n}}$, with h and w being the height and width of the electrode.

6

On Training-Conditional Conformal Prediction and Binomial Proportion Confidence Intervals

Estimating the expectation of a Bernoulli random variable based on N independent trials is a classical problem in statistics, typically addressed using Binomial Proportion Confidence Intervals (BPCI). In the control systems community, many critical tasks—such as certifying the statistical safety of dynamical systems—can be formulated as BPCI problems.

Conformal Prediction (CP), a distribution-free technique for uncertainty quantification, has gained significant attention in recent years and has been applied to various control systems problems, particularly to address uncertainties in learned dynamics or controllers. A variant known as training-conditional CP *was recently employed to tackle the problem of safety certification.*

In this chapter, we highlight that the use of training-conditional CP in this context does not provide valid safety guarantees. We demonstrate why CP is unsuitable for BPCI problems and argue that traditional BPCI methods are better suited for statistical safety certification.

This chapter is based on the publication [58].

6.1. Introduction

Uncertainty quantification is a critical aspect in fields where predictions influence safety and performance guarantees, such as in control systems. Probabilistic guarantees, including those derived from the theory of Probably Approximately Correct (PAC) learning, play an important role in providing bounds on the accuracy of predictions under limited training data.

Conformal Prediction (CP) is one of the approaches that has gained visibility due to its ability to provide valid prediction sets without requiring strong distributional assumptions. A distinctive characteristic of CP is that, rather than providing a point prediction of the variable of interest, it provides set predictions with a valid bound on the probability that the predicted set contains the true variable [42]. This chapter focuses on a specific formulation of CP known as training-conditional CP [113]. However, existing applications in areas such as safety verification for dynamical systems have shown limitations in the interpretation of these guarantees. In particular, recent works have applied training-conditional CP to safety verification problems in control systems [114–116]. While promising, these applications have misinterpreted the implications of CP's set prediction framework, especially in cases where the underlying data can be modeled as Bernoulli random variables. This chapter aims to rigorously analyze these limitations and provide an alternative framework for interpreting PAC-based guarantees in such contexts. In 6.2 we recall existing methods for estimating the expectation of a Bernoulli random variable. In 6.3 we introduce the formalism of training-conditional CP, followed by a detailed analysis of its PAC guarantees. In 6.4 we present a special case of interest, where the nonconformity measure corresponds to an indicator function, leading to Bernoulli-distributed conformity scores. We demonstrate that the PAC guarantees derived from this setting are unsuitable for estimating the expectation of a Bernoulli random variable.

6.2. Binomial Proportion Confidence Intervals

Consider a setting where there are $N+1$ independent and identically distributed (i.i.d.) Bernoulli random variables (r.v.) $R_1, R_2, ..., R_N, R_{N+1}$ with parameter b, i.e. $R_i \sim \mathrm{Bern}_b$ and $\Pr_{\mathrm{Bern}_b}(R_i = 1) \doteq b$. Given a realization of the first N r.v., the problem is to estimate an interval of values for the probability that the $N+1$-th variable will be equal to 1, or in other words we want to estimate the parameter b. This is a very well studied problem and it is known in the literature under the name of Binomial Proportion Confidence Intervals (BPCI), see [117] for a survey. We give below a quick overview of the setting.

Define the new r.v. $Y \doteq \sum_{i=1}^{N} R_i$. It is well known that Y has a binomial distribution $Y \sim \mathrm{Bin}_{N,b}$ with N trials and probability of success b, defined by $\Pr_{\mathrm{Bin}_{N,b}}(Y = y) \doteq \binom{N}{y} b^y (1-b)^{N-y}$ for $y \in \mathbb{Z}_{[0,N]}$, where $\mathbb{Z}_{[0,N]}$ denotes the integers $0, 1, ..., N$. Let $\check{b} : \mathbb{Z}_{[0,N]} \to [0,1]$ and $\hat{b} : \mathbb{Z}_{[0,N]} \to [0,1]$ be two random variables serving as interval estimators. The coverage probability of the interval estimator

$[\check{b}, \hat{b}]$ for $Y \sim \mathrm{Bin}_{N,b}$ is defined as

$$\rho(b, \check{b}, \hat{b}) \doteq \Pr_{\mathrm{Bin}_{N,b}}(\check{b}(Y) \leq b \leq \hat{b}(Y)). \tag{6.1}$$

In the expression above b is fixed and it's the true parameter of the binomial distribution describing Y. Note that $\check{b}$ and $\hat{b}$ are a transformation of the same random variable Y. This expression can also be rewritten equivalently as

$$\rho(b, \check{b}, \hat{b}) = \sum_{y \in I} \Pr_{\mathrm{Bin}_{N,b}}(Y = y),$$

where $I \doteq \{y \in \mathbb{Z}_{[0,N]} : \check{b}(y) \leq b \leq \hat{b}(y)\}$. For $\alpha \in (0, 1)$ an interval estimator $[\check{b}, \hat{b}]$ is a *conservatively valid* (sometimes also called 'exact' or 'secure') $1 - \alpha$ confidence interval if the coverage probability $\rho(b, \check{b}, \hat{b})$ is greater or equal to $1 - \alpha$ for all the values of b. An example of a conservatively valid interval estimator is given by the Clopper-Pearson method [118], see also [117] for more estimators.

Before concluding this section, we rewrite equation (6.1) in an equivalent form that is more commonly found in the literature on Probably Approximately Correct (PAC) bounds. First, note that $\Pr_{\mathrm{Bern}_b}(R_{N+1} = 1) = b$. Second, since Y is a r.v. obtained as a transformation of the i.i.d. Bernoulli random variables $R_1, \ldots, R_N$, the probability of any event $M \subseteq \mathbb{Z}_{[0,N]}$, $\Pr_{\mathrm{Bin}_{N,b}}(Y \in M)$ can be equivalently described by $\Pr^N_{\mathrm{Bern}_b}(\{(r_1, ..., r_N) : \sum_{i \leq N} r_i \in M\})$, where $\Pr^N_{\mathrm{Bern}_b}$ is the product probability measure induced by the N i.i.d. Bernoulli random variables. Hence, we rewrite the definition of a conservatively valid $1 - \alpha$ confidence interval by revisiting equation (6.1):

$$\Pr_{\mathrm{Bin}_{N,b}}(\check{b}(Y) \leq b \leq \hat{b}(Y)) = \Pr^N_{\mathrm{Bern}_b}\left(\check{b}\left(\sum_{i \leq N} R_i\right) \leq \Pr_{\mathrm{Bern}_b}(R_{N+1} = 1) \leq \hat{b}\left(\sum_{i \leq N} R_i\right)\right) \geq 1 - \alpha, \tag{6.2}$$

for all $b \in [0, 1]$. We will use this form of the coverage probability to draw a comparison with the guarantees given by training-conditional CP.

6.3. Training-conditional Conformal Prediction

Conformal Prediction is a statistical tool that uses the available data sampled from identically and independently from an underlying distribution to output predictions for which an error probability can be computed. The original formulation of CP can be informally explained as follows. Suppose that we want to solve a classification problem and we have method that given a feature x outputs a label $\hat{y}$. Given a desired error probability ϵ, conformal prediction uses the available data to generate a *set of labels*, typically containing $\hat{y}$, containing the true label y corresponding to the feature x with a probability not smaller than $1 - \epsilon$ [119]. It is a method capable of augmenting a (usually unreliable) point prediction to a set prediction with probabilistic guarantees of correctness, i.e. it construct a *set predictor*. The original formulation of CP has been successfully applied to both classification and regression problems, see [120, 121] for a recent survey.

In this section we introduce instead the basic concepts of *training-conditional* CP [113]. Training-conditional CP is a variant of the original formulation of CP. While the quality of the guarantees differs from the original, the core idea remains the same, that is, constructing set predictions with some form of guarantees: training-conditional CP produces PAC-style guarantees. In the following, we give a self-contained overview of the theoretical details of training-conditional CP.

Let $(\Delta, \mathcal{F}, \mathbb{P})$ be a probability space where $\mathbf{Z}$, $\mathcal{F}$ and $\mathbb{P}$ denote a sample set, a σ-algebra, and a probability measure respectively, and consider $L+1$ i.i.d. random variables (r.v.) $Z'_1, ..., Z'_M, Z_1, ..., Z_N$ and Z_{N+1} with $L = N + M$. Let Z'_i for $i = 1, ..., M$ be the *training set* and Z_i for $i = 1, ..., N$ be the *calibration set*. Note that Z_{N+1} is not part of either set. We use the lower case of a r.v. to denote a realization[1]. An *Inductive Nonconformity M-measure* (INM) is a measurable function $A : \mathbf{Z}^M \times \mathbf{Z} \to \mathbb{R}$. While no additional requirements are needed for A, intuitively an effective INM will assign a high real number to any element in $\mathbf{Z}$ that does not conform to a training set (in $\mathbf{Z}^M$). An *Inductive Nonconformal Predictor* (INP) is a set predictor defined as

$$\Gamma^\epsilon(z_1, ..., z_N, z'_1, ..., z'_M) \doteq \{z \in \mathbf{Z} \ : \ p^z > \epsilon\}, \tag{6.3}$$

where $\epsilon \in [0,1]$ is the *significance level*, the p-values are defined as

$$p^z \doteq \frac{|\{i : R_i \geq R^z\}| + 1}{N+1}, \tag{6.4}$$

and

$$R_i \doteq A((z'_1, ..., z'_M), z_i) \text{ for } i = 1, ...N, \qquad R^z \doteq A((z'_1, ..., z'_M), z), \tag{6.5}$$

are the *nonconformity scores.*

In the following, when it is clear from the context we omit the arguments of the INP and write Γ^ϵ instead of $\Gamma^\epsilon(z_1, ..., z_N, z'_1, ..., z'_M)$. Intuitively, z belongs to the INP Γ^ϵ if there are strictly more than $\lfloor \epsilon(N+1) - 1 \rfloor$ elements R_i in the calibration set with a higher (worse) or equal nonconformity score than R^z. It is easy to see that $\epsilon' < \epsilon''$ implies that $\Gamma^{\epsilon''} \subseteq \Gamma^{\epsilon'}$. The INP is the set predictor mentioned in the discussion at the beginning of this section: similarly to the original formulation of CP, given some prediction method depending on the training set, the INP uses the available calibration set to produce a set prediction guaranteed to contain the correct prediction. The elements included in the set prediction are all the $z \in \mathbf{Z}$ that conform well enough with the calibration set, according to the chosen INM. The following theorem specifies the PAC-style guarantees for training-conditional CP.

Theorem 2 ([113]). *Choose $\epsilon, E \in [0,1]$[2], fix the training set $Z'_1 = z'_1, ..., Z'_M = z'_M$,*

[1] Our considerations hold also for the case where $\mathbf{Z} = \mathbf{X} \times \mathbf{Y}$ where $\mathbf{X}$ and $\mathbf{Y}$ represent a measurable feature space and label space respectively and each $z \in \mathbf{Z}$ may be written as $z = (x, y)$ where $x \in \mathbf{X}$ is some feature and $y \in \mathbf{Y}$ a label. For clarity we omit the exact structure of $\mathbf{Z}$.

[2] A brief note on the notation. In the original formulation of CP ϵ has a double role: it is the significance level (appearing as the index to the INP Γ^ϵ) and it describes the coverage probability as $1 - \epsilon$, see [119] for details. In the training-conditional formulation the latter role is covered by E, that is $1 - E$ is the coverage probability and ϵ remains the significance level, see [113].

let N be the size of the calibration set, and consider the event

$$S_E \doteq \{(z_1, ..., z_N) \in \mathbf{Z}^N : \mathbb{P}(Z_{N+1} \in \Gamma^\epsilon(z_1, ..., z_N, z'_1, ..., z'_M)) \geq 1 - E\} \tag{6.6}$$

in the σ-algebra $\mathcal{F}^N$ of the product probability space $(\mathbf{Z}^N, \mathcal{F}^N, \mathbb{P}^N)$, where Γ^ϵ is defined according to equations (6.3-6.5). It holds that

$$\mathbb{P}^N(S_E) \geq 1 - \delta, \tag{6.7}$$

where $\delta \doteq Bin_{N,E}(J) = \sum_{j=0}^{J} \binom{N}{j} E^j (1-E)^{N-j}$ is the cumulative binomial distribution with N trials and probability of success E, with $J \doteq \lfloor \epsilon(N+1) - 1 \rfloor$.

The quantities $1 - \delta$ and $1 - E$ are sometimes referred to as the *confidence* and *coverage probability* (which is not the coverage probability mentioned in 6.2). Theorem 2 is to be understood in the following way. Given two values ϵ and E, for the given training set, the event S_E is the subset of $\mathbf{Z}^N$ containing all the tuples $(z_1, ..., z_N)$ such that the INP Γ^ϵ contains a realization of Z_{N+1} with probability at least $1 - E$, or, in other words, Γ^ϵ returns a subset of $\mathbf{Z}$ of measure at least $1 - E$. By equation (6.7) the measure of this set of tuples S_E is at least $1 - \delta$, where δ depends on ϵ, β and N. This form of guarantees where a double layer of nested probabilities is present is called Probably Approximately Correct (PAC). Moreover, this is a distribution-free result, that is, it holds for every $\mathbb{P}$ as long as the samples used to construct the INP are i.i.d. and $\mathbb{P}$-distributed. In particular, in this work we focus on Bernoulli-distributed r.v.'s, and Theorem 2 holds for any value of the parameter b of a Bernoulli distribution. Observe that the confidence $1 - \delta$ and the quantity $1 - \alpha$ mentioned in 6.2 play a similar role in that they described the outmost layer of probability, compare for instance equations equation (6.7) and equation (6.2). Finally, we note that ϵ and E are chosen *a priori*; in other words, they cannot be defined as random variables depending on a realization of the calibration set, as is erroneously done in [115].

6.4. A Special Case of Interest

In this section we draw a parallel between the BPCI and training-conditional CP and show the fundamental difference between the two approaches.

Let the INM be an indicator function for the set $Q \subset \mathbf{Z}$, that is

$$A((z'_1, ..., z'_M), z) \doteq \begin{cases} 1 \text{ if } z \in Q, \\ 0 \text{ if } z \in \overline{Q}. \end{cases} \tag{6.8}$$

Typically Q depends on $z'_1, ..., z'_M$. For example, in binary classification problems, the training set may be used to train a parameterized function that assigns one of two labels to all $z \in Q$, as in Support Vector Machines. However, since Theorem 2 assumes a given training set, we omit this dependency here. A point z with a high nonconformity score is interpreted as poorly conforming to the training set. For this reason, in 6.4.1 the set Q will represent the unsafe region of a dynamical system. Given a fixed training set, the nonconformity scores of the calibration set follow an i.i.d. Bernoulli distribution with parameter b, i.e. $R_i \sim \text{Bern}_b$, where $b \doteq \mathbb{P}(Q)$. Using a BPCI method it is directly possible to derive a conservatively valid confidence interval for the parameter b describing the probability of drawing a sample in Q, as shown in 6.2. Can a training-conditional CP approach also provide a conservatively valid confidence interval for b based on the calibration set? The answer is no. We illustrate this with an example.

Example 5. *Part 1*

Suppose that the calibration set has size 2, i.e. $N = 2$. Up to reindexing, there are three distinct outcomes.

Case 1: With probability $(1-b)^2$ we have $z_1, z_2 \notin Q$, resulting in nonconformity scores $R_1 = R_2 = 0$. We construct the prediction set Γ^ϵ following its definition equation (6.3).

- For all $z \in Q$, we have that $R^z = 1$, meaning z has the highest (worst) nonconformity score. Since $|\{i \leq 2 : R_i \geq R^z\}| = 0$ the corresponding p-value is $p^z = \frac{1}{3}$.
- For all $z \in \overline{Q}$ we have that $R^z = 0$ resulting in and $p^z = 1$.

The inclusion of z in the predicted set Γ^ϵ depends on the significance level ϵ.

- If $\epsilon \in [\frac{1}{3}, 1)$ then any $z \in Q$ is excluded from Γ^ϵ since $p^z = \frac{1}{3} \leq \epsilon$, while all $z \in \overline{Q}$ are included since $p^z = 1 > \epsilon$. Thus, $\Gamma^\epsilon = \overline{Q}$.
- If $\epsilon \in [0, \frac{1}{3})$ then any $z \in Q \cup \overline{Q} = \mathbf{Z}$ has a sufficiently high p-value, meaning $\Gamma^\epsilon = \mathbf{Z}$.

Case 2: With probability $2b(1-b)$ we have $(z_1 \in Q \wedge z_2 \in \overline{Q})$ or $(z_2 \in Q \wedge z_1 \in \overline{Q})$ hence $R_1 \cup R_2 = \{0, 1\}$.

- If $R^z = 1$ then $p^z = \frac{2}{3}$.
- If $R^z = 0$ then $p^z = 1$.

Thus:

- If $\epsilon \in [0, \frac{2}{3})$ then $\Gamma^\epsilon = \mathbf{Z}$.
- If $\epsilon \in [\frac{2}{3}, 1)$ then $\Gamma^\epsilon = \overline{Q}$.

Case 3: With probability b^2 we have $z_1, z_2 \in Q$ and $R_1 = R_2 = 1$.

- If $R^z = 1$ then $p^z = 1$.
- If $R^z = 0$ then $p^z = 1$ as well.

Then for any significance level $\epsilon \in [0, 1)$ it holds $\Gamma^\epsilon = \mathbf{Z}$.

In summary, for any fixed ϵ the INP is fully determined by the calibration set through equations 6.3-6.5; as a result, Γ^ϵ can be thought equivalently as a discrete random variable with support Q, $\overline{Q}$ and $\mathbf{Z}$, see Figure 6.1. Part 2.

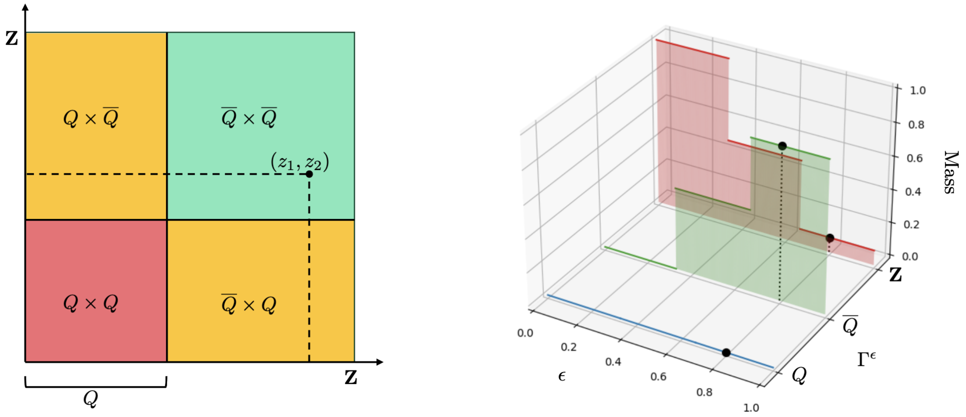


Figure 6.1.: On the left, a representation of the product space $\mathbf{Z}^2 = \mathbf{Z} \times \mathbf{Z}$, partitioned accordingly to the sets Q and $\overline{Q}$, and a hypothetical calibration set (z_1, z_2) as in Case 1. On the right, a summary of Case 1, 2 and 3. On the x-,y-,z-axes are represented the values of ϵ, the prediction (or support) of the INP, and the probability mass function respectively. For any given ϵ, the INP Γ^ϵ can be viewed as a discrete random variable with support Q, $\overline{Q}$ and $\mathbf{Z}$. In the figure, for $\epsilon = 0.8$ and $b = 0.3$, the INP predicts Q with probability 0, $\mathbf{Z}$ with probability b^2, and $\overline{Q}$ with probability $1 - b^2$.

Now, fix $E \in [0, 1]$ and consider any $\epsilon \in [\frac{2}{3}, 1)$. Theorem 2 implies that

$$\mathbb{P}^2(S_E) \geq E^2, \tag{6.9}$$

where

$$S_E \doteq \{(z_1, z_2) \in \mathbf{Z}^2 : \mathbb{P}(Z_3 \in \Gamma^\epsilon(z_1, z_2, z_1', \ldots, z_M')) \geq 1 - E\}.$$

However, equation (6.9) does not provide a confidence interval for the probability of drawing a new sample in Q, or conversely in $\overline{Q}$. For $\epsilon \in [\frac{2}{3}, 1)$, $\Gamma^\epsilon = \mathbf{Z}$ with probability b^2 (from Case 3) and $\Gamma^\epsilon = \overline{Q}$ with probability $1 - b^2$ (from Case 1 and 2)[3], see Figure 6.1. Theorem 2 is a distribution-free result and as such it holds for all values of b, leading to two cases $b \leq E$ and $b > E$:

1. If $b \leq E$ (i.e. $1 - b \geq 1 - E$):
   - If $z_1, z_2 \in Q$ (Case 3) we have that $\mathbb{P}(Z_{N+1} \in \Gamma^\epsilon) = \mathbb{P}(\mathbf{Z}) = 1 \geq 1 - E$, hence $Q \times Q \subseteq S_E$.
   - If at least one of z_1 and z_2 belongs to $\overline{Q}$ (Case 1 and 2) we have that $\mathbb{P}(Z_{N+1} \in \Gamma^\epsilon) = \mathbb{P}(\overline{Q}) = 1 - b \geq 1 - E$, hence $\overline{Q \times Q} \subseteq S_E$.

[3] If Γ^ϵ predicts $\overline{Q}$ it implies that the nonconformity score of Z_{N+1} is predicted to be 0, whereas if it predicts $\mathbf{Z}$ then all we know is that the nonconformity score of Z_{N+1} is predicted to be in $\{0, 1\}$ which is uninformative.

- Thus, $S_E = \mathbf{Z}^2$. Trivially, $\mathbb{P}^2(S_E) = \mathbb{P}^2(\mathbf{Z}^2) = 1 \geq E^2$.

2. If $b > E$ (i.e. $1 - b < 1 - E$)
   - If $z_1, z_2 \in Q$ (Case 3), as before, $\mathbb{P}(Z_{N+1} \in \Gamma^\epsilon) = \mathbb{P}(\mathbf{Z}) = 1 \geq 1 - E$, and once again $Q \times Q \subseteq S_E$.
   - If at least one of z_1 and z_2 belongs to $\overline{Q}$ (Case 1 and 2) then $\mathbb{P}(Z_{N+1} \in \Gamma^\epsilon) = \mathbb{P}(\overline{Q}) = 1 - b < 1 - E$, hence such z_1 and z_2 do not belong to S_E by definition.
   - Thus, $\mathbb{P}^2(S_E) = \mathbb{P}^2(Q \times Q) = b^2 \geq E^2$.

Theorem 2 holds for both cases, since we have either $\mathbb{P}^2(\mathbf{Z}^2) = 1 \geq E^2$ or $\mathbb{P}^2(Q \times Q) = b^2 \geq E^2$. Now, assume $b > E$ and that the calibration set gives $R_1 = 0$ and $R_2 = 1$. What can we say about b?

For the given calibration set and significance level the INP predicts $\Gamma^\epsilon = \overline{Q}$, hence it is tempting to say that $\mathbb{P}^2(\mathbb{P}(\overline{Q}) \geq 1 - E) = \mathbb{P}^2(1 - b \geq 1 - E) \geq E^2$, or equivalently $\mathbb{P}^2(b \leq E) \geq E^2$: recalling equation (6.2), we may conclude that $[0, E]$ is a E^2 confidence interval for b. But this is clearly not true: since we assumed that $b > E$ the interval $[0, E]$ will never contain the parameter b (note that none of the arguments of $\mathbb{P}^2()$ depends on (z_1, z_2) in the preceding statement, unlike equation (6.2)). We conclude from this example that this is not a viable path to obtain a PAC bound for b comparable to equation (6.2).

The example above leads us to the following remark and main message of this chapter.

Remark 15. Theorem 2 guarantees the correctness of the *set* predictor Γ^ϵ. Adopting the frequentist perspective, it is a statement on how often the set predictor Γ^ϵ constructed from N samples attains the desired coverage level $1 - E$ for a new realization of Z_{N+1}. In other words, since b is unknown, if $b > E$ the INP attains the desired coverage level only when $\Gamma^\epsilon = \mathbf{Z}$ (which is a trivial prediction), and it does not attain the desired coverage level when $\Gamma^\epsilon = \overline{Q}$. Essentially, the confidence level of E^2 is attained by making trivial predictions sufficiently often. If instead $b \leq E$, the INP is always correct. Thus, Theorem 2 does not estimate b or provide information on the probability of a specific score or class, which is the goal of BPCI methods. See the appendix for a graphical representation.

To further clarify, consider the equivalent set predictor mapping the elements z predicted by Γ^ϵ to their respective nonconformity score

$$\overline{\Gamma^\epsilon}(z_1, ..., z_N, z'_1, ..., z'_M) \doteq \bigcup_{z \in \Gamma^\epsilon(z_1, ..., z_N, z'_1, ..., z'_M)} A((z'_1, ..., z'_M), z),$$

which amounts to $\overline{\Gamma^\epsilon} = \{0, 1\}$ when $\Gamma^\epsilon = \mathbf{Z}$ and $\overline{\Gamma^\epsilon} = \{0\}$ when $\Gamma^\epsilon = \overline{Q}$. Let $R_{N+1} \doteq A((z'_1, ..., z'_M), Z_{N+1})$ be the score of the $N+1$-th sample. Then, we can replace the event $R_{N+1} \in \overline{\Gamma^\epsilon}$ with $Z_{N+1} \in \Gamma^\epsilon$ in equation (6.6). In essence, both BPCI methods and training-conditional CP provide PAC guarantees but differ in scope: while BPCI methods compute an interval containing the true value b describing the probability of *the event that the* $N+1$*-th score equals* 1*, i.e.* $R_{N+1}=1$ (with probability not less than $1-\alpha$), training-conditional CP computes a lower bound for the probability of *the event that the* $N+1$*-th score is contained in the predicted set of scores, i.e.* $R_{N+1}\in\overline{\Gamma^{\epsilon}}$ (with probability not less than $1-\delta$).

Remark 15 extends to any scenario where the nonconformity score takes values from a finite set, effectively defining a classification problem. Training-conditional CP provides a framework for constructing a set predictor that guarantees the desired coverage level with a minimum confidence. The predictor adapts to the calibration data: for 'good' calibration data, it produces tight sets (few classes), while for 'poor' calibration data, it outputs loose sets (many classes). On average, the probability that the calibration data yields a predictor attaining the coverage level of $1-E$ is at least $1-\delta$.

Depending on the choice of ϵ and E, we have shown that the $1-\delta$ confidence level may be achieved simply by predicting the entire sample space (i.e., all classes) sufficiently often (see Figure 6.2). However, this approach does not provide meaningful information about the probability of a specific class, which is the focus of equation (6.2) and, more generally, BPCI methods.

6.4.1. A Note on Safety Verification for Dynamical Systems

Recent studies have applied training-conditional CP, particularly Theorem 2, to provide PAC guarantees on the safety of control systems with neural network-based controllers [114, 115], and more broadly, on the safety of autonomous systems [116]. In this section we show that these works follow the reasoning outlined in 6.4, and are therefore incorrect. Below, we follow the notation used in [115], but the same applies to the other works.

Consider a dynamical system defined by $\dot{x}=f(x)$ where $x\in X\subseteq\mathbb{R}^n$, a fixed time horizon $T\in\mathbb{R}_{>0}$. Denote by $\xi_x(\tau)$ for $\tau\in[0,T]$ the state trajectory of the system at time τ when initialized at x (for simplicity we assume that the solution to the differential equation exists and is unique)[4]. Let $X_A\subset X$ represent a set of undesirable states, and consider the cost function defined as

$$J(x)\doteq\min_{\tau\in[0,T]}d(\xi_x(\tau)),$$

where $d:X\to\mathbb{R}$ is a function satisfying

$$d(x)\leq\gamma\iff x\in X_A,\ d(x)>\gamma\iff x\in X\setminus X_A,$$

for some threshold $\gamma\in\mathbb{R}$. The function d measures the distance between a point in the domain and the unsafe set X_A. An instructive example for the discussion is below is to choose $\gamma=0$ and $d:X\to\{0,1\}$, with $d=0\iff x\in X_A$ and $d=1\iff x\in X\setminus X_A$, but the same applies for any different choice. In this case, J assigns a positive real number to a point $x\in X$ if and only if the state trajectory

[4] In the original paper the trajectory ξ depends on a learned controller and depends on a training set $Z'_1,\ldots,Z'_M$. For clarity we omit this dependence here, since the training set is given and is fixed.

from x never intersects with X_A. Let $(X, \mathcal{F}, \mathbb{P})$ be a probability space. To quantify system safety probabilistically, we seek to estimate $\mathbb{P}(\{x : J(x) > 0\})$, i.e. the probability of sampling an initial state that leads to a safe trajectory. In [115] the authors define the nonconformity score as $R_i \doteq J(x_i)$ for $i = 1, ..., N$, and are therefore interested in estimating $\mathbb{P}(\{x : R^x > 0\})$. However, this is equivalent to defining a nonconformity measure as

$$A(x) \doteq \begin{cases} 1 \text{ if } x \in X_A, \\ 0 \text{ if } x \in X \setminus X_A, \end{cases} \tag{6.10}$$

and we have shown that this line of reasoning is not suitable for estimating the parameter b of a Bernoulli r.v. given N i.i.d. realizations $R_i \sim \text{Bern}_b$ of it.

Since [114] relies on the framework of [115], it suffers from the same issue. Additionally, in [116, Theorem 1], the authors re-derive Theorem 2, originally from [113]. They claim that training-conditional CP reduces to the Clopper-Pearson confidence interval when the underlying i.i.d. random variables are Bernoulli-distributed (see their Sec. Proofs-D). However, we have disproved this claim.

6.5. Conclusion

In this chapter we examined existing methodologies to use training-conditional CP for statistical safety verification, a problem that can be reduced to estimating the expectation of a Bernoulli random variable. While training-conditional CP remains a powerful tool for uncertainty quantification we have shown that it is not appropriate for BPCI problems. Specifically, we clarified the correct interpretation of confidence intervals and PAC-style guarantees for training-conditional CP. We do not rule out the possibility that a different formulation of CP could be applied to BPCI problems.

6.6. Appendix

6.6.1. Numerical Validation

We validate empirically equation (6.9) as follows and represent the results graphically in Figure 6.2.

We define a list of values for E by $E_q = 0.01 + 0.01 * q$ for $q = 0, ..., 98$. For every value of E_q we consider an underlying Bernoulli distribution with parameter $b_{1,q} = E_q - \alpha E_q < E_q$ (right figure) and an underlying Bernoulli distribution with parameter $b_{2,q} = E + \alpha E_q\% > E_q$ (left figure) with $\alpha = 0.005$. For every value of $q = 0, ..., 98$ we examine the two situations $b_{1,q} \leq E_q$ and $b_{2,q} > E_q$, as mentioned in Example 1 - Part 2. The significance level ϵ is set to $2/3$. We draw $n_{\text{cal}} = 5 \cdot 10^4$ pairs of calibration points $\{z_1^{(i)}, z_2^{(i)}\}_{i=1}^{n_{\text{cal}}}$. For every pair of calibration points $z_1^{(i)}, z_2^{(i)}$ we construct the resulting INP as $\Gamma^{\epsilon}_{(i)} \doteq \Gamma^{\epsilon}(z_1^{(i)}, z_2^{(i)}, ...)$, draw $n_{\text{test}} = 5 \cdot 10^4$ test points $\{z_{N+1}^{(j)}\}_{i=j}^{n_{\text{test}}}$ and compute the empirical frequency $\hat{g}_i = \frac{|\{j=1,...n_{\text{test}} : z_{N+1}^{(j)} \in \Gamma^{\epsilon}_{(i)}\}|}{n_{\text{test}}}$

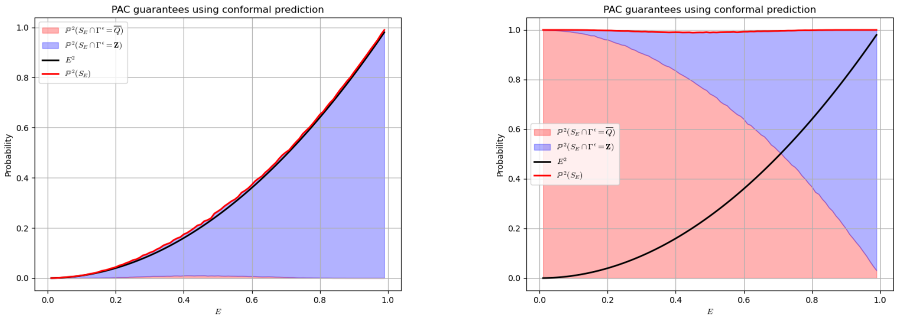


Figure 6.2.: On the left the curves resulting from $b_{2,q} > E_q$, on the right the curves resulting from $b_{1,q} \leq E_q$, for $q = 0, ..., 98$.

as an approximation for $\mathbb{P}(Z_{N+1} \in \Gamma^{\epsilon}_{(i)})$; finally we compute $\hat{h} = \frac{|\{i=j,...n_{\text{cal}}:\hat{g}_i \geq 1-E\}|}{n_{\text{cal}}}$ as an approximation to $\mathbb{P}^2(S_E)$ shown in the plots as the solid red line. The solid black line represents the curve given by E^2, which remains always below the red line in both plots, as expected. The area shaded in blue represents the fraction of the $\hat{g}_i$'s for which the INP $\Gamma^{\epsilon}_{(i)}$ is equal to $\mathbf{Z}$, whereas the area shaded in red represents the fraction of the $\hat{g}_i$'s for which the INP $\Gamma^{\epsilon}_{(i)}$ is equal to $\overline{Q}$ *and* $\hat{g}_i$ is greater or equal than $1 - E$. It is visible in the left plot that the only reason why the solid red line (approximating $\mathbb{P}^2(S_{E_q}) = b^2_{2,q}$) is above E^2_q is that the INP is allowed to predict the entire set $\mathbf{Z}$. In contrast, on the right the solid red line approximates $\mathbb{P}^2(S_{E_q}) = 1$ since any pair of $z^{(i)}_1, z^{(i)}_2$ results in a prediction $\Gamma^{\epsilon}_{(i)}$ satisfying $\mathbb{P}(Z_{N+1} \in \Gamma^{\epsilon}_{(i)}) \geq 1 - E_q$; accordingly, for a fixed q, the area shaded in red covers approximately $1 - b^2_{1,q}$ of the 'Probability' axis and the area shaded in blue approximately $b^2_{1,q}$.

In summary, in both situations the theorem is confirmed empirically, since the red line is always above the black line. In the first case, where $b_{2,q} > E_q$, the minimum confidence level of E^2_q is attained by predicting sufficiently often the entire sample space $\mathbf{Z}$, precisely with a frequency of $b^2_{2,q}$, as this is the only set prediction attaining the required coverage probability of $1 - E_q$. Unfortunately, a prediction of the entire sample space is uninformative. In the second case, where $b_{1,q} \leq E_q$, any predicted set between $\overline{Q}$ and $\mathbf{Z}$ attains the required coverage probability of $1 - E_q$.

II

Stochastic Systems with Partially Known Dynamics

7

Enhancing Data-Driven Stochastic Control via Bundled Interval MDP

This chapter investigates a novel scheme to obtain data-driven abstractions of discrete-time stochastic processes in terms of finite-state stochastic models, whose actions lead to nondeterministic transitions over the space of probability measures. The data-driven component of the proposed methodology lies in the fact that we only assume samples from an unknown probability distribution. We also rely on the model of the underlying dynamics to build our abstraction through backward reachability computations. The nondeterminism in the probability space is captured by a collection of Markov Processes, and we identify how this model can improve upon existing abstraction techniques in terms of satisfying temporal properties, such as safety or reach-avoid. The connection between the discrete and the underlying dynamics is made formal through the use of the scenario approach theory. Numerical experiments illustrate the advantages and main limitations of the proposed techniques with respect to existing approaches.

This chapter is based on the publication [59].

7.1. Introduction

In the previous chapters, we focused on abstracting systems with unknown deterministic dynamics from symbolic external behaviours. Here, we shift our focus to abstract stochastic systems, whose dynamics are modelled as a known deterministic part plus an unknown stochastic part; in this setting, we characterise the unknown distribution of the latter from a data set of noise realisations.

Scenario theory, and more broadly concentration-inequality-based approaches, for systems with stochastic dynamics, is also an active field of research. In [12], the authors synthesise barrier certificates for stochastic systems using templated candidates, based on a data set of one-step transitions. In [35] the authors study probabilistic reachability specifications on uncertain MDPs parametrized by a set of random variables with unknown distribution, and exploiting results from scenario theory provide PAC guarantees for the desired specification. In [38] the authors propose synthesizing a Stochastic Bisimulation Function (SBF) relating an infinite state MDP with a finite state MDP approximation, based on a data set of one-step transitions to find the correct parameters; similarly to works cited in Chapter 3 they establish a connection between a ROP and a scenario program exploiting the fundamental results of [65]: this, however, requires knowledge of the Lipschitz constants of the SBF, and, therefore, of the system's dynamics. Instead, in [8, 36], the systems' dynamics are modelled as a known deterministic part with additive unknown noise and Markov models are created using the scenario approach to evaluate transition probabilities; in addition, [37] also tackles epistemic uncertainty in the deterministic dynamics. With the exception of [36], the works cited above differ from ours in what is assumed to be known about the dynamics of the system.

In this chapter, we revisit the approach presented in [36] to abstract a discrete-time dynamical system with additive noise as an IMDP, using techniques from the scenario approach, with the overall goal of studying reach-avoid control problems. In doing so, we introduce an instance of Robust MDP [31] where the ambiguity set has a particular structure. Building upon the results therein, we present a new strategy to construct such an abstraction by incorporating nondeterminism in the transitions: this allows us to search for policies over a larger action space and, therefore, to synthesise controllers for a wider variety of scenarios, with in particular the attainment of the specification of interest with a possibly higher probability, if compared to [36].

7.2. Notation and Modelling Framework

Consider a stochastic control system represented by a stochastic difference equation, where the dynamics of the state $X_{k+1} \in \mathcal{X} \subset \mathbb{R}^n$ at time $k+1$ depends on a known function $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ of the previous state and input, and on the noise W_k. We formally define the model below.

Definition 18. *Consider a probability space* $(\Omega, \mathcal{F}, \mathbb{P})$ *and an independent and identically distributed random process* $\{W_k(\omega) \in \mathbb{R}^n : k \in \mathbb{N}_0, \omega \in \Omega\}$*. A* Stochastic Difference Equation *(SDE) with additive noise is a sequence of random variables*

(RVs) defined as

$$X_{k+1} = f(X_k, u_k) + W_k, \tag{7.1}$$

where $u_k : \mathbb{N}_0 \to \mathcal{U} \subseteq \mathbb{R}^m$ *and* $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$*.*

We denote by $\mathcal{P}(S)$ the set of all probability distributions on a discrete or continuous set S. Let us define a stochastic kernel $T : \mathbb{R}^n \times \mathcal{U} \to \mathcal{P}(\mathbb{R}^n)$ to describe the distribution of X_{k+1} given x_k and u_k as

$$X_{k+1} \sim T(\cdot \mid x_k, u_k). \tag{7.2}$$

Furthermore we denote the next state under the *nominal dynamics* of the SDE without additive noise as

$$\hat{x}_{k+1} = f(x_k, u_k). \tag{7.3}$$

We focus on synthesising a controller for a SDE enforcing a reach-avoid specification $\varphi_{x_0}^H$ over a finite time horizon [5, 37]. Let $\mathcal{X}_G \subset \mathcal{X}$ be a *goal* set and let $\mathcal{X}_U \subset \mathcal{X}$ be an *unsafe* set. Similarly to equation (2.11), $\varphi_{x_0}^H$ denotes the trajectories initialized at x_0 reaching the goal set $\mathcal{X}_G$ within H steps, while avoiding the unsafe set $\mathcal{X}_U$. Given a controller $\phi : \mathbb{R}^n \times \mathbb{N}_0 \to \mathcal{U}$ the state of the system at time k, X_k, is a RV, hence we denote the *probability* of satisfying a specification as $\mathbb{P}_\phi\{\varphi_{x_0}^H\}$.

Problem Statement 4. Given a reach-avoid specification $\varphi_{x_0}^H$ and an SDE with *unknown* additive noise, compute a controller ϕ by abstracting the SDE to a finite-state Markov model and provide a lower bound on the probability of satisfying $\varphi_{x_0}^H$.

Our goal is to produce an abstraction for equation (7.1), where we assume to have full knowledge of the nominal dynamics $f(\cdot)$, whilst the distribution of the noise W_k, and hence the distribution of X_k, is unknown. In the next sections, we demonstrate how this problem can be solved by translating the infinite-state SDE to a finite-state Markov model, recalled in Section 2.5. Below, we introduce a novel Markov model and, in Section 7.4, we illustrate how it improves the flexibility of existing abstraction schemes for SDEs.

Definition 19. *A* Bundled Interval Markov Decision Process *(bIMDP) is a tuple* $M_{\vdots} = (\mathcal{S}, \mathcal{A}, P_{\vdots}, r)$ *where* $\mathcal{S}$, $\mathcal{A}$, *and* r *are defined as in Definition 7,* $P_{\vdots} : \mathcal{S} \times \mathcal{A} \to \wp(\mathcal{P}(\mathcal{S}))$ *is an* uncertain transition probability function *such that for all* s, s' *and* a *there exists some* $K \in \mathbb{N}$ *such that* $P_{\vdots}(s, a)(s') := \bigcup_{k=1}^{K} [\underline{p}_k, \overline{p}_k] \subseteq [0, 1]$.

IMDPs and bIMDPs are instances of Robust MDPs where the *ambiguity set* has a special structure, see [31]. As the name suggests, bIMDPs can be thought of as a collection of IMDPs. Both can be thought of as collections of MDPs, each represented by an instance of a transition probability function $P \in P_{\vdots}$ or $P \in P_{\updownarrow}$, respectively.

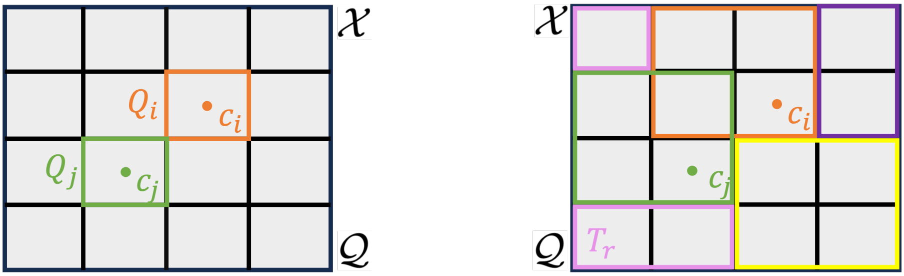
7.3. Finite-State Abstraction

Next, we describe the components required to construct a finite-state abstraction of an SDE: discretisation of the state space, transitions among the abstract states, and the evaluation of the probability associated with each transition.

7.3.1. State Space Discretization

Let $\mathcal{Q} = \{Q_i\}_{i=1}^N$ be a partition of $\mathcal{X} \subset \mathbb{R}^n$ such that every Q is an n-dimensional convex polytope and let Q_0, the closure of $\mathbb{R}^n \setminus \mathcal{X}$, be a so-called *absorbing region*. We define an abstract state for each element of $\{Q_i\}_{i=0}^N$, yielding a set of $N+1$ discrete states $\mathcal{S} = \{s_i\}_{i=0}^N$. We define the relation $R \subseteq \mathbb{R}^n \times \mathcal{S}$ where $(x, s_i) \in R$ if and only if $x \in Q_i$, and the notation $R(x) := \{s : (x, s) \in R\}$ and $R^{-1}(s_i) := \{x : (x, s_i) \in R\} = Q_i$. Given a finite collection of *reference* points in $\mathcal{X}$ denoted by $\{c_i\}_{i=1}^N$ such that $R(c_i) = s_i$, we define a bijective map $\psi : \mathcal{S} \setminus s_0 \to \{c_i\}_{i=1}^N$ as $\psi(s_i) = c_i$. For simplicity, the points $\{c_i\}_{i=1}^N$ represent the centres of each cell of a uniform grid, as shown in Figure 7.1a.

Remark 16. Without loss of generality, suppose that the goal set $\mathcal{X}_G$ and the unsafe set $\mathcal{X}_U$ align with the partition, meaning they can be represented as a union of elements from $\mathcal{Q}$. This allows to translate a specification from the concrete system $\varphi_{x_0}^H$ to an equivalent specification on a MDP $\varphi_s'^H$, as described by equation (2.11).



(a) (b)

Figure 7.1.: (a) Partition $\mathcal{Q}$ of the domain of interest $\mathcal{X}$, where Q_i and Q_j are two elements of the partition, c_i and c_j are the respective reference points. (b) Cover of target sets $\mathcal{T}$ of the partition $\mathcal{Q}$. Each color represents a different target set.

7.3.2. Actions

Below, we define actions linking a single abstract state to (possibly) a *set* of abstract states, named the target set.

Let $\mathcal{T} = \{T_i\}_{i=1}^M$ be a finite collection of *target sets* covering $\mathcal{Q}$, in particular $\mathcal{T} \subseteq \wp(\mathcal{Q})$, see Figure 7.1b. Further, for every $i = 1, ..., M$, let C_i be the set of reference states associated with T_i, that is $C_i = \bigcup_{Q \in T_i} \{\psi(R(x)) : x \in Q\}$.

The collection of target sets defines a set of M (arbitrary) elements $\mathcal{A} = \{a_r : r = 1, ..., M\}$, termed *abstract actions*, where a_r is associated with the target set T_r as shown below. We construct the set of enabled actions at s_i as follows.

Action a_r is enabled at state s_i if for every state $x_k \in R^{-1}(s_i)$ there exists a control input u_k such that the next state under nominal dynamics equation (7.3) belongs to the set of reference points associated with T_r, or, in other words, $\hat{x}_{k+1} \in C_r$. Formally, we define the (nominal) *backward reachable set* of a point $x' \in \mathcal{X}$, and, with a slight abuse notation, the *backward reachable set* of a set $C_r \subset \mathcal{X}$ as

$$\text{Pre}(x') := \{x \in \mathcal{X} : \exists u \in \mathcal{U}, f(x, u) = x'\} \tag{7.4}$$

$$\text{Pre}(C_r) := \bigcup_{c_j \in C_r} \text{Pre}(c_j). \tag{7.5}$$

We require that backward reachable sets of reference points, or a union of those, can contain regions Q_i.

Assumption 3. *The backward reachable set of any reference point has a non-empty interior.*

For instance, if the system's dynamics is linear, namely, $f(x_k, u_k) = Ax_k + Bu_k$, if A and B are invertible, and U has a non-empty interior then Assumption 3 holds, since equation (7.4) results in an affine transformation of U, see also [36].

Action a_r is enabled at state s_i iff $R^{-1}(s_i)$ is contained in $\text{Pre}(C_r)$, as shown in Figure 7.2a:

$$Q_i = R^{-1}(s_i) \subseteq \text{Pre}(C_r) \iff a_r \in \mathcal{A}(s_i). \tag{7.6}$$

If Q_i satisfies equation (7.6) for some C_r, for $x \in Q_i$ there may exist multiple control inputs leading x to C_r, that is the sets $\{\text{Pre}(c_j)\}_{c_j \in C_r}$ need not be disjoint. We exploit the ordering of the partitioning sets $\{Q_i\}_{i=1}^{N}$ to assign a unique control input driving the state to C_r.

Let $c^* : \mathcal{X} \times \mathcal{A} \to \mathcal{X}$ be a function mapping a continuous state and abstract action to a continuous reference point indexed by the lowest integer, that is

$$c^*(x, a_r) := \arg\min_{c_j \in C_r} j \qquad \text{s.t.} \quad x \in \text{Pre}(c_j). \tag{7.7}$$

Let us define a control law $u^* : \mathcal{X} \times \mathcal{A} \to \mathcal{U}$ such that

$$u^*(x, a_r) \in \{u : f(x, u) = c^*(x, a_r)\}. \tag{7.8}$$

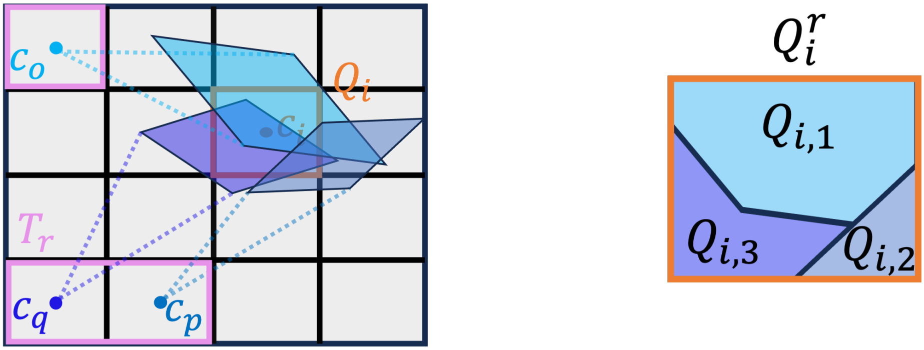
For every C_r, operation equation (7.7) naturally induces a partition on a set $R^{-1}(s_i)$ satisfying equation (7.6), defined as

$$Q_i^r := \{X \subseteq Q_i : \forall x, x' \in X, c^*(x, a_r) = c^*(x', a_r)\}. \tag{7.9}$$

In other words, the partition $Q_i^r = \{Q_{i,l}\}_{l=1}^{L_r}$ is a collection of L_r sets defined by the points sharing the same next state under nominal dynamics and control law u^*, see Figure 7.2b.

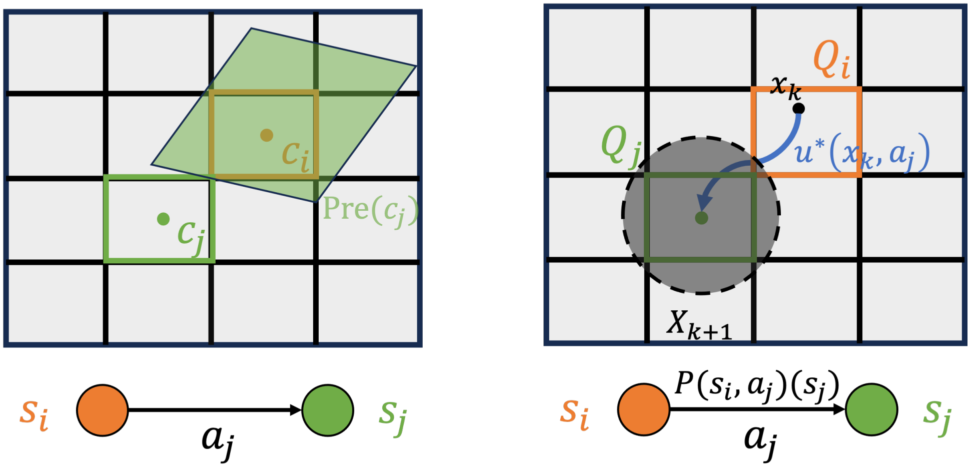
7.3.3. Transition Probabilities

In the remaining part of this section, we recall the construction scheme proposed by [36] to abstract a SDE with additive noise to an MDP and introduce a shortcoming of such a procedure. Suppose that the collection of target sets $\mathcal{T}$ coincides with



(a) (b)

Figure 7.2.: (a) $\text{Pre}(C_r)$ represented as the union of $\text{Pre}(c_o)$, $\text{Pre}(c_p)$, and $\text{Pre}(c_q)$; $a_r \in \mathcal{A}(s_i)$ (b) Assuming the ordering $o < p < q$, the partition Q_i^r induced on Q_i by C_r.



(a) (b)

Figure 7.3.: (a) Consider $T_j = Q_j$. If action $a_j \in \mathcal{A}(s_i)$ then $\text{Pre}(c_j) \supseteq Q_i$. (b) For every $x_k \in Q_i$ there exists an input $u_k = u^*(x, a_j)$ driving the state to c_j. The shaded area represents the support of $T(dx_{k+1}|x_k, u_k)$.

the partition $\mathcal{Q}$, more precisely, $T_r = \{Q_r\}$ for every $r = 1, ..., N$. In this tailored setting, we can simplify our discussion: every set C_r contains a single element, namely c_r, hence action a_r is enabled in the abstract state s_i if and only if $R^{-1}(s_i) \subseteq \text{Pre}(C_r) = \text{Pre}(c_r)$. Similarly, Q_i^r is the trivial partition and contains a single element, namely $Q_i^r = \{Q_i\}$ – cfr. equation (7.9) – as depicted in Figure 7.3a. A finite-state abstraction that describes this framework is an MDP $M = (\mathcal{S}, \mathcal{A}, P, R)$, where given $x_k \in R^{-1}(s_i)$, an abstract action $a_j \in \mathcal{A}(s_i)$, and $u_k = u^*(x_k, a_j)$ the probability of transitioning to the abstract state s_j can be computed as:

$$P(s_i, a_j)(s_j) := \int_{R^{-1}(s_j)} T(dx_{k+1}|x_k, u_k). \tag{7.10}$$

Due to the noise being additive equation (7.1) and given the control law u^* we can express equation (7.10) as

$$P(s_i, a_j)(s_j) = \mathbb{P}\{\omega \in \Omega : c_j + W_k(\omega) \in R^{-1}(s_j)\}, \tag{7.11}$$

where we have used the fact that under nominal dynamics $f(x_k, u^*(x, a_j)) = c_j$. This situation is depicted in Fig. 7.3b.

7.3.4. Shortcomings and Motivating Example

One shortcoming of this approach is that it may lead to a significant *under-approximation* of the dynamics of the concrete system. Formally expressed in equation (7.6), if Q_i is not fully contained in $\text{Pre}(c_j)$, a_j is not enabled for s_i. As such, one may have to exclude a large set of actions if the dynamics are not well aligned with the chosen partition.

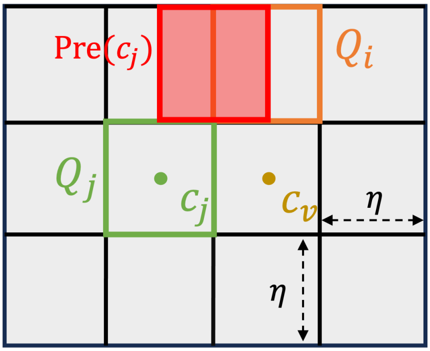


Figure 7.4.: The dynamics are misaligned with the partition.

Example 6. Consider the SDE given by

$$X_{k+1} = X_k - u_k + W_k, \tag{7.12}$$

where $X_k \in \mathbb{R}^2$, W_k is a RV taking values in $\mathbb{R}^2$, $u_k \in \mathcal{U} = [0, \eta] \times [\eta/2, 3\eta/2] \subset \mathbb{R}^2$ for some $\eta > 0$. The partition $\mathcal{Q}$ of $\mathcal{X}$ is a uniform grid where each set Q_i is a $\eta \times \eta$ box. Let $\mathcal{T} = \mathcal{Q}$. Consider a reference state c_j, the center of the box Q_j, and let us examine $\text{Pre}(C_j) = \text{Pre}(c_j)$: by inverting the nominal dynamics, we can characterise such a set as

$$\text{Pre}(c_j) = \{x : \exists u \in \mathcal{U}, c_j + u = x\},$$

which represents a copy of Q_j with its center shifted by $[\eta/2, \eta]$, as shown in Figure 7.4. Considering only one reference point leads to an empty MDP: i.e., all abstract states s have an empty action set. If instead every target set comprises a pair of adjacent cells in the same row, it is easy to see that Q_i is included in $\text{Pre}(\{c_j, c_v\})$. This observation motivates the following Section, containing our main contribution[1].

7.4. Uncertain Transition Probabilities

In contrast to Section 7.3.3, let us now consider a general cover $\mathcal{T}$ of the partition Q , where at least one of the M target sets, say T_r, contains more than one element of Q; equivalently, C_r contains more than one reference state. Let $Q_i \subseteq \text{Pre}(C_r)$, as depicted in Figure 7.2a, and consider the non-trivial partition $Q_i^r = \{Q_{i,l}\}_{l=1}^{L_r}$ induced on Q_i by equation (7.7) and described by equation (7.9). We know that for every $l = 1, ..., L_r$ and for every $x \in Q_{i,l}$ there exists a control law $u^*(x, a_r)$ that drives the state to one of the reference points in C_r.

In this new setting, it is not possible to describe the transition from s_i under action a_r to a future abstract state $s_j = R(c)$ for $c \in T_r$ by a *single* transition probability function as in equation (7.11), but rather by a *set* of transition probability functions. Indeed, the probability of reaching s_j from s_i under action a_r depends on the actual continuous state $x \in Q_i$ from which the transition takes place.

In order to encompass this framework, we define an uncertain probability transition function $P_{\vdots}$ which encapsulates all possible cases and captures the *nondeterminism* introduced by clustering multiple reference points. Consider an abstract state s_i, an action $a_r \in \mathcal{A}(s_i)$, the partitioning Q_i^r, and suppose that $x_k \in Q_{i,l}$ for some $l \in \{1, ..., L_r\}$: under the control input $u_k = u^*(x_k, a_r)$ the next state under nominal dynamics is the reference point $c^*(x, a_r) \in C_r$. Let us define

$$P^l(s_i, a_r)(s_j) := \int_{R^{-1}(s_j)} T(dx_{k+1}|x_k, u_k), \tag{7.13}$$

By enumerating $l = 1, ..., L_r$ we obtain a set of transition probability functions which describes all cases, namely $x \in Q_{i,1}, ..., x \in Q_{i,L_r}$. Accordingly, we define the bIMDP $M_{\vdots} = (\mathcal{S}, \mathcal{A}, P_{\vdots}, R_{\vdots})$ where

$$P_{\vdots}(s_i, a_r)(s_j) = \bigcup_{l=1}^{L_r} P^l(s_i, a_r)(s_j). \tag{7.14}$$

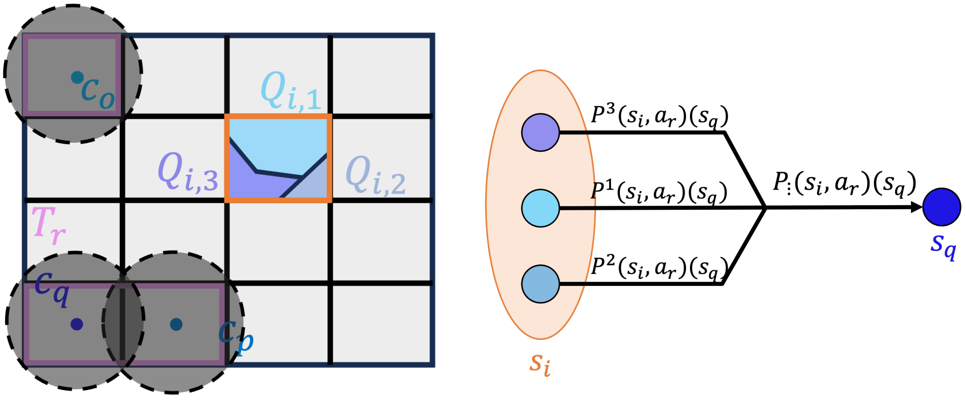
This is shown graphically in Figure 7.5a and Figure 7.5b.

Remark 17. The target sets can be selected arbitrarily. A simple choice is to select adjacent cells, creating 'neighborhoods' of increasing size.

7.5. PAC Probability Intervals via Sampling

Computing equation (7.13) is possible only when the distribution of the additive noise W_k is perfectly known. Additionally, even if it were known, computing the

[1] A qualitatively different approach to mitigate the illustrated shortcoming is to consider the backward reachable set of reference polytopes, as in [37], instead of reference points.



(a) (b)

Figure 7.5.: (a) Computation of $P^l(s_i, a_r)(s_q)$ as per equation (7.13) for $l = 1, 2, 3$. (b) The uncertain transition probability function from a state s_i to s_q under action a_r is a set with L_r values.

integral explicitly could be difficult or undesirable in certain cases. Instead we provide a lower and upper bound of $P^l(s_i, a_r)(s_j)$ using the *sampling-and-discarding* scenario approach proposed in [122] and improved in [123]. In particular, we adopt the framework presented in [36], under the following necessary assumption

Assumption 4 (Non-degeneracy)**.** *For every k, W_k has a density with respect to the Lebesgue measure.*

We summarise the results therein here. Let us collect a set of $Z \in \mathbb{N}$ i.i.d. samples of W_k, denoted $w_k^{(1)}, ..., w_k^{(Z)}$ and define the quantities

$$Z_{s_j}^{\text{in}} = |\{w_k^{(i)} : w_k^{(i)} + c_j \in Q_j\}|, \qquad Z_{s_j}^{\text{out}} = Z - Z_{s_j}^{\text{in}}.$$

In words, $Z_{s_j}^{\text{in}}$ is the number of samples $w_k^{(i)}$ which, when shifted by c_j, fall within region Q_j.

Theorem 3. *(PAC probability intervals [36, Theorem 1]) Given Z samples of the noise W_k, compute $Z_{s_j}^{out}$ and fix a confidence parameter β. It holds that*

$$\mathbb{P}^Z\{\underline{p}_{j,l} \leq P^l(s_i, a_r)(s_j) \leq \overline{p}_{j,l}\} \geq 1 - \beta,$$

where $\underline{p}_{j,l} = 0$ if $Z_{s_j}^{out} = Z$, $\overline{p}_{j,l} = 1$ if $Z_{s_j}^{out} = 0$, and otherwise $\underline{p}_{j,l}$ and $\overline{p}_{j,l}$ are *respectively the solutions of*

$$\frac{\beta}{2Z} = \sum_{i=0}^{Z_{s_j}^{out}} \binom{Z}{i} (1 - \underline{p}_{j,l})^i \underline{p}_{j,l}{}^{Z-i},$$

$$\frac{\beta}{2Z} = 1 - \sum_{i=0}^{Z_{s_j}^{out}-1} \binom{Z}{i} (1 - \overline{p}_{j,l})^i \overline{p}_{j,l}{}^{Z-i}.$$

Theorem 3 allows us to provide an upper and lower bound on the individual transition probabilities $P^l(s_i, a_r, s_j)$.

We can then describe the resulting abstraction as a bIMDP $M'_{\vdots} = (\mathcal{S}, \mathcal{A}, P'_{\vdots}, R_{\vdots})$ where

$$P'_{\vdots}(s_i, a_r)(s_j) := \bigcup_{l=1}^{L_r} [\underline{p}_{j,l}, \overline{p}_{j,l}]. \tag{7.15}$$

In order to leverage existing algorithms for value iteration on IMDPs, following the approach in [32, 33], we can embed (abstract) the resulting bIMDP into an IMDP $M_{\updownarrow} = (\mathcal{S}, \mathcal{A}, P_{\updownarrow}, r)$ where the uncertain transition probability from s_i to state s_j under action a_r is defined as

$$P_{\updownarrow}(s_i, a_r)(s_j) = [\underline{p}_j, \overline{p}_j], \tag{7.16}$$

with $\underline{p}_j = \min P'_{\vdots}(s_i, a_r)(s_j)$ and $\overline{p}_j = \max P'_{\vdots}(s_i, a_r)(s_j)$.

It is obvious from equation (7.15) and equation (7.16) that the collection of MDPs described by $M'_{\vdots}$ is a subset of the MDPs described by $M_{\updownarrow}$. Indeed if $P \in P'_{\vdots}$ it implies that $P \in P_{\updownarrow}$. Let $\overline{\pi}$ denote the optimal policy for $M_{\updownarrow}$. It follows that

$$\min_{P \in P_{\updownarrow}} \mathbb{P}_{\overline{\pi},P}\{\varphi_s'^H\} \leq \min_{P \in P'_{\vdots}} \mathbb{P}_{\overline{\pi},P}\{\varphi_s'^H\}.$$

This embedding allows us to employ existing tools to obtain a policy with a valid lower bound on the probability of satisfaction of the reach-avoid property for the bIMDP, see Remark 3. We compute the optimal policy $\overline{\pi}$ with respect to the IMDP, which in general differs from the optimal policy with respect to the bIMDP. We leave this for future work.

We conclude this section with the following theorem connecting the probability of satisfaction of the reach-avoid property on the IMDP given an optimal policy with the probability of satisfaction of the reach-avoid property on the underlying dynamical system by refining the policy to a time-varying feedback controller. The proof follows the rationale in [36, Theorem 2], and is omitted for brevity.

Theorem 4 (Adapted from [36])**.** *Let $\overline{\pi}$ denote the optimal policy equation (2.12) for the IMDP obtained according to equation (7.16), and let $L_{\max} := \max_r L_r$. For $\alpha := \beta N M L_{\max}$, the controller $\phi := u^*(x, \overline{\pi}(R(x), k))$, and $x_0 \in R^{-1}(s)$ it holds*

$$\min_{P \in P_{\updownarrow}} \mathbb{P}_{\overline{\pi},P}\{\varphi_s'^H\} \geq \eta \Rightarrow \mathbb{P}^Z\{\mathbb{P}_{\phi}\{\varphi_{x_0}^H\} \geq \eta\} \geq 1 - \alpha. \tag{7.17}$$

7.6. Experimental Evaluation

We demonstrate our results on two systems. Our approach is suitable for nonlinear systems; however, we focus on linear dynamics to simplify the computation of equation (7.4). Our code is based upon [36], which for brevity is denoted as “single-target procedure” (STP) where the target sets are chosen as in Section 7.3.3, and has been modified in order to include multiple-target transitions (denoted as MTP) according to Section 7.4. We use PRISM [124] to compute optimal IMDP policies. The interval transition probabilities are computed from $Z = 2 \cdot 10^4$ samples and a confidence $\beta = 10^{-8}$.

Example 1 (Cont’d). We consider the dynamical model equation (7.12), over the domain $\mathcal{X} = [-25, 25]^2$, partitioned into 2500 square regions, where the goal set is the region $\mathcal{X}_G = [-25, 25] \times [-25, -24]$. The control input lies in the set $\mathcal{U} = [0, 1] \times [0.5, 1.5]$, and the noise follows a Gaussian distribution $W_k \sim \mathcal{N}(0, 0.15 \cdot I)$. Our goal is the computation of a control policy making the dynamics reach the goal in 50 time steps at most. As outlined in Example 6, the STP returns an IMDP with no actions enabled. In contrast, as argued at the end of Section 7.3.4, if we define every target set as the union of two adjacent cells on the same row, the Pre set covers an entire cell region. Our procedure creates an IMDP equipped with 42391 transitions, and computes a policy whose lower bounds on the satisfaction probability is shown in Fig. 7.6, with a confidence (see equation (7.17)) $\alpha \simeq 0.12$.

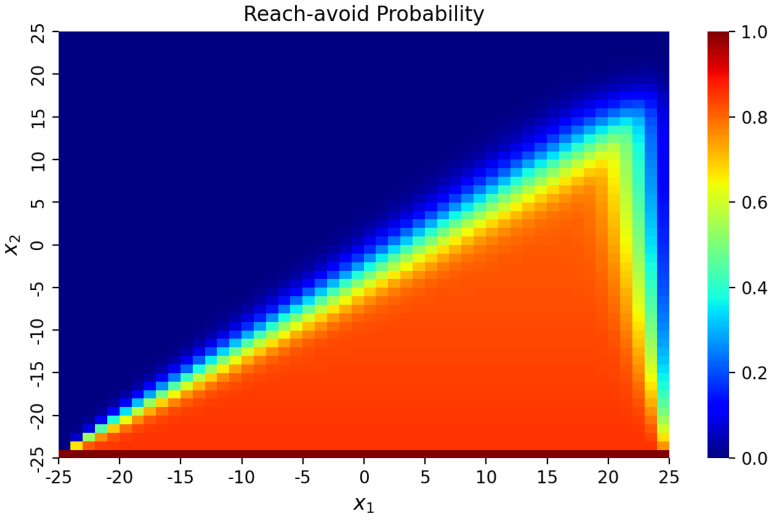


Figure 7.6.: Lower bound on the probability of reaching the goal set (represented by the lowest row of states) for Example 6.

Double Integrator. Let us consider the stochastic model

$$X_{k+1} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} X_k + \begin{bmatrix} 0.5 & 0 \\ 0 & 1 \end{bmatrix} u_k + W_k, \tag{7.18}$$

where $W_k \sim \mathcal{N}(0, 0.15 \cdot I)$. The reach-avoid task is to reach the set $[-2, 2]^2$ in 5 time steps, while avoiding states $X \notin [-11, 11]^2$. The control input is limited by the set $[-2, 4] \times [-3, 3]$. We partition the domain $\mathcal{X} = [-11, 11]^2$ into square partitions, in five different configurations: 11x11, 15x15, 18x18, 20x20, and 25x25 regions. For the MTP the target sets are all the pairs of adjacent cells, vertically or horizontally. The complete results are reported in Table 7.1, in terms of computational time, number of transitions of the resulting abstractions, and in percentage of states with a positive probability of reaching the goal set, along with the confidence α (see equation (7.17)) for the MTP approach. Due to the coarseness of the first two partitions (11x11 and 15x15) the STP returns an IMDP with no enabled actions, as motivated in Section 7.3.4, while our new approach successfully returns a policy. For finer partitions, the STP returns smaller abstractions than the MTP; this is expected, as the MTP considers significantly more target sets – this is reflected in the higher time needed to construct the abstract models. In turn, the MTP yields a larger portion of states with a positive probability of reaching the goal set, thanks to the additional actions available. With finer partitioning, the difference between the STP and the MTP diminishes; the benefit of larger backward reachable sets in the MTP is offset by the smaller cell volume within the partition.

Partition	Transitions		Time [s]		Reach [%]		$\alpha \cdot 10^3$
	STP	MTP	STP	MTP	STP	MTP	MTP
11^2	–	1907	–	7.9	–	61.9	0.5
15^2	–	5592	–	11.2	–	62.2	1.9
18^2	6423	26511	8.4	28.6	54.3	66.0	4.0
20^2	13155	52580	9.8	35.1	62.5	68.5	6.1
25^2	77507	262952	15.5	63.7	67.7	71.0	15

Table 7.1.: Comparison between the STP and MTP abstractions, in terms of number of transitions, computational time, and percentage of states that have a positive probability to reach the goal set, and confidence α for the MTP.

7.7. Discussion and Conclusions

This chapter presented a novel abstraction procedure for discrete-time stochastic systems, exploiting nondeterministic transitions to generate finite-state abstract models. By allowing target sets to comprise multiple cells, rather than a single cell, we show that we can build an abstraction for a greater variety of situations, thus generalising the scope of earlier results. Our experiments show that this flexibility comes at the cost of generating larger (in terms of transitions) models than the existing single-target approach, and hence introducing more behaviours in the abstraction. The computation of equation (7.5) may return a nonconvex set despite the arguments of the union being convex, complicating verifying whether the LHS of equation (7.6) holds. The selection of target sets and the embedding of a bIMDP into an IMDP affect the performance of our method: a deeper study of tailored algorithms for bIMDPs exploiting the structure of the uncertain transition function obtained by this scheme is a matter of future efforts.

III

Deterministic Systems with Known Dynamics

8

Multi-resolution Approximate Bisimulation Learning

This chapter introduces a fully automatic framework for synthesising compact, finite-state deterministic abstractions of deterministic, continuous-state autonomous systems under locally specified resolution requirements.

This framework builds on multi-resolution approximate bisimulations, a generalisation of classical ϵ-approximate bisimulations, that support state-dependent error bounds and subsumes both variable- and uniform-resolution relations. We show that some systems admit multi-resolution bisimulations but no ϵ-approximate bisimulation.

Furthermore, this chapter studies the existence of multi-resolution approximately bisimilar abstractions for all incrementally uniformly bounded (δ-UB) systems, thereby broadening the applicability of symbolic verification to a larger class of dynamics; as a trivial special case, this result also covers incrementally globally asymptotically stable (δ-GAS) systems.

The Multi-resolution Abstraction Synthesis Problem (MRASP) is solved via a scalable Counterexample-Guided Inductive Synthesis (CEGIS) loop, combining mesh refinement with counterexample-driven refinement. This ensures soundness for all δ-UB systems, and ensures termination in certain special cases.

Experiments on linear and nonlinear benchmarks, including non-δ-GAS and non-differentiable cases, demonstrate that our algorithm yields abstractions up to 50% smaller than Lyapunov-based grids while enforcing tighter, location-dependent error guarantees.

This chapter is based on the publication [60].

8.1. Introduction

In the previous chapters, we considered abstractions of systems with entirely or partially unknown deterministic dynamics; data is employed to compensate for the lack of knowledge. In contrast, here we investigate a setting where the dynamics are known but standard approaches are computationally expensive or inapplicable altogether. We combine model knowledge and data to propose an abstraction synthesis algorithm for a broad class of systems.

A successful paradigm that enabled the abstraction of increasingly complex dynamical systems is that of approximate relations [125–127].In particular, the notion of ϵ-approximate bisimulation marked a turning point: by relaxing the strict equivalence required by exact bisimulation, it allowed for a broad class of infinite-state dynamical systems to be represented by finite-state models while still preserving essential behavioural properties within a controlled error margin [6, 9, 43].

The first concrete instantiations of this approach were based on uniform gridding of the state space. These constructions, however, quickly encounter scalability issues due to the curse of dimensionality. In addition, they often rely on certificates such as Lyapunov-like functions [44, 45], whose existence and computation can be restrictive in practice. These limitations have spurred the search for more adaptable abstraction methods capable of handling a wider variety of systems while mitigating the curse of dimensionality.

Rather than using a uniform grid, [46, 47] propose using a set of grids of different coarseness to establish a ϵ-approximate bisimulation for δ-GAS switched systems, extending the framework of [44] to multi-scale abstractions for safety and reachability controller synthesis. This allows for generating abstraction with smaller state sets, compared to the original approach based on a uniform grid. In [49] propose using a set of grids of different coarseness to abstract systems for which a growth bound is known. Additionally, they rely on the notion of feedback-refinement relation (FRR) and tackle controller synthesis for ω-regular specifications. In [128], the authors introduce a functional to predict the computational effort to generate an abstraction for which an FRR can be established, and propose a method to select the hyper-intervals characterising the relation, so as to minimise the number of transitions. Another relevant work based on FRR is given by [129], where the authors construct goal-specific sparse abstractions using a backward search and ellipsoidal coverings of L-smooth dynamical systems while relying on a set of affine controllers, allowing them to reduce the size of the resulting abstraction. In [130], several of these approaches are implemented as an efficient tool.

Our setting relies on the notion of multi-resolution approximate bisimulation relations, originally proposed in [50]. Multi-resolution approximate bisimulation relations allow the user to specify a heterogeneous resolution by means of a resolution function, allowing for local adaptations of the abstraction's precision. The authors propose a local linearization of the dynamics under a differentiability assumption, and given a candidate abstraction and a norm-based relation template, they cast the existence of a multi-resolution approximate abstraction as a linear optimisation program. When the program is infeasible, a greedy heuristic refines the candidate abstraction. Their approach is very fast on linear systems; however, computing local linearizations can be computationally intensive [10]. In our work, we do not rely on local linearizations of the system's dynamics and use sampled transitions instead, allowing us to apply our solution to a broader class of systems.

In the context of data-driven abstractions with known dynamics [53] propose a notion of neural simulation relation; given an abstraction obtained from a discretised concrete state and input set, they parametrise a relation and a controller as neural networks, train them on the system's transition, and exploit Lipschitz continuity to guarantee an ϵ-approximate simulation between from the abstraction to the concrete system. In [52], the authors abstract continuous-time deterministic dynamical systems using samples of the system's vector field to train a neural network and obtain a hybrid automaton whose continuous dynamics are linear with additive disturbance, and the correctness of the automaton is checked by an SMT solver. [51, 131] propose a methodology to abstract discrete-time deterministic systems with discrete state spaces to a stutter-insensitive bisimilar model via queries to a Satisfiability Modulo Theories (SMT) prover, based on a data set of the system's transitions. In analogy with their work, we frame the synthesis of an abstraction and a multi-resolution approximate bisimulation as an SMT query and solve it via CEGIS.

8.2. Preliminaries

Given a metric d on a set $\mathcal{X} \subseteq \mathbb{R}^n$, we denote an open ball of radius r centered at $x \in \mathcal{X}$ by $B_r(x) \doteq \{x' \in \mathcal{X} : d(x, x') < r\}$. For $x \in \mathbb{R}^n$, $x[i]$ denotes the i-th component, and $x[i : j]$ the subvector consisting of components i through j, inclusive. A function $\alpha : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ is of class $\mathcal{K}$ if it is continuous, strictly increasing, and $\alpha(0) = 0$. A function $\beta : \mathbb{R}_{\geq 0} \times \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ is of class $\mathcal{KL}$ if, for each fixed $t \geq 0$, $\beta(\cdot, t) \in \mathcal{K}$, and for each fixed $s \geq 0$, $\beta(s, t) \to 0$ as $t \to \infty$.

In this chapter we consider autonomous deterministic TSs, where the output map is the identity. For this reason, we simply the notation and denote such systems as $S = (\mathcal{X}, \mathcal{X}_0, f)$ where $\mathcal{X}_0 \subseteq \mathcal{X} \subset \mathbb{R}^n$ define the (initial) state set and $f : \mathcal{X} \to \mathcal{X}$ is the transition function. For $i \geq 1$ we denote by f^i the composition of f i-times, i.e. $f^1 = f$, $f^2 = f \circ f$, and so on.

Definition 20. *A set T is called* transient *for S if there exists $\tau \in \mathbb{N}$ called* transient time *such that for all $x \in T$ there exists $q \leq \tau$ such that $f^q(x) \notin T$.*

Definition 21. *Let $\hat{\mathcal{X}}$ be a collection of $k \in \mathbb{N}$ elements in $\mathcal{X}$. A quantizer of $\mathcal{X}$ is a function $\kappa : \mathcal{X} \to \hat{\mathcal{X}}$ mapping every element of $\mathcal{X}$ to one of the elements of $\hat{\mathcal{X}}$ (e.g. nearest neighbour).*

A quantizer can be equivalently represented by a *mesh.*

Definition 22. *The mesh associated to κ is defined as $\mathcal{M} \doteq \{(\hat{x}_1, C_1), (\hat{x}_2, C_2), \dots (\hat{x}_k, C_k)\}$, where C_i is the i-th cell of the mesh, and $\{C_i \doteq \{x \in \mathcal{X} : \hat{x}_i = \kappa(x)\}\}_{i=1}^{k}$ is a partition of $\mathcal{X}$.*

We aim to represente S by a deterministic finite-state abstraction $\hat{S}$.

Definition 23. *A deterministic abstraction of S is the TS $\hat{S} \doteq (\hat{\mathcal{X}}, \hat{\mathcal{X}}_0, \hat{f})$ where $\hat{\mathcal{X}}_0 \doteq \{\hat{x} \in \hat{\mathcal{X}} : \exists x \in \mathcal{X}_0 \, . \, \hat{x} = \kappa(x)\}$ and $\hat{f} : \hat{\mathcal{X}} \to \hat{\mathcal{X}} : \hat{x} \to \kappa(f(\hat{x}))$.*

8.3. Multi-resolution Bisimulation

Exact bisimulation relations between systems require their behaviours to be identical [127]. In the context of abstracting continuous-state dynamical systems, this notion is restrictive. Instead, when there is a metric on the behaviours, *approximate* relations have proven to be successful, as they only require the behaviours of the concrete system and its abstraction to not differ more than a prescribed margin of error [6, 44]. Below, we recall the definition of ϵ-approximate bisimulation relation, adapted for deterministic TS.

Definition 24 (Adapted from [6])**.** *A relation $R \subseteq \mathcal{X} \times \hat{\mathcal{X}}$ is an ϵ-approximate bisimulation relation between S and $\hat{S}$, written $S \simeq^{\epsilon}_{R} \hat{S}$ if*

1. *for all $x \in \mathcal{X}_0$ there exists $\hat{x} \in \hat{\mathcal{X}}_0$ such that $(x, \hat{x}) \in R$, and for all $\hat{x} \in \hat{\mathcal{X}}_0$ there exists $x \in \mathcal{X}_0$ such that $(x, \hat{x}) \in R$*

2. *for all $(x, \hat{x}) \in R$ it holds that $(f(x), \hat{f}(\hat{x})) \in R$,*

3. *for all $(x, \hat{x}) \in R$ it holds that $\boldsymbol{d}(x, \hat{x}) \leq \epsilon$.*[1]

The definition above has been used successfully to prove that Incrementally Globally Asymptotically Stable (δ-GAS) systems (satisfying an additional minor assumption) admit an abstraction that is ϵ-approximately bisimilar to the system [6, 44].

This chapter follows the seminal work of [50], focusing on a generalisation of the notion of ϵ-approximate relations, to construct *compact* abstractions $\hat{S}$. The resulting abstractions must approximate the original system S with a sufficiently high resolution specified by the user as a *space-dependent resolution specification*, expressed as a relation between the concrete and abstract states. Formally, let $\boldsymbol{d}$ be a metric on $\mathcal{X}$, $\epsilon : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_{>0}$ be a *resolution function*, and $\overline{R} \subseteq \mathcal{X} \times \mathcal{X}$ be a relation of interest, or *resolution relation*, of the form

$$\overline{R} \doteq \{(x, \hat{x}) \in \mathcal{X} \times \mathcal{X} : d(x, \hat{x}) \leq \epsilon(x, \hat{x})\}. \tag{8.1}$$

We define the following sufficient conditions to establish a resolution relation between the TS and its abstraction.

Definition 25 ([50])**.** *A relation R is a multi-resolution approximate simulation relation from S to $\hat{S}$ with resolution $\overline{R}$, written $S \preceq^{\overline{R}}_{R} \hat{S}$, if:*

1. *for all $x \in \mathcal{X}_0$ there exists $\hat{x} \in \hat{\mathcal{X}}_0$ such that $(x, \hat{x}) \in R$,*

[1] Since we restrict our attention to deterministic concrete systems and deterministic abstractions, condition 2 is stated in a simplified form when compared to the standard ϵ-approximate simulation relation.

2. *for all* $(x, \hat{x}) \in R$ *it holds that* $(f(x), \hat{f}(\hat{x})) \in R$,
3. $R \subseteq \overline{R}$.

Definition 26. *A relation R is a multi-resolution approximate bisimulation relation between S and $\hat{S}$ (with resolution $\overline{R}$), written $S \simeq_{R}^{\overline{R}} \hat{S}$, if:*

1. *R is a multi-resolution approximate simulation relation from S to $\hat{S}$,*
2. *for all $\hat{x} \in \hat{\mathcal{X}}_0$ there exists $x \in \mathcal{X}_0$ such that $(x, \hat{x}) \in R$*

Intuitively, equation (8.1) can be thought of as a specification of the coarseness entailed by approximating the dynamics of S by those of $\hat{S}$. For example, suppose that $\epsilon(x, \hat{x}) = u \min(||x||, ||\hat{x}||) + v$ for some $u, v \in \mathbb{R}_{\geq 0}$; such a relation allows for more coarseness far from the origin, requiring higher resolution near the origin. The reason to introduce the relation R is to enable us to fix a template for relations that are a subset of $\overline{R}$.

Note that if $\overline{R}$ is obtained by fixing a uniform $\epsilon \in \mathbb{R}_{\geq 0}$ resolution, i.e. $\epsilon(x, \hat{x}) = \epsilon$ for all x and $\hat{x}$ we recover the definition of ϵ-approximate bisimulation relation given in Definition 24.

Definition 27. *A transition system S, a resolution relation $\overline{R}$ and a deterministic abstraction of S: $\hat{S}$, define an* Approximate Relation Synthesis Problem *(ARSP). Formally, we denote the set of its solutions as,*

$$\mathcal{W}(\hat{S}, \overline{R}) \doteq \{R \subseteq \mathcal{X} \times \mathcal{X} : S \simeq_{R}^{\overline{R}} \hat{S}\}. \tag{8.2}$$

A transition system S and a resolution relation $\overline{R}$ define a Multi-resolution Abstraction Synthesis Problem *(MRASP), with solution space,*

$$\mathcal{W}(\overline{R}) \doteq \{(\hat{S}, R) : S \simeq_{R}^{\overline{R}} \hat{S}\} \tag{8.3}$$

When referring to ϵ-approximate bisimulation relations we overload the notation and write $\mathcal{W}(\epsilon) \doteq \{(\hat{S}, R) : S \simeq_{R}^{\epsilon} \hat{S}\}$.

Problem Statement 5. Characterise sufficient conditions on S and $\overline{R}$ ensuring that $\mathcal{W}(\overline{R})$ is nonempty.

The following statements are a direct consequence of Definitions 24 and 26

Proposition 9. *If the resolution function $\epsilon(x, \hat{x})$ has global extrema then for all $\hat{S}$:*

- $\mathcal{W}(\hat{S}, \overline{R}) \supseteq \mathcal{W}(\hat{S}, \epsilon)$, *and $\overline{R}$ for all* $\epsilon \leq \min_{(x,\hat{x}) \in \overline{R}} \epsilon(x, \hat{x})$.
- $\mathcal{W}(\hat{S}, \overline{R}) \subseteq \mathcal{W}(\hat{S}, \epsilon)$, *and $\overline{R}$ for all* $\epsilon \geq \max_{(x,\hat{x}) \in \overline{R}} \epsilon(x, \hat{x})$.

8.3.1. Incrementally Globally Asymptotically Stable Systems

Definition 28 ([132]). *A system S is δ-GAS if there exists a class $\mathcal{KL}$ function β such that for all $k \in \mathbb{N}$ and $x_0, x'_0 \in \mathcal{X}_0$ it holds that*

$$|x_k - x'_k| \leq \beta(|x_0 - x'_0|, k) \tag{8.4}$$

with $x_0 x_1 \ldots x_k$ and $x'_0 x'_1 \ldots x'_k$ (finite) behaviors of S.

It is well known that δ-GAS systems defined on a compact subset of $\mathbb{R}$ admit the construction of an abstraction for which it is possible to define an ϵ-approximate bisimulation relation with the original system. We summarise below this result.

Theorem 5 ([6]). *For a δ-GAS system $S = (\mathcal{X}, \mathcal{X}_0, f)$ with $\mathcal{X} \subset \mathbb{R}^n$, for any $\epsilon > 0$ there exists an positive scalar η and an ϵ-approximate bisimulation relation between S and the deterministic abstraction $\hat{S} = (\hat{\mathcal{X}}, \hat{\mathcal{X}}_0, \hat{f})$ with $\hat{\mathcal{X}} = \{x \in \mathcal{X} : x[i] = k_i \frac{2}{\sqrt{n}} \eta, k_i \in \mathbb{Z}, i \in \mathbb{N}\}$.*

The proof of Theorem 5 relies on the existence of a δ-GAS Lyapunov function $V : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$. Finding such a function proving that a system is δ-GAS is not a simple task, and in practice it is often expressed as function of $x - x'$, e.g. $V(x, x') = \sqrt{(x - x')^T P (x - x')}$, see [44]. The relation R used to prove the existence of an ϵ-approximate bisimulation is defined as a level set of the δ-GAS Lyapunov function; consequently, given two abstract states $\hat{x} \neq \hat{x}'$ the sets are $R(\hat{x})$ and $R(\hat{x}')$ are identical, modulo a translation. This, however, is not strictly required by Definition 24: we exploit this fact to show that δ-GAS systems are not the only systems accepting approximate bisimulation relations.

From Proposition 9 we obtain the following corollary.

Corollary 2. *For every δ-GAS system S, and for every resolution specification satisfying the conditions of Propostion 9 it holds that $\mathcal{W}(\overline{R}) \neq \emptyset$.*

8.3.2. Non-Incrementally Globally Asymptotically Stable Systems

While most of the existing literature focuses on approximate relations for systems exhibiting some type of 'contractive' dynamics, we shift our focus to a simple unstable system.

Example 7 (ϵ-approximate bisimulation on non-δ-GAS system). Consider the system defined as $x_{k+1} = 2x_k$ with $\mathcal{X} = [1, 16]$, and suppose that we are interested in finding an abstraction for which there exists an ϵ-approximate bisimulation relation with the original system, with $\epsilon = 1$. The partition given by the diameters of the circles together with the points $\hat{\mathcal{X}} = \{x \in \mathcal{X} : x = 0.5^k x', x' \in \{9, 11, 13, 15\}, 0 \leq k \leq 3\}$ form a mesh, shown in Figure 8.1: the resulting deterministic abstraction constructed according Definition 23 together with the relation induced by the mesh satisfies the design specification. It is easy to see that for $\epsilon = 1$ the abstraction shown above is optimal in the sense that among all abstractions that are ϵ-approximately bisimilar to S it is the one with the lowest number of states, precisely 16.

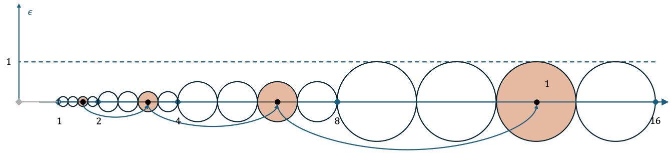
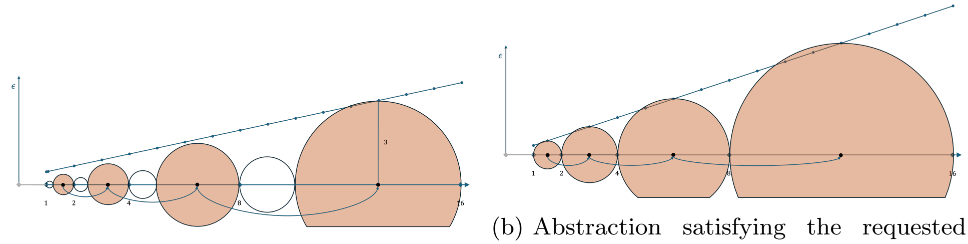


Figure 8.1.: Abstraction satisfying the requested resolution. The abstract state are the midpoint of each segment highlighted by the respective circle, which in turn represent the sought ϵ-approximate relation.

Example 8 (multi-resolution approximate bisimulation on non-δ-GAS system (bounded domain)). Consider again the system of Example 7, but now instead the design specification is $\overline{R} \doteq \{(x, \hat{x}) \in \mathcal{X} \times \mathcal{X} : |x - \hat{x}| \leq \epsilon(x, \hat{x})\}$ with $\epsilon(x, \hat{x}) \doteq p_1(\hat{x}) \doteq u\hat{x} + v$, and such that $p_1(13) = 3$, and $p_1(1) = 0.5$. If we were to construct a deterministic abstraction of S and an ϵ-approximate bisimulation relation, we would need to set $\epsilon = \min_{\hat{x} \in \mathcal{X}} p_1(\hat{x}) = 0.5$, according to Proposition 9. Following an analogous construction to what was shown in Example 7, we can show that the optimal abstraction (in the sense of a minimal cardinality) would comprise 32 abstract states. By shifting our attention to multi-resolution approximate bisimulation relations, we can craft an abstraction with only 8 states, satisfying the resolution specification. The solution shown in Figure 8.2a was obtained from the solution in Example 7 and by merging states 11, 13 and 15 into a single state at 13 with $R(13) = [10, 16]$. Similarly, we can merge the predecessor states and so on. Note that the solution presented in Example 7 also satisfies the specification. However, it does not hold that $\mathcal{W}(\overline{R}) \supseteq \mathcal{W}(1)$.



(a) Abstraction satisfying the requested resolution function.

(b) Abstraction satisfying the requested resolution function, over the unbounded domain.

Example 9 (multi-resolution approximate bisimulation on non-δ-GAS system (unbounded domain)). Finally, consider the system with the same dynamics of Example 7, but with the domain $\mathcal{X} = [1, \infty)$, and with resolution specification $\overline{R}$ defined by $\epsilon(x, \hat{x}) = \frac{\hat{x}}{3}$. A simple solution to the MRASP is given by the state set $\hat{x}_i = 1.5 \cdot 2^i$ and $R(\hat{x}_i) = [\hat{x}_i - 2^{i-1}, \hat{x}_i + 2^{i-1}]$ for $i = 0, 1, \ldots$, shown in Figure 8.2b. Since the resolution function grows unbounded over the unbounded domain, Proposition 9 does not apply: indeed, due to the unstable dynamics over the unbounded domain, it is easy to see that $\mathcal{W}(\epsilon) = \emptyset$ for all $\epsilon > 0$, i.e. it is not possible to construct an abstraction that is ϵ-approximately bisimilar to the original system, yet we have shown that, by considering the larger class of multi-resolution approximate relations, $\mathcal{W}(\overline{R})$ admits a solution.

The examples above motivate our attention on multi-resolution approximate bisimulation relations. In particular, we are interested in exploiting the local dynamics and the local resolution to generate compact abstractions, that is, abstractions satisfying the resolution specification using a minimal number of states. We defer to Section 8.6 the statement and proof of sufficient conditions for general nonlinear systems to admit a solution to an MRASP. Instead, we move on to defining our second problem statement.

8.4. Compact Abstractions

We frame the search for compact abstractions as a minimisation problem. Typically, the size of an abstraction is measured by the number of transitions. By focusing on deterministic abstractions, we can equivalently minimise the number of states of the abstraction.

Problem Statement 6. Given a transition system S and a resolution relation $\overline{R}$, find (one of) the smallest deterministic abstractions $\hat{S} = (\hat{\mathcal{X}}, \hat{\mathcal{X}}_0, \hat{f})$ and relation $R \subseteq \mathcal{X} \times \hat{\mathcal{X}}$ satisfying Definition 26, that is

$$\min_{(\hat{S},R)\in\mathcal{W}(\overline{R})} |\hat{\mathcal{X}}|. \tag{8.5}$$

More explicitly,

$$\begin{aligned}
&\min_{\hat{\mathcal{X}}\subset\mathcal{X},R} \quad |\hat{\mathcal{X}}| && && (8.6)\\
&\text{s.t.} \quad \forall x \in \mathcal{X}_0 \exists \hat{x} \in \hat{\mathcal{X}} \;.\; (x,\hat{x}) \in R) && \text{(coverage)} && (8.7)\\
&\qquad (x,\hat{x}) \in R \implies (f(x), \hat{f}(\hat{x})) \in R && \text{(transition consistency)} && (8.8)\\
&\qquad R \subseteq \overline{R} && \text{(minimum resolution)} && (8.9)
\end{aligned}$$

Finding a solution to the minimisation problem shown above is challenging. For this reason, we introduce some simplifications.

Assumption 5. *A maximum budget k for the cardinality of $\hat{\mathcal{X}} \doteq \{\hat{x}_1, \dots, \hat{x}_k\}$ is given.*

With Assumption 5 in place we are effectively relaxing the minimisation problem 8.6 to a feasibility problem. In order to simplify the exposition, we introduce the following assumption.

Assumption 6. *Let $R_{\hat{\boldsymbol{\epsilon}}} \doteq \{(x,\hat{x}_i) \in \mathcal{X} \times \hat{\mathcal{X}} : d(x,\hat{x}_i) \leq \hat{\epsilon}_i\}$ be the template for R, parametrized by the k positive scalars $\hat{\epsilon}_i$'s, one for every $\hat{x}_i$, with $\hat{\boldsymbol{\epsilon}} \doteq \hat{\epsilon}_1, \dots, \hat{\epsilon}_k$.*

Throughout this work we rely on templating the relation R for computational reasons; however, in Section TBD we choose more general templates, therefore relaxing Assumption 6. Finally, from here on, we also assume for conciseness that $\mathcal{X}_0 = \mathcal{X}$.

Under Assumption 5 and 6, we rewrite our problem statement as below.

Relaxed Problem 1. Find $\hat{x}_1, \ldots, \hat{x}_k \in \mathcal{X}$, and $\hat{\epsilon}_1, \ldots, \hat{\epsilon}_k$ s.t. $\forall x \in \mathcal{X}$:

$$\Phi_{\text{cov}}(x) \doteq \bigvee_{i=1}^{k} \left[d(x, \hat{x}_i) \leq \hat{\epsilon}_i\right], \qquad \text{(coverage)} \tag{8.10}$$

$$\Phi_{\text{con}}(x) \doteq \bigwedge_{i=1}^{k} \left[d(x, \hat{x}_i) \leq \hat{\epsilon}_i \implies d(f(x), \hat{x}_{g(i)}) \leq \hat{\epsilon}_{g(i)}\right], \qquad \text{(transition consistency)} \tag{8.11}$$

$$\Phi_{\text{res}}(x) \doteq \bigwedge_{i=1}^{k} \left[\hat{\epsilon}_i \leq \epsilon(x, \hat{x}_i)\right], \qquad \text{(minimum resolution)} \tag{8.12}$$

where $g\colon [1, \ldots, k] \to [1, \ldots, k]$ maps the index of an abstract state to that of its successor, i.e. $\hat{x}_{g(i)} = \hat{f}(\hat{x}_i)$.

Relaxed Problem 1 represents an MRASP. In the absence of Assumption 6, Relaxed Problem 1 would give rise to a second-order logic formula of the form $\exists \hat{x}_1, \ldots, \hat{x}_k, R \subseteq \mathcal{X} \times \hat{\mathcal{X}}$ s.t. $\forall x \ldots$, whose satisfiability is undecidable in general [133]. Even by parametrising the relation R, we obtain a formula with a quantifier alternation $\exists\forall$ over a set of conjunctions and disjunctions of (in the best case) linear inequalities, which can be, in general, computationally expensive to check even for mature first-order solvers [134]. CEGIS provides a computationally tractable approach by decomposing the quantifier alternation into two subproblems, addressing the existential and universal quantification individually, as detailed in the next section.

8.5. Counterexample-Guided Multi-resolution Abstraction Synthesis

In this section, we describe how we construct a solution for Relaxed Problem 1. We adopt a CEGIS scheme as a workaround to the presence of quantifier alternation. Typically, a CEGIS loop involves a *learner* and a *verifier*. The learner proposes a candidate parameter assignment $\hat{x}_1, \ldots, \hat{x}_k, \hat{\epsilon}_1, \ldots, \hat{\epsilon}_k$ satisfying equation (8.10), equation (8.11), and equation (8.16) over a finite number of states $D \doteq \{x_j\}_{j=1}^{N}$; this allows to transform the quantifier alternation into a single existential query over a finite set of conjunctions as

$$\exists \hat{x}_1, \ldots, \hat{x}_k \in \mathcal{X}, \hat{\epsilon}_1, \ldots, \hat{\epsilon}_k \in \mathbb{R}_{\geq 0} \text{ s.t. } \bigwedge_{x_j \in D} \Phi_{\text{cov}}(x_j) \wedge \Phi_{\text{con}}(x_j) \wedge \Phi_{\text{res}}(x_j). \tag{8.13}$$

The verifier checks whether there exists a counterexample in the domain $\mathcal{X}$ for the candidate parameter assignment. Observe that the learner may use an arbitrary methodology to craft a candidate, potentially based on heuristics; the verifier provides a formal guarantee of correctness in case the check returns no counterexamples. The learner must assign a total of $k \cdot n$ real variables for the abstraction's state set and k real variables for the scalars $\hat{\epsilon}_i$. While gradient-based methods could be used to craft candidate solutions, encoding the constraints in equation (8.13) as a loss function is challenging, particularly due to the implication in the transition consistency condition and the nearest neighbour operation required by $\hat{f}$ being nondifferentiable. Alternatively, when $f(x)$ and $d(x, \hat{x})$ are polynomial functions of their arguments, the learner and the verifier may consist of an SMT solver, which natively handles logical constraints. Unfortunately, obtaining abstractions of practical use typically requires a large k, making the learner excessively slow. As a result, we further split equation (8.13) into two simpler problems. The learner comprises a Clustering stage and a Relation Learner stage. The verifier consists of a single SMT stage.

Relaxed Problem 2.

1. (Clustering) Select $\hat{x}_1, \dots, \hat{x}_k \in \mathcal{X}$.
2. (Relation Learner) Given $\hat{x}_1, \dots, \hat{x}_k \in \mathcal{X}$, find $\hat{\epsilon}_1, \dots, \hat{\epsilon}_k$ s.t. $\forall x \in \mathcal{X}$:

$$\Phi_{\text{cov}}(x) \doteq \bigvee_{i=1}^{k} \left[d(x, \hat{x}_i) \le \hat{\epsilon}_i \right], \qquad \text{(coverage)} \tag{8.14}$$

$$\Phi_{\text{con}}(x) \doteq \bigwedge_{i=1}^{k} \left[d(x, \hat{x}_i) \le \hat{\epsilon}_i \implies d(f(x), \hat{x}_{g(i)}) \le \hat{\epsilon}_{g(i)} \right], \qquad \text{(transition consistency)} \tag{8.15}$$

$$\Phi_{\text{res}}(x) \doteq \bigwedge_{i=1}^{k} \left[\hat{\epsilon}_i \le \epsilon(x, \hat{x}_i) \right] \qquad \text{(minimum resolution)} \tag{8.16}$$

In this second problem relaxation, the choice of the abstract states $\hat{x}_1, \dots, \hat{x}_k$ is separated from the query designing the scalars $\hat{\epsilon_1}, \dots, \hat{\epsilon_k}$; the former are instead computed beforehand by a fast heuristic, specifically by a clustering algorithm. Problem Relaxation 2 transforms the MRASP in Problem Relaxation 1 in 1) finding a deterministic abstraction of S with k states, driven by heuristics, and 2) solving an ARSP. While this reduces the space of solutions, in practice, it dramatically speeds up the computation of a solution and the scalability of the approach. A solution to Problem Relaxation 2 constitutes a solution to Problem Relaxation 1 as well.

8.5.1. Proposed Algorithm - Overview

In this section, we provide a brief overview of the algorithmic scheme represented in Figure 8.3, designed to solve Relaxed Problem 2. In Section 8.5.3 we provide an efficient implementation of the scheme and describe it in detail.

The starting point is a dataset $D \doteq \{x_j\}_{i=j}^{N}$, parsed to the learner.

Learner

The learner comprises two stages, a clustering stage and an SMT stage.

Clustering Stage. The clustering stage uses the dataset to generate k anchor points $\hat{x}_1, \dots \hat{x}_k$ and corresponding partition of $\mathcal{X}$ (clusters) $C_1, \dots, C_k$. Together we obtain the mesh $\mathcal{M} \doteq \{(\hat{x}_1, C_1), \dots (\hat{x}_k, C_k)\}$. Intuitively, each anchor point and corresponding cluster are chosen to group neighbouring data points in D sharing similar dynamics. This can be done using a fast clustering algorithm of choice.
Input: dataset of transitions $\{(x, f(x)) : x \in D\}$, max anchor points k, resolution relation $\hat{R}$.
Output: mesh $\mathcal{M} \doteq \{(\hat{x}_1, C_1), \dots (\hat{x}_k, C_k)\}$, abstraction $\hat{S}$.

Relation Learner Stage. The mesh $\mathcal{M}$ and $\hat{x}$ uniquely define the candidate deterministic abstraction $\hat{S}$; we are left to find a suitable assignment for the $\hat{\epsilon}_i$'s. Let Q be the equivalence relation $Q \doteq \{(x, \hat{x}_i) : x \in C_i\}$ induced by the mesh. We exploit Q to conveniently embed the coverage condition Φ_{cov} by imposing $Q \subseteq R$. One possibility to achieve this is to set $\hat{\epsilon}_i \geq \max_{x \in C_i} d(x, \hat{x}_i) \doteq \gamma_i$. Then, the query becomes:

$$\exists \hat{\epsilon}_1 \geq \gamma_1 \dots \hat{\epsilon}_k \geq \gamma_k \in \mathbb{R}_{\geq 0} \text{ s.t. } \bigwedge_{x_j \in D} \Phi_{\text{con}}(x_j) \wedge \Phi_{\text{res}}(x_j) \tag{8.17}$$

with:

$$\Phi_{\text{con}}(x_j) \doteq \bigwedge_{i=1}^{k} \left[d(x_j, \hat{x}_i) \leq \hat{\epsilon}_i \implies d(f(x_j), \hat{x}_{g(i)}) \leq \hat{\epsilon}_{g(i)} \right]$$

$$\Phi_{\text{res}}(x_j) \doteq \bigwedge_{i=1}^{k} \left[\hat{\epsilon}_i \leq \epsilon(x_j, \hat{x}_i) \right]$$

Note that, in contrast with the Relaxed Problem 1, the constraints are much simpler, since the only variables are the $\hat{\epsilon}_i$'s. Under Assumption 6 the scalars $\hat{\epsilon}_1, \dots, \hat{\epsilon}_k$ uniquely determine the relation.
Input: mesh $\mathcal{M} \doteq \{(\hat{x}_1, C_1), \dots (\hat{x}_k, C_k)\}$, abstraction $\hat{S}$, dataset of transitions $\{(x, f(x)) : x \in D\}$.
Output: candidate relation R.

Verifier

The verifier consists of a single SMT query. Given a candidate design of the learner, it checks

$$\exists x \in \mathcal{X} \ . \ \neg(\Phi_{\text{con}}(x) \wedge \Phi_{\text{res}}(x)). \tag{8.18}$$

Recall that coverage is ensured by a successful assignment of the SMT query in the learner. Additionally, if the resolution function $\epsilon(x, \hat{x})$ depends only on $\hat{x}$ (e.g. $\epsilon(x, \hat{x}) = a|\hat{x}| + b$, the verification can be simplified further: in this case the check becomes

$$\exists x \in \mathcal{X} \ . \ \neg\Phi_{\text{con}}(x). \tag{8.19}$$

Input: candidate abstraction $\hat{S}$, candidate relation R.
Output: counterexample x_{ctx}.

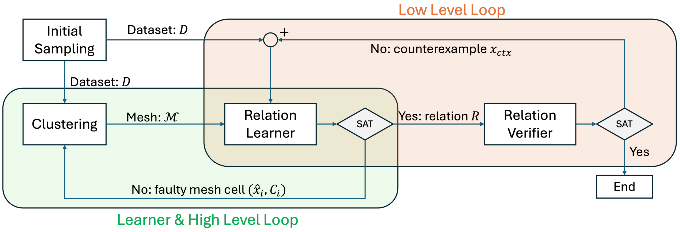


Figure 8.3.: Block diagram of the CEGIS scheme providing a solution to Problem Relaxation 2.

8.5.2. Counterexample Feedback

If the verifier does not provide a counterexample, the CEGIS loop terminates, having successfully synthesized an abstraction $\hat{S}$ and relation R satisfying Definition 26. If the verifier provides a counterexample x_{ctx}, that is, a point satisfying equation (8.18), it is introduced in the dataset D; the new dataset $D' = D \cup \{x_{ctx}\}$ is then fed back to the Relation Learner for a new learning iteration, until the verifier runs out of counterexamples. This cycle is highlighted in orange in Figure 8.3, and we refer to it as *Low Level Loop*. If the Relation Learner fails at synthesising a candidate relation, the algorithm identifies the set of cells of the current mesh $\mathcal{M}$ and refines it, resulting in a candidate abstraction $\hat{S}$ with an increased number of states, facilitating a successful synthesis as we motivate in the following sections. This cycle is highlighted in green in Figure 8.3, and we refer to it as *High Level Loop*. Typically, the High Level Loop iterates a smaller number of times when compared to the Low Level Loop.

8.5.3. Proposed Algorithm - In Detail

Referring to Figure 8.3, we detail now each block in the diagram and its interconnections. While both, the Clustering Stage and the Relation Learner Stage are grouped in the *learner* block, the latter is interlocked directly with the Relation Verifier: these two stages constitute the core of the CEGIS scheme (Low Level Loop). In a nutshell, the Clustering Stage outputs a candidate abstraction $\hat{S}$; next, the Relation Learner designs the candidate relation by addressing the counterexamples of the Relation Verifier, until a multi-resolution ϵ-approximate bisimulation between S and $\hat{S}$ is found. The candidate abstraction $\hat{S}$ is updated only when the Relation Learner is unable to find a solution (High Level Loop).

From now on, for simplicity, let the resolution function ϵ be solely a function of the abstract states, that is $\overline{R} \doteq \{(x, \hat{x}) : d(x, \hat{x}) \leq \epsilon(\hat{x})\}$.

Clustering Stage. We implement the clustering stage using K-Means [135]. We first augment the dataset D to $D^+ \doteq \{(x_j, f(x_j))\}_{j=1}^N$. Optionally, we can define a weighting function $w : x \longrightarrow \mathbb{R}$ for the first component of every pair in D^+, assigning higher weights to data points in locations where higher resolution is requested as specified by the resolution function, resulting in the following clustering cost function to be minimised

$$J(\mu_1, \ldots, \mu_k) = \sum_{j=1}^{N} w(x_j) ||z_j - \mu_{c_j}||,$$

where $z_j = [x_j^T, f(x_j)^T]^T$ and $c_j \in [1, ..., k]$ is the centroid assigned to z_j. A simple choice for the weighting function is the inverse of the resolution function, i.e. $w = \epsilon^{-1}$; in Figure 8.4 we show the effect of such a choice for $\epsilon(x) = 0.3|x| + 0.3$ on th domain $[-1, 1]^2$.

We define the abstraction's state set using the first n components of each centroid, i.e. $\hat{x}_i \doteq \mu_i[1 : n]$ for $i \in [1, \ldots, k]$, and the mesh $\mathcal{M}$ as the Voronoi diagram obtained from the abstraction's states

$$\mathcal{M} \doteq \text{Voronoi}(\hat{x}_1, \ldots, \hat{x}_k) = \{(\hat{x}_1, C_1), \ldots (\hat{x}_k, C_k)\} \tag{8.20}$$

where each Voronoi cell C_i is a convex polytope. As the mesh $\mathcal{M}$ implicitly defines a quantizer κ, according to Definition 23, the Clustering Stage effectively returns the deterministic abstraction $\hat{S} \doteq (\hat{\mathcal{X}}, \hat{\mathcal{X}}, \hat{f})$, where $\hat{\mathcal{X}} \doteq \{\hat{x}_1, \ldots, \hat{x}_k\}$.

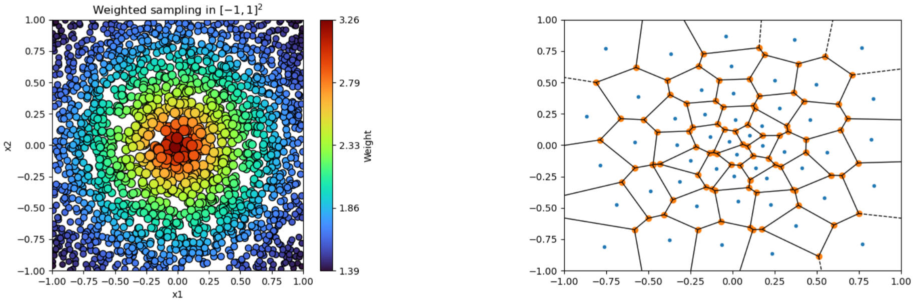


Figure 8.4.: (Left) Effect of the weighting function on the data points: more centroids are drawn near the origin. (Right) Example of a Voronoi diagram resulting from a clustering with 50 centroids.

Remark 18. The role of the Clustering Stage is to provide a good first guess for the mesh and the placement of the abstract states. The presence of the High Level Loop ensures that if the Relation Learner can not find a candidate relation, the mesh is suitably refined where needed. This is critical, since, going from Relaxed Problem 1 to Relaxed Problem 2, we have separated the selection of abstract states from the selection of the relation. The High Level Loop introduces a means of communication between the two stages; we show that this guarantees the convergence to a solution for a class of systems, see Theorem 6.

Given the candidate abstraction $\hat{S}$, it remains to compute the relation $R \subseteq \mathcal{X} \times \hat{\mathcal{X}}$ pairing concrete states x and abstract states $\hat{x}$: this is offloaded to the Low Level Loop, as described in Section 8.5.5.

8.5.4. Relation Templates

From here on, we relax Assumption 6, and consider more general relation templates.

Assumption 7. *Let $L_{\theta_i}(\hat{x}_i)$ denote a compact semialgebraic set parametrized by the parameter vector $\theta_i \in \mathbb{R}^{n_i}$ and associated with $\hat{x}_i$, for $i = 1, \dots, k$. We parametrise the space of candidate relations as*

$$R_{\theta_1,\dots,\theta_k,\hat{\epsilon}_1,\dots,\hat{\epsilon}_k} \doteq R_{\boldsymbol{\theta},\hat{\boldsymbol{\epsilon}}} \doteq \{(x, \hat{x}_i) \in \mathcal{X} \times \hat{\mathcal{X}} : x \in L_{\theta_i}(\hat{x}_i) \wedge d(x, \hat{x}_i) \leq \hat{\epsilon}_i\}, \tag{8.21}$$

where $\boldsymbol{\theta} \doteq \theta_1, \dots, \theta_k$ and $\hat{\boldsymbol{\epsilon}} \doteq \hat{\epsilon}_1, \dots, \hat{\epsilon}_k$.

In the next section, we discuss different choices for the templates.

Purely Metric Relation Template

In the simplest situation, the sets $L_{\theta_i}(\hat{x}_i)$ can themselves be defined by the distance; for instance if $d(x, \hat{x}_i) = ||x - \hat{x}_i||_2$ and $L_{\theta_i}(\hat{x}_i) = ||x - \hat{x}_i||_2^2 - \theta_i^2 \leq 0$ (with $\theta_i \in \mathbb{R}$), then the space of relations reduces to

$$R_{\boldsymbol{\theta},\hat{\boldsymbol{\epsilon}}} \doteq \{(x, \hat{x}_i) \in \mathcal{X} \times \hat{\mathcal{X}} : ||x - \hat{x}_i||_2 \leq \min(\hat{\epsilon}_i, \theta_i)\}, \tag{8.22}$$

thus we recover the template in Assumption 6. Recall that the three conditions of interest are *coverage* (equation (8.14)), *transition consistency* (equation (8.15)), and *minimum resolution* (equation (8.16)). To reduce the search space for the SMT stage and reduce the number of iterations between *learner* and *verifier* we exploit the partition of $\mathcal{X}$ provided by the mesh $\mathcal{M}$ and we embed the coverage condition directly as linear constraints on the scalars $\hat{\epsilon}$, in the form of lower bounds, as mentioned in Section 8.5.1. To ensure that $R_{\boldsymbol{\theta},\hat{\boldsymbol{\epsilon}}} \supseteq Q$, where $Q \doteq \{(x, \hat{x}_i) : x \in C_i\}$ is the relation induced by the mesh, one can simply compute the smallest 2-norm ball containing each mesh cell and use such value to compute lower bounds for the ϵ_i's (and θ_i's).

Example 10. Consider a subset of states $\hat{x}_1$, $\hat{x}_2$, and $\hat{x}_3$ of the abstraction $\hat{S}$ depicted in Figure 8.5, where the mesh cells are given by the polygons in solid line, and the abstract transition function $\hat{f}$ is given by the arrows. Assuming R is parametrised as in equation (8.22), to ensure that $R \supseteq Q$, we extract a lower bound on the values of the ϵ_i's by computing the radius of the smallest ball containing the mesh cell. In Figure 8.5 we show graphically the *transition consistency* condition: once the radii $\hat{\epsilon}_1$, $\hat{\epsilon}_2$ of the predecessors of $\hat{x}_3$ are chosen, the radius $\hat{\epsilon}_3$ must be large enough so that for all $x \in R(\hat{x}_1) \cup R(\hat{x}_2)$ it holds that $f(x) \in R(\hat{x}_3)$. In this particular case equation (8.15) requires that if $||x - \hat{x}_1|| \leq \hat{\epsilon}_1$ or $||x - \hat{x}_2|| \leq \hat{\epsilon}_2$ then $||f(x) - \hat{x}_3|| \leq \hat{\epsilon}_3$. Note that the dotted circle indicates the maximum allowable radius enforced by the minimum resolution condition; clearly it conflicts with the transition consistency condition.

Example 10 shows that ensuring the coverage condition when using p-norm balls can cause a severe overapproximation of the mesh cells, therefore unnecessarily inflating the radii of the successor abstract states. Every overapproximation has a cascading effect; the radius selected for $\hat{x}_3$ will affect its successor, and so on.

To mitigate this issue, we follow a different approach, aiming at minimising the overapproximation of the mesh cells while ensuring coverage.

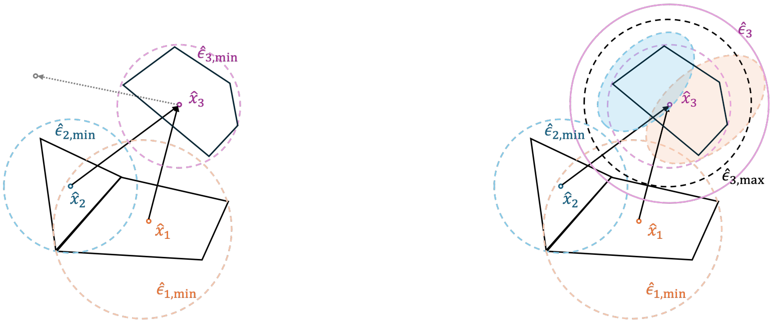

$\hat{\epsilon}_{3,\min}$
$\hat{x}_3$
$\hat{\epsilon}_{2,\min}$
$\hat{x}_2$
$\hat{x}_1$
$\hat{\epsilon}_{1,\min}$
$\hat{\epsilon}_3$
$\hat{x}_3$
$\hat{\epsilon}_{2,\min}$
$\hat{\epsilon}_{3,\max}$
$\hat{x}_2$
$\hat{x}_1$
$\hat{\epsilon}_{1,\min}$

Figure 8.5.: Left: sub-graph of an abstraction with abstract states $\hat{x}_1$, $\hat{x}_2$, and $\hat{x}_3$. Minimum radius per abstract state $\hat{\epsilon}_{1,\min}$, $\hat{\epsilon}_{2,\min}$, and $\hat{\epsilon}_{3,\min}$ respectively to guarantee $R \supseteq Q$. Right: *transition consistency* condition: set $\hat{\epsilon}_1 = \hat{\epsilon}_{1,\min}$ and $\hat{\epsilon}_2 = \hat{\epsilon}_{2,\min}$; $\hat{\epsilon}_3$ must be large enough to contain the image of $R(\hat{x}_1)$ and $R(\hat{x}_2)$, shown as the orange and blue shaded areas respectively.

Polytopic Relation Template

The clustering stage outputs a mesh obtained from the Voronoi diagram induced by the abstract states. Every Voronoi cell can be expressed in its half-space representation, that is

$$C_i = \{x \in \mathcal{X} : A_i x \leq b_i, A_i \in \mathbb{R}^{p \times n}, b_i \in \mathbb{R}^p\}.$$

The rows of the A_i matrix identify the set of normal vectors defining the cell. We can scale each direction independently using a vector of scaling factors so that the scaled cell contains the images of the projection of R at its predecessors. The scaling of a cell C_i around a point $\hat{x}_i$ by a factor $\theta_i > 0$ is defined as

$$L_{\theta_i}(\hat{x}_i) \doteq \{x \in \mathcal{X} : A_i(x - \hat{x}_i) \leq \theta_i(b_i - A_i\hat{x}_i)\}. \tag{8.23}$$

Note that for θ_i equal to 1, $L_{\theta_i}(\hat{x}_i) = C_i$.

Example 11. Consider again the system of Example 10. Let $R_{\boldsymbol{\theta},\hat{\boldsymbol{\epsilon}}}$ be defined as in equation 8.21 but, in contrast with Section 8.5.4, let $L_{\theta_i}(\hat{x}_i)$ be defined as in equation 8.23, and let $\hat{\epsilon}_i = \max_{x \in L_{\theta_i}(\hat{x}_i)} d(x, \hat{x}_i)$. To ensure that the coverage condition is satisfied for $\hat{x}_1$ and $\hat{x}_2$, it is sufficient to select θ_1 and θ_2 equal to 1. The sets $W_1 = \{x : \exists x' \in C_1(x = f(x'))\}$ and $W_2\{x : \exists x' \in C_2(x = f(x'))\}$, the polygons in orange and blue solid line respectively in Figure 8.6, represent the set of all successors of $R_{\boldsymbol{\theta},\hat{\boldsymbol{\epsilon}}}(\hat{x}_1)$ and $R_{\boldsymbol{\theta},\hat{\boldsymbol{\epsilon}}}(\hat{x}_2)$ respectively. To ensure that the coverage condition holds and that the transition consistency condition holds for $\hat{x}_3$, it must hold that $\theta_3 \geq \max(\max_{x \in W_1 \cup W_2} \frac{A_3(x - \hat{x}_3)}{b_3 - A_3\hat{x}_3}, 1)$. This allows us to generate a much tighter set $R_{\boldsymbol{\theta},\hat{\boldsymbol{\epsilon}}}(\hat{x}_3)$, given by the polygon containing $\hat{x}_3$, compared to what we obtained in Example 10, while ensuring coverage and a measure of resolution, given by $\hat{\epsilon}_3$. By using the additional sets $L_{\theta_i}(\hat{x}_i)$ we decrease the coupling between transition consistency condition, and the minimum resolution condition: the predecessors of $\hat{x}_3$ yield the same resolution as in Example 10, but they allow for a higher resolution in their successor in this case.

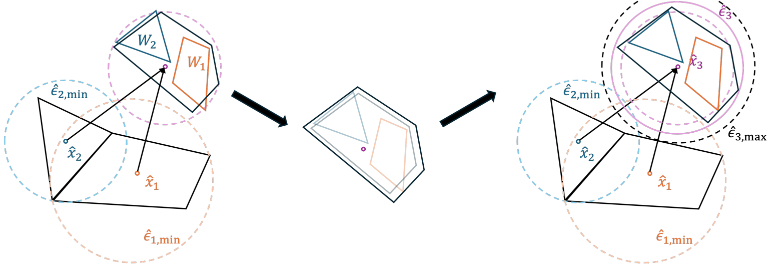

W_2
W_1
$\hat{\epsilon}_{2,\min}$
$\hat{x}_2$
$\hat{x}_1$
$\hat{\epsilon}_{1,\min}$
$\hat{\epsilon}_{2,\min}$
$\hat{\epsilon}_3$
$\hat{x}_3$
$\hat{\epsilon}_{3,\max}$
$\hat{x}_2$
$\hat{x}_1$
$\hat{\epsilon}_{1,\min}$

Figure 8.6.: Scaled Voronoi cells. The vectors n_i are normal to the edges of the cell, and their magnitude is the distance from the center.

8.5.5. Low Level Loop - One Shot

The Low Level Loop encodes and solves by CEGIS the following problem

$$\exists \theta_1, \dots, \theta_k \geq 1, \forall x \in \mathcal{X} \ . \ \Phi_{\text{con}}(x) \wedge \Phi_{\text{res}}(x). \tag{8.24}$$

It receives as input the initial dataset D, the candidate abstraction $\hat{S}$, the mesh $\mathcal{M}$, and a relation template $R_{\boldsymbol{\theta},\hat{\boldsymbol{\epsilon}}}$. Each $R_{\boldsymbol{\theta},\hat{\boldsymbol{\epsilon}}}(\hat{x}_i)$ describes a scaled mesh cell as described in Section 8.5.4. The pseudocode for the Low Level Loop is provided by Algorithm 1;

Recall that every θ_i influences the scaling of the mesh cell C_i containing $\hat{x}_i$. By lower bounding each θ_i by 1, we ensure that the coverage condition always holds. Additionally, let us fix $\hat{\epsilon}_i = \max_{x \in L_{\theta_i}(\hat{x}_i)} d(x, \hat{x}_i)$; we can rewrite the radius as

$$\hat{\epsilon}_i = \theta_i \max_{x \in L_1(\hat{x}_i)} d(x, \hat{x}_i) = \theta_i \gamma_i \tag{8.25}$$

We have reduced the number of parameters to be decided in the Low Level Loop to 1 per cluster. This is analogous to what was shown in Relaxed Problem 2: instead of directly choosing the radii $\hat{\epsilon}_i$, we now equivalently choose the appropriate scaling factors θ_i.

Relation Learner stage. The Relation Learner must provide a solution to:

$$\exists \theta_1, \dots, \theta_k \geq 1 \ . \bigwedge_{x_j \in D} \Phi_{\text{con}}(x_j) \wedge \Phi_{\text{res}}(x_j), \tag{8.26}$$

where

$$\Phi_{\text{con}}(x_j) \doteq \bigwedge_{i=1}^{k} \left[x_j \in R_{\boldsymbol{\theta},\hat{\boldsymbol{\epsilon}}}(\hat{x}_i) \implies f(x_j) \in R_{\boldsymbol{\theta},\hat{\boldsymbol{\epsilon}}}(\hat{x}_{g(i)}) \right]$$

$$\Phi_{\text{res}}(x_j) \doteq \bigwedge_{i=1}^{k} \left[\theta_i \leq \frac{\epsilon(\hat{x}_i)}{\gamma_i} \right].$$

Algorithm 1 Low Level Loop - One Shot

```
procedure LowLevelLoopOneShot($D, f, \mathcal{M}, R_{\boldsymbol{\theta},\hat{\boldsymbol{\epsilon}}}$)
    Inputs: $D$: set of one-step transitions; $f$: transition function; $\mathcal{M}$: mesh; $R_{\boldsymbol{\theta},\hat{\boldsymbol{\epsilon}}}$: relation template.
    Outputs: $\boldsymbol{\theta}$: parameters for $R_{\boldsymbol{\theta},\hat{\boldsymbol{\epsilon}}}$ to satisfy Definition 26 | None: call for abstraction refinement.
    $\hat{S} \leftarrow$ BuildAbstraction($\mathcal{M}, f$)
    while True do
        $SATL, \boldsymbol{\theta} \leftarrow$ RelationLearner($D, \hat{S}, R_{\boldsymbol{\theta},\hat{\boldsymbol{\epsilon}}}$)
        if $SATL$ then
            $SATV, x_{ctx} \leftarrow$ Verifier($f, \hat{S}, R_{\boldsymbol{\theta},\hat{\boldsymbol{\epsilon}}}, \boldsymbol{\theta}$)
        else
            return False, None
        end if
        if $SATV$ then
            $D \leftarrow D \cup \{(x_{ctx}, f(x_{ctx}))\}$
        else
            return True, $\boldsymbol{\theta}$
        end if
    end while
end procedure
```

Equivalently, for the chosen template, we have

$$\begin{aligned}\Phi_{\text{con}}(x_j) \doteq \bigwedge_{i=1}^{k} [A_i(x_j - \hat{x}_i) \leq \theta_i(b_i - A_i\hat{x}_i) \implies \\ A_{g(i)}(f(x_j) - \hat{x}_{g(i)}) \leq \theta_{g(i)}(b_{g(i)} - A_{g(i)}\hat{x}_{g(i)})] \\ \Phi_{\text{res}}(x_j) \doteq \bigwedge_{i=1}^{k} \left[\theta_i \leq \frac{\epsilon(\hat{x}_i)}{\gamma_i}\right].\end{aligned}$$

Observe that, independently of the transition function f, the predicates in the constraints listed above are linear in $\theta_1, \dots, \theta_k$. The same applies to any template L_{θ_i} that is linear in θ_i. As such, we have encoded the Relation Learner stage in a single existential sentence in the theory of Linear Real Arithmetic (∃-LRA).

This can be efficiently solved with SMT solvers like Z3, see [136].

Verifier stage. The verifier follows verbatim Section 8.5.1. It checks

$$\exists x \in \mathcal{X} \text{ s.t. } \neg\Phi_{\text{con}}(x). \tag{8.27}$$

where

$$\begin{aligned}\Phi_{\text{con}}(x) \doteq \bigwedge_{i=1}^{k} [A_i(x - \hat{x}_i) \leq \theta_i(b_i - A_i\hat{x}_i) \implies \\ A_{g(i)}(f(x) - \hat{x}_{g(i)}) \leq \theta_{g(i)}(b_{g(i)} - A_{g(i)}\hat{x}_{g(i)})].\end{aligned}$$

The antecedents in the constraint above are always linear in the variable x; the consequent is linear only when f is linear in x. If f is polynomial, the problem is in the existential fragment of nonlinear real arithmetic (∃-NRA). While the fragment is decidable, Z3's implementation is incomplete, meaning that it is not guaranteed that the search for a counterexample will terminate. For this fragment, Z3 relies on heuristics, and the solution time depends on the number of constraints, number of variables, and degree of polynomials; usually it can handle quickly small (usually less than 6-8 variables), low-degree (usually less than 4) polynomials. As the number of constraints rapidly grows with the number of abstract states and the dimensionality of the system, in the next Section, we propose an alternative algorithm to alleviate the slowdown due to the number of constraints.

8.5.6. Low Level Loop - Parallelised

We prove now that we can compute an assignment of $\boldsymbol{\theta}$ solving formula 8.24 by solving a sequence of ordered minimisation problems.

Definition 29. *Denote by $\mathcal{A}_i$ and $\mathcal{D}_i$ the set of abstract states eventually reaching $\hat{x}_i$ (ancestors) and the set of abstract states reached by $\hat{x}_i$ (descendants) respectively, formally,*

$$\mathcal{A}_i \doteq \{\hat{x} \in \hat{\mathcal{X}} : \exists q \in \mathbb{N} \,.\, \hat{f}^q(\hat{x}) = \hat{x}_i\}, \tag{8.28}$$

$$\mathcal{D}_i \doteq \{\hat{x} \in \hat{\mathcal{X}} : \exists q \in \mathbb{N} \,.\, \hat{f}^q(\hat{x}_i) = \hat{x}\}. \tag{8.29}$$

The graph of the abstraction is a Directed Acyclic Graph (DAG) if for all $\hat{x}_i$'s $\mathcal{A}_i \cap \mathcal{D}_i = \emptyset$.

With a slight abuse of notation, let $R_{\theta_i}(\hat{x}_i) = R_{\boldsymbol{\theta},\hat{\boldsymbol{\varepsilon}}}(\hat{x}_i)$ to emphasise the fact that the projection of $R_{\boldsymbol{\theta},\hat{\boldsymbol{\varepsilon}}}(\hat{x}_i)$ at $\hat{x}_i$ on $\mathcal{X}$ depends solely on the parameter θ_i, and let $g^{-1} : [1, \dots, k] \to 2^{[1,\dots,k]}$ be the inverse of g, mapping each index i to the indexes of the predecessors of the abstract state $\hat{x}_i$.

Proposition 10. *Consider the following subproblem:*

$$\theta_i^* = \min_{\theta_i \geq 1} \quad \theta_i \tag{8.30}$$

$$s.t. \quad \forall x \in \mathcal{X}, j \in g^{-1}(i) \,.\, x \in R_{\theta_j^*}(\hat{x}_j) \implies f(x) \in R_{\theta_i}(\hat{x}_i) \tag{8.31}$$

$$\theta_i \leq \frac{\epsilon(\hat{x}_i)}{\gamma_i}$$

If the graph of the abstractions forms a DAG, Problem 8.24 admits a solution if and only if $\boldsymbol{\theta}^* = \theta_1^*, \dots, \theta_k^*$ is a solution to 8.30 for all $i = 1, \dots, k$.

Proposition 10 allows us to exploit the structure of the abstraction, in particular when it forms a DAG, to recast 8.24 into an ordered sequence of one-dimensional minimisation problems. We postpone to Section 8.5.6 a discussion on how Proposition 10 can be generalised to cases where the abstraction is not a DAG. Furthermore, it is easy to infer whether a solution θ_i^* does or does not depend on a different solution θ_j^*.

Corollary 3. *Let $\mathcal{A}_i$ be the set of ancestors of $\hat{x}_i$ and let θ_i^* be the solution of equation (8.30). If $\hat{x}_j$ is not an ancestor of $\hat{x}_i$ then θ_i^* does not depend on θ_j^*.*

Remark 19. Corollary 3 relies on the imposition of the coverage condition by design, $\theta_i \geq 1$. Corollary 3 highlights a computational advantage in that it allows for parallelising the solution of each θ_i^*, according to its dependencies.

The feasibility set for the first constraint equation (8.31) is given by a logical formula with quantifier alternation ($\exists \theta_i \forall x$). Similarly to Section 8.5.5, this can be solved by a CEGIS where the Relation Learner is given by a MILP solver solving:

$$\begin{aligned} \theta_i^* = \min_{\theta_i \geq 1} \quad & \theta_i \\ \text{s.t.} \quad & \forall x_j \in D, \bigwedge_{j \in g^{-1}(i)} \Big(x_j \in R_{\theta_j^*}(\hat{x}_j) \implies f(x_j) \in R_{\theta_i}(\hat{x}_i) \Big), \\ & \theta_i \leq \frac{\epsilon(\hat{x}_i)}{\gamma_i}. \end{aligned} \tag{8.32}$$

The verifier checks the candidate solution for the current subproblem and the algorithm proceeds to the next subproblem when no counterexamples are found. The pseudocode is given in Algorithm 2. Finally, the following corollary to Proposition 10 provides a comparison between a solution obtained by Algorithm 1 and a solution obtained by Algorithm 2.

Algorithm 2 Low Level Loop - Parallelised

```
procedure LLLParallelised($D^+, f, \mathcal{M}, R_{\boldsymbol{\theta},\hat{\boldsymbol{\epsilon}}}$)
    Input: $D^+$: transitions; $f$: flow; $\mathcal{M}$: mesh; $R_{\boldsymbol{\theta},\hat{\boldsymbol{\epsilon}}}$: relation template.
    Output: $\boldsymbol{\theta}$: parameters; None: refinement needed.
    $\hat{S} \leftarrow$ BuildAbstraction($\mathcal{M}, f$)
    $G \leftarrow$ BuildAbstractionGraph($\hat{S}$)
    $ordered\_states \leftarrow$ TopologicalSort($G$)
    $\boldsymbol{\theta} \leftarrow$ empty map
    for each $\hat{x}_i$ in $ordered\_states$ do
        dependencies $\leftarrow$ GetPredecessors($G, \hat{x}_i$)
        $\boldsymbol{\theta}_{deps} \leftarrow$ LookupParams($\boldsymbol{\theta}$, dependencies)
        $\theta_i$, success $\leftarrow$ CEGISLoop(($D^+, f, R_{\boldsymbol{\theta},\hat{\boldsymbol{\epsilon}}}, \boldsymbol{\theta}_{deps}$))
        if not success then
            return None
        end if
        $\boldsymbol{\theta}[\hat{x}_i] \leftarrow \theta_i$
    end for
    return $\boldsymbol{\theta}$
end procedure
```

Corollary 4. *Let $\theta_1, \ldots, \theta_k$ be any solution satisfying Problem 8.24 and let $\hat{\epsilon}_1, \ldots, \hat{\epsilon}_k$ be the radii defined according to equation (8.25); let $\theta_1^*, \ldots, \theta_k^*$ be the solution obtained* *by equation* (8.30) *and* $\hat{\epsilon}^*_1, \dots, \hat{\epsilon}^*_k$ *be the corresponding radii. For all* $i \leq k$ *it holds that* $\hat{\epsilon}^*_i \leq \hat{\epsilon}_i$.

Corollary 4 states that, for the given candidate abstractions $\hat{S}$ and relation template $R_{\boldsymbol{\theta}}$, solving Problem 8.24 using Proposition 10 yields the relation with the highest resolution among the ones in the feasible set.

General Directed Graphs

Consider now an abstraction whose graph G does not form a DAG. A set of nodes is called *strongly connected* if every node can be reached from every other node. For the case of deterministic abstractions, a set of strongly connected nodes forms a cycle. In this case, we can compute the condensation of G. The condensation of G returns a DAG by construction, where the nodes are partitioned and grouped (or condensed) according to the strongly connected components into supernodes. Let Y denote the indexes of a set of m abstract states forming a cycle, and thus grouped into a supernode; then instead of having a one-dimensional minimisation problem, we now have a multidimensional problem as

$$\begin{aligned} \theta^*_{y_1}, \dots, \theta^*_m = \arg \min_{\theta^*_y \geq 1, y \in Y} \quad & \sum_{i \in Y} \theta_i \\ \text{s.t.} \quad & \forall x_j \in D, \bigwedge_{i \in Y} \bigwedge_{j \in g^{-1}(i)} \left(x_j \in R_{\theta^*_j}(\hat{x}_j) \implies f(x_j) \in R_{\theta_i}(\hat{x}_i) \right) \\ & \bigwedge_{i \in Y} \theta_i \leq \frac{\epsilon(\hat{x}_i)}{\gamma_i} \end{aligned} \tag{8.33}$$

The condensation of G can be used to sort the sequence of minimisation problems, analogously to what was shown in Proposition 10. Similarly, it is easy to show that Problem 8.24 admits a solution, if and only if it can be solved by cascading the solution subproblems described by equation (8.33).

Remark 20. In this section, we discussed the case where the template for each cell is parametrised by a single scalar value representing the scaling of the corresponding mesh cell. It is easy to extend the discussion to the case where the template $L_{\theta_i}(\hat{x}_i)$ describes a polytopic template and, accordingly, θ_i is a vector.

Mesh Refinement

Leveraging Proposition 10 to solve Problem 8.24 provides an immediate scheme to refine the abstraction whenever Problem 8.24 is infeasible. Whenever the i-th subproblem results infeasible, we select the set of abstract states given by $\hat{x}_i$ and all of its ancestors; then we split each corresponding mesh cell C_i in two, assign new centroids, and update the abstraction $\hat{S}$.

Remark 21. Given a data set D and the corresponding transitions, it is possible to quickly assess whether the current candidate abstraction $\hat{S}$ *can* satisfy the conditions described in Relaxed Problem 2: indeed, we can run the Relation Learner stage, without running the Relation Verifier. The returned parameters $\theta'_1, \dots, \theta'_k$ will only be valid on the given data set, but they provide a lower bound on the actual values of $\theta_1^*, \ldots, \theta_k^*$. If any of these lower bounds returns $\hat{\epsilon}_i = \theta_i' \gamma_i$ greater than the locally allowed resolution $\epsilon(\hat{x}_i)$, we can already conclude that the candidate abstraction can not satisfy the conditions of Relaxed Problem 2. Accordingly, we can either refine the current abstraction as described in Section 8.5.6, or increase the budget of abstract states k altogether. This step allows to reduce the number of iterations between the High Level Loop and the Low Level Loop, reducing the number of queries to the SMT verifier, and speeding up the construction of a solution.

8.6. Incrementally Uniformly Bounded Systems

In this section, we present a class of systems admitting the existence of a deterministic abstraction and a multi-resolution bisimulation relation for any resolution relation $\overline{R}$. We rely on the notion of Incrementally Uniformly Bounded systems (δ-UB).

Definition 30. *A transition system $S = (\mathcal{X}, \mathcal{X}_0, f)$ is δ-UB if*

$$\forall x_0, x_0' \in \mathcal{X}_0, k \in \mathbb{N} \quad d(x_k, x_k') \leq \alpha(d(x_0, x_0')), \tag{8.34}$$

where α is a class $\mathcal{K}$ function.

The condition above is an adaptation of [132, Definition 17] for autonomous systems. Note that δ-UB implies continuity of f.

Theorem 6. *Let $S = (\mathcal{X}, \mathcal{X}_0, f)$ be a δ-UB transition system with asymptotically stable equilibrium at x^*, where $\mathcal{X}$ is a compact subset of its region of attraction. Then for all resolution functions $\epsilon(x, \hat{x})$ there exists a finite state deterministic abstraction $\hat{S}$ and a multi-resolution approximate bisimulation relation between S and $\hat{S}$ satisfying $\overline{R}$, that is $\mathcal{W}(\overline{R}) \neq \emptyset$.*

The proof of Theorem 6 relies on the existence of an abstraction $\hat{S}$ whose graph has a single cycle at the equilibrium. Then, by providing a refinement scheme for the cells of the underlying mesh that monotonically shrinks the size of the cells, hindering the solution of any subproblem 8.30, we show that by exploiting δ-UB, any desired resolution can be achieved anywhere in the domain, see Appendix 8.9.1.

Corollary 5. *Let $S = (\mathcal{X}, \mathcal{X}_0, f)$ be a δ-UB transition system where $\mathcal{X}$ is a compact transient set. Then $\mathcal{W}(\overline{R}) \neq \emptyset$.*

Corollary 5 drops the requirement for the presence of an asymptotically stable equilibrium, requiring instead that the domain of abstraction $\mathcal{X}$ is a transient set: if the system is δ-UB it guarantees that for any specified resolution function, the elicited MRASP always admits a solution. The proof of this result is immediate from the proof of Theorem 6.

8.7. Numerical Examples

8.7.1. Linear System

$$x_{k+1} = 0.4 \cdot \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix} x_k \tag{8.35}$$

The dynamics of the system represents a rotation by 45° counter-clockwise and scaling by $0.4 * \sqrt{2}$ on the domain $\mathcal{X} = [-1, 1]^2$. We consider the resolution specification $\overline{R} \doteq \{(x, \hat{x}) : ||x - \hat{x}||_2 \leq 0.3||\hat{x}|| + 0.5\}$. And we select an initial number of abstract states of $k = 30$ and a data set of $N = 5000$ transitions. In a single iteration, our algorithm constructs an abstraction and a relation R establishing a multi-resolution approximate bisimulation with the concrete system. The time necessary for the synthesis is ~ 6 seconds. In Figure 8.7 we show the mesh obtained from the Clustering stage, and the resulting abstraction. Figure 8.8 (left) shows the obtained relation, where each polytope represents the projection of R at a corresponding abstract state. Figure 8.8 (right) shows how the resolution is obtained as a function of the norm of the abstract states. The green line shows the minimal resolution necessary to guarantee the coverage condition, that is, the resolution of the mesh. The red line shows the lower bound obtained using the pre-processing step described in Remark 21, and the orange line represents the resolution obtained after the Low Level Loop terminates, which is clearly below the prescribed resolution, represented by the blue line.

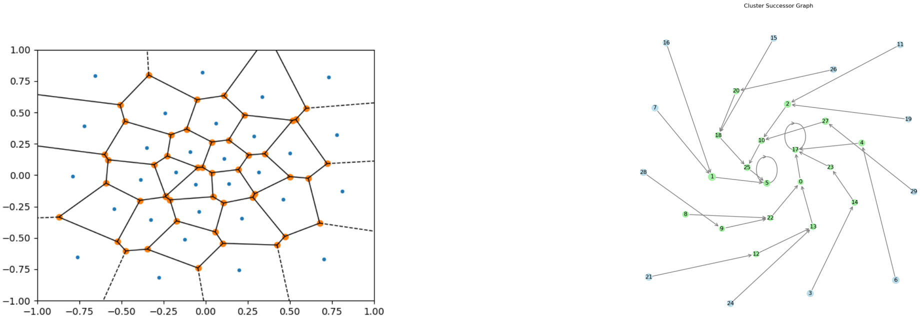


Figure 8.7.: Resulting mesh and abstract states (left), and graph of the abstraction (right).

We repeat the experiment using $k = 200$, and a tighter specification $\overline{R} \doteq \{(x, \hat{x}) : ||x - \hat{x}||_2 \leq 0.3||\hat{x}|| + 0.3\}$. The results are shown in Figure 8.9b. The synthesis took ≈ 18 seconds.

8.7.2. Compactness Comparison

We compare the size of the abstraction resulting from our method with the example in [6, Example 10.9], where the authors consider the system

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} -7 & 1 \\ 8 & -10 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \tag{8.36}$$

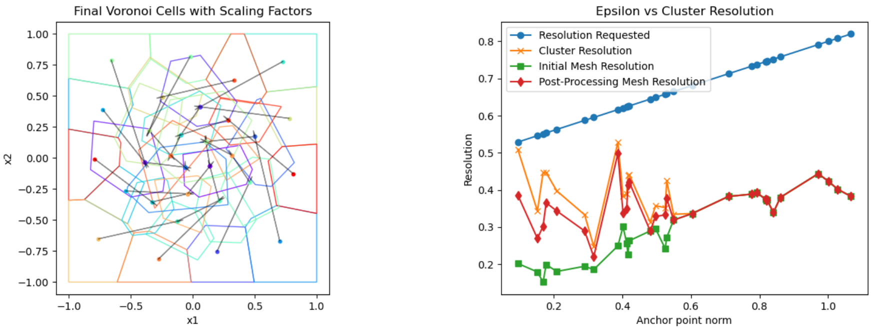
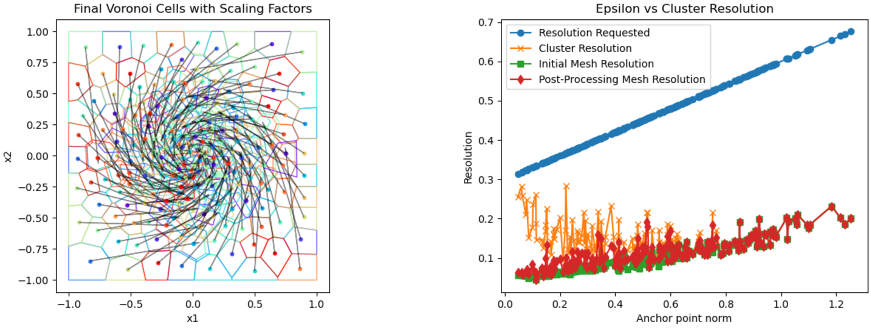


Figure 8.8.: Final relation verified (left), and achieved resolution as a function of the abstract state's norm (right).



(a) Final Verified relation

(b) Obtained resolution

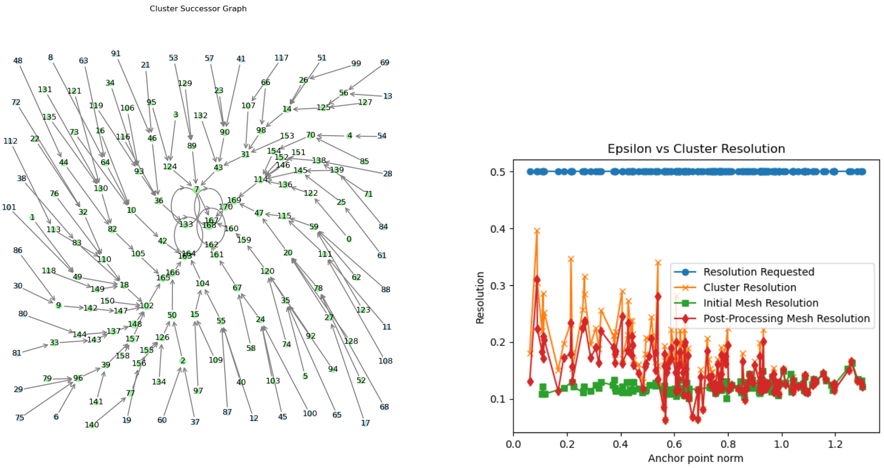
discretized in time with $\tau = 0.05$. We restrict our focus to the domain $\mathcal{X} = [-1, 1]^2$. After identifying a δ-GAS Lyapunov function, the authors select a (uniform) resolution of $\epsilon = 0.5$, resulting in $\hat{\mathcal{X}} = \{x \in \mathcal{X} : x[i] = k_i \frac{2}{\sqrt{n}}\eta, k_i \in \mathbb{Z}, i \in \mathbb{N}\}$ with $\eta = \frac{\sqrt{2}}{20}$, for a total of 400 abstract states.

In contrast, we select as initial guess $k = 150$ abstract states, which, after refinement, lead to 171 abstract states, and a successful synthesis in ≈ 17 seconds. The results are shown in Figures 8.10a and 8.10b. Using less than 50% of the abstract states, the obtained relation provides a worst-case resolution 20% higher than the desired value, whereas the rest of the states show an even greater improvement.

8.7.3. Non-δ-GAS System

$$\begin{bmatrix} x_{k+1} \\ y_{k+1} \end{bmatrix} = \begin{bmatrix} 0.5x_k \\ 0.5y_k + 0.5x_k^2 \end{bmatrix} \tag{8.37}$$

Domain $\mathcal{X} = [-0.5, 3] \times [-0.5, 5]$. The system above is asymptotically stable but it is not δ-GAS (this can be seen by considering two initial conditions



(a) Failure in Low Level Loop

(b) Minimal branch refinement

(x_0, y_0) and $(x_0', y_0') = (x_0 + \alpha, y_0)$ with $x_0 > 2)$. Resolution specification $\overline{R} \doteq \{(x, \hat{x}) : ||x - \hat{x}||_2 \leq 0.3||\hat{x}|| + 0.5\}$. Parameters: initial number of abstract states $k = 400$; number of samples $N = 10000$. Time: 118 seconds.

To illustrate the efficacy of the abstraction refinement protocol, we decrease the initial abstract states to $k = 300$. During the first run of the Low Level Loop fails while solving that abstract state labeled '296' and marked in color red in Figure 8.11, which in turn causes the failure of all its descendants, in this case '252'. This triggers an abstraction refinement step: all of the ancestors of node '296' marked in orange undergo a splitting, after which the abstraction's graph is reconstructed. NNote that only the set of nodes that were split and all of its descendants need to be resubmitted to the Low Level Loop, shown in Figure 8.11 (right). The remaining nodes where already solved in the first run. This greatly reduces the overall solution time. The results are shown in Figure 8.12. The first run took 69 seconds, while the second one only 21 seconds. The final abstraction has 358 abstract nodes.

8.7.4. Non-Differentiable System

$$\begin{bmatrix} x_{k+1} \\ y_{k+1} \end{bmatrix} = \begin{bmatrix} 0.8x_k \\ 3.2y_k^3 \end{bmatrix} \text{ if } |y_k| < 0.5, \text{ else } \begin{bmatrix} 0.8x_k \\ 0.8y_k \end{bmatrix} \tag{8.38}$$

Finally, we consider a continuous but nondifferentiable system. With resolution specification $\overline{R} \doteq \{(x, \hat{x}) : ||x - \hat{x}||_2 \leq 0.3||\hat{x}|| + 0.5\}$ and $k = 300$, in 56 seconds we obtain a valid abstraction and relation, shown in Figure 8.12. Note that this system can not be abstracted using the algorithm proposed in [10].

8.7.5. Comparison with Tazaki et al.

We compare the efficacy of our approach with the algorithm proposed by [50], using the system from the linear system in Section 8.7.1. We select different

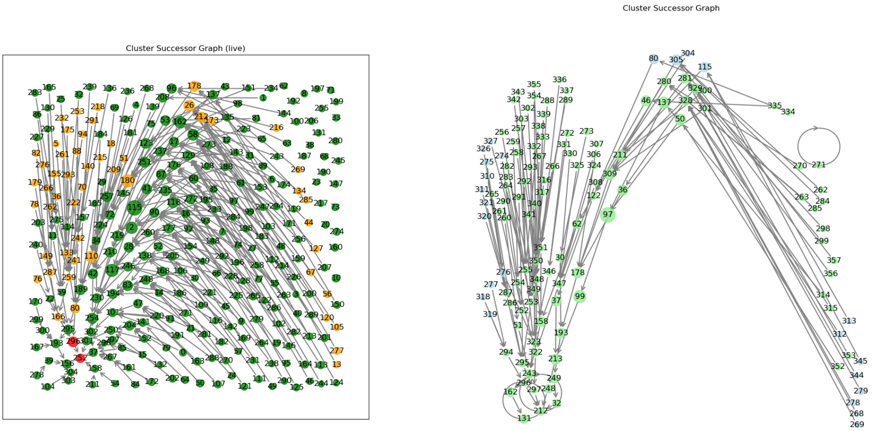


Figure 8.11.: (Left) Failed nodes in Low Level Loop are highlighted in red; their ancestors for which a solution was found are shown in orange. (Right) Refinement of the red and orange nodes.

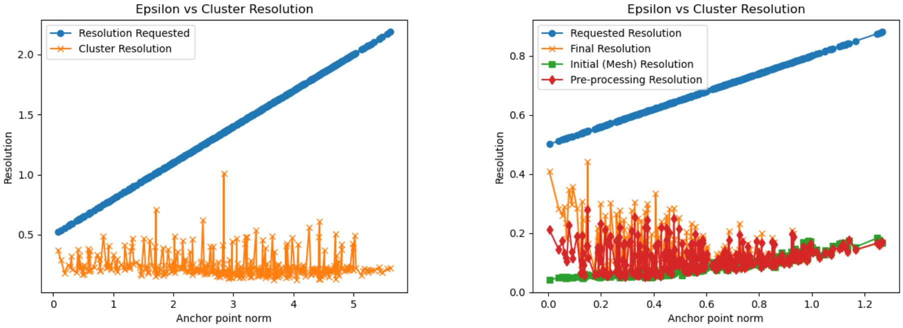


Figure 8.12.: Final resolution of the examples in Section 8.7.3 (left) and Section 8.7.4 (right)

resolution specification I)$\epsilon(x) = 0.3$,II) $\epsilon(x) = 0.2$, and III) $\epsilon(x) = 0.1$. The results are summarised in Table 8.1. We observe that the approach of [50] provides smaller abstractions compared to our approach. In turn, as the number of abstract states grows, our approach returns a solution in a shorter time.

8.8. Conclusions and Discussion

This chapter presented a novel framework for the synthesis of multi-resolution approximate bisimulation relations, designed to construct provably correct abstractions for a broad class of continuous systems. We introduced the first

Resolution $\epsilon(x)$	#States (Ours)	Time (s, Ours)	#States (Tazaki et al.)	Time (s, Tazaki et al.)
0.3	152	9.6	60	1.5
0.2	203	8.6	124	10.0
0.1	1431	93.4	491	148.2

Table 8.1.: Comparison of our approach with [50] for different resolution specifications.

existential proofs for multi-resolution approximate bisimulation relations. By showing that the class of δ-UB systems, which strictly contains δ-GAS systems, admits such relations (Theorem 6), we extend the class of systems for which approximate bisimulations can be found.

Importantly, the existential proof of Theorem 6 is constructive, providing the backbone for a sound and complete algorithm with guaranteed termination to compute an abstraction $\hat{S}$ and a multi-resolution approximate relation R between S and $\hat{S}$ for any specified resolution relation, qualitatively described in Algorithm 2, for the systems described in Corollary 5. For systems with an asymptotically stable equilibrium, it is necessary to compute a forward invariant set around the equilibrium: if the template chosen for the candidate relation can not capture such a set, Algorithm 2 is sound, but it may not terminate. While this may be alleviated by selecting a more flexible polytopic template for the cell containing the equilibrium (see Remark 20) to approximate polyhedral Lyapunov functions [137], or by allowing nondeterminism for the cell, we leave these adaptations for future work. Note that if nondeterminism were allowed for the cell containing the equilibrium, it would be easy to guarantee termination of Algorithm 2.

The size of the abstraction in terms of its number of states is, however, guided by heuristics in the Clustering stage and by the mesh refinement procedure described in Section 8.5.6 and detailed in the proof of Theorem 6. Nevertheless, we show in our numerical examples, that the synthesised abstractions can, in practice, be considerably smaller when compared with the established Lyapunov-based approach of Theorem 5. Additionally, our approach does not require explicit knowledge of a Lyapunov function, nor the explicit form of α in equation (8.34). In turn, our algorithm relies on sampled transitions to iteratively and sequentially select the parameters of a relation template. If the size of the abstraction is not of concern and a δ-GAS Lyapunov function for the concrete system is known, Theorem 5 allows for a computationally inexpensive abstraction synthesis, as it boils down to computing the successor state of every abstract state and mapping it to the nearest neighbour.

In contrast with [10, 50], our approach does not rely on a local linearization of the system's dynamics, making our algorithm applicable to a larger set of systems. As mentioned [10], computing such linearizations can be computationally expensive for abstractions with a large state set. In fact, differentiability is not strictly required by Theorem 6, as shown in Example 8.7.4. Moreover, Algorithm 2 exploits the abstraction's graph structure and allows for a parallelised solution. The algorithm presented in [10, 50] is not parallelizable, as the refinement procedure relies on the solution of an optimisation program involving the entire state set of the abstraction.

We compared our approach with [50] on a linear system, for which [50] does not require any linearization. We observe that for large abstractions, our algorithm can converge to a solution faster. However, the heuristics-based refinement technique presented in [50] performs well in practice, yielding smaller abstract state sets. Finally, we note that the Verifier stage is currently implemented by an SMT solver; accordingly, as the number of dimensions grows, the number of constraints representing the template grows as well. Indeed, we observe that in the Low Level Loop, most of the time is spent on verification, whereas learning is extremely fast.

Our current refinement scheme makes the overall computation time sensitive to the output of the Clustering stage, since High Level Loop iterations have proven to be time-consuming. Possible extensions of this direction may combine the advantage of dividing the solution of an ARSP by exploiting the graph structure of the abstraction with a more efficient abstraction refinement scheme, allowing for faster communication between High and Low Level Loop, therefore combining the advantages of the approach in [50] with ours. Another relevant extension may consider nondeterministic multi-resolution approximate abstractions, potentially enabling the abstraction of a broader set of systems, as systems with unstable equilibria and topological attractors.

8.9. Appendix

8.9.1. Proofs

Proposition 10

$\Longleftarrow:$ trivial.
$\Longrightarrow:$ Let $\tilde{\boldsymbol{\theta}} = \tilde{\theta}_1, \dots, \tilde{\theta}_k$ be a solution of 8.24. Let $\hat{x}_i$ be a state with no predecessors (at least one such state exists since the abstraction forms a DAG), $\tilde{\theta}_i$ be the associated parameter, $\hat{x}_i$ be its abstract successor state with associated parameter $\tilde{\theta}_j$. Since $\hat{x}_i$ has no predecessors, from 8.24 we conclude that the only constraint on $\tilde{\theta}_i$ is the resolution condition, i.e. $1 \le \tilde{\theta}_i \le \epsilon(\hat{x}_i)/\gamma_i$, hence $\theta_i^* = 1$. If $\boldsymbol{\theta} = \tilde{\theta}_1, \dots \tilde{\theta}_i, \dots, \tilde{\theta}_k$ is a solution to 8.24, then so is $\tilde{\boldsymbol{\theta}}' = \tilde{\theta}_1, \dots, \theta_i^*, \dots \tilde{\theta}_k$. Indeed, since $x \in R_{\tilde{\theta}_i}(\hat{x}_i) \implies f(x) \in R_{\tilde{\theta}_j}(\hat{x}_j)$ holds, then $x \in R_{\theta_i^*}(\hat{x}_i) \implies f(x) \in R_{\tilde{\theta}_j}(\hat{x}_j)$ also holds, since $R_{\theta_i^*}(\hat{x}_i) \subseteq R_{\tilde{\theta}_i}(\hat{x}_i)$. Similarly, we can set equal to 1 the parameter associated with any abstract state with no predecessors. Now, the predecessors of $\hat{x}_j$ have their associated parameters fixed and computed according to 8.30. We can repeat the argument inductively to show that we can substitute $\tilde{\theta}_l$ with θ_l^* to obtain a new improved solution for 8.24, by solving a cascade of one-dimensional optimization problems following the topological ordering of the abstract states, eventually obtaining the complete solution $\boldsymbol{\theta}^*$.

Corollary 3

By definition of equation (8.30), θ_i^* depends on θ_j^* if and only if $j \in g^{-1}(i)$; in turn, for any such j, θ_j^* depends on θ_l^* if and only if $j \in g^{-1}(j)$, and so on. Accordingly, θ_i^* can be written as a function of the set of θ_j^*'s such that $\hat{x}_j \in \mathcal{A}_i$.

Corollary 4

This follows immediately from the proof of Proposition 10.

Theorem 6

We begin by enunciating two lemmas.

Lemma 5. *Let $\mathcal{M}$ be a mesh such that for every cell C it holds that*

$$x \in C \implies f(x) \notin C. \tag{8.39}$$

Let Refine^p be an operator defined on a pair $(\hat{x}_j, C_j) \in \mathcal{M}$ as

$$\mathit{Refine}^p(\hat{x}_j, C_j) \doteq \{(\hat{x}_j^l, C_j^l) : C_j^l \subseteq C_j, \hat{x}_j^l \in C_j^l, \max_{x \in C_j^l} d(x, \hat{x}_j^l) \leq p\} \tag{8.40}$$

such that the set of C_j^l's forms a partition of C_j. Then it holds that,

$$x \in C_j^l \implies f(x) \notin C_j. \tag{8.41}$$

Intuitively, the lemma above states that, if every cell of a mesh $\mathcal{M}$ is such that any state starting in it leaves it in one step, then any subsequent refinement of the $\mathcal{M}$ will preserve this property. We omit the proof.

Lemma 6. *Let $S = (\mathcal{X}, \mathcal{X}_0, f)$ be a TS with asymptotically stable equilibrium at x^*, where $\mathcal{X}$ is a compact subset of its region of attraction, and f is continuous. Then, for all $r > 0$ every compact subset $T \subseteq \mathcal{X} \setminus B_r(x^*)$ is transient. Moreover, any such T admits a mesh $\mathcal{M}$ satisfying equation (8.39).*

Proof: For every $x \in \mathcal{X}$ it holds that $\lim_{k\to\infty} f^k(x) = x^*$. Define $\tau(x) \doteq \min\{k \in \mathbb{N} : f^k(x) \in B_r(x^*)\}$. By asymptotic stability $\tau(x) < \infty$. The function $\tau : T \to \mathbb{N}$ is upper-semicontinuous. To see this note that, for any k $\{x \in T : \tau(x) < k\}$ is open. Indeed the set $\{x \in T : \exists 1 \leq n \leq k \;.\; f^n(x) \in B_r(x^*)\}$ is open, as the composition f^n of the continuous function f is continuous and $B_r(x^*)$ is open. By the Extreme Value Theorem, we conclude that $\sup_{x \in T} \tau(x) = \tau_{max} < \infty$, hence T is transient. Finally, the mesh $\{(\hat{x}_i, C_i)\}_{i=1}^{\tau_{max}}$ where $C_i = \{x \in T : \tau(x) = i\}$ and $\hat{x}_i \in C_i$ satisfies equation (8.39).

We proceed now to prove Theorem 6.

Proof: By Lemma 6, let $\mathcal{M}$ be a mesh of $\mathcal{X} \setminus B_v(x^*)$ satisfying equation (8.39), let $\mathcal{M}' \doteq \mathcal{M} \cup \{(x^*, B_v(x^*))\}$ be a mesh of $\mathcal{X}$ and let $\hat{S}$ be the resulting deterministic abstraction. Consider an arbitrary abstract state $\hat{x}_i$, and the template for R given by $R_{\hat{\epsilon}}$ in Assumption 6. Since we can incorporate the coverage condition by a suitable lower bound γ_i for $\hat{\epsilon}_i$ as $\gamma_i = \max_{x \in C_i} d(x, \hat{x}_i)$, we focus on the transition consistency and minimum resolution condition. The transition consistency condition can be rewritten as

$$R_{\epsilon_i}(\hat{x}_i) \supseteq \bigcup_{j \in g^{-1}(i)} \mathrm{Post}(R_{\epsilon_j}(\hat{x}_j)), \tag{8.42}$$

where $g^{-1}(i)$ is the set of indices of i's predecessors. Let $f(x)$ be the successor of a point x related to the abstract predecessor $\hat{x}_j$. We have,

$$d(\hat{x}_i, f(x)) \leq d(\hat{x}_i, f(\hat{x}_j)) + d(f(\hat{x}_j), f(x)) \leq \gamma_i + \alpha(d(\hat{x}_j, x)),$$
$$\hat{\epsilon}_j \doteq \max_{x \in R_{\hat{\epsilon}_j}(\hat{x}_j)} d(\hat{x}_j, x),$$

where the first inequality is a consequence of the triangular inequality and the second inequality follows by the definition of γ_i and equation (8.34). Then, setting

$$\hat{\epsilon}_i \geq \gamma_i + \max_{j \in g^{-1}(i)} \alpha(\hat{\epsilon}_j) \tag{8.43}$$

ensures that equation (8.42) is satisfied. Note that $\hat{\epsilon}_j$ is generally greater than its lower bound γ_j, and can not necessarily be reduced simply by splitting the cell j, as $\hat{\epsilon}_j$ depends in turn on its predecessors, and so on. We show now that $\hat{\epsilon}_i$ can be made arbitrarily small.

Denote by $\mathcal{A}_i$ and $\mathcal{D}_i$ the set of abstract states eventually reaching i (ancestors) and the set of abstract states reached by i (descendants) respectively, according to equation (8.28) and equation (8.29). There are two cases:

1. i is not on a cycle: $\mathcal{A}_i \cap \mathcal{D}_i = \emptyset$
2. i is on a cycle: $\mathcal{A}_i \cap \mathcal{D}_i \neq \emptyset$

Case 1: We prove that with a sufficient refinement of the set of ancestors of i we can achieve an arbitrary resolution at i. Let $\mathcal{M}_{\mathcal{A}_i}$ denote the sub-mesh associated with the ancestors of i, i.e. $\mathcal{M}_{\mathcal{A}_i} \doteq \{(\hat{x}_j, C_j) \in \mathcal{M} : j \in \mathcal{A}_i\}$, and let τ be the transient time of their union. We construct a refinement $\mathcal{M}'_{\mathcal{A}_i}$ of $\mathcal{M}_{\mathcal{A}_i}$ by subdividing every pair $(\hat{x}_j, C_j)$ with the refinement operator defined in equation (8.40) as

$$\mathcal{M}'_{\mathcal{A}_i} \doteq \bigcup_{(\hat{x}_j, C_j) \in \mathcal{M}_{\mathcal{A}_i}} \text{Refine}^p(\hat{x}_j, C_j),$$

and let $\hat{S}'$ denote the abstraction obtained by replacing $\mathcal{M}_{\mathcal{A}_i}$ with $\mathcal{M}'_{\mathcal{A}_i}$, and let $\mathcal{A}'_i$ denote the new set of ancestors of i. It holds that $\mathcal{A}'_i \cap \mathcal{D}_i = \emptyset$. Since the refinement does not change the transient time of a set, the new furthest ancestor of i is at distance $q \leq \tau$. Then, setting

$$\hat{\epsilon}_i \geq \gamma_i + \Phi^q(p) \tag{8.44}$$

satisfies the transition consistency condition for the state $\hat{x}_i$ of $\hat{S}'$, where $\Phi(p) = p + \alpha(p)$ is a class $\mathcal{K}$ function of p and Φ^q is the composition of Φ q times. Since both γ_i and p can be made arbitrarily small, we can achieve an arbitrary resolution at $\hat{x}_i$, therefore guaranteeing the satisfaction of $\hat{\epsilon}_i \leq \epsilon(\hat{x}_i)$.

Case 2: The set $\mathcal{D}_i$ is the set of states forming the cycle. By assumption, S does not admit periodic trajectories. Therefore, the cycle in the abstraction $\hat{S}$ is a spurious behaviour due to the quantisation of abstract the transition function $\hat{f} \doteq \kappa \circ f$ and there exists a refinement of $\mathcal{D}_i$ that eliminates the cycle. Let C be the region defined by the union of the cells associated with $\mathcal{D}_i$; there exists $q \in \mathbb{N}$ such that $f^q(\hat{x}_i) \notin C$. It follows that, by adding to the set of states of the abstraction the points $f^j(\hat{x}_i)$ for $j \leq q$, the cycle is eliminated, therefore leading back to Case 1.

It remains to address the cell containing the equilibrium $B_r(x^*)$. Let $\hat{x}_0 \doteq x^*$; clearly $\hat{x}_0$ is its own predecessor, that is $0 \in g^{-1}(0)$. As we have argued in equation (8.44), the lower bound elicited by the abstract states $\{\hat{x}_j \in \hat{\mathcal{X}} : j \neq 0 \in g^{-1}(0)\}$ on $\hat{\epsilon}_0$ can be made arbitrarily small following the same reasoning. Finally, by asymptotic stability, f admits a Lyapunov function $V : \mathcal{X} \to \mathbb{R}$. Let $\beta = \min_{d(x,x^*)=\epsilon(\hat{x}_0)} V(x)$.Then the set $\Omega_\beta = \{x \in \mathcal{X} : V(x) \leq \beta\}$ is positively invariant. Setting $R(\hat{x}_0) = \Omega_\beta$ satisfies the condition. Note that, in contrast with the abstract states in $\mathcal{M}$, the concrete states related to the equilibrium x^*, $R(\hat{x}_0)$, are not necessarily in the form $\{x \in \mathcal{X} : d(x, x^*) \leq \hat{\epsilon}_0\}$, since Ω_β is a sub-level set of a Lyapunov function □

8.9.2. Safety and Reachability

Once we have established that a relation R is a multi-resolution ϵ-approximate bisimulation relation between S and $\hat{S}$ we can analyze safety and reachability problems defined for the system S using instead a corresponding problem defined for the abstractions $\hat{S}$.

Safety

Let Reach(S) denote the set of reachable states for S, that is

$$\text{Reach}(S) \doteq \{x \in \mathcal{X} : \exists x_0 \in \mathcal{X}_0, k \in \mathbb{Z}_{\geq 0}(x_0 x_1 \ldots \in \mathcal{B}_{x_0}(S) \wedge x_k = x)\} \tag{8.45}$$

By definition, for any $x \in \text{Reach}(S)$ there exists a behaviour $x_0 x_1 \ldots x_k \ldots$ such that $x_k = x$. Then there exists a behaviour $\hat{x}_0 \hat{x}_1 \ldots \hat{x}_k \ldots$ of $\hat{S}$ satisfying $(x_i, \hat{x}_i) \in R$ for all $i \leq k$, hence $\hat{x}_k \in \text{Reach}(\hat{S})$. Moreover, we have $d(x_k, \hat{x}_k) \leq \delta_{\hat{x}_k}$. Let $\text{Reach}(\hat{S})_R \supseteq \text{Reach}(\hat{S})$ be the set defined as

$$\text{Reach}(\hat{S})_R \doteq \{x \in \mathcal{X} : \exists \hat{x} \in \text{Reach}(\hat{S})(d(x, \hat{x}) \leq \delta_{\hat{x}})\}. \tag{8.46}$$

Consequently, $x \in \text{Reach}(\hat{S})_R$. We conclude that

$$\text{Reach}(S) \subseteq \text{Reach}(\hat{S})_R. \tag{8.47}$$

Let A denote some *avoid* or *unsafe* states. If $\text{Reach}(\hat{S})_R \cap A = \emptyset$ then $\text{Reach}(S) \cap A = \emptyset$.

Reachability

Similarly, for any $\hat{x} \in \text{Reach}(\hat{S})$ there exists a behaviour $\hat{x}_0 \hat{x}_1 \ldots \hat{x}_k \ldots$ such that $\hat{x}_k = \hat{x}$. Then there exists a behaviour $x_0 x_1 \ldots x_k \ldots$ of S satisfying $(x_i, \hat{x}_i) \in R$ for all $i \leq k$, hence $\hat{x}_k \in \text{Reach}(\hat{S})$. Moreover, we have $d(x_k, \hat{x}_k) \leq \delta_{\hat{x}_k}$. Let G be some *reach* or *goal* states. If there exists $\hat{x}$ in $\text{Reach}(\hat{S}) \cap G \neq \emptyset$ such that $R^{-1}(\hat{x}) \subseteq G$ then $\text{Reach}(S) \cap G \neq \emptyset$.

9

Conclusions and Research Outlook

9.1. General Conclusions

This thesis investigated how data can be incorporated into formal methods for system abstractions. We considered three progressively richer settings for model knowledge: essentially absent (Part I), partial (Part II), and total (Part III).

In Parts I and II, data was used to compensate for missing knowledge of the dynamics, yielding PAC-style guarantees despite uncertainty. In Part III, by contrast, data guided the construction of abstractions that were subsequently corroborated with known dynamics. This progression highlights a central theme: while data enables formal guarantees across a spectrum of modelling assumptions, the cost of substituting missing knowledge with data grows rapidly in terms of sample complexity.

Although many other scenarios are actively studied in the broader community of data-driven abstractions, the three settings addressed in this thesis illustrate the fundamental trade-offs between model knowledge, data requirements, and computational tractability. The results suggest that careful integration of analytical insight with data-driven methods is essential to make abstraction-based techniques both expressive and scalable.

In what follows, we discuss the main limitations of the proposed approaches and outline directions for future research.

9.2. Part I - Deterministic Systems with Unknown Dynamics

In Part I of this thesis, we explored how finite-time external trajectories can be turned into finite-state abstractions that enable verification and control of deterministic systems with unknown dynamics. Across the chapters, we moved from fundamental questions about how sampled trajectories relate to formal notions in abstraction-based verification and control synthesis, to practical constructions that demonstrate these ideas.

At a high level, this part of the thesis illustrates one way to establish a middle ground between two worlds that often appear disjoint: on one side, formal methods, which provide rigorous guarantees but typically require explicit system models or exhaustive analysis; on the other, data-driven approaches, which bypass explicit modeling but often lack reliability beyond empirical performance.

Our main contribution has been to demonstrate that finite data, when combined with concepts such as behavioural inclusion and alternating simulation, can support rigorous guarantees even in the absence of system models. In this way, we bring the tools of formal verification closer to black-box and learning-based settings.

Several key themes recur throughout Part I:

1. **Abstraction from behaviours rather than states.** We developed a theoretical framework for systems where direct state measurements are unavailable and where the output does not necessarily admit a meaningful notion of distance. This represents a significant shift: abstraction is reframed as something that lives in the space of behaviours rather than the set of states. This was made possible by adopting the SAlCA formalism. To the best of our knowledge, this perspective had not previously been studied within the control community.

2. **Probabilistic reasoning as a bridge.** In settings where no knowledge of the dynamics is available, we assumed a random distribution over initial conditions to collect system behaviours. While not without limitations, this assumption enabled us to formally reason about the reliability of abstractions built from incomplete knowledge. Scenario theory provided the language to connect finite data with universal statements about system behaviour, via the data-driven SAlCA.

3. **The importance of model knowledge.** Assuming a distribution over initial conditions also imposes restrictions: guarantees may only hold for the time horizon corresponding to the collected data, unless coarse knowledge of the system dynamics is available.

4. **Specification-agnostic design.** The abstractions we constructed can be reused across multiple verification and synthesis problems without the need for resampling. This is particularly valuable in settings where specifications evolve or where several tasks must be solved from the same dataset.

The framework developed here also reveals a number of limitations and open questions at the intersection of data-driven and formal approaches:

1. **Sample complexity and conservativeness of scenario guarantees.** Without a model of the dynamics, the amount of data required to obtain high-certainty abstractions is generally large, even for relatively simple systems. At the same time, our experiments showed that the bounds obtained via scenario theory are often conservative: the empirical violation (the probability of observing a behaviour absent from the data-driven SAlCA) is typically one or two orders of magnitude lower than the PAC bounds. Constructing a data-driven SAlCA requires solving a simple yet degenerate scenario optimization program. As a result, probabilistic bounds may only be computed *a posteriori*, after the scenario complexity has been determined. This should not be entirely surprising: as argued in Remark 7, bounding the probability of observing a new behaviour absent from the data-driven SAlCA is closely related to bounding the probability of observing a new symbol after drawing N samples from a finite alphabet under an unknown distribution. Put differently, bounding the probability of observing a new l-sequence can be viewed as estimating the support size of an unknown distribution—a problem with a rich history in statistics and biology [138]. Establishing connections with this line of research could open new paradigms for guarantees on data-driven abstractions.

   Another key observation is that sample complexity depends on the *behavioural richness* of the system. This reflects not only the transition dynamics but also the output map. For example, if a system contains a strange attractor (e.g., the Lorenz system), its state dynamics are chaotic; if the invariant set is finely partitioned, then external behaviours will reflect this. Conversely, a coarser partition may obscure such richness. More generally, finer partitions yield more local information but also increase scenario complexity, leading to looser

PAC guarantees. When the output map is designable, an interesting research direction is to select it in ways that avoid unnecessary behavioural richness.

2. **Discarding behaviours.** In Chapter 3, we showed that if the data-driven SAlCA satisfies a property, then the same property can be guaranteed (with probability) for the original system. However, if the abstraction does *not* satisfy a property, can we discard some l-sequences to obtain a partial SAlCA that does, while still retaining probabilistic guarantees? This question is closely related to the sample-and-discard approach in scenario theory. To the best of our knowledge, existing theory does not cover non-degenerate scenario programs. Recently, a link between sample-and-discard methods and conformal prediction was established [139]; since conformal prediction rests on weaker assumptions, it may provide a pathway to answering this question.

3. **Control is exponentially harder than verification.** Our abstractions for control synthesis rely on randomly sampled input sequences. This leads to a degradation in scenario guarantees that is exponential in the time horizon and polynomial in the input space cardinality. Consequently, much larger datasets are required to obtain non-trivial bounds for control compared to verification.

   Restricting the set of input sequences could mitigate this limitation, though it raises challenges for ensuring freedom in control synthesis. Another potential avenue is to divide abstraction construction across separate regions of the state domain, potentially reducing the exponential penalty on horizon length.

4. **The gap to infinite-horizon guarantees.** Although we provided sufficient conditions to extend guarantees to arbitrarily long but finite horizons, reliance on finite-length trajectories limits direct applicability to infinite-horizon specifications, which are especially relevant for safety-critical systems. One possible direction is to exploit ergodicity and invariant measures [140], ensuring that the probability of observing a new l-sequence stabilizes after some horizon H (or changes only marginally).

9.3. Part II - Stochastic Systems with Partially Known Dynamics

In Chapter 7, we developed a new abstraction procedure for discrete-time stochastic dynamical systems, combining elements of backward reachability, scenario theory, and nondeterministic transitions. The resulting models take the form of Bundled Interval Markov Decision Processes (bIMDP), where nondeterminism in the probability space is explicitly represented. This allows us to capture uncertainty more flexibly and to design controllers that can address a broader range of reach-avoid problems compared to existing single-target constructions.

A central insight of this work is that introducing nondeterministic transitions over sets of probability measures enriches the action space, which in turn makes it possible to synthesise policies that achieve specifications with higher probability. At the same time, this comes with trade-offs: the resulting abstract models are larger.

This tension between expressiveness and tractability is a recurring theme in the design of data-driven abstractions.

From a methodological perspective, similiarly to [36], a controllability assumption together with the separation of the dynamics in a deterministic and additive random component allowed us to avoid sampling multi-step transitions for the application of the scenario theory: by constranining the deterministic dynamics to only target reference points, we ensure that the addition of the noise generates a distribution that was characterized with noise realizations offline.

Overall, this chapter illustrates how nondeterminism can be used as a resource in the abstraction of stochastic systems, rather than merely as a source of conservatism. By enlarging the set of admissible strategies, our approach generalises earlier results and highlights new opportunities for correct-by-design control under uncertainty. At the same time, it underscores the challenges that remain in making such abstractions scalable, precise, and broadly applicable.

Several directions emerge naturally from this work:

1. **Algorithmic development for bIMDPs.** Current methods rely on embedding bIMDPs into IMDPs. Tailored algorithms that directly exploit the structure of the uncertain transitions in bIMDPs would allow us to reduce conservatism in the final guarantees.

2. **Managing model size.** While richer action sets improve specification satisfaction, they also inflate the abstract model. Techniques such as state aggregation, adaptive refinement, or learning-based partitioning could help strike a better balance between precision and scalability.

3. **Handling nonconvexity.** Enabling actions requires checking a set inclusion: the nonconvex sets that may arise in backward reachability can slow down the construction of the bIMDP. Developing approximation schemes or convex relaxations that retain guarantees is a key open challenge.

4. **Beyond additive noise.** The current framework assumes a known deterministic component with additive uncertainty. Extending the method to more general stochastic dynamics, possibly with multiplicative or correlated noise, would substantially broaden applicability.

9.4. Part III - Deterministic Systems with Known Dynamics

Chapter 8 introduced a fully automated, counterexample-guided framework for synthesising multi-resolution abstractions of deterministic, continuous-state dynamical systems. At its core is the notion of multi-resolution approximate bisimulation, which generalises classical ϵ-bisimulations by allowing state-dependent error bounds. This strictly enlarges the class of systems that can be abstracted: in particular, we proved that incrementally uniformly bounded systems admit such abstractions, even when no uniform relation exists.

By adapting resolution locally to system behaviour, multi-resolution bisimulations preserve rigorous guarantees while producing abstractions that are often significantly more compact than uniform grids. The proposed CEGIS algorithm realises this idea constructively, without requiring Lyapunov functions or linearisation, and scales through graph-based parallelisation. Experiments confirmed both the efficiency and the practical flexibility of the method.

At the same time, important challenges remain:

1. **Presence of heuristics.** In the Relaxed Problem 2 we separated the selection of abstract states from the selection of the relation for computational reasons. This prevents us from answering whether for a fixed budget of abstract states there exists solution to the MRASP. To overcome this we couple a clustering and refinement strategy, both heuristics-based. A scalable learning approach that can solve Relaxed Problem 1 would improve the compactness of the resulting abstraction.

2. **Richer templates.** The proof of Theorem 6 suggests that more general templates could ensure termination of the abstraction algorithm, particularly near stable equilibria.

3. **Scalable verification.** Experiments show that verifying candidate relations through SMT solving incurs significant slowdowns in higher dimensions, even with linear dynamics and polytopic templates. While our parallelisable algorithm mitigates this to some extent, more efficient verification methods are essential to broaden applicability.

4. **Beyond determininistic abstractions.** The current framework generates deterministic abstractions by construction, which limits its ability to capture systems with unstable equilibria, limit cycles, or inherent nondeterminism. Extending the method to nondeterministic abstractions would enlarge its scope, though defining minimal-size abstractions in this setting presents nontrivial challenges.

5. **Tool integration.** Embedding the algorithm into existing verification and control toolchains would enable real-world applications and provide benchmarks to guide future theory.

Closing Comments

Taken together, the results of this thesis show that data and formal methods need not be opposing paradigms but can be combined into complementary approaches for system abstraction. Whether in the complete absence of a model, under partial knowledge, or when full dynamics are available, the integration of data reshapes both the opportunities and the challenges of correct-by-design verification and control. The hope is that the insights developed here contribute to a broader research agenda: building scalable, rigorous, and flexible tools that bring the reliability of formal methods into data-rich, real-world systems.

Bibliography

[1] M. Dowson. 'The Ariane 5 software failure'. In: *ACM SIGSOFT Software Engineering Notes* 22.2 (1997), p. 84.

[2] Ü. Çetinkaya, M. Yeşil, R. Bayındır and E. Irmak. 'Technical Analysis and Strategic Insights from the 2025 Spain Blackout'. In: *2025 13th International Conference on Smart Grid (icSmartGrid)*. IEEE. 2025, pp. 809–815.

[3] T. Nakajo and H. Kume. 'A case history analysis of software error cause-effect relationships'. In: *IEEE Transactions on Software Engineering* 17.8 (1991), p. 830.

[4] N. G. Leveson. 'Role of software in spacecraft accidents'. In: *Journal of spacecraft and Rockets* 41.4 (2004), pp. 564–575.

[5] C. Baier and J.-P. Katoen. *Principles of model checking*. MIT press, 2008.

[6] P. Tabuada. *Verification and control of hybrid systems: a symbolic approach*. Springer Science & Business Media, 2009.

[7] D. Reijsbergen, P.-T. De Boer, W. Scheinhardt and B. Haverkort. 'On hypothesis testing for statistical model checking'. In: *International journal on software tools for technology transfer* 17.4 (2015), pp. 377–395.

[8] T. S. Badings, A. Abate, N. Jansen, D. Parker, H. A. Poonawala and M. Stoelinga. 'Sampling-based robust control of autonomous systems with non-Gaussian noise'. In: *Workshops at the Thirty-Sixth AAAI Conference on Artificial Intelligence*. 2022.

[9] A. Girard and G. J. Pappas. 'Approximate bisimulations for nonlinear dynamical systems'. In: *Proceedings of the 44th IEEE Conference on Decision and Control*. IEEE. 2005, pp. 684–689.

[10] Y. Tazaki and J.-i. Imura. 'Discrete abstractions of nonlinear systems based on error propagation analysis'. In: *IEEE Transactions on Automatic Control* 57.3 (2011), pp. 550–564.

[11] P. Akella and A. D. Ames. 'A Barrier-Based Scenario Approach to Verifying Safety-Critical Systems'. In: *IEEE Robotics and Automation Letters* 7.4 (2022), pp. 11062–11069.

[12] A. Salamati, A. Lavaei, S. Soudjani and M. Zamani. 'Data-Driven Safety Verification of Stochastic Systems via Barrier Certificates'. In: *IFAC-PapersOnLine* 54.5 (2021). 7th IFAC Conference ADHS 2021, pp. 7–12. ISSN: 2405-8963. DOI: https://doi.org/10.1016/j.ifacol.2021.08.466. URL: https://www.sciencedirect.com/science/article/pii/S2405896321012416.

[13] A. Devonport and M. Arcak. ‘Estimating reachable sets with scenario optimization’. In: *Learning for dynamics and control.* PMLR. 2020, pp. 75–84.

[14] A. Devonport and M. Arcak. ‘Data-driven estimation of forward reachable sets’. In: *Proceedings of the Workshop on Computation-Aware Algorithmic Design for Cyber-Physical Systems.* 2021, pp. 11–12.

[15] A. Devonport, F. Yang, L. El Ghaoui and M. Arcak. ‘Data-Driven Reachability Analysis with Christoffel Functions’. In: *2021 60th IEEE Conference on Decision and Control (CDC).* 2021, pp. 5067–5072. DOI: 10.1109/CDC45484.2021.9682860.

[16] A. Devonport, A. Saoud and M. Arcak. ‘Symbolic Abstractions From Data: A PAC Learning Approach’. In: *2021 60th IEEE Conference on Decision and Control (CDC).* 2021, pp. 599–604. DOI: 10.1109/CDC45484.2021.9683316.

[17] S. Sadraddini and C. Belta. ‘Formal guarantees in data-driven model identification and control synthesis’. In: *Proceedings of the 21st HSCC (part of CPS Week).* 2018, pp. 147–156.

[18] A. Makdesi, A. Girard and L. Fribourg. ‘Data-Driven Models of Monotone Systems’. In: *IEEE Transactions on Automatic Control* (2023).

[19] M. Kazemi, R. Majumdar, M. Salamati, S. Soudjani and B. Wooding. ‘Data-driven abstraction-based control synthesis’. In: *Nonlinear Analysis: Hybrid Systems* 52 (2024), p. 101467.

[20] D. Ajeleye, A. Lavaei and M. Zamani. ‘Data-driven controller synthesis via finite abstractions with formal guarantees’. In: *IEEE Control Systems Letters* 7 (2023), pp. 3453–3458.

[21] B. Xue, M. Zhang, A. Easwaran and Q. Li. ‘PAC model checking of black-box continuous-time dynamical systems’. In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 39.11 (2020), pp. 3944–3955.

[22] Z. Wang and R. M. Jungers. ‘Scenario-Based Set Invariance Verification for Black-Box Nonlinear Systems’. In: *IEEE Control Systems Letters* 5.1 (2021), pp. 193–198. DOI: 10.1109/LCSYS.2020.3001882.

[23] Z. Wang and R. M. Jungers. ‘A data-driven method for computing polyhedral invariant sets of black-box switched linear systems’. In: *IEEE Control Systems Letters* (2020).

[24] M. C. Campi and S. Garatti. ‘The exact feasibility of randomized solutions of uncertain convex programs’. In: *SIAM Journal on Optimization* 19.3 (2008), pp. 1211–1230.

[25] G. C. Calafiore and M. C. Campi. ‘The scenario approach to robust control design’. In: *IEEE Transactions on automatic control* 51.5 (2006), pp. 742–753.

[26] A. Abate and M. Prandini. ‘Approximate abstractions of stochastic systems: A randomized method’. In: *2011 50th IEEE Conference on Decision and Control and European Control Conference.* IEEE. 2011, pp. 4861–4866.

[27] K. G. Larsen and A. Skou. 'Bisimulation through probabilistic testing (preliminary report)'. In: *Proceedings of the 16th ACM SIGPLAN-SIGACT symposium on Principles of programming languages*. 1989, pp. 344–352.

[28] A. Abate. 'Approximation metrics based on probabilistic bisimulations for general state-space Markov processes: a survey'. In: *Electronic Notes in Theoretical Computer Science* 297 (2013), pp. 3–25.

[29] J. Desharnais, F. Laviolette and M. Tracol. 'Approximate analysis of probabilistic processes: Logic, simulation and games'. In: *2008 Fifth International Conference on Quantitative Evaluation of Systems*. IEEE. 2008, pp. 264–273.

[30] M. Lahijanian, S. B. Andersson and C. Belta. 'Formal verification and synthesis for discrete-time stochastic systems'. In: *IEEE Transactions on Automatic Control* 60.8 (2015), pp. 2031–2045.

[31] A. Nilim and L. El Ghaoui. 'Robust control of Markov decision processes with uncertain transition matrices'. In: *Operations Research* 53.5 (2005), pp. 780–798.

[32] R. Givan, S. Leach and T. Dean. 'Bounded-parameter Markov decision processes'. In: *Artificial Intelligence* 122.1 (2000), pp. 71–109. ISSN: 0004-3702. DOI: https://doi.org/10.1016/S0004-3702(00)00047-3. URL: https://www.sciencedirect.com/science/article/pii/S0004370200000473.

[33] T. Dean, R. Givan and S. Leach. 'Model reduction techniques for computing approximately optimal solutions for Markov decision processes'. In: *Proceedings of the Thirteenth conference on Uncertainty in artificial intelligence*. 1997, pp. 124–131.

[34] T. G. Dietterich and J. Hostetler. 'Conformal prediction intervals for markov decision process trajectories'. In: *arXiv preprint arXiv:2206.04860* (2022).

[35] M. Cubuktepe, N. Jansen, S. Junges, J.-P. Katoen and U. Topcu. 'Scenario-based verification of uncertain mdps'. In: *International Conference on Tools and Algorithms for the Construction and Analysis of Systems*. Springer. 2020, pp. 287–305.

[36] T. Badings, L. Romao, A. Abate, D. Parker, H. A. Poonawala, M. Stoelinga and N. Jansen. 'Robust control for dynamical systems with non-gaussian noise via formal abstractions'. In: *Journal of Artificial Intelligence Research* 76 (2023), pp. 341–391.

[37] T. Badings, L. Romao, A. Abate and N. Jansen. 'Probabilities are not enough: Formal controller synthesis for stochastic dynamical models with epistemic uncertainty'. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 37. 12. 2023, pp. 14701–14710.

[38] A. Lavaei, S. Soudjani, E. Frazzoli and M. Zamani. 'Constructing MDP Abstractions Using Data with Formal Guarantees'. In: *IEEE Control Systems Letters* (2022).

[39] A. Lavaei. ‘MDP abstractions from data: Large-scale stochastic networks’. In: *2023 62nd IEEE Conference on Decision and Control (CDC)*. IEEE. 2023, pp. 6058–6063.

[40] A. Banse, L. Romao, A. Abate and R. M. Jungers. ‘Data-driven Abstractions via Adaptive Refinements and a Kantorovich Metric’. In: *2023 62nd IEEE Conference on Decision and Control (CDC)*. IEEE. 2023, pp. 6038–6043. DOI: 10.1109/CDC49753.2023.10383513.

[41] A. Banse, L. Romao, A. Abate and R. Jungers. ‘Data-driven memory-dependent abstractions of dynamical systems’. In: *Learning for Dynamics and Control Conference*. PMLR. 2023, pp. 891–902.

[42] V. Vovk, A. Gammerman and G. Shafer. *Algorithmic learning in a random world*. Vol. 29. Springer, 2005.

[43] A. Girard and G. J. Pappas. ‘Approximate bisimulation: A bridge between computer science and control theory’. In: *European Journal of Control* 17.5-6 (2011), pp. 568–578.

[44] A. Girard, G. Pola and P. Tabuada. ‘Approximately bisimilar symbolic models for incrementally stable switched systems’. In: *IEEE Transactions on Automatic Control* 55.1 (2009), pp. 116–126.

[45] M. Zamani, G. Pola, M. Mazo and P. Tabuada. ‘Symbolic models for nonlinear control systems without stability assumptions’. In: *IEEE Transactions on Automatic Control* 57.7 (2011), pp. 1804–1809.

[46] J. Camara, A. Girard and G. Gössler. ‘Safety controller synthesis for switched systems using multi-scale symbolic models’. In: *2011 50th IEEE Conference on Decision and Control and European Control Conference*. IEEE. 2011, pp. 520–525.

[47] J. Cámara, A. Girard and G. Gössler. ‘Synthesis of switching controllers using approximately bisimilar multiscale abstractions’. In: *Proceedings of the 14th international conference on Hybrid systems: computation and control*. 2011, pp. 191–200.

[48] A. Girard, G. Gössler and S. Mouelhi. ‘Safety controller synthesis for incrementally stable switched systems using multiscale symbolic models’. In: *IEEE Transactions on Automatic Control* 61.6 (2015), pp. 1537–1549.

[49] K. Hsu, R. Majumdar, K. Mallik and A.-K. Schmuck. ‘Multi-layered abstraction-based controller synthesis for continuous-time systems’. In: *Proceedings of the 21st International Conference on Hybrid Systems: Computation and Control (part of CPS Week)*. 2018, pp. 120–129.

[50] Y. Tazaki and J.-i. Imura. ‘Approximately bisimilar discrete abstractions of nonlinear systems using variable-resolution quantizers’. In: *Proceedings of the 2010 American Control Conference*. IEEE. 2010, pp. 1015–1020.

[51] A. Abate, M. Giacobbe and Y. Schnitzer. ‘Bisimulation learning’. In: *International Conference on Computer Aided Verification*. Springer. 2024, pp. 161–183.

[52] A. Abate, A. Edwards and M. Giacobbe. 'Neural abstractions'. In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 26432–26447.

[53] A. Nadali, B. Zhong, A. Trivedi and M. Zamani. 'Transfer Learning for Control Systems via Neural Simulation Relations'. In: *arXiv preprint arXiv:2412.01783* (2024).

[54] A.-K. Schmuck, P. Tabuada and J. Raisch. 'Comparing asynchronous l-complete approximations and quotient based abstractions'. In: *2015 54th IEEE Conference on Decision and Control (CDC)*. IEEE. 2015, pp. 6823–6829.

[55] R. Coppola, A. Peruffo and M. Mazo Jr. 'Data-driven Abstractions for Verification of Linear Systems'. In: *IEEE Control Systems Letters* (2023).

[56] R. Coppola, A. Peruffo and M. Mazo Jr. 'Data-Driven Abstractions for Control Systems via Random Exploration'. In: *arXiv preprint arXiv:2402.10668* (2024).

[57] R. Coppola, H. Touloujian, P. Ombrini and M. Mazo Jr. 'Reinforcement Learning for Robust Ageing-Aware Control of Li-ion Battery Systems with Data-Driven Formal Verification'. In: *arXiv preprint arXiv:2509.04288* (2025).

[58] R. Coppola and M. Mazo Jr. 'On Training-Conditional Conformal Prediction and Binomial Proportion Confidence Intervals'. In: *Transactions on Machine Learning Research* (2025). ISSN: 2835-8856. URL: https://openreview.net/forum?id=pSk5qyt1ob.

[59] R. Coppola, A. Peruffo, L. Romao, A. Abate and M. Mazo Jr. 'Enhancing Data-Driven Stochastic Control via Bundled Interval MDP'. In: *IEEE Control Systems Letters* 8 (2024), pp. 2069–2074.

[60] R. Coppola, Y. Schnitzer, M. Giacobbe, A. Abate and M. Mazo Jr. *Existence and Synthesis of Multi-Resolution Approximate Bisimulations for Continuous-State Dynamical Systems*. 2025. arXiv: 2509.17739 [eess.SY]. URL: https://arxiv.org/abs/2509.17739.

[61] A.-K. Schmuck and J. Raisch. 'Asynchronous l-complete approximations'. In: *Systems & Control Letters* 73 (2014), pp. 67–75.

[62] G. A. De Gleizer and M. Mazo. 'Computing the sampling performance of event-triggered control'. In: *24th ACM International Conference on Hybrid Systems Computation and Control (HSCC)*. Association for Computing Machinery (ACM). 2021.

[63] M. C. Campi, S. Garatti and F. A. Ramponi. 'A general scenario theory for nonconvex optimization and decision making'. In: *IEEE Transactions on Automatic Control* 63.12 (2018), pp. 4067–4078.

[64] S. Garatti and M. C. Campi. 'The risk of making decisions from data through the lens of the scenario approach'. In: *IFAC-PapersOnLine* 54.7 (2021), pp. 607–612.

[65] P. M. Esfahani, T. Sutter and J. Lygeros. ‘Performance bounds for the scenario approach and an extension to a class of non-convex programs’. In: *IEEE Transactions on Automatic Control* 60.1 (2014), pp. 46–58.

[66] A. Lavaei. ‘Symbolic abstractions with guarantees: A data-driven divide-and-conquer strategy’. In: *2023 62nd IEEE Conference on Decision and Control (CDC)*. IEEE. 2023, pp. 7994–7999.

[67] A. Lavaei and E. Frazzoli. ‘Data-Driven Synthesis of Symbolic Abstractions with Guaranteed Confidence’. In: *IEEE Control Systems Letters* (2022).

[68] S. Ghosh, S. Bansal, A. Sangiovanni-Vincentelli, S. A. Seshia and C. Tomlin. ‘A new simulation metric to determine safe environments and controllers for systems with unknown dynamics’. In: *Proceedings of the 22nd ACM International Conference on Hybrid Systems: Computation and Control*. 2019, pp. 185–196.

[69] A. Peruffo and M. Mazo. ‘Data-driven Abstractions with Probabilistic Guarantees for Linear PETC Systems’. In: *IEEE Control Systems Letters* (2022).

[70] A. Peruffo and M. Mazo Jr. ‘Sampling Performance of Periodic Event-Triggered Control Systems: a Data-driven Approach’. In: *IEEE Transactions on Control of Networked Systems, to appear* (2023).

[71] J.-C. Fernandez. ‘An implementation of an efficient algorithm for bisimulation equivalence’. In: *Science of Computer Programming* 13.2-3 (1990), pp. 219–236.

[72] C. Belta, B. Yordanov and E. A. Gol. *Formal Methods for Discrete-Time Dynamical Systems*. Vol. 89. Springer, 2017.

[73] C. J. Watkins and P. Dayan. ‘Q-learning’. In: *Machine learning* 8.3 (1992), pp. 279–292.

[74] R. Alur, T. A. Henzinger, O. Kupferman and M. Y. Vardi. ‘Alternating refinement relations’. In: *CONCUR’98 Concurrency Theory: 9th International Conference Nice, France, September 8–11, 1998 Proceedings 9*. Springer. 1998, pp. 163–178.

[75] A. Care, S. Garatti and M. C. Campi. ‘FAST: an algorithm for the scenario approach with reduced sample complexity’. In: *IFAC Proceedings Volumes* 44.1 (2011), pp. 9236–9241.

[76] C. L. Lim, A. Moffat and A. Wirth. ‘Lazy and eager approaches for the set cover problem’. In: *Proceedings of the Thirty-Seventh Australasian Computer Science Conference-Volume 147*. 2014, pp. 19–27.

[77] A. Lasota and M. C. Mackey. *Chaos, fractals, and noise: stochastic aspects of dynamics*. Vol. 97. Springer Science & Business Media, 2013.

[78] R. Coppola, A. Peruffo and M. Mazo Jr. ‘Data-driven Abstractions for Verification of Deterministic Systems’. In: *arXiv preprint arXiv:2211.01793* (2022).

[79] F. Bullo. *Contraction Theory for Dynamical Systems*. 1.1. Kindle Direct Publishing, 2023. ISBN: 979-8836646806. URL: https://fbullo.github.io/ctds.

[80] *Mountaimn Car Environment.* https://gymnasium.farama.org/environments/classic_control/mountain_car/. Accessed: 2023-09-30.

[81] J. Weng, A. Jossen, A. Stefanopoulou, J. Li, X. Feng and G. Offer. 'Fast-charging lithium-ion batteries require a systems engineering approach'. In: *Nature Energy* (2025), pp. 1–2.

[82] K. A. Smith, C. D. Rahn and C.-Y. Wang. 'Control oriented 1D electrochemical model of lithium ion battery'. In: *Energy Conversion and management* 48.9 (2007), pp. 2565–2578.

[83] A. Barré, B. Deguilhem, S. Grolleau, M. Gérard, F. Suard and D. Riu. 'A review on lithium-ion battery ageing mechanisms and estimations for automotive applications'. In: *Journal of power sources* 241 (2013), pp. 680–689.

[84] R. Darling and J. Newman. 'Modeling Side Reactions in Composite LiyMn2 O 4 Electrodes'. In: *Journal of The Electrochemical Society* 145.3 (1998), p. 990.

[85] Z. Khalik, H. J. Bergveld and M. Donkers. 'Ageing-aware charging of lithium-ion batteries using a surrogate model'. In: *2021 American Control Conference (ACC)*. IEEE. 2021, pp. 4414–4420.

[86] Z. Khalik, H. J. Bergveld and M. Donkers. 'Ageing-aware charging of lithium-ion batteries using an electrochemistry-based model with capacity-loss side reactions'. In: *2020 American Control Conference (ACC)*. IEEE. 2020, pp. 2213–2218.

[87] Z. Khalik. 'Modeling and Optimal Control for Aging-Aware Charging of Batteries'. PhD Thesis (Research TU/e / Graduation TU/e). Eindhoven: Eindhoven University of Technology, Nov. 2021. ISBN: 9789038653822.

[88] S. Kolluri, S. V. Aduru, M. Pathak, R. D. Braatz and V. R. Subramanian. 'Real-time nonlinear model predictive control (NMPC) strategies using physics-based models for advanced lithium-ion battery management system (BMS)'. In: *Journal of The Electrochemical Society* 167.6 (2020), p. 063505.

[89] S. Park, A. Pozzi, M. Whitmeyer, H. Perez, A. Kandel, G. Kim, Y. Choi, W. T. Joe, D. M. Raimondo and S. Moura. 'A Deep Reinforcement Learning Framework for Fast Charging of Li-ion Batteries'. In: *IEEE Transactions on Transportation Electrification* 8.2 (2022), pp. 2770–2784.

[90] T. Gao, Y. Han, D. Fraggedakis, S. Das, T. Zhou, C.-N. Yeh, S. Xu, W. C. Chueh, J. Li and M. Z. Bazant. 'Interplay of lithium intercalation and plating on a single graphite particle'. In: *Joule* 5.2 (2021), pp. 393–414.

[91] X. Lu, M. Lagnoni, A. Bertei, S. Das, R. E. Owen, Q. Li, K. O'Regan, A. Wade, D. P. Finegan, E. Kendrick *et al.* 'Multiscale dynamics of charging and plating in graphite electrodes coupling operando microscopy and phase-field modelling'. In: *Nature Communications* 14.1 (2023), p. 5127.

[92] M. A. Chowdhury, S. S. Al-Wahaibi and Q. Lu. ‘Adaptive safe reinforcement learning-enabled optimization of battery fast-charging protocols’. In: *AIChE Journal* 71.1 (2025), e18605.

[93] A. Abate, C. David, P. Kesseli, D. Kroening and E. Polgreen. ‘Counterexample guided inductive synthesis modulo theories’. In: *International Conference on Computer Aided Verification.* Springer. 2018, pp. 270–288.

[94] H. Ravanbakhsh and S. Sankaranarayanan. ‘Learning control lyapunov functions from counterexamples and demonstrations’. In: *Autonomous Robots* 43.2 (2019), pp. 275–307.

[95] D. Ahmed, A. Peruffo and A. Abate. ‘Automated and sound synthesis of Lyapunov functions with SMT solvers’. In: *Tools and Algorithms for the Construction and Analysis of Systems: 26th International Conference, TACAS 2020, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2020, Dublin, Ireland, April 25–30, 2020, Proceedings, Part I 26.* Springer. 2020, pp. 97–114.

[96] M. Doyle, T. F. Fuller and J. Newman. ‘Modeling of galvanostatic charge and discharge of the lithium/polymer/insertion cell’. In: *Journal of the Electrochemical society* 140.6 (1993), p. 1526.

[97] F. Brosa Planella, W. Ai, A. M. Boyce, A. Ghosh, I. Korotkin, S. Sahu, V. Sulzer, R. Timms, T. G. Tranter, M. Zyskin *et al.* ‘A continuum of physics-based lithium-ion battery models reviewed’. In: *Progress in Energy* 4.4 (2022), p. 042003.

[98] C.-H. Chen, F. B. Planella, K. O’regan, D. Gastol, W. D. Widanage and E. Kendrick. ‘Development of experimental techniques for parameterization of multi-scale lithium-ion battery models’. In: *Journal of The Electrochemical Society* 167.8 (2020), p. 080534.

[99] A. Nyman, M. Behm and G. Lindbergh. ‘Electrochemical characterisation and modelling of the mass transport phenomena in LiPF6–EC–EMC electrolyte’. In: *Electrochimica Acta* 53.22 (2008), pp. 6356–6365.

[100] K. O’Regan, F. B. Planella, W. D. Widanage and E. Kendrick. ‘Thermal-electrochemical parameters of a high energy lithium-ion cylindrical battery’. In: *Electrochimica Acta* 425 (2022), p. 140700.

[101] P. Ramadass, B. Haran, P. M. Gomadam, R. White and B. N. Popov. ‘Development of first principles capacity fade model for Li-ion cells’. In: *Journal of the Electrochemical Society* 151.2 (2004), A196.

[102] V. Sulzer, S. G. Marquis, R. Timms, M. Robinson and S. J. Chapman. ‘Python battery mathematical modelling (PyBaMM)’. In: *Journal of Open Research Software* 9.1 (2021).

[103] T. Haarnoja, A. Zhou, P. Abbeel and S. Levine. ‘Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor’. In: *International conference on machine learning.* Pmlr. 2018, pp. 1861–1870.

[104] D. Xu and Y. Tian. ‘A comprehensive survey of clustering algorithms’. In: *Annals of data science* 2 (2015), pp. 165–193.

[105] D. H. P. C. C. (DHPC). *DelftBlue Supercomputer (Phase 1)*. https://www.tudelft.nl/dhpc/ark:/44463/DelftBluePhase1. 2022.

[106] P. Karanth, M. Weijers, P. Ombrini, D. Ripepi, F. Ooms and F. M. Mulder. ‘A phase inversion strategy for low-tortuosity and ultrahigh-mass-loading nickel-rich layered oxide electrodes’. In: *Cell Reports Physical Science* 5.6 (2024).

[107] M. Z. Bazant. ‘Unified quantum theory of electrochemical kinetics by coupled ion–electron transfer’. In: *Faraday Discussions* 246 (2023), pp. 60–124.

[108] P. Bai, D. A. Cogswell and M. Z. Bazant. ‘Suppression of phase separation in LiFePO4 nanoparticles during battery discharge’. In: *Nano letters* 11.11 (2011), pp. 4890–4896.

[109] P. Ombrini, Q. Wang, A. Vasileiadis, F. Wu, Z. Gao, X. Hu, M. van Hulzen, B. Li, C. Zhao and M. Wagemaker. ‘Kinetically induced memory effect in Li-ion batteries’. In: *EES Batteries* (2025).

[110] P. Ombrini, M. Z. Bazant, M. Wagemaker and A. Vasileiadis. ‘Thermodynamics of multi-sublattice battery active materials: from an extended regular solution theory to a phase-field model of LiMnyFe1-yPO4’. In: *npj Computational Materials* 9.1 (2023), p. 148.

[111] A. Karger, S. E. O’Kane, M. Rogge, C. Kirst, J. P. Singer, M. Marinescu, G. J. Offer and A. Jossen. ‘Modeling particle versus SEI cracking in lithium-ion battery degradation: Why calendar and cycle aging cannot simply be added’. In: *Journal of The Electrochemical Society* 171.9 (2024), p. 090512.

[112] V. Sulzer, S. G. Marquis, R. Timms, M. Robinson and S. J. Chapman. ‘Python Battery Mathematical Modelling (PyBaMM)’. In: *Journal of Open Research Software* 9.1 (2021).

[113] V. Vovk. ‘Conditional validity of inductive conformal predictors’. In: *Asian conference on machine learning*. PMLR. 2012, pp. 475–490.

[114] V. K. Chilakamarri, Z. Feng and S. Bansal. ‘Reachability Analysis for Black-Box Dynamical Systems’. In: *arXiv preprint arXiv:2410.07796* (2024).

[115] A. Lin and S. Bansal. ‘Verification of neural reachable tubes via scenario optimization and conformal prediction’. In: *6th Annual Learning for Dynamics & Control Conference*. PMLR. 2024, pp. 719–731.

[116] J. A. Vincent, A. O. Feldman and M. Schwager. ‘Guarantees on robot system performance using stochastic simulation rollouts’. In: *IEEE Transactions on Robotics* (2024).

[117] N. Dean and M. Pagano. ‘Evaluating confidence interval methods for binomial proportions in clustered surveys’. In: *Journal of Survey Statistics and Methodology* 3.4 (2015), pp. 484–503.

[118] C. J. Clopper and E. S. Pearson. ‘The use of confidence or fiducial limits illustrated in the case of the binomial’. In: *Biometrika* 26.4 (1934), pp. 404–413.

[119] G. Shafer and V. Vovk. ‘A tutorial on conformal prediction.’ In: *Journal of Machine Learning Research* 9.3 (2008).

[120] A. N. Angelopoulos, S. Bates *et al.* ‘Conformal prediction: A gentle introduction’. In: *Foundations and Trends® in Machine Learning* 16.4 (2023), pp. 494–591.

[121] M. Fontana, G. Zeni and S. Vantini. ‘Conformal prediction: a unified review of theory and new challenges’. In: *Bernoulli* 29.1 (2023), pp. 1–23.

[122] M. C. Campi and S. Garatti. ‘A sampling-and-discarding approach to chance-constrained optimization: feasibility and optimality’. In: *Journal of optimization theory and applications* 148.2 (2011), pp. 257–280.

[123] L. Romao, A. Papachristodoulou and K. Margellos. ‘On the exact feasibility of convex scenario programs with discarded constraints’. In: *IEEE Transactions on Automatic Control* 68.4 (2022), pp. 1986–2001.

[124] M. Kwiatkowska, G. Norman and D. Parker. ‘PRISM 4.0: Verification of probabilistic real-time systems’. In: *Computer Aided Verification: 23rd International Conference, CAV 2011, Snowbird, UT, USA, July 14-20, 2011. Proceedings 23.* Springer. 2011, pp. 585–591.

[125] G. J. Pappas. ‘Bisimilar linear systems’. In: *Autom.* 39.12 (2003), pp. 2035–2047.

[126] A. J. van der Schaft. ‘Bisimulation of Dynamical Systems’. In: *HSCC.* Vol. 2993. Lecture Notes in Computer Science. Springer, 2004, pp. 555–569.

[127] A. van der Schaft. ‘Equivalence of dynamical systems by bisimulation’. In: *IEEE Trans. Autom. Control.* 49.12 (2004), pp. 2160–2172.

[128] A. Weber, M. Rungger and G. Reissig. ‘Optimized state space grids for abstractions’. In: *IEEE Transactions on Automatic Control* 62.11 (2016), pp. 5816–5821.

[129] J. Calbert, L. N. Egidio and R. M. Jungers. ‘Smart abstraction based on iterative cover and non-uniform cells’. In: *IEEE Control Systems Letters* 8 (2024), pp. 2301–2306.

[130] J. Calbert, A. Banse, B. Legat and R. M. Jungers. ‘Dionysos. jl: a modular platform for smart symbolic control’. In: *arXiv preprint arXiv:2404.14114* (2024).

[131] A. Abate, M. Giacobbe, C. Micheletti and Y. Schnitzer. ‘Branching Bisimulation Learning’. In: *CAV (4).* Vol. 15934. Lecture Notes in Computer Science. Springer, 2025, pp. 161–184.

[132] D. N. Tran, B. S. Rüffer and C. M. Kellett. ‘Incremental stability properties for discrete-time systems’. In: *2016 IEEE 55th Conference on Decision and Control (CDC).* IEEE. 2016, pp. 477–482.

[133] H. Ebbinghaus and J. Flum. *Finite model theory.* Perspectives in Mathematical Logic. Springer, 1995.

[134] P. Eirinakis, S. Ruggieri, K. Subramani and P. Wojciechowski. 'On quantified linear implications'. In: *Annals of Mathematics and Artificial Intelligence* 71.4 (2014), pp. 301–325.

[135] S. P. Lloyd. 'Least squares quantization in PCM'. In: *IEEE Trans. Inf. Theory* 28.2 (1982), pp. 129–136.

[136] L. De Moura and N. Bjørner. 'Z3: An efficient SMT solver'. In: *International conference on Tools and Algorithms for the Construction and Analysis of Systems.* Springer. 2008, pp. 337–340.

[137] M. Lazar. 'On infinity norms as Lyapunov functions: Alternative necessary and sufficient conditions'. In: *49th IEEE Conference on Decision and Control (CDC).* IEEE. 2010, pp. 5936–5942.

[138] G. Valiant and P. Valiant. 'Estimating the unseen: an n/log (n)-sample estimator for entropy and support size, shown optimal via new CLTs'. In: *Proceedings of the forty-third annual ACM symposium on Theory of computing.* 2011, pp. 685–694.

[139] N. O'Sullivan, L. Romao and K. Margellos. 'Bridging conformal prediction and scenario optimization'. In: *arXiv preprint arXiv:2503.23561* (2025).

[140] J. Hawkins. *Ergodic Dynamics.* Springer, 2021.

[141] R. Coppola, A. Peruffo, L. Lindemann and M. Mazo Jr. 'Scenario approach and conformal prediction for verification of unknown systems via data-driven abstractions'. In: *2024 European Control Conference (ECC).* IEEE. 2024, pp. 558–563.

[142] R. Coppola, S. Ahmed and M.-S. Alouini. 'Road users classification based on bi-frame micro-Doppler with 24-GHz FMCW radar'. In: *Frontiers in Signal Processing* 2 (2022), p. 864538.

Acknowledgements

I would like to thank the committee members for taking the time to carefully review my thesis and for their valuable feedback.

I am deeply grateful to my supervisor, Manuel Mazo Jr. From the very beginning, Manuel has shown me kindness, attention, and patience, guiding me as I learned what it truly means to be a researcher. He helped me shape and develop my academic work, and working closely with him allowed me to appreciate not only his scientific rigor, but also his integrity and his approach to doing science. Beyond his invaluable mentorship, I greatly appreciated the freedom he consistently entrusted to me, our enjoyable and stimulating conversations, and, above all, the care, respect, and friendship he has always shown.

A special thanks goes to Andrea Peruffo, with whom I worked side by side for most of my PhD. Andrea taught me a great deal about how to turn ideas into results and how to “get things done”. I truly enjoyed the collaborative dynamic we built together, the many laughs we shared during our discussions, and the adventures we experienced while travelling for work.

I am also deeply indebted to Gabriel Gleizer and Giannis Delimpaltadakis. Their passion for mathematics has profoundly influenced me and has played a significant role in shaping me as a researcher. Beyond the academic impact, the time we spent together left a lasting mark on the way I think, both scientifically and personally, and strongly influenced my broader perspective on life.

I would also like to express my sincere gratitude to Alessandro Abate for our fruitful collaborations and for his generous hospitality. I am thankful to him for welcoming me into his research group at the University of Oxford and for hosting me during my research stay.

I would also like to thank my friend and collaborator, Pierfrancesco Ombrini. I have met very few people with such a strong drive for research, and our exchanges have been both motivating and intellectually enriching. I am equally grateful to Hovsep Touloujian for his excellent work, dedication, and professionalism, which made the supervision experience smooth and highly productive.

Finally, I would like to thank my parents, Giuseppe and Tanja, and my sister, Lara, as well as all my friends. Capturing all the subtle ways in which they have shaped and influenced the person I have become is difficult, but it is clear that their constant support, patience, and encouragement have been fundamental. Without them, I would not be where I am today.

Curriculum Vitæ

Rudi Coppola

22-11-1997	Born in Poggibonsi, Italy.

Education

2016–2019	BSc in Electronics Engineering University of Pisa, Pisa, IT	
	Grade:	110/110 cum laude
2019–2021	MSc in Electrical Engineering King Abdullah University of Science and Technology, Thuwal, SA	
	Grade:	4.92/5.00
	Thesis:	Road Users Classification Based on Bi-Frame Micro-Doppler With 24-GHz FMCW Radar
	Supervisor:	Mohamed-Slim Alouini
2021-2025	PhD in Systems and Control Delft University of Technology	
	Thesis:	Abstraction Learning with Guarantees: Data-driven Approaches to Symbolic Control and Verification
	Promotor:	Manuel Mazo Espinoza
	Co-promotor:	Luca Laurenti

Awards and Certificates

2021	First prize winner - IEEE Radar Applications Challenge: IEEE AESS Radar Conference 2021, Atlanta.
2021-2024	DISC Certificate for Graduate Studies, Dutch Institute of Systems and Control.

List of Publications

1. R. Coppola, Y. Schnitzer, M. Giacobbe, A. Abate and M. Mazo Jr. *Existence and Synthesis of Multi-Resolution Approximate Bisimulations for Continuous-State Dynamical Systems*. 2025. arXiv: 2509.17739 [eess.SY]. URL: https://arxiv.org/abs/2509.17739

2. R. Coppola, H. Touloujian, P. Ombrini and M. Mazo Jr. ‘Reinforcement Learning for Robust Ageing-Aware Control of Li-ion Battery Systems with Data-Driven Formal Verification’. In: *arXiv preprint arXiv:2509.04288* (2025)

3. R. Coppola and M. Mazo Jr. ‘On Training-Conditional Conformal Prediction and Binomial Proportion Confidence Intervals’. In: *Transactions on Machine Learning Research* (2025). ISSN: 2835-8856. URL: https://openreview.net/forum?id=pSk5qyt1ob

4. R. Coppola, A. Peruffo, L. Lindemann and M. Mazo Jr. ‘Scenario approach and conformal prediction for verification of unknown systems via data-driven abstractions’. In: *2024 European Control Conference (ECC)*. IEEE. 2024, pp. 558–563

5. R. Coppola, A. Peruffo and M. Mazo Jr. ‘Data-Driven Abstractions for Control Systems via Random Exploration’. In: *arXiv preprint arXiv:2402.10668* (2024)

6. R. Coppola, A. Peruffo, L. Romao, A. Abate and M. Mazo Jr. ‘Enhancing Data-Driven Stochastic Control via Bundled Interval MDP’. in: *IEEE Control Systems Letters* 8 (2024), pp. 2069–2074

7. R. Coppola, A. Peruffo and M. Mazo Jr. ‘Data-driven Abstractions for Verification of Linear Systems’. In: *IEEE Control Systems Letters* (2023)

8. R. Coppola, A. Peruffo and M. Mazo Jr. ‘Data-driven Abstractions for Verification of Deterministic Systems’. In: *arXiv preprint arXiv:2211.01793* (2022)

9. R. Coppola, S. Ahmed and M.-S. Alouini. ‘Road users classification based on bi-frame micro-Doppler with 24-GHz FMCW radar’. In: *Frontiers in Signal Processing* 2 (2022), p. 864538