Proceedings of Machine Learning Research vol 283:1–15, 2025 7th Annual Conference on Learning for Dynamics and Control

Data-Driven Yet Formal Policy Synthesis for Stochastic Nonlinear Dynamical Systems

Mahdi Nazeri[1,2] MAHDI.NAZERI@CS.OX.AC.UK
Thom Badings[1] THOM.BADINGS@CS.OX.AC.UK
Sadegh Soudjani[2] SADEGH@MPI-SWS.ORG
Alessandro Abate[1] ALESSANDRO.ABATE@CS.OX.AC.UK
[1]*Department of Computer Science, University of Oxford, Oxford, United Kingdom*
[2]*Max Planck Institute for Software Systems, Kaiserslautern, Germany*

Editors: N. Ozay, L. Balzano, D. Panagou, A. Abate

Abstract

The automated synthesis of control policies for stochastic dynamical systems presents significant challenges. A standard approach is to construct a finite-state abstraction of the continuous system, typically represented as a Markov decision process (MDP). However, generating abstractions is challenging when (1) the system's dynamics are nonlinear, and/or (2) we do not have complete knowledge of the dynamics. In this work, we introduce a novel data-driven abstraction technique for nonlinear Lipschitz continuous dynamical systems with additive stochastic noise that addresses both of these issues. As a key step, we use samples of the dynamics to learn the enabled actions and transition probabilities of the abstraction. We represent abstractions as MDPs with intervals of transition probabilities, known as interval MDPs (IMDPs). These abstractions enable the synthesis of policies for the concrete nonlinear system, with probably approximately correct (PAC) guarantees on the probability of satisfying a specified control objective. Our numerical experiments illustrate the effectiveness and robustness of our approach in achieving reliable control under uncertainty.

Keywords: Data-driven abstraction, Nonlinear dynamical systems, Stochastic systems, Formal controller synthesis, Markov decision processes

1. Introduction

Formal policy synthesis is an area of control theory focusing on designing controllers that provably meet specific requirements (Belta et al., 2017). One such requirement is the *(stochastic) reach-avoid task*: Compute a (control) policy such that, with at least a specified probability, the system reaches a set of goal states while avoiding unsafe states (Fan et al., 2018; Summers and Lygeros, 2010). The state-of-the-art in policy synthesis for stochastic systems is, arguably, to abstract the system into a finite-state model that appropriately captures its behaviour (Lavaei et al., 2022; Abate et al., 2008; Tabuada, 2009). However, conventional abstractions often rely on precise and explicit representations of the system's dynamics, which are unavailable in many cases.

Fuelled by increasing data availability and advances in machine learning, *data-driven abstractions* have emerged as an alternative to conventional model-based abstractions (Makdesi et al., 2021; Coppola et al., 2023; Lavaei et al., 2023; Kazemi et al., 2022; Hashimoto et al., 2022; Devonport et al., 2021; Banse et al., 2023; Peruffo and Mazo, 2023; Schön et al., 2024). By incorporating techniques from formal verification (Baier and Katoen, 2008), temporal logic (Pnueli, 1977), and reachability analysis (Althoff et al., 2021), data-driven abstractions can be used to synthesise policies despite incomplete knowledge of the system dynamics. However, with only a few recent exceptions (Gracia et al., 2024a,b; Jackson et al., 2021), these data-driven abstractions apply to nonstochastic systems only, thus leaving an important gap in the literature.

In this paper, we study discrete-time dynamical systems whose dynamics are composed of a deterministic nonlinear term and an additive stochastic noise term. Following a data-driven paradigm, we assume only (black-box) sampling access to the stochastic noise. For the nonlinear term, we require sampling access plus partial knowledge in the form of knowing the Lipschitz constant. Given such a system and a reach-avoid task, we focus on the following problem: *Compute a policy such that the reach-avoid task is satisfied with at least a specific threshold probability* $\rho \in [0, 1]$.

We address this problem by abstracting the system into a finite-state Markov decision process (MDP) (Puterman, 1994). Inspired by Badings et al. (2023a), we define the abstract actions via *backward reachability computations* on the dynamical system. However, the approach from Badings et al. (2023a) only applies to systems with linear dynamics and leads to overly conservative abstractions. To overcome these limitations, we introduce two data-driven aspects in our approach:

1. **Data-driven backward reachability analysis:** Performing backward reachability computations on nonlinear systems is generally challenging (Mitchell, 2007; Rober et al., 2022). We develop a novel data-driven method to *underapproximate* backward reachable sets based on forward simulations of the dynamical system. Our method only requires differentiability of the nonlinear dynamics and leads to sound underapproximations of backward reachable sets.
2. **Data-driven probability intervals:** We use statistical techniques to compute *probably approximately correct* (PAC) intervals of transition probabilities, which we capture in an *interval MDP* (IMDP) (Givan et al., 2000; Nilim and Ghaoui, 2005). While Badings et al. (2023a) also uses sampling techniques (using the scenario approach (Campi et al., 2021; Romao et al., 2023)), their intervals are very loose. We instead use the classical Clopper-Pearson confidence interval (Clopper and Pearson, 1934), yielding tighter intervals (Meggendorfer et al., 2024).

In summary, our main contribution is a novel data-driven IMDP abstraction technique for nonlinear stochastic systems with incomplete knowledge of the dynamics. Due to the PAC guarantee on each individual probability interval of the IMDP, we can use our abstraction to synthesise policies with PAC reach-avoid guarantees. We showcase our abstraction technique on multiple benchmarks.

Related work. Control of nonlinear systems against temporal tasks is an active research area (Khalil and Grizzle, 2002; Belta et al., 2017). Since computing optimal policies is generally infeasible (Bertsekas and Shreve, 1978), many model-based abstraction techniques have been developed, often representing abstractions as (I)MDPs (Soudjani and Abate, 2013; Lahijanian et al., 2015; Soudjani et al., 2015; van Huijgevoort et al., 2023; Mathiesen et al., 2024; Delimpaltadakis et al., 2023). Particularly related here is Gracia et al. (2024a), who generate data-driven abstractions of switched stochastic systems into robust MDPs, by estimating the (unknown) noise distribution as a Wasserstein ball. However, Gracia et al. (2024a) resorts to a model-based approach to abstract the deterministic part of the dynamics, while we use sampling instead. Also closely related are Badings et al. (2023b,a), who, however, require the (deterministic) dynamics to be linear and known.

Existing methods over/underapproximate backward reachable sets by level set functions (Yin et al., 2019; Stipanovic et al., 2003), approximating operators on, e.g., zonotopes Yang et al. (2022), piecewise affine bounding of the dynamics (Rober et al., 2022), and Hamilton-Jacobi reachability analysis (Bansal et al., 2017). However, most approaches are computationally expensive and are model-based, whereas we focus on data-driven techniques to obtain sound underapproximations.

Apart from abstraction, others recently studied control of stochastic systems using Lyapunov-like functions learned represented as neural networks (Mathiesen et al., 2023; Abate et al., 2024; Zikelic et al., 2023), or using robust and scenario optimization (Salamati et al., 2024; Nejati et al., 2023).

2. Problem Formulation

A probability space $(\Omega, \mathcal{F}, \mathbb{P})$ consists of an uncertainty space Ω, a σ-algebra $\mathcal{F}$, and a probability measure $\mathbb{P} \colon \mathcal{F} \to [0, 1]$. A random variable z is a measurable function $z \colon \Omega \to \mathbb{R}^n$ for some $n \in \mathbb{N}$. The set of all distributions for a (continuous or discrete) set X is $\Delta(X)$. The Cartesian product of an interval is $[a, b]^n$, for $a \leq b$, $n \in \mathbb{N}$. The element-wise absolute value of $x \in \mathbb{R}^n$ is written as $|x|$.

Stochastic systems. Consider a discrete-time nonlinear system $\mathcal{S}$ with additive stochastic noise:

$$\mathcal{S}: \; x_{k+1} = f(x_k, u_k) + w_k, \quad x_0 = x_I, \tag{1}$$

where $x_k \in \mathbb{R}^n$ and $u_k \in \mathcal{U} \subset \mathbb{R}^p$ are the state and control input at discrete time step $k \in \mathbb{N}$, where $\mathcal{U} \subset \mathbb{R}^p$ is compact. The (deterministic) dynamics function $f \colon \mathbb{R}^n \times \mathcal{U} \to \mathbb{R}^n$ is also called *nominal dynamics*, and $x_I \in \mathbb{R}^n$ is the initial state. Moreover, $w_0, w_1, \ldots$ is a sequence of independent and identically distributed (i.i.d.) random variables, defined on the same probability space $(\Omega, \mathcal{F}, \mathbb{P})$.[1]

Assumption 1 *The transition function f is differentiable with bounded first-order partial derivatives, and the measure $\mathbb{P}$ is absolutely continuous w.r.t. the Lebesgue measure. However, $\mathbb{P}$ itself is unknown.*

Thus, while f and $\mathbb{P}$ can be unknown, our method requires: (1) the noise w_k being additive, (2) knowledge of, e.g., the Lipschitz constant of f, and (3) independent sampling access to f and w_k.

The inputs $u_k \in \mathbb{R}^n$ are chosen by a *(Markovian) policy* $\pi := (\pi_0, \pi_1, \pi_2, \ldots)$, where each $\pi_k \colon \mathbb{R}^n \to \mathcal{U}$, $k \in \mathbb{N}$, is a measurable map from states to inputs. We denote the set of all policies by $\Pi^{\mathcal{S}}$. Fixing a policy π defines a Markov process in the probability space of all trajectories (Bertsekas and Shreve, 1978; Puterman, 1994), whose probability measure we denote by $\mathbb{P}_\pi^{\mathcal{S}}$.

Given a policy π, we are interested in the probability of reaching a *goal set* $X_G \subseteq \mathbb{R}^n$ within $h \in \mathbb{N} \cup \{\infty\}$ steps, while never reaching an *unsafe set* $X_U \subseteq \mathbb{R}^n$.[2] We call the triple (X_G, X_U, h) a *reach-avoid specification*. The *reach-avoid probability* $\mathrm{Pr}_\pi^{\mathcal{S}}(X_G, X_U, h)$ for this specification is

$$\mathrm{Pr}_\pi^{\mathcal{S}}(X_G, X_U, h) := \mathbb{P}_\pi^{\mathcal{S}}\big\{\exists k \in \{0, \ldots, h\} \;:\; x_k \in X_G \wedge (\forall k' \in \{0, \ldots, k\} : x_{k'} \notin X_U)\big\}. \tag{2}$$

We now have all the ingredients to formalise the problem that we wish to solve:

Problem 1 *Suppose we are given a dynamical system $\mathcal{S}$, a reach-avoid specification (X_G, X_U, h), and a threshold probability $\rho \geq 0$. Compute a policy $\pi \in \Pi^{\mathcal{S}}$ such that $\mathrm{Pr}_\pi^{\mathcal{S}}(X_G, X_U, h) \geq \rho$.*

Interval MDPs. We will abstract system $\mathcal{S}$ into an MDP with intervals of transition probabilities, known as an interval MDP (IMDP). For an introduction to IMDPs, we refer to Suilen et al. (2025).

Definition 1 (IMDP) *An* interval MDP *(IMDP) $\mathcal{M}_{\mathbb{I}}$ is a tuple $\mathcal{M}_{\mathbb{I}} := (S, Act, s_I, \mathcal{P})$, where S is a finite set of states, $s_I \in S$ is the initial state, Act is a finite set of actions, with $Act(s) \subseteq Act$ the actions enabled in state $s \in S$, and $\mathcal{P} \colon S \times Act \rightharpoonup 2^{\Delta(S)}$ is a transition function[3] defined for all $s \in S, a \in Act(s)$ as $\mathcal{P}(s, a) = \big\{\mu \in \Delta(S) : \forall s' \in S, \; \mu(s') \in [\check{p}(s, a, s'), \hat{p}(s, a, s')] \subset [0, 1]\big\}$.*

Without loss of generality, we assume that $\mathcal{P}(s, a) \neq \emptyset$ for all $s \in S$, $a \in Act(s)$. We also call $[\check{p}(s, a, s'), \hat{p}(s, a, s')] \subseteq [0, 1]$ the *probability interval* for transition (s, a, s'). Actions in an IMDP are chosen by a (Markovian) *scheduler*[4] $\sigma = (\sigma_0, \sigma_1, \ldots)$, where each $\sigma_k \colon S \to Act$ is defined such that $\sigma_k(s) = a \implies a \in Act(a)$. The set of all Markov schedulers for $\mathcal{M}_{\mathbb{I}}$ is denoted by $\mathfrak{S}^{\mathcal{M}_{\mathbb{I}}}$.

1. For brevity, we assume Ω is a subset of $\mathbb{R}^n$ and directly write $w \in \Omega$ to say that the random variable takes a value.
2. Formally, X_G and X_U must be Borel-measurable (Salamon, 2016), but we glance over measurability details here.
3. To model that not all actions may be enabled in a state, the transition function $\mathcal{P}$ is a partial map, denoted by $\rightharpoonup$.
4. For clarity, we use the word *scheduler* for (finite) IMDPs, whereas we use *policy* for (continuous) dynamical systems.

An IMDP can be interpreted as a game between a scheduler that chooses actions and an *adversary* that fixes distributions $P(s, a) \in \mathcal{P}(s, a)$ for all $s \in S$, $a \in Act(s)$. We assume a different probability can be chosen every time the same pair (s, a) is encountered (called the *dynamic* uncertainty model (Iyengar, 2005)). We overload notation and write $P \in \mathcal{P}$ for fixing an adversary.

Fixing $\sigma \in \mathfrak{S}^{\mathcal{M}_\mathbb{I}}$ and $P \in \mathcal{P}$ for $\mathcal{M}_\mathbb{I}$ yields a Markov chain with (standard) probability measure $\mathbb{P}_{\sigma,P}^{\mathcal{M}_\mathbb{I}}$ (Baier and Katoen, 2008). A reach-avoid specification for $\mathcal{M}_\mathbb{I}$ is a tuple (S_G, S_U, h) of goal and unsafe states $S_G, S_U \subseteq S$ and a horizon $h \in \mathbb{N} \cup \{\infty\}$ (we use this notation later in Sect. 3.1). The probability of satisfying this specification is written as $\mathrm{Pr}_{\sigma,P}^{\mathcal{M}_\mathbb{I}}(S_G, S_U, h)$, and is defined based on $\mathbb{P}_{\sigma,P}^{\mathcal{M}_\mathbb{I}}$ analogously to Eq. (2). An optimal (robust) scheduler $\sigma^\star \in \mathfrak{S}^{\mathcal{M}_\mathbb{I}}$ is defined as

$$\sigma^\star \in \arg\max_{\sigma \in \mathfrak{S}^{\mathcal{M}_\mathbb{I}}} \min_{P \in \mathcal{P}} \mathrm{Pr}_{\sigma,P}^{\mathcal{M}_\mathbb{I}}(S_G, S_U, h). \tag{3}$$

In practice, $\sigma^\star$ can be computed using, e.g., robust value iteration (Wolff et al., 2012; Iyengar, 2005).

3. Finite-State IMDP Abstraction

We present an abstraction of the system $\mathcal{S}$ into a finite IMDP. Our approach extends Badings et al. (2023a) from *linear* to *nonlinear* systems, which has fundamental consequences for the practical computability of the abstraction. For clarity, we first define the IMDP's states, actions, and transition function in this section, while we defer our novel contributions to compute these to Sects. 4 and 5.

Partition. Let $\mathcal{X} \subset \mathbb{R}^n$ be a compact subset of the state space we want to capture by the abstraction. We create a partition[5] of $\mathcal{X}$ into $v \in \mathbb{N}$ convex polytopes $\{R_1, R_2, \dots, R_v\}$, such that each region is defined as $R_i = \{x \in \mathbb{R}^n : M_i x \leq b_i\}$, where $M_i \in \mathbb{R}^{\xi_i \times n}$, $b_i \in \mathbb{R}^{xi_i}$, $\xi_i \in \mathbb{N}$. We append one element $R_\star = \mathbb{R}^n \setminus \mathcal{X}$ to the partition, called the *absorbing region*, which represents all states outside of $\mathcal{X}$. Thus, the collection $\{R_1, R_2, \dots, R_v\} \cup \{R_\star\}$ covers the entire state space.

Definition 2 (Scaled polytope) *Let $d_i \in R_i$ be the centre*[6] *point of the region R_i. The convex polytope $R_i(\lambda) \subset \mathbb{R}^n$ is defined as the version of R_i scaled around d_i by a factor of $\lambda \geq 0$, i.e., $R_i(\lambda) = \{x \in \mathbb{R}^n \mid M_i x \leq \lambda(b_i - M_i d_i) + M_i d_i\}$.*

Thus, for $\lambda = 1$, the scaled region is the same, i.e., $R_i(1, d) = R_i$. Similarly, for any $\lambda < 1$, we have $R_i(\lambda, d_i) \subseteq R_i$. We will use scaled regions later to define the abstract actions.

3.1. Abstract IMDP definition

States. We define one IMDP state s_i for each element R_i, plus one *absorbing state* for $R_\star$, resulting in $S \coloneqq \{s_1, \dots, s_v, s_\star\}$. An *abstraction function* maps between continuous and abstract states:

Definition 3 (Abstraction function) *The* abstraction function $\mathcal{T} \colon \mathbb{R}^n \to S$ *is defined as $\mathcal{T}(x) = s_i$ if $x \in R_i$.*[7] *We also define the preimage of s_i under $\mathcal{T}$ as $\mathcal{T}^{-1}(s_i) = R_i$ for all $i = 1, \dots, v$.*

In other words, the IMDP state s_i represents all continuous states $x \in R_i$. The initial IMDP state is then defined as $s_I \coloneqq \mathcal{T}(x_I)$. We map the reach-avoid specification (X_G, X_U, h) to the abstract IMDP by under- and overapproximating the goal and unsafe states, respectively, as $S_G \coloneqq \{s \in S : \forall x \in \mathcal{T}^{-1}(s),\ x \in X_G\}$, and $S_U \coloneqq \{s \in S : \exists x \in \mathcal{T}^{-1}(s),\ x \in X_U\}$.

5. The sets $\{R_1, \dots, R_v\}$ form a partition of $\mathcal{X}$ if their union covers $\mathcal{X}$ and the interiors of all elements R_i are disjoint.
6. In fact, we can choose any $d_i \in R_i$, but the centre is often convenient in practice.
7. If x is on the boundary of two regions R_i, R_j, we arbitrarily choose $\mathcal{T}(x) = s_i$ or $\mathcal{T}(x) = s_i$. However, Assumption 1 implies that this occurs with probability zero, so this arbitrary choice does not affect the correctness of our algorithm.

Actions. Each IMDP action does not represent a single input $u_k \in \mathcal{U}$ (as is typically done in abstraction) but *a collection* of inputs leading to a *common state* x_{k+1}. Without loss of generality, we define one action for each IMDP state (except $s_\star$), such that $Act \coloneqq \{a_1, \dots, a_v\}$. We then associate every pair $(s_i, a_j) \in S \times Act$ with a so-called *target set* $R_j(\lambda_{i\to j})$ that represents region R_j scaled by a factor $\lambda_{i\to j} \geq 0$; see Definition 2. We discuss how we compute each factor $\lambda_{i\to j}$ in Sect. 4.

Suppose the system's state is $x \in \mathbb{R}^n$, associated with IMDP state $\mathcal{T}(x) = s_i$. Choosing action $a_j \in Act$ in s_i corresponds to choosing an input $u \in \mathcal{U}$ such that $f(x,u) \in R_j(\lambda_{i\to j})$. Thus, this IMDP action a_j must only be enabled in the state s_i if for all $x \in \mathcal{T}^{-1}(s_i)$, there exists such an input $u \in \mathcal{U}$ for which $f(x,u)$ is contained in the target set $R_j(\lambda_{i\to j})$ of the state-action pair (s_i, a_j). To formalise this requirement, let $\mathsf{Pre}(X)$ be the *backward reachable set* for a set $X \subset \mathbb{R}^n$:

$$\mathsf{Pre}(X) = \left\{x' \in \mathbb{R}^n \mid \exists u \in \mathcal{U} : f(x', u) \in X\right\}. \tag{4}$$

Then, for every state $s_i \in S$, the set of enabled actions $Act(s_i)$ is defined as

$$Act(s_i) = \{a_j \in Act : \exists \lambda_{i\to j} \in [0, \Lambda],\ \mathcal{T}^{-1}(s_i) \subseteq \mathsf{Pre}(R_j(\lambda_{i\to j}))\}, \tag{5}$$

where $\Lambda \in \mathbb{R}_{\geq 0}$ is a global hyperparameter that prevents $R_j(\lambda_{i\to j})$ from becoming too large.

Computing $\mathsf{Pre}(\cdot)$ exactly is challenging for nonlinear dynamics (Mitchell, 2007; Rober et al., 2022). However, any *underapproximation* preserves correction of our abstraction (albeit increasing conservatism). In Sect. 4, we present a data-driven method to compute such underapproximations.

Transition function. Due to the stochastic noise in system $\mathcal{S}$, choosing the IMDP action a_j in IMDP state $s_i \in S$ leads to the continuous successor state $f(x,u) + w \in \mathbb{R}^n$, where $f(x,u) \in R_j(\lambda_{i\to j})$. That is, for every possible $\hat{x} \in R_j(\lambda_{i\to j})$, we obtain a *different probability distribution* over the continuous successor state $\hat{x} + w$. Mathematically, let $\eta(\hat{x}, \bar{X}) \in [0,1]$ denote the probability that $\hat{x} + w$ is contained in a compact set $\bar{X} \subset \mathbb{R}^n$, i.e. $\eta(\hat{x}, \bar{X}) = \mathbb{P}\left\{w \in \Omega : \hat{x} + w \in \bar{X}\right\}$. Then, the IMDP transition function $\mathcal{P}$ is defined for all (s_i, a_j) by taking the min/max over η as follows:

$$\mathcal{P}(s_i, a_j) = \Big\{\mu \in \Delta(S) : \forall s' \in S,\ \min_{\hat{x} \in R_j(\lambda_{i\to j})} \eta\big(\hat{x}, \mathcal{T}^{-1}(s')\big) \leq \mu(s') \leq \max_{\hat{x} \in R_j(\lambda_{i\to j})} \eta\big(\hat{x}, \mathcal{T}^{-1}(s')\big)\Big\}. \tag{6}$$

In Sect. 5, we will compute these bounds using samples of the noise $w \in \Omega$.

3.2. Correctness of the abstraction

In Sect. 3.1, we defined an IMDP abstraction $\mathcal{M}_{\mathbb{I}} = (S, Act, s_I, \mathcal{P})$ of system $\mathcal{S}$. As is common in abstraction-based control, any IMDP scheduler $\sigma \in \mathfrak{S}^{\mathcal{M}_{\mathbb{I}}}$ can be *refined* into a policy $\pi \in \Pi^{\mathcal{S}}$ for system $\mathcal{S}$. Crucially, the reach-avoid probability for σ on $\mathcal{M}_{\mathbb{I}}$ is a *lower bound* on that for π on $\mathcal{S}$:

Theorem 4 (Policy synthesis (Badings et al., 2023b)) *Let $\mathcal{M}_{\mathbb{I}}$ be the IMDP abstraction for dynamical system $\mathcal{S}$. For every IMDP scheduler $\sigma \in \mathfrak{S}^{\mathcal{M}_{\mathbb{I}}}$, there exists a policy $\pi \in \Pi^{\mathcal{S}}$ for $\mathcal{S}$ such that*

$$\min_{P \in \mathcal{P}} \Pr\nolimits_{\sigma,P}^{\mathcal{M}_{\mathbb{I}}}(S_G, S_U, h) \ \leq\ \Pr\nolimits_{\pi}^{\mathcal{S}}(X_G, X_U, h). \tag{7}$$

As we discuss in Nazeri et al. (2024, App. A), Theorem 4 is based on a probabilistic extension of an *alternating simulation relation* (Alur et al., 1998), The policy $\pi \in \Pi^{\mathcal{S}}$ for which Theorem 4 holds can be derived recursively, by choosing inputs at every step $k \in \mathbb{N}$ such that this relation is preserved. Concretely, the *refined policy* $\pi = (\pi_0, \pi_1, \dots)$ for system $\mathcal{S}$ is defined for all $x \in \mathbb{R}^n$ and $k \in \mathbb{N}$ as

$$\pi_k(x) \in \{u \in \mathcal{U} : f(x,u) \in R_j(\lambda_{i\to j})\}, \tag{8}$$

where $s_i = \mathcal{T}(x)$ and $a_j = \sigma_k(s_i)$ are the current IMDP state and action. In Sect. 4, we discuss how we obtain $\pi_k(x)$ directly from the data-driven underapproximation of $\mathsf{Pre}(R_j(\lambda_{i\to j}))$.

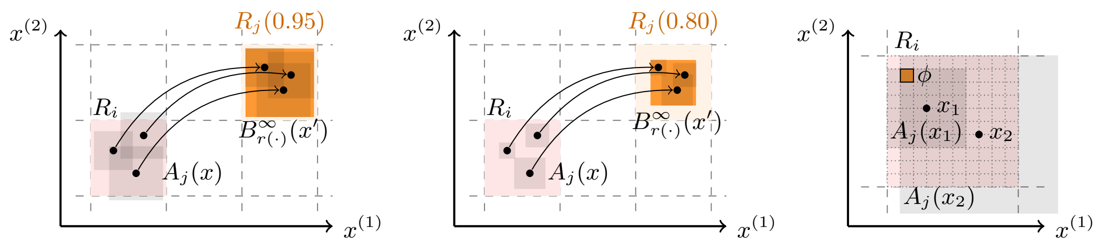


Figure 1: Three samples (x, u, x') with $x' \in R_j(\lambda_{i\to j})$, and the balls $B^{\infty}_{r(\cdot)}(x')$ around each x' and $A_j(x)$ around each x (shown in gray) for fixed values of $\lambda_{i\to j} = 0.95$ (left) and $\lambda_{i\to j} = 0.80$ (middle). On the right, we show sets $A_j(x)$ for two samples such that voxel $\phi \in \Phi(R_i)$ is contained.

4. Data-Driven Underapproximations of Backward Reachable Sets

In this section, we compute the enabled actions $Act(s_i) \subseteq Act$ in each IMDP state $s_i \in S$. Recall from Eq. (5) that action $a_j \in Act$ is enabled in state $s_i \in S$ if $\mathcal{T}^{-1}(s_i) \subseteq \mathsf{Pre}(R_j(\lambda_{i\to j}))$. As a key contribution, we present a data-driven method to compute the scaling factor $\lambda_{i\to j}$ and an underapproximation of $\mathsf{Pre}(R_j(\lambda_{i\to j}))$ for all $s_i \in S$ and $a_j \in Act(s_i)$.

Data collection. The core idea is to underapproximate each set $\mathsf{Pre}(R_j(\lambda_{i\to j}))$, based on forward simulations of the nominal dynamics function $\hat{x}_\ell = f(x_\ell, u_\ell)$. Since we assumed sampling access to f, we can easily obtain such a set of samples. Let us denote the resulting set of $K \in \mathbb{N}$ samples by

$$\mathcal{D}_K = \{(x_\ell, u_\ell, f(x_\ell, u_\ell)) : \ell = 1, \dots, K,\ x_\ell \in \mathcal{X},\ u_\ell \in \mathcal{U}\}\,. \tag{9}$$

Without loss of generality, we assume to obtain these samples by a uniform gridding of $\mathcal{X}$ and $\mathcal{U}$. While more sophisticated approaches may lead to better results, we leave this for future work.

We describe how we use the dataset $\mathcal{D}_K$ to underapproximate $\mathsf{Pre}(R_j(\lambda_{i\to j}))$ for a fixed $s_i \in S$ and $a_j \in Act$. We repeat this procedure for all other state-action pairs to compute all enabled actions.

Fixing $\lambda_{i\to j}$ upfront. Consider a fixed value for $\lambda_{i\to j} > 0$, which fixes the target set $R_j(\lambda_{i\to j})$ that defines the semantics of action a_j. By definition, the x-component of every sample $(x, u, x') \in \mathcal{D}_K$ for which $x' \in R_j(\lambda_{i\to j})$ is contained in $\mathsf{Pre}(R_j(\lambda_{i\to j}))$. Moreover, due to the differentiability of the dynamics (see Assumption 1), there exists a region Y around x such that, for all $y \in Y$, $f(y, u)$ is also contained in $R_j(\lambda_{i\to j})$. Thus, this region Y is also contained in $\mathsf{Pre}(R_j(\lambda_{i\to j}))$.

For a fixed input $\hat{u}$, the Jacobian of $f(x, \hat{u})$ is the matrix $J \in \mathbb{R}^{n\times n}$, whose entries are defined as $J_{pq} = \frac{\partial f(x,\hat{u})_p}{\partial x_q}$, $p, q = 1, \dots, n$. We define $J^+(R_i) \in \mathbb{R}^{n\times n}$ as the matrix whose entries $J^+(R_i)_{pq} \in \mathbb{R}_{\geq 0}$ are defined as the supremum over the absolute value of $J_{pq}(x)$ for all $x \in R_i$, i.e., $J^+(R_i)_{pq} = \sup\{|J_{pq}(x)| : x \in R_i\}$. We use the matrix $J^+(R_i)$ to derive the following theorem.

Theorem 5 *For all $x_1, x_2 \in R_i$ it holds that $|f(x_1, u_\ell) - f(x_2, u_\ell)| \leq J^+(R_i) \cdot |x_1 - x_2|$.*

We prove Theorem 5 in Nazeri et al. (2024, App. B). Note that the maximum element of $J^+(R_i)$ is a local Lipschitz constant of f with respect to changes in x, within the region R_i. Thus, if computing the Jacobian J is difficult, e.g., when f is not known explicitly, we can use an (upper bound on the) Lipschitz constant instead. We use Theorem 5 to underapproximate $\mathsf{Pre}(R_j(\lambda_{i\to j}))$ as follows:

Definition 6 *Let $(x, u, x') \in \mathcal{D}_K$ with $x \in R_i$, $x' \in R_j$. The radius $r_j(x', \lambda_{i\to j})$ of the largest x'-centered L^∞-ball[8] contained in $R_j(\lambda_{i\to j})$ is $r_j(x', \lambda_{i\to j}) \coloneqq \max\{\epsilon \geq 0 : B^\infty_\epsilon(x') \subseteq R_j(\lambda_{i\to j})\}$.*

8. The (open) L^∞-ball $B^\infty_\epsilon(x')$ of size $\epsilon \geq 0$ centered at x' is defined as $B^\infty_\epsilon(x) = \{y \in \mathbb{R}^n : \|x' - y\|_\infty < \epsilon\}$.

Theorem 7 (Underapproximate backward reach. set) *Fix $s_i \in S$, $a_j \in Act$, and $\lambda_{i\to j} \geq 0$. Let $(x, u, x') \in \mathcal{D}_K$ be a sample with $x' \in R_j(\lambda_{i\to j})$ and define the set $A_j(x) \coloneqq \big\{y \in R_i : \big\|J^+(R_i) \cdot |x - y|\big\|_\infty \leq r_j(x', \lambda_{i\to j})\big\}$. Then, it holds that $A_j(x) \subseteq \mathsf{Pre}(R_j(\lambda_{i\to j}))$.*

By using Theorem 7 for multiple samples, we obtain $\cup_x A_j(x) \subseteq \mathsf{Pre}(R_j(\lambda_{i\to j}))$. This idea is visualised in Figure 1, showing $R_j(\lambda_{i\to j})$ in orange and three samples (x, u, x'). The shaded squares around each x' are the balls $B^\infty_{r_j(x',\lambda_{i\to j})}(x')$, and the squares around each x are the sets $A_j(x)$ that form the underapproximation of the backward reachable set. Observe that, for the higher value of $\lambda_{i\to j} = 0.95$, we obtain larger sets $A_j(x)$ and thus a larger underapproximation. However, a higher $\lambda_{i\to j}$ also leads to a larger target set $R_j(\lambda_{i\to j})$, and thus to a more conservative abstraction.

Algorithm with variable $\lambda_{i\to j}$. Fix a state s_i and an action a_j. To determine if a_j is enabled in s_i, we need to check if the union of all sets $A_j(x)$ covers $\mathcal{T}^{-1}(s_i) = R_i$. Moreover, the question remains what value of $\lambda_{i\to j}$ we should use in practice. To this end, we propose an algorithm that chooses $\lambda_{i\to j}$ based on the samples $(x, u, x') \in \mathcal{D}_K$ available. For brevity, define $\mathcal{D}_K(R_j) \subset \mathcal{D}_K$ as the subset of samples for which $x' \in R_j$, i.e., $\mathcal{D}_K(R_j) = \{(x, u, x') \in \mathcal{D}_K : x' \in \mathcal{T}^{-1}(s_j)\}$. As described in Algorithm 1, we compute the enabled actions $Act(s_i)$ for all $s_i \in S$ as follows:

1. As shown in Figure 1 (right), we create a uniform tiling of R_i into hyperrectangles that we call *voxels*. The set of all $m_i \in \mathbb{N}$ voxels for R_i is $\Phi(R_i) = \{\phi_1, \dots, \phi_{m_i}\}$, where each $\phi_\ell \subset \mathbb{R}^n$ and $\cup_{\ell=1}^{m_i} \phi_\ell = R_i$. Let $c_\phi, \delta_\phi \in \mathbb{R}^n$ be the centre and radius of voxel $\phi \in \Phi(R_i)$, respectively.
2. Fix a voxel $\phi \in \Phi(R_i)$ and a sample $(x, u, x') \in \mathcal{D}_K(R_j)$. With overloading of notation, let $\lambda^\star(\phi, (x, u, x'))$ be the smallest value of $\lambda_{i\to j}$ such that ϕ is completely contained in $A_j(x)$, i.e., $\phi \subseteq A_j(x)$. In practice, we find an overapproximation of $\lambda^\star(\phi, (x, u, x'))$ defined as
$$\lambda^+(\phi, (x, u, x')) \coloneqq 1 + \frac{\big\|J^+(R_i) \cdot (|x - c_\phi| + \delta_\phi)\big\|_\infty - r_j(x', 1)}{r_j(x', 2) - r_j(x', 1)} \geq \lambda^\star(\phi, (x, u, x')). \quad (10)$$
3. Then, we compute the actual value of $\lambda_{i\to j}$ as the maximum of $\lambda^+(\phi, (x, u, x'))$ over all $\phi \in \Phi(R_i)$ and the minimum over all $(x, u, x') \in \mathcal{D}_K(R_j)$:
$$\lambda_{i\to j} = \max_{\phi \in \Phi(R_i)} \min_{(x,u,x') \in \mathcal{D}_K(R_j)} \lambda^+(\phi, (x, u, x')). \quad (11)$$
As shown in Figure 1 (right), we thus find a factor $\lambda_{i\to j}$ such that *every voxel ϕ is covered by a ball around some sampled state x'*. Since $\cup_{\phi\in\Phi(R_i)} = R_i$, it follows that $R_i \subseteq \mathsf{Pre}(R_j(\lambda_{i\to j}))$.
4. We check whether $\lambda_{i\to j}$ satisfies the global upper bound Λ. If $\lambda_{i\to j} \leq \Lambda$, then we conclude that $a_j \in Act(s_i)$ for $\lambda_{i\to j}$. If, on the other hand, $\lambda_{i\to j} > \Lambda$, then we set $a_j \notin Act(s_i)$.

Remark 8 (Role of Λ) *The hyperparameter Λ controls the number of actions enabled in the IMDP. A higher Λ results in more enabled IMDP actions, resulting in a more accurate but larger abstraction.*

Policy refinement. Our approach leads directly to a strategy for obtaining the refined policy $\pi_k(x)$ in Eq. (8). Suppose that at time k, the continuous state $x_k \in R_i$ corresponds with IMDP state $s_i = \mathcal{T}(x_k)$, and suppose the optimal IMDP action is $a_j = \sigma(s_i)$. Let $\phi \in \Phi(R_i)$ be the voxel containing x_k. Then, we choose $\pi_k(x_k)$ as the input $u \in \mathcal{U}$ that attains the minimal $\lambda^+(\phi, (x, u, x'))$ over all samples $(x, u, x') \in \mathcal{D}_K(R_j)$ in Eq. (11). As required, this leads to $f(x_k, u) \in R_j(\lambda_{i\to j})$ by construction. Thus, we obtain a refined policy that is constant within each voxel.

Algorithm 1 Computing enabled actions by underapproximating backward reachable sets

Data: Samples $\mathcal{D}_K = \{(x_\ell, u_\ell, \hat{x}_\ell = f(x_\ell, u_\ell))\}_{\ell=1}^K$; max. scaling factor $\Lambda > 0$
Result: Enabled actions $Act(s_i) \subseteq Act$ for all IMDP states $s_i \in S$

$Act(s_i) \leftarrow \emptyset \; \forall s_i \in S$ ▷ Initialise enabled actions
for $j = 1, \dots, v$ **do**
  $\mathcal{D}_K(R_j) \leftarrow \{(x, u, x') \in \mathcal{D}_K : x' \in \mathcal{T}^{-1}(s_j)\}$ ▷ Find samples leading to successor in s_j
  for $i = 1, \dots, v$ **do**
    $\Phi(R_i) \leftarrow \{\phi_1, \dots, \phi_{m_i}\}$ ▷ Define voxelised representation of R_i
    for $\phi \in \Phi(R_i)$ **do**
      for $(x, u, x') \in \mathcal{D}_K(R_j)$ **do**
        if $x \in R_i$ **then**
          $\lambda^+(\phi, (x, u, x')) \leftarrow 1 + \frac{\left\| J^+(R_i) \cdot (|x - c_\phi| + \delta_\phi) \right\|_\infty - r_j(x', 1)}{r_j(x', 2) - r_j(x', 1)}$
        end
      end
    end
    $\lambda_{i \to j} \leftarrow \max_{\phi \in \Phi(R_i)} \min_{(x,u,x') \in \mathcal{D}_K(R_j)} \lambda^+(\phi, (x, u, x'))$ ▷ Compute scaling factor
    if $\lambda_{i \to j} \leq \Lambda$ **then**
      $Act(s_i) \leftarrow Act(s_i) \cup \{a_j\}$ ▷ Enable action if $\lambda_{i \to j}$ is below max. scaling factor Λ
    end
  end
end

5. Computing Probability Intervals with Data

We compute bounds on the probability intervals in Eq. (6) by sampling the noise $w \in \Omega$. We focus on a fixed transition (s_i, a_j, s') and repeat the procedure for all state-action pairs. First, observe that[9]

$$\check{P}_{i,j}(s') := \mathbb{P}\left\{w \in \Omega : R_j(\lambda_{i \to j}) + w \subseteq \mathcal{T}^{-1}(s')\right\} \leq \min_{\hat{x} \in R_j(\lambda_{i \to j})} \eta\big(\hat{x}, \mathcal{T}^{-1}(s')\big), \tag{12}$$

$$\hat{P}_{i,j}(s') := \mathbb{P}\left\{w \in \Omega : R_j(\lambda_{i \to j}) + w \cap \mathcal{T}^{-1}(s') \neq \emptyset\right\} \geq \max_{\hat{x} \in R_j(\lambda_{i \to j})} \eta\big(\hat{x}, \mathcal{T}^{-1}(s')\big), \tag{13}$$

where the probabilities $\check{P}_{i,j}(s')$, $\hat{P}_{i,j}(s')$ are the (unknown) success probabilities of a Bernoulli random variable. Thus, fixing a set $\{w^{(1)}, \dots, w^{(N)}\} \in \Omega^N$ of $N \in \mathbb{N}$ noise samples induces the binomial distributions $B(N, \check{P}_{i,j}(s'))$ and $B(N, \hat{P}_{i,j}(s'))$.[10] Observing $\{w^{(1)}, \dots, w^{(N)}\} \in \Omega^N$ yields samples $\check{N}_{i,j}(s') \sim B(N, \check{P}_{i,j}(s'))$ and $\hat{N}_{i,j}(s') \sim B(N, \hat{P}_{i,j}(s'))$ from these binomials. (see Nazeri et al. (2024, App. D) for an explicit definition). Badings et al. (2023a, 2024) leverage the scenario approach (Campi et al., 2021) to estimate $\check{P}_{i,j}(s')$ and $\hat{P}_{i,j}(s')$ based on $\check{N}_{i,j}(s')$ and $\hat{N}_{i,j}(s')$ as intervals. However, as recently pointed out by (Meggendorfer et al., 2024, Theorem 2), tighter intervals can be obtained by using the *Clopper-Pearson interval*, a well-known statistical method for calculating binomial confidence intervals (Clopper and Pearson, 1934; Newcombe, 1998):

Theorem 9 (Clopper-Pearson interval) *Let $\{w^{(1)}, \dots, w^{(N)}\} \in \Omega^N$, and let $\beta \in (0, 1)$. For fixed $s_i, s' \in S$ and $a_j \in Act(s_i)$, compute $\check{N}_{i,j}(s')$ and $\hat{N}_{i,j}(s')$. Then, it holds that*

$$\mathbb{P}^N\Big\{\{w^{(1)}, \dots, w^{(N)}\} \in \Omega^N : \check{P}_{lb} \leq \mathcal{P}(s_i, a_j)(s') \leq \hat{P}_{ub}\Big\} \geq 1 - \beta, \tag{14}$$

9. We write $R_j(\lambda_{i \to j}) + w = \{\alpha + w : \alpha \in R_j(\lambda_{i \to j})\}$ for the Minkowski sum between $R_j(\lambda_{i \to j}$ and w.
10. We write $B(n, p)$ to denote a binomial distribution with $n \in \mathbb{N}$ experiments and success probability $p \in [0, 1]$.

where $\check{P}_{lb} = 0$ if $\check{N}_{i,j}(s') = 0$, and otherwise, $\check{P}_{lb}$ is the solution to

$$\frac{\beta}{2} = \sum\nolimits_{i=\check{N}_{i,j}(s')}^{N} \binom{N}{i} \cdot (\check{P}_{lb})^i \cdot (1 - \check{P}_{lb})^{N-i}, \tag{15}$$

and $\hat{P}_{ub} = 1$ if $\check{N}_{i,j}(s') = N$, and otherwise, $\hat{P}_{ub}$ is the solution to

$$\frac{\beta}{2} = \sum\nolimits_{i=0}^{\hat{N}_{i,j}(s')} \binom{N}{i} \cdot (\hat{P}_{ub})^i \cdot (1 - \hat{P}_{ub})^{N-i}. \tag{16}$$

Proof The proof follows by applying the Clopper-Pearson interval (Clopper and Pearson, 1934; Thulin, 2014) to the binomials $\check{N}_{i,j}(s') \sim B(N, \check{P}_{i,j}(s'))$ and $\hat{N}_{i,j}(s') \sim B(N, \hat{P}_{i,j}(s'))$, which yields

$$\mathbb{P}^N\big\{\check{P}_{\text{lb}} \leq \check{P}_{i,j}(s')\big\} \geq 1 - {}^{\beta}\!/_{2}, \quad \text{and} \quad \mathbb{P}^N\big\{\hat{P}_{i,j}(s') \leq \hat{P}_{\text{ub}}\big\} \geq 1 - {}^{\beta}\!/_{2}. \tag{17}$$

Combining Eq. (17) with Eqs. (12) and (13) through the union bound, we obtain Eq. (14). ■

Theorem 9 asserts that each interval is *correct with probability* $\geq 1 - \beta$. Combining Theorems 4 and 9 leads to a statistical solution to Problem 1; the proof is analogous to Badings et al. (2023b, Thm. 2):

Corollary 10 *Let $\mathcal{M}_{\mathbb{I}}{}'$ be the IMDP abstraction with probability intervals obtained via Theorem 9. For every IMDP scheduler $\sigma \in \mathfrak{S}^{\mathcal{M}_{\mathbb{I}}{}'}$, there exists a policy $\pi \in \Pi^{\mathcal{S}}$ for system $\mathcal{S}$ such that*

$$\mathbb{P}\Big\{\min_{P \in \mathcal{P}} \Pr_{\sigma,P}^{\mathcal{M}_{\mathbb{I}}{}'}(S_G, S_U, h) \leq \Pr_{\pi}^{\mathcal{S}}(X_G, X_U, h)\Big\} \geq \max(0, 1 - \beta \cdot |S|^2 \cdot |Act|).$$

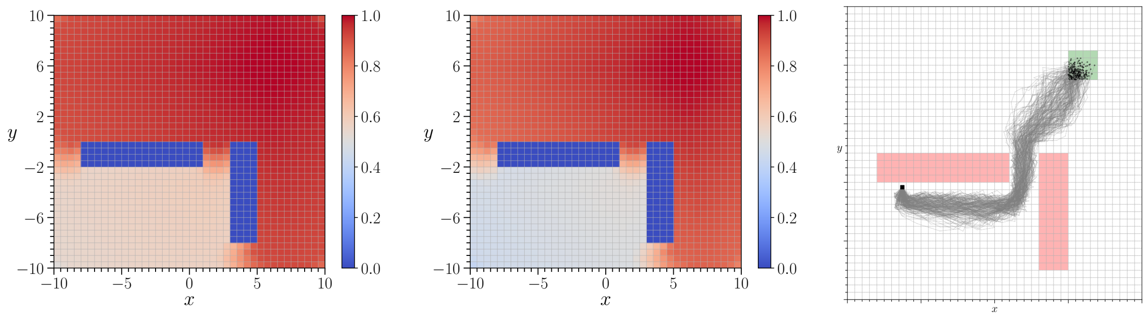
The factor of $|S|^2 \cdot |Act|$ in Theorem 10 comes from the maximum possible number of IMDP transitions, which is the number of (s, a, s') triples such that $s \in S$, $a \in Act(s)$, and $\mathcal{P}(s, a)(s') > 0$.

Corollary 10 carries an important message: For any system $\mathcal{S}$ (that satisfies Assumption 1), the IMDP abstraction $\mathcal{M}_{\mathbb{I}}{}'$ with probability intervals given by Theorem 9 leads, with at least probability $1-\beta\cdot|S|^2\cdot|Act|$, to a solution to Problem 1. In practice, we can again use Eq. (8) to refine any IMDP scheduler $\sigma \in \mathfrak{S}^{\mathcal{M}_{\mathbb{I}}{}'}$ into the corresponding policy $\pi \in \Pi^{\mathcal{S}}$ for system $\mathcal{S}$ that solves Problem 1.

6. Experimental Evaluation

We conduct experiments on (1) an inverted pendulum, (2) a harmonic oscillator with nonlinear damping, and (3) a car parking benchmark with nonlinear control. Details on each benchmark are in Nazeri et al. (2024, App. E). We implement our approach in Python, using robust value iteration (Wolff et al., 2012; Iyengar, 2005), implemented in the model checker PRISM (Kwiatkowska et al., 2011) to compute optimal schedulers as per Eq. (3). All experiments ran parallelized on a computer with 32 3.3 GHz cores and 128 GB of RAM. For all experiments, we use an error rate of $\beta = 1 - \frac{0.05}{T}$ on every IMDP transition, leading to an overall confidence probability (as per Corollary 10) of 0.95 for an IMDP abstraction with T transitions.

Lower bounds on reach-avoid probabilities. We investigate whether our IMDP abstractions lead to sound and non-trivial lower bounds on the reach-avoid probability $\Pr_{\pi}^{\mathcal{S}}(X_G, X_U, h)$. A heatmap of these probabilities for the car parking benchmark (probability intervals obtained using Theorem 9 with $N = 10\,000$ samples) is shown in Figure 2(*a*) (results for the other benchmarks and/or a lower number of samples of $N = 1\,000$ are in Nazeri et al. (2024, App. E)). For this case, Figure 2(*c*)



(a) Clopper-Pearson intervals (b) Scenario approach intervals (c) Simulation under (a)

Figure 2: Reach-avoid probabilities $\Pr_\pi^{\mathcal{S}}(X_G, X_U, h)$ for the car benchmark with (a) probability intervals from Theorem 9 and (b) the approach from Badings et al. (2022), both with $N = 10\,000$ samples. Fig. (c) shows simulated trajectories under the resulting policy from Eq. (8) for our method.

shows a simulated trajectory under the resulting policy obtained from Eq. (3). These results confirm that our method yields reliable policies with non-trivial reach-avoid guarantees in practice.

Comparison to scenario approach. We benchmark our IMDPs with probability intervals from Theorem 9 against the approach from Badings et al. (2023a), which instead uses the scenario approach. The resulting reach-avoid probabilities for the car benchmark are in Figure 2(b) (again, see Nazeri et al. (2024, App. E) for the other benchmarks). Using Clopper-Pearson leads to tighter intervals than those from Badings et al. (2023a), thus leading to policies with better reach-avoid guarantees.

The role of the scaling factors $\lambda_{i\to j}$. Finally, we demonstrate the importance of choosing the scaling factors $\lambda_{i\to j}$ defining the IMDP actions. To this end, we run all three benchmarks with a smaller Λ that upper bounds $\lambda_{i\to j}$, as defined in Sect. 4 (we report the precise values of Λ in Nazeri et al. (2024, App. E)). Our results, which we present in Nazeri et al. (2024, App. E), demonstrate that a lower Λ generally leads to decreased reach-avoid probabilities. This is likely because a lower Λ leads to smaller backward reachable sets, which causes the system taking more steps until it reaches the goal – investigating this effect in more detail is an important aspect for future research.

7. Conclusion

We presented a data-driven approach for the automated synthesis of policies for nonlinear systems with additive stochastic noise. Our method only requires samples of the system and the Lipschitz constant, thus overcoming the limitations of model-based abstractions when the dynamics are not fully known. Our numerical experiments show our approach yields robust and reliable policies.

This work opens pathways to enhance data-driven methods for controller synthesis in real-world settings where traditional modelling is infeasible. The main limitations of this work are the restriction to additive and i.i.d. stochastic noise, and the sensitivity of the framework to hyperparameters. Given the high amount of data required to construct the IMDP, this work only scales to systems with a small number of state variables. Future directions include investigating tighter bounds for PAC guarantees and integrating our techniques with other frameworks to reduce the computational complexity. In particular, exploiting structural properties of the dynamics and integrating our abstraction technique with a learning framework are two ideas to overcome the scalability limitations.

Acknowledgments

This paper was supported by the Horizon Europe EIC project SymAware (101070802), the ERC grant 101089047, and the EPSRC grant EP/Y028872/1, Mathematical Foundations of Intelligence: An "Erlangen Programme" for AI.

References

Alessandro Abate, Maria Prandini, John Lygeros, and Shankar Sastry. Probabilistic reachability and safety for controlled discrete time stochastic hybrid systems. *Autom.*, 44(11):2724–2734, 2008.

Alessandro Abate, Mirco Giacobbe, and Diptarko Roy. Stochastic omega-regular verification and control with supermartingales. In *CAV (3)*, volume 14683 of *LNCS*, pages 395–419. Springer, 2024.

Matthias Althoff, Goran Frehse, and Antoine Girard. Set propagation techniques for reachability analysis. *Annu. Rev. Control. Robotics Auton. Syst.*, 4:369–395, 2021.

Rajeev Alur, Thomas A. Henzinger, Orna Kupferman, and Moshe Y. Vardi. Alternating refinement relations. In *CONCUR*, volume 1466 of *LNCS*, pages 163–178. Springer, 1998.

Thom S. Badings, Alessandro Abate, Nils Jansen, David Parker, Hasan A. Poonawala, and Mariëlle Stoelinga. Sampling-based robust control of autonomous systems with non-gaussian noise. In *AAAI*, pages 9669–9678. AAAI Press, 2022.

Thom S. Badings, Licio Romao, Alessandro Abate, and Nils Jansen. Probabilities are not enough: Formal controller synthesis for stochastic dynamical models with epistemic uncertainty. In *AAAI*, pages 14701–14710. AAAI Press, 2023a.

Thom S. Badings, Licio Romao, Alessandro Abate, David Parker, Hasan A. Poonawala, Mariëlle Stoelinga, and Nils Jansen. Robust control for dynamical systems with non-gaussian noise via formal abstractions. *J. Artif. Intell. Res.*, 76:341–391, 2023b.

Thom S. Badings, Licio Romao, Alessandro Abate, and Nils Jansen. A stability-based abstraction framework for reach-avoid control of stochastic dynamical systems with unknown noise distributions. In *ECC*, pages 564–570. IEEE, 2024.

Christel Baier and Joost-Pieter Katoen. *Principles of model checking*. MIT Press, 2008.

Somil Bansal, Mo Chen, Sylvia L. Herbert, and Claire J. Tomlin. Hamilton-jacobi reachability: A brief overview and recent advances. In *CDC*, pages 2242–2253. IEEE, 2017.

Adrien Banse, Licio Romao, Alessandro Abate, and Raphaël M. Jungers. Data-driven abstractions via adaptive refinements and a kantorovich metric. *CoRR*, abs/2303.17618, 2023.

Calin Belta, Boyan Yordanov, and Ebru Aydin Gol. *Formal methods for discrete-time dynamical systems*, volume 15. Springer, 2017.

Dimitri P. Bertsekas and Steven E. Shreve. *Stochastic Optimal Control: The Discrete-time Case*. Athena Scientific, 1978. ISBN 1-886529-03-5.

Marco C. Campi, Algo Carè, and Simone Garatti. The scenario approach: A tool at the service of data-driven decision making. *Annu. Rev. Control.*, 52:1–17, 2021.

Charles J Clopper and Egon S Pearson. The use of confidence or fiducial limits illustrated in the case of the binomial. *Biometrika*, 26(4):404–413, 1934.

Rudi Coppola, Andrea Peruffo, and Manuel Mazo Jr. Data-driven abstractions for verification of linear systems. *IEEE Control. Syst. Lett.*, 7:2737–2742, 2023.

Giannis Delimpaltadakis, Morteza Lahijanian, Manuel Mazo Jr., and Luca Laurenti. Interval markov decision processes with continuous action-spaces. In *HSCC*, pages 12:1–12:10. ACM, 2023.

Alex Devonport, Adnane Saoud, and Murat Arcak. Symbolic abstractions from data: A PAC learning approach. In *CDC*, pages 599–604. IEEE, 2021.

Chuchu Fan, Umang Mathur, Sayan Mitra, and Mahesh Viswanathan. Controller synthesis made real: Reach-avoid specifications and linear dynamics. In *CAV (1)*, volume 10981 of *LNCS*, pages 347–366. Springer, 2018.

Robert Givan, Sonia M. Leach, and Thomas L. Dean. Bounded-parameter markov decision processes. *Artif. Intell.*, 122(1-2):71–109, 2000.

Ibon Gracia, Dimitris Boskos, Luca Laurenti, and Morteza Lahijanian. Data-driven strategy synthesis for stochastic systems with unknown nonlinear disturbances. In *L4DC*, volume 242 of *Proceedings of Machine Learning Research*, pages 1633–1645. PMLR, 2024a.

Ibon Gracia, Luca Laurenti, Manuel Mazo Jr., Alessandro Abate, and Morteza Lahijanian. Temporal logic control for nonlinear stochastic systems under unknown disturbances. *CoRR*, abs/2412.11343, 2024b.

Kazumune Hashimoto, Adnane Saoud, Masako Kishida, Toshimitsu Ushio, and Dimos V. Dimarogonas. Learning-based symbolic abstractions for nonlinear control systems. *Autom.*, 146:110646, 2022.

Garud N. Iyengar. Robust dynamic programming. *Math. Oper. Res.*, 30(2):257–280, 2005.

John Jackson, Luca Laurenti, Eric W. Frew, and Morteza Lahijanian. Strategy synthesis for partially-known switched stochastic systems. In *HSCC*, pages 6:1–6:11. ACM, 2021.

Milad Kazemi, Rupak Majumdar, Mahmoud Salamati, Sadegh Soudjani, and Ben Wooding. Data-driven abstraction-based control synthesis. *CoRR*, abs/2206.08069, 2022.

Hassan K Khalil and Jessy W Grizzle. *Nonlinear systems*, volume 3. Prentice hall Upper Saddle River, NJ, 2002.

Marta Z. Kwiatkowska, Gethin Norman, and David Parker. PRISM 4.0: Verification of probabilistic real-time systems. In *CAV*, volume 6806 of *LNCS*, pages 585–591. Springer, 2011.

Morteza Lahijanian, Sean B. Andersson, and Calin Belta. Formal verification and synthesis for discrete-time stochastic systems. *IEEE Trans. Autom. Control.*, 60(8):2031–2045, 2015.

Abolfazl Lavaei, Sadegh Soudjani, Alessandro Abate, and Majid Zamani. Automated verification and synthesis of stochastic hybrid systems: A survey. *Autom.*, 146:110617, 2022.

Abolfazl Lavaei, Sadegh Soudjani, and Emilio Frazzoli. A compositional dissipativity approach for data-driven safety verification of large-scale dynamical systems. *IEEE Trans. Autom. Control.*, 68 (12):7240–7253, 2023.

Anas Makdesi, Antoine Girard, and Laurent Fribourg. Efficient data-driven abstraction of monotone systems with disturbances. In *ADHS*, volume 54 of *IFAC-PapersOnLine*, pages 49–54. Elsevier, 2021.

Frederik Baymler Mathiesen, Simeon C. Calvert, and Luca Laurenti. Safety certification for stochastic systems via neural barrier functions. *IEEE Control. Syst. Lett.*, 7:973–978, 2023.

Frederik Baymler Mathiesen, Sofie Haesaert, and Luca Laurenti. Scalable control synthesis for stochastic systems via structural imdp abstractions, 2024.

Tobias Meggendorfer, Maximilian Weininger, and Patrick Wienhöft. What are the odds? improving the foundations of statistical model checking. *CoRR*, abs/2404.05424, 2024.

Ian M. Mitchell. Comparing forward and backward reachability as tools for safety analysis. In *HSCC*, volume 4416 of *LNCS*, pages 428–443. Springer, 2007.

Mahdi Nazeri, Thom Badings, Alessandro Abate, and Sadegh Soudjani. Data-driven yet formal policy synthesis for stochastic nonlinear dynamical systems (extended version). Technical report, 2024.

Ameneh Nejati, Abolfazl Lavaei, Pushpak Jagtap, Sadegh Soudjani, and Majid Zamani. Formal verification of unknown discrete- and continuous-time systems: A data-driven approach. *IEEE Trans. Autom. Control.*, 68(5):3011–3024, 2023.

Robert G Newcombe. Two-sided confidence intervals for the single proportion: comparison of seven methods. *Statistics in medicine*, 17(8):857–872, 1998.

Arnab Nilim and Laurent El Ghaoui. Robust control of markov decision processes with uncertain transition matrices. *Oper. Res.*, 53(5):780–798, 2005.

Andrea Peruffo and Manuel Mazo. Data-driven abstractions with probabilistic guarantees for linear PETC systems. *IEEE Control. Syst. Lett.*, 7:115–120, 2023.

Amir Pnueli. The temporal logic of programs. In *FOCS*, pages 46–57. IEEE Computer Society, 1977.

Martin L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Wiley Series in Probability and Statistics. Wiley, 1994.

Nicholas Rober, Sydney M. Katz, Chelsea Sidrane, Esen Yel, Michael Everett, Mykel J. Kochenderfer, and Jonathan P. How. Backward reachability analysis of neural feedback loops: Techniques for linear and nonlinear systems. *CoRR*, abs/2209.14076, 2022.

Licio Romao, Antonis Papachristodoulou, and Kostas Margellos. On the exact feasibility of convex scenario programs with discarded constraints. *IEEE Trans. Autom. Control.*, 68(4):1986–2001, 2023.

Ali Salamati, Abolfazl Lavaei, Sadegh Soudjani, and Majid Zamani. Data-driven verification and synthesis of stochastic systems via barrier certificates. *Autom.*, 159:111323, 2024.

Dietmar Salamon. *Measure and Integration*. European Mathematical Society, 2016, 2016.

Oliver Schön, Shammakh Naseer, Ben Wooding, and Sadegh Soudjani. Data-driven abstractions via binary-tree Gaussian processes for formal verification. *IFAC-PapersOnLine*, 58(11):115–122, 2024.

Sadegh Esmaeil Zadeh Soudjani and Alessandro Abate. Adaptive and sequential gridding procedures for the abstraction and verification of stochastic processes. *SIAM J. Appl. Dyn. Syst.*, 12(2): 921–956, 2013.

Sadegh Esmaeil Zadeh Soudjani, Caspar Gevaerts, and Alessandro Abate. FAUST ^{2} : Formal abstractions of uncountable-state stochastic processes. In *TACAS*, volume 9035 of *LNCS*, pages 272–286. Springer, 2015.

Dusan M. Stipanovic, Inseok Hwang, and Claire J. Tomlin. Computation of an over-approximation of the backward reachable set using subsystem level set functions. In *ECC*, pages 300–305. IEEE, 2003.

Marnix Suilen, Thom Badings, Eline M. Bovy, David Parker, and Nils Jansen. *Robust Markov Decision Processes: A Place Where AI and Formal Methods Meet*, pages 126–154. Springer Nature Switzerland, Cham, 2025. ISBN 978-3-031-75778-5. doi: 10.1007/978-3-031-75778-5_7.

Sean Summers and John Lygeros. Verification of discrete time stochastic hybrid systems: A stochastic reach-avoid decision problem. *Autom.*, 46(12):1951–1961, 2010.

Paulo Tabuada. *Verification and Control of Hybrid Systems - A Symbolic Approach*. Springer, 2009.

Måns Thulin. The cost of using exact confidence intervals for a binomial proportion. *Electronic Journal of Statistics*, 8(1), January 2014. ISSN 1935-7524. doi: 10.1214/14-ejs909.

Birgit van Huijgevoort, Oliver Schön, Sadegh Soudjani, and Sofie Haesaert. Syscore: Synthesis via stochastic coupling relations. In *HSCC*, pages 13:1–13:11. ACM, 2023.

Eric M. Wolff, Ufuk Topcu, and Richard M. Murray. Robust control of uncertain markov decision processes with temporal logic specifications. In *CDC*, pages 3372–3379. IEEE, 2012.

Liren Yang, Hang Zhang, Jean-Baptiste Jeannin, and Necmiye Ozay. Efficient backward reachability using the minkowski difference of constrained zonotopes. *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst.*, 41(11):3969–3980, 2022.

He Yin, Andrew K. Packard, Murat Arcak, and Peter J. Seiler. Finite horizon backward reachability analysis and control synthesis for uncertain nonlinear systems. In *ACC*, pages 5020–5026. IEEE, 2019.

Dorde Zikelic, Mathias Lechner, Thomas A. Henzinger, and Krishnendu Chatterjee. Learning control policies for stochastic systems with reach-avoid guarantees. In *AAAI*, pages 11926–11935. AAAI Press, 2023.