2024 European Control Conference (ECC)
June 25-28, 2024. Stockholm, Sweden

A Stability-Based Abstraction Framework for Reach-Avoid Control of Stochastic Dynamical Systems with Unknown Noise Distributions

Thom Badings, Licio Romao, Alessandro Abate, Nils Jansen

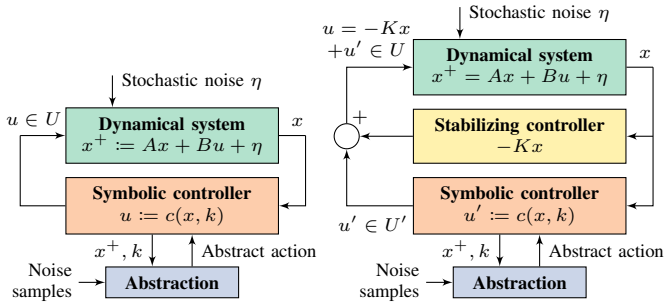
***Abstract*— Finite-state abstractions are widely studied for the automated synthesis of correct-by-construction controllers for stochastic dynamical systems. However, existing abstraction methods often lead to prohibitively large finite-state models. To address this issue, we propose a novel abstraction scheme for stochastic linear systems that exploits the system's stability to obtain significantly smaller abstract models. As a unique feature, we first stabilize the open-loop dynamics using a linear feedback gain. We then use a model-based approach to abstract a known part of the stabilized dynamics while using a data-driven method to account for the stochastic uncertainty. We formalize abstractions as Markov decision processes (MDPs) with intervals of transition probabilities. By stabilizing the dynamics, we can further constrain the control input modeled in the abstraction, which leads to smaller abstract models while retaining the correctness of controllers. Moreover, when the stabilizing feedback controller is aligned with the property of interest, then a good trade-off is achieved between the reduction in the abstraction size and the performance loss. The experiments show that our approach can reduce the size of the graph of abstractions by up to 90% with negligible performance loss.**

I. Introduction

The automated synthesis of correct-by-construction controllers for stochastic dynamical systems is crucial for their deployment in safety-critical scenarios. Synthesizing such controllers is challenging due to continuous and stochastic dynamics and the complexity of control tasks [1]. One solution is to abstract the system into a finite-state (also called symbolic) model [2]–[4]. Under an appropriate behavioral relation (e.g., a feedback refinement relation [5]), trajectories of the abstraction are related to those of the dynamical system. Thus, a controller (i.e., policy) in the abstraction can, by construction, be refined to a controller for the original system.

Conventional abstraction methods, however, rely on a precise mathematical model of the system. In relaxing this assumption, *data-driven abstractions* have recently gained momentum [6]–[13]. These methods take a black-box (or sometimes a gray-box, e.g., [10]) perspective and construct abstractions from sampled system trajectories. Several papers provide *probably approximately correct* (PAC) guarantees [7,12], whereas others return controllers with hard (non-statistical) guarantees [6,8,11]. However, except from [13], none of these methods can handle stochastic systems.

This research has been partially funded by an NWO grant NWA.1160.18.238 (PrimaVera), ERC Starting Grant 101077178 (DEUCE), and a JPMC faculty research award. T. Badings and N. Jansen are with the Institute for Computing and Information Sciences, Radboud University, Nijmegen, The Netherlands; {thom.badings, nils.jansen}@ru.nl. N. Jansen is also with the Faculty of Computer Science, Ruhr-Universität Bochum, Germany. L. Romao and A. Abate are with the Department of Computer Science, Oxford University; {licio.romao, alessandro.abate}@cs.ox.ac.uk.



(a) Single-layer abstraction. (b) Two-layer abstraction.

Fig. 1: A single-layer abstraction (a), versus our two-layer feedback design framework, which combines a stabilizing controller (linear feedback gain) with a symbolic controller (obtained from the abstraction). We impose constraints on both the total input $-Kx+u' \in U$ and on the input $u' \in U'$.

In this paper, we take a middle route between these model-based and data-driven abstraction methods. Specifically, we consider control problems for linear systems with *known* deterministic dynamics but stochastic noise of an *unknown distribution*. This *hybrid* setting is similar to [14,15], which develop a method to construct abstractions with PAC guarantees by sampling of the noise. However, due to their exhaustive discretization of the state space, the application to large-scale, industrial and realistic systems remains elusive.

A promising way to improve scalability is to exploit classical system properties, such as stability [16]. For example, [17] shows that for any stable discrete-time linear system with input constraints, there exists an approximately bismilar finite abstraction of any desired precision. Similar results hold for incrementally stable continuous-time switched [18] and nonlinear systems [19,20]. A related notion is that of incremental forward completeness, which enables the abstraction of nonlinear discrete-time systems [21]. We observe that these results guarantee *the existence* of a certain type of abstraction *if* the system is stable. However, we postulate that stability may also be beneficial to construct finite-state abstractions with *smaller* underlying graphs.

Thus, the question central to this paper is: "How can the stability of a stochastic dynamical system be exploited to synthesize controllers via finite-state abstractions with smaller underlying graphs?" Our approach builds upon the hybrid abstraction technique for discrete-time stochastic linear systems developed in [14]. We consider tasks as *reach-avoid properties*, i.e., reach a set of goal states while always avoiding

unsafe states. The control objective is to design a feedback controller such that the closed-loop system satisfies the reach-avoid task with at least a desired threshold probability.

Inspired by [14], we create an abstraction for discrete-time stochastic linear systems into an interval Markov decision process (iMDP) with PAC intervals of transition probabilities, which we compute using data-driven techniques for scenario programs with discarded constraints [22,23]. A defining characteristic of this abstraction is that each abstract action is associated with a *fixed distribution* over (continuous) successor states. By contrast, other abstractions typically associate each abstract action with a *fixed control input*, such that the distribution over successor states depends on the precise continuous state where the abstract action is chosen. With our approach, we avoid this issue at the cost of more restrictive assumptions on the dynamics (Assumption 1).

Instead of abstracting the open-loop dynamics directly (as in Fig. 1a), we propose the two-layer feedback control design framework in Fig. 1b. In this framework, we first stabilize the system with a linear feedback gain and then abstract the stabilized dynamics. This approach delegates part of the control effort to the stabilizing controller, which allows us to further constrain the control input synthesized in the abstraction. Especially if the stabilizing controller contributes to satisfying the reach-avoid task, we can reduce the number of edges in the graph of the abstraction significantly (by up to 90%; see Sect. V) with negligible performance loss.

Contributions: As our main contribution, we extend the abstraction method from [14] to the two-layer abstraction in Fig. 1b and show that this new scheme can be used to construct abstractions with smaller underlying graphs. We show that the formal relation induced by the abstraction from [14] carries over to our setting. Our experiments exemplify the conditions necessary for a good trade-off between abstraction size and controller performance.

II. Preliminaries

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 x is a measurable function $x\colon \Omega \to \mathbb{R}^n$ for some $n \in \mathbb{N}$, which takes value $x(\omega) \in \mathbb{R}^n$ for $\omega \in \Omega$. We denote the set of all distributions for both a continuous and discrete set X by $\mathcal{P}(X)$. The convex hull of a set of points $\{v_1, \dots, v_m\}$ in $\mathbb{R}^n$ is $\mathrm{conv}(v_1, \dots, v_n)$. We denote the interior of $V \subset \mathbb{R}^n$ by $\mathrm{int(V)}$ and the pseudoinverse of matrix B by $B^\dagger$. The indicator function $\mathbb{1}_V(x)$ for a set $V \subset \mathbb{R}^n$ is one if $x \in V$ and zero otherwise. The Cartesian product of an interval is written as $[a,b]^n$, for $a \le b$, $n \in \mathbb{N}$.

A. Stochastic dynamical systems

Consider a discrete-time, stochastic linear dynamical system $\mathcal{S}$ where the state space variable $x_k \in \mathbb{R}^n$ evolves as

$$\mathcal{S}: \; x_{k+1} = Ax_k + Bu_k + \eta_k, \quad x_0 = \bar{x}, \tag{1}$$

where $\bar{x} \in \mathbb{R}^n$ is the initial condition, $u_k \in \mathbb{R}^p$ is the control input, and $\eta_k \in \mathbb{R}^n$ is a stochastic noise. Matrices A and B have the appropriate dimensions. The control input is constrained to a convex polytope $U = \{u \in \mathbb{R}^p : Gu \le g\} \subset \mathbb{R}^p$ called the admissible control input, where $G \in \mathbb{R}^{q \times p}$ and $g \in \mathbb{R}^q$. Moreover, $(\eta_k)_{k \in \mathbb{N}_0}$ is a discrete-time stochastic process defined on a probability space $(\Omega, \mathcal{F}, \mathbb{P})$, with its natural filtration (see [24] for details). Thus, $(x_k)_{k \in \mathbb{N}_0}$ is also a stochastic process in the same probability space.

Assumption 1 (Non-singular and controllable)**.** *Matrix $A \in \mathbb{R}^{n \times n}$ is non-singular, and the pair (A, B) is controllable.*

Assumption 2 (Noise distribution)**.** *For all $(x,u) \in \mathbb{R}^n \times U$ and all (Borel measurable [25]) sets $V \subset \mathbb{R}^n$, let $\mu_k(V; x, u) = \mathbb{P}\{\omega \in \Omega : Ax + Bu + \eta_k(\omega) \in V\} \in [0,1]$ be the conditional probability that the next state belongs to V, given the current state-input pair. Then we have that:*

- *(identically distributed): $\mu_k(V; x, u)$ is time-invariant; hence, we may drop the time index and write $\mu(V; x, v)$;*
- *(independence): For any finite collection $V_1, \dots, V_m \subset \mathbb{R}^n$ and state-action pairs $\{(x_i, u_i)\}_{i=1}^m$, we have that*

$$\mathbb{P}\{\omega \in \Omega : \bigcap_{i=1}^m Ax_i + Bu_i + \eta(\omega) \in V_i\} = \prod_{i=1}^m \mu(V_i; x_i, u_i);$$

- *(density): The Radon-Nikodym derivative of $\mu(V; x, u)$ exists for all pairs (x, u), and $\mu(V'; x, u)$ is a measurable function from $\mathbb{R}^n \times U$ to $[0,1]$ for all $V' \subset \mathbb{R}^n$. However (and importantly), the density μ is unknown.*

Assumption 2 requires process $(\eta_k)_{k \in \mathbb{N}}$ to be i.i.d. and to possess a well-defined probability density. However, we do not assume knowledge of its density. Under Assumption 2, system (1) can be equivalently expressed using the stochastic kernel (see [26, Chapter 7] for details) $T\colon \mathbb{R}^n \times U \to \mathcal{P}(\mathbb{R}^n)$, which maps each state-input pair to a distribution over states:

$$x_{k+1} \sim T(\cdot \mid x_k, u_k), \quad x_0 = \bar{x}. \tag{2}$$

For example, if $\eta_k \sim \mathcal{N}(\mu, \Sigma)$, $k \in \mathbb{N}$ is Gaussian, the stochastic kernel is given by $T(\cdot \mid x_k, u_k) = \mathcal{N}(Ax_k + Bu_k + \mu, \Sigma)$. A time-varying feedback controller chooses the inputs $u_k \in U$ by measuring the current state.

Definition 1. *A time-varying feedback controller is a measurable function $c\colon \mathbb{R}^n \times \mathbb{N}_0 \to U$, which maps a state $x \in \mathbb{R}^n$ and a time step $k \in \mathbb{N}_0$ to a control input $u \in U$.*

B. Problem statement

Given a system as in (1), our goal is to find a time-varying controller c such that the closed-loop system with $u_k = c(x_k, k)$, for all $k \in \mathbb{N}$, satisfies some objective. Specifically, we consider the objective of reaching a desired region X_G of the state space while always avoiding unsafe states X_U.

Definition 2 (Reach-avoid property)**.** *A reach-avoid property is a tuple (X_G, X_U, H) of a set of goal states X_G and unsafe states X_U (such that $X_G \cap X_U = \varnothing$), and horizon $H \in \mathbb{N}$.*

For system $\mathcal{S}$ in (1), we consider $X_G, X_U \subset \mathbb{R}^n$ as compact subsets of $\mathbb{R}^n$ and use the notation $\varphi = (X_G, X_U, H)$ for this reach-avoid property. A trajectory $x_0, x_1, \dots, x_H$ of length H for system $\mathcal{S}$ satisfies φ if there exists a $k \in \{0, \dots, H\}$ such that $x_k \in X_G$ and $x_{k'} \notin X_U$ for all

$k' \in \{0, \dots, k\}$. Under a fixed controller, system $\mathcal{S}$ induces a stochastic process $(x_k)_{k \in \mathbb{N}_0}$ for which we can reason over the probability of satisfying a reach-avoid property [26].

Definition 3 (Satisfaction of φ)**.** *For a fixed time-varying feedback controller $c : \mathbb{R}^n \times \mathbb{N} \to U$ and a given initial condition $\bar{x}$, the satisfaction probability of φ is denoted by*

$$\Pr^c_{\mathcal{S}}(\bar{x} \models \varphi) := \mathbb{P}\Big\{\omega \in \Omega : \exists k \in \{0, \dots, H\}, x_k(\omega) \in X_G, \; x_{k'}(\omega) \notin X_U \; \forall k' \in \{0, \dots, k\}, \; x_0 = \bar{x}, \, x_{k+1} \sim T(\cdot \mid x_k, c(x_k, k))\Big\}. \tag{3}$$

We are now able to describe our control objective.

Problem 1. *Given a linear stochastic system $\mathcal{S}$ as in* (1)*, with initial state $\bar{x}$, a reach-avoid property φ as in Definition 2, and a desired threshold probability $\rho \in [0, 1]$, design a time-varying feedback controller c such that $\Pr^c_{\mathcal{S}}(\bar{x} \models \varphi) \geq \rho$.*

C. Markov decision processes

Our approach for solving Problem 1 is to create a discrete abstraction of system $\mathcal{S}$ into an MDP with imprecise transition probabilities. To distinguish from system $\mathcal{S}$, we call abstract states *locations* and a controller for the abstraction a *policy*.

Definition 4. *An* interval MDP *(iMDP) $\mathcal{I}$ is defined as a tuple $\mathcal{I} := (S, \bar{s}, \mathcal{A}, \check{P}, \hat{P})$, where*

- *S is a finite set of locations, with initial condition $\bar{s} \in S$,*
- *$\mathcal{A}$ is a finite set of actions, with $\mathcal{A}(s) \subset \mathcal{A}$ denoting the actions enabled in location $s \in S$, and*
- *$P \colon S \times \mathcal{A} \rightrightarrows \mathcal{P}(S)$ maps each pair (s, a) to a set of distributions defined by $\check{P}(s, a), \hat{P}(s, a) \in [0, 1]^{|S|}$ as*

$$P(s, a) = \Big\{p \in [0, 1]^{|S|} : \check{P}_{s'}(s, a) \leq p_{s'} \leq \hat{P}_{s'}(s, a), \; \forall s' \in S, \; \sum_{s' \in S} p_{s'} = 1\Big\}. \tag{4}$$

For any iMDP, we require that $\check{P}_{s'}(s, a) \leq \hat{P}_{s'}(s, a)$ for all $s, s' \in S, a \in \mathcal{A}(s)$, and that $\sum_{s' \in S} \check{P}_{s'}(s, a) \leq 1 \leq \sum_{s' \in S} \hat{P}_{s'}(s, a)$ for all $s \in S, a \in \mathcal{A}(s)$; otherwise, the set (4) may be empty. An *adversary* fixes a probability $P'(s, a)(s') \in P(s, a)$ for all pairs $(s, a) \in S \times \mathcal{A}$. Importantly, a different P' can be chosen every time the same pair (s, a) is encountered. For brevity, we overload notation and use $P' \in P$ to denote choosing an adversary in the set of all adversaries.

Actions are chosen by a time-varying policy $\pi \colon S \times \mathbb{N}_0 \to \mathcal{P}(\mathcal{A})$, which maps every location $s \in S$ and time $k \in \mathbb{N}_0$ to an action $a \in \mathcal{A}$.[1] The set of all admissible policies[2] is

$$\Pi = \big\{\pi \colon S \times \mathbb{N}_0 \to \mathcal{P}(\mathcal{A}) \mid \pi(s, k)(a) > 0 \implies a \in \mathcal{A}(s)\big\}.$$

Thus, any policy $\pi \in \Pi$ requires that for all $k \in \mathbb{N}$ and $s \in S$, the support of the distribution $\pi(s, k)$ is contained in $\mathcal{A}(s)$.

For an iMDP, a reach-avoid property $\varphi' = (S_G, S_U, H)$ (cf. Def. 2) is defined over the locations, i.e., $S_G, S_U \subseteq S$. The semantics over trajectories $s_0, s_1, \dots, s_H$ are the same as for system $\mathcal{S}$. Similar to Def. 3, for any policy $\pi \in \Pi$ and transition function $P' \in P$, we denote the probability of satisfying φ' by $\Pr^\pi_{P'}(\bar{s} \models \varphi')$.[3] An optimal (robust) policy $\pi^\star \in \Pi$ optimizes the next min-max problem:

$$\pi^* \in \arg\max_{\pi \in \Pi} \min_{P' \in P} \Pr^\pi_{P'}(\bar{s} \models \varphi'). \tag{5}$$

Remark 1. *We can alternatively express reach-avoid properties by extending the iMDP with a reward function $R \colon S \to \mathbb{R}_{\geq 0}$ defined as $R(s) = \mathbb{1}_{S_G}(s)$, and making all locations $s \in S_U$ absorbing, i.e., $\check{P}(s, a, s) = \hat{P}(s, a, s) = 1 \, \forall s \in S_U$, $a \in \mathcal{A}(s)$. For details, we refer to [27, Def. 10.71].*

III. Abstraction-Based Controller Synthesis

We formally relate the dynamics in (1) to a finite iMDP, using a probabilistic variant of a feedback refinement relation [5]. Then, we establish that the abstraction proposed by [14] induces this relation. A measurable set $R \subseteq \mathbb{R}^n \times S$ is called a binary relation, for which we use notations $R(x) = \{s \in S : (x, s) \in R\}$ and $R^{-1}(s) = \{x \in \mathbb{R}^n : (x, s) \in R\}$.

Definition 5 ([15])**.** *A binary relation $R \subset \mathbb{R}^n \times S$ is a* probabilistic feedback refinement relation *from iMDP $\mathcal{I} = (S, \bar{s}, \mathcal{A}, \check{P}, \hat{P})$ to system $\mathcal{S}$ defined by* (2) *if*

1) *for the initial state-location, we have $(\bar{x}, \bar{s}) \in R$, and*
2) *for all $(x, s) \in R$ and $a \in \mathcal{A}(s)$, there exists a $u \in U$ such that for all $s' \in S$, it holds that*

$$\check{P}_{s'}(s, a) \leq \int_{\mathbb{R}^n} \mathbb{1}_{R^{-1}(s')}(\xi) T(d\xi \mid x, u) \tag{6}$$
$$= \mathbb{P}\big\{\omega \in \Omega : Ax + Bu + \eta(\omega) \in R^{-1}(s')\big\} \leq \hat{P}_{s'}(s, a).$$

Similar to [5], we denote a probabilistic feedback refinement relation R from $\mathcal{I}$ to $\mathcal{S}$ by $\mathcal{I} \preceq_R \mathcal{S}$. Moreover, we also use the relation R in Def. 5 to relate reach-avoid properties between system $\mathcal{S}$ and an iMDP.

Definition 6. *A pair of reach-avoid properties $\varphi = (X_G, X_U, H)$ and $\varphi' = (S_G, S_U, H)$ is consistent under a relation $R \subset \mathbb{R}^n \times S$, denoted by $\varphi' \preceq_R \varphi$ if*

$$S_G = \{s \in S : R^{-1}(s) \subseteq X_G\}, \tag{7}$$
$$S_U = \{s \in S : R^{-1}(s) \cap X_U \neq \varnothing\}. \tag{8}$$

Intuitively, given an iMDP path $s_0, s_1, \dots, s_H$ that satisfies φ', the relation $\varphi' \preceq_R \varphi$ implies that *all related trajectories* $x_0, x_1, \dots, x_H$, i.e., trajectories for which $(x_i, s_i) \in R$ for all $i = 0, \dots, H$, must satisfy φ. The following result, which is proven in [15], shows that Def. 5 and 6 can be used to synthesize correct-by-construction controllers for system $\mathcal{S}$.

Theorem 1 ([15])**.** *Consider a system $\mathcal{S}$ as in Eq.* (1)*, an iMDP as in Def. 4, and a relation $R \subset \mathbb{R}^n \times S$ such that $\mathcal{I} \preceq_R \mathcal{S}$. Also let properties $\varphi = (X_G, X_U, H)$ and $\varphi' = (S_G, S_U, H)$ be such that $\varphi' \preceq_R \varphi$. Then, for any policy $\pi \in \Pi$, there exists a controller c as in Def. 1 such that*

$$\Pr^c_{\mathcal{S}}(\bar{x} \models \varphi) \geq \min_{P' \in P} \Pr^\pi_{P'}(\bar{s} \models \varphi'). \tag{9}$$

[1] Time-varying policies are needed to attain optimal values for the time-bounded properties we consider [27, Ch. 10.6]. An equivalent approach is to encode the time step explicitly in the iMDP by defining the set of locations $S' = S \times \{0, \dots, H\}$ and using memoryless policies $\pi \colon S' \to \mathcal{A}$ instead.

[2] The policy class Π suffices to obtain optimal policies for iMDP [28].

[3] Fixing a policy $\pi \in \Pi$ and an adversary with $P' \in P$ induces a Markov chain with probability measure $\Pr^\pi_{P'}$; see [27, Def. 10.10] for details.

The proof of Theorem 1 uses that, for MDPs, the relation R preserves the satisfaction of probabilistic computation tree logic (PCTL), in which the reach-avoid property in Def. 2 can be expressed [29]. For iMDPs, this preservation of probabilistic satisfaction leads to the inequality in (9). In fact, Theorem 1 can be extended to any PCTL formula; see, e.g., [30]. However, since our contributions are unrelated to the property, we focus on reach-avoid properties for simplicity.

A. Abstraction procedure

We revisit the abstraction developed in [14,15], which first uses a model-based approach to compute the abstract locations and actions. Second, a data-driven approach is used to capture the stochastic uncertainty into intervals of transition probabilities. The resulting abstraction is an iMDP that creates a relation as in Def. 6 with a pre-defined confidence level.

1) Model-based locations and actions: The locations of the abstraction are given by a polyhedral partition of a bounded portion $\mathcal{X} \subset \mathbb{R}^n$ of the state space of system $\mathcal{S}$:

Definition 7. *A polyhedral partition of $\mathcal{X} \subset \mathbb{R}^n$ is a finite collection of sets $\{\mathcal{V}_1, \dots, \mathcal{V}_L, \mathbb{R}^n \setminus \mathcal{X}\}$ such that*

1) *Each $\mathcal{V}_i$ is a convex polytope, i.e., $\mathcal{V}_i = \{x \in \mathbb{R}^n : H_i x \leq h_i\}$ for $H_i \in \mathbb{R}^{p_i \times n}$, $h_i \in \mathbb{R}^{p_i}$, and $p_i \in \mathbb{N}$;*
2) *$\mathcal{X} = \bigcup_{i=1}^{L} \mathcal{V}_i$;*
3) *$\mathrm{int}(\mathcal{V}_\mathrm{i}) \bigcap \mathrm{int}(\mathcal{V}_\mathrm{j}) = \emptyset,\ \forall i, j \in \{1, \dots, L\},\ i \neq j$.*

Adding the final element $\mathbb{R}^n \setminus \mathcal{X}$ ensures that the partition covers $\mathbb{R}^n$. A partition creates an equivalence relation [31].

Remark 2 (Equivalence relation)**.** *A polyhedral partition of $\mathcal{X} \subset \mathbb{R}^n$ creates an equivalence relation $\sim\ \subset \mathbb{R}^n \times \mathbb{R}^n$, such that $[x]_\sim := \{x' \in \mathbb{R}^n \mid x \sim x'\}$ is the equivalence class of state $x \in \mathbb{R}^n$, where $x \sim x'$ denotes that $(x, x') \in \sim$. The set of all equivalence classes $\mathbb{R}^n/\sim\ = \{[x]_\sim \mid x \in \mathbb{R}^n\} = \{\mathcal{V}_1, \dots, \mathcal{V}_L, \mathbb{R}^n \setminus \mathcal{X}\}$ is the partition itself.*

The locations of the abstraction are the equivalence classes of the partition, i.e., $S := \mathbb{R}^n/\sim$. Next, the set of actions is $\mathcal{A} := \{a_1, \dots, a_q\}$, $q \in \mathbb{N}$, where each $a \in \mathcal{A}$ is associated with a target point $d_a \in \mathbb{R}^n$ in the state space of $\mathcal{S}$. For each point d_a, for $a \in \mathcal{A}$, we define the backward reachable set as

$$\begin{aligned} \mathcal{R}^{-1}(d_a) &= \{x \in \mathbb{R}^n : d_a = Ax + Bu,\, u \in U\} \\ &= \mathrm{conv}\big(A^{-1}(d_a - Bv^i)\ :\ i = 1, \dots, q\big), \end{aligned} \tag{10}$$

where the second equality follows from Assumption 1. The set $\mathcal{A}(s) \subseteq \mathcal{A}$ of actions enabled in location $s \in S$ is

$$\mathcal{A}(s) = \big\{a \in \mathcal{A} \mid s \subseteq \mathcal{R}^{-1}(d_a)\big\}. \tag{11}$$

Thus, action $a \in \mathcal{A}$ is enabled in location $s \in S$ only if the equivalence class s is contained in the backward reachable set $\mathcal{R}^{-1}(d_a)$. Choosing abstract action $a \in \mathcal{A}$ is defined such that $d_a = Ax + Bu$, which implies that $u = B^\dagger(d_a - Ax)$.[4] Since the noise is additive, the successor state is $d_a + \eta$, which is a random variable with distribution $T(\cdot \mid x, B^\dagger(d_a - Ax))$.

[4]Action a is only enabled in equivalence classes that are a subset of the backward reachable set $\mathcal{R}^{-1}(d_a)$. Thus, we have $u = B^\dagger(d_a - Ax) \in U$ by construction for any state $x \in \bigcup \{s \in S \mid a \in \mathcal{A}(s)\} \subset \mathbb{R}^n$.

Remark 3. *Other abstraction methods typically associate each $a \in \mathcal{A}$ with a fixed input $\hat{u} \in U$. Thus, the distribution $T(\cdot \mid x, \hat{u})$ over successor states associated with choosing action a depends on the precise state $x \in \mathbb{R}^n$. By contrast, we associate each abstract action $a \in \mathcal{A}$ with a fixed distribution $T(\cdot \mid x, B^\dagger(d_a - Ax))$ over successor states. Since $Ax + Bu + \eta = Ax + BB^\dagger(d_a - Ax) + \eta = d_a + \eta$, this distribution is the same for any state $x \in \mathbb{R}^n$ for which $a \in \mathcal{A}([x])$, i.e., where abstract actions a is enabled.*

2) Data-driven transition probabilities: As the distribution of the noise is unknown, we use a finite set of samples of η_k to compute an interval on the probability of reaching each location $s \in S$. Formally, let $\{\delta_1, \dots, \delta_N\} \in \Omega^N$, where $N \in \mathbb{N}$ is the number of samples[5] of η_k. For each pair $s, s' \in S$ and enabled action $a \in \mathcal{A}(s)$, the interval $[\check{P}_{s'}(s,a), \hat{P}_{s'}(s,a)]$ is computed such that the exact probability is contained with at least a desired probability of $1 - \beta$, with $\beta \in (0, 1)$:

$$\begin{aligned} \mathbb{P}^N \Big\{\ \check{P}_{s'}(s,a) \leq \int_{\mathbb{R}^n} \mathbb{1}_{s'}(\xi) T(d\xi \mid x, [B^\dagger(d_a - Ax)]) \\ \leq \hat{P}_{s'}(s,a)\ \Big\} \geq 1 - \beta, \end{aligned} \tag{12}$$

where x is such that $[x] = s$, state $s' \in S = \mathbb{R}^n/\sim$ is interpreted as a subset of $\mathbb{R}^n$, and the outer probability is taken w.r.t. the upper bound $\hat{P}_{s'}(s,a)$ and lower bound $\check{P}_{s'}(s,a)$ of the interval, which are random variables in the space Ω^N.

In practice, we compute these intervals using the method from [15], which leverages the scenario approach [22]. This method implicitly solves a set of $2N$ convex scenario programs with discarded constraints [32] and uses [23] to compute tight bounds on the probability of *constraint violation* for each of these programs. By construction, one of these $2N$ probabilities of constraint violations lower bounds the transition probability in Eq. (12), and another one is an upper bound. By choosing the confidence level on the probability of constraint violation for each scenario program as $1 - \frac{\beta}{2N}$, we obtain a probability interval such that Eq. (12) holds. It is shown in [15] that these scenario programs can be solved analytically based on its geometry, making the approach highly efficient. Due to space restrictions, we refer to [15, Theorem 1] for formal details.

3) Complete abstraction: Putting all elements together, we define the iMDP abstraction $(S, \bar{s}, \mathcal{A}, \check{P}, \hat{P})$, with

- Set of locations $S = \mathbb{R}^n/\sim$, with $\bar{s} = [\bar{x}]$;
- Actions $\mathcal{A} = \{a_1, \dots, a_q\}$, for $q \in \mathbb{N}$, with the enabled actions $\mathcal{A}(s)$ defined by (11) for all $s \in S$;
- For each $s, s' \in S$ and $a \in \mathcal{A}(s)$, the lower and upper bound probabilities $\check{P}_{s'}(s,a)$ and $\hat{P}_{s'}(s,a)$ are such that (12) holds for a desired value of $\beta \in (0, 1)$.

B. Controller synthesis

We show that the equivalence relation $\sim\ \subset \mathbb{R}^n \times S$ created by the partition (cf. Remark 2) is (with a certain probability) a probabilistic feedback refinement relation R from iMDP $\mathcal{I}$ to system $\mathcal{S}$, as defined by the conditions in Def. 5.

[5]Since (1) is time-invariant and has additive noise, we can obtain these samples from a *single* trajectory of length N starting at an arbitrary state $\bar{x}$.

Theorem 2 ([15, Thm. 2])**.** *For a given polyhedral partition that creates an equivalence relation $\sim$, let $\mathcal{I}$ be the iMDP abstraction for system $\mathcal{S}$ with $\beta \in (0,1)$. Then, it holds that $\mathbb{P}^N\{\mathcal{I} \preceq_\sim \mathcal{S}\} \geq 1 - \beta \cdot |\mathcal{A}| \cdot |S|$.*

Proof. The iMDP has at most $|\mathcal{A}| \cdot |S|$ unique probability intervals (see [15] for details). We have that $\mathcal{I} \preceq_\sim \mathcal{S}$ if all of these intervals contain the exact probability, which (by applying the union bound) is satisfied with a probability of at least $1 - \beta \cdot |\mathcal{A}| \cdot |S|$. Thus, the claim follows. □

Under Assumption 1, we can refine any policy for the abstract iMDP into a controller of the form in Def. 1.

Definition 8 (Controller refinement)**.** *Let $\pi \in \Pi$ be any iMDP policy. The refined controller $c\colon \mathbb{R}^n \times \{0,\dots,H\} \to U$ is piece-wise affine in $x \in \mathbb{R}^n$ and is defined for all $x \in \mathbb{R}^n$ as*

$$c(x,k) = B^\dagger(d_a - Ax), \;\; a \in \pi(s,k) \in \mathcal{A}(s), \tag{13}$$

where $s \in S$ is the iMDP location such that $[x]_\sim \in s$.[6]

Finally, we obtain the following key result for the iMDP abstraction and the refined controller defined above.

Theorem 3. *Let $\mathcal{S}$ be a stochastic linear system $\mathcal{S}$, φ a reach-avoid property, and $\sim$ the equivalence relation for a polyhedral partition. Then, for the iMDP abstraction $\mathcal{I}$, a reach-avoid property $\varphi_\mathcal{I}$ such that $\varphi_\mathcal{I} \preceq_\sim \varphi$, and any policy $\pi \in \Pi_\mathcal{I}$ with refined controller c (as per Def. 8), it holds that*

$$\mathbb{P}^N\Big\{ \Pr^c_\mathcal{S}(x_0 \models \varphi) \geq \min_{P \in \mathcal{P}} \Pr^\pi_{\mathcal{I}[P]}(\bar{s} \models \varphi_\mathcal{I}) \Big\} \geq 1 - \beta \cdot |\mathcal{A}| \cdot |S|.$$

Thus, the satisfaction probability on the iMDP is a *lower bound* on the satisfaction probability for system $\mathcal{S}$ under the refined controller, with probability at least $1 - \beta \cdot |\mathcal{A}| \cdot |S|$.

IV. Exploiting Stability in Abstraction

The size of the iMDP abstraction (which can be expressed by the number of edges, or transitions, in the underlying graph) from Sect. III grows exponentially with the dimension of the state space. In this section, we develop an extension to the method from [14] to create smaller abstractions. As our key contribution, we leverage the two-layer control design framework in Fig. 1b, which first stabilizes the dynamics and then creates an abstraction of the closed-loop dynamics. Specifically, we use the feedback control law given by

$$u_k = -Kx_k + u'_k, \tag{14}$$

where the gain matrix $K \in \mathbb{R}^{m \times n}$ represents a stabilizing control law, and u' is the control input captured by the abstraction. In this paper, we obtain the feedback gain matrix by solving an instance of a linear quadratic regulator (LQR) [16] control problem. Applying the feedback control law in (14) to system (1) yields the closed-loop dynamics given by

$$x_{k+1} = A_{\text{cl}}x_k + Bu'_k + \eta_k, \tag{15}$$

where $A_{\text{cl}} = A - BK$. We assume that the feedback gain K satisfies the input constraints in the following way.

[6] If $x \in \mathbb{R}^n$ is on the boundary of multiple partition elements, the refined controller can select any location $s = [x]_\sim \in S$.

Assumption 3. *The gain matrix $K \in \mathbb{R}^{m \times n}$ is such that $-Kx \in U$ for all $x \in \mathcal{X}$ and the matrix A_{cl} is non-singular.*

A. Backward reachable set for stabilized dynamics

We show how the iMDP abstraction described in Sect. III can be employed together with the two-layer feedback control law in (14). The key step is that we modify the backward reachable set computation in (10), replacing it by

$$\begin{aligned}\mathcal{R}^{-1}_{\text{cl}}(d_a, U') = \big\{x \in \mathbb{R}^n : d_a = A_{\text{cl}}x + Bu', \\ -Kx + u' \in U,\, u' \in U'\big\},\end{aligned} \tag{16}$$

where the constraint $-Kx + u' \in U = \{u \in \mathbb{R}^m : Gu \leq h\}$ enforces that the total input u is admissible, and the constraint $u' \in U' = \{u \in \mathbb{R}^m : G'u \leq h'\}$ controls the size of the abstraction. Matrices G and G' and vectors h and h' define the admissible control inputs; their sizes are omitted for brevity.

Assumption 4. *The set U' contains the origin, i.e., $0 \in U'$.*

Observe that Eq. (16) is of the same form as Eq. (10) (despite imposing additional constraints) and can thus be computed similarly, as shown by the following lemma.

Lemma 1. *Under Assumptions 3 and 4, the following holds:*

i) For any $d_a \in \mathbb{R}^n$, the set $\mathcal{R}^{-1}_{\text{cl}}(d_a, U')$ is non-empty;

ii) $\mathcal{R}^{-1}_{\text{cl}}(d_a, U') = \{x \in \mathbb{R}^n \mid d_a = A_{\text{cl}}x + Bu',\, u' \in \tilde{U}\}$, where $\tilde{U} \subset \mathbb{R}^m$ is a convex polytope defined as

$$\tilde{U} = \big\{u \in \mathbb{R}^m \,:\, G(\alpha + \beta u) \leq h,\; G'u \leq h'\big\}, \tag{17}$$

with $\alpha = -KA^{-1}_{\text{cl}}d_a$ and $\beta = I + KA^{-1}_{\text{cl}}B$, where I the identity matrix of appropriate size.

Proof. Item i): We will show that the point $\tilde{x} = A^{-1}_{\text{cl}}d_a \in \mathcal{R}^{-1}_{\text{cl}}(d_a, U')$. Note that this point $\tilde{x}$ is obtained for $u' = 0$ in (16), which is an admissible input due to Assumption 4. Moreover, due to Assumption 3, we have that $-Kx \in U$, and thus, it holds that $\tilde{x} \in \mathcal{R}^{-1}_{\text{cl}}(d_a, U')$, which concludes the proof of item i). Item ii): Solving the equality constraint in (16) for x yields $x = A^{-1}_{\text{cl}}(d_a - Bu)$, so the input constraint $-Kx + u' \in U$ can be written as

$$-KA^{-1}_{\text{cl}}d_a + (I + KA^{-1}_{\text{cl}}B)u' = \alpha + \beta u' \in U. \tag{18}$$

Thus, we have two convex polyhedral constraints on u, given by $\alpha + \beta u' \in U = \{u \in \mathbb{R}^m : Gu \leq h\}$ and $u' \in U' = \{u \in \mathbb{R}^m : G'u \leq h'\}$. The intersection of the feasible sets for u is the set $\tilde{U}$ in (17), concluding the proof of item ii). □

Observe that, while we can compute $\mathcal{R}^{-1}_{\text{cl}}(d_a, U')$ similarly as in Sect. III, the number of vertices to consider is generally higher due to the additional input constraint $u' \in U'$.

B. Constructing smaller abstractions

We can use the modified backward reachable set in (16) to construct abstractions with fewer enabled actions, as illustrated by the following lemma.

Lemma 2. *Consider the backward sets defined by* (10) *and* (16)*. If $U' = \mathbb{R}^p$ in* (16)*, then we have that $\mathcal{R}^{-1}_{\text{cl}}(d_a, U') = \mathcal{R}^{-1}(d_a)$. Moreover, for any two subsets $U'' \subset U' \subseteq \mathbb{R}^p$, it holds that $\mathcal{R}^{-1}_{\text{cl}}(d_a, U'') \subseteq \mathcal{R}^{-1}_{\text{cl}}(d_a, U')$.*

Proof. Letting $U' = \mathbb{R}^p$ and $A_{\text{cl}} = A - BK$ in (16) gives

$$\begin{aligned}\mathcal{R}_{\text{cl}}^{-1}(d_a, U') &= \{x \in \mathbb{R}^n : d_a = Ax + B(-Kx + u'), \\ &\qquad -Kx + u' \in U\} \\ &= \{x \in \mathbb{R}^n : d_a = Ax + Bu,\ u \in U\} \\ &= \mathcal{R}^{-1}(d_a),\end{aligned}$$

thus proving the first claim. For $U'' \subset U'$, observe from (16) that $u' \in U'' \subset U'$. Thus, we obtain that $\mathcal{R}_{\text{cl}}^{-1}(d_a, U'') \subseteq \mathcal{R}_{\text{cl}}^{-1}(d_a, U')$, which proves the second claim. □

Recall from Sect. III that an action $a \in \mathcal{A}$ is enabled in location $s \in S$ if and only if the corresponding partition element is contained in the backward reachable set $\mathcal{R}^{-1}(d_a)$. In the modified backward reachability set in (16), we can control the size of $\mathcal{R}_{\text{cl}}^{-1}(d_a)$ through U'. In fact, Lemma 2 shows that by shrinking the set U', we may reduce the number of enabled actions at each state and, consequently, the size of the graph of the iMDP. In the next section, we show how suitable choices for the feedback gain K and input constraint U' may lead to significantly smaller iMDP abstractions.

V. Numerical Experiments

We implement our method in a Python tool, which is available at `https://github.com/LAVA-LAB/DynAbs`. We use the model checker PRISM [33] to compute optimal policies as per (5) for iMDPs. In all experiments, we apply Theorem 3 with an overall confidence of $1-\beta\cdot|\mathcal{A}|\cdot|S| = 0.99$. For simplicity, we use partitions into rectangular regions.

A. Double integrator

Consider a stochastic system with dynamics given as

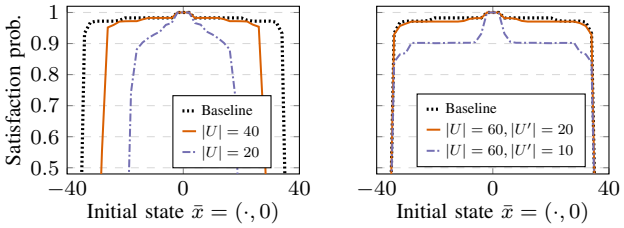
$$x_{k+1} = \begin{bmatrix} \frac{1}{\rho^2} & \frac{1+\rho}{+}1 \\ 0 & 1 \end{bmatrix} x_k + \begin{bmatrix} \frac{0.5+\rho}{\rho} & 0.5 \\ 1 & 1 \end{bmatrix} (-Kx_k + u'_k) + \eta_k,$$

where we apply the control law given in (14), and the noise $\eta_k \sim \mathcal{N}(0, I_2)$ has a standard normal distribution and satisfies the conditions in Assumption 2. We select $\rho = 2$ to render the system unstable when a trivial control of $-Kx_k + u'_k = 0$ is applied. The reach-avoid task is to reach a state $x \in [-3, 3]^2$ while avoiding states $x \notin [-41, 41]^2$ within $H = 16$ steps.[7] We partition the set $\mathcal{X} = [-41, 41]^2$ into 41 by 41 rectangular regions of width two. The input constraint is $U = [-60, 60]^2$.

Baseline: As a baseline, we set $K = 0$ and construct the single-layer iMDP abstraction (as outlined in Sect. III) with input constraint $U = [-60, 60]^2$. The resulting iMDP has 39.8 million transitions, and the lower bounds on the satisfaction probabilities (obtained from Theorem 3) are shown in Fig. 2 for a range of initial states $\bar{x} = (x_1, 0)$ for $x_1 \in [-41, 41]$.

Stabilizing controller: We now use our two-layer abstraction scheme, where we compute the gain K with an LQR with cost matrices $Q = R = I_2$, yielding $(A - BK)$ having eigenvalues of $\lambda = 0.178 \pm 0.136i$. We construct the iMDP for the different sets U and U' shown in Table I, presenting the lower bound satisfaction probabilities for two cases in Fig. 2. With our method, we construct significantly smaller abstractions with little loss in probabilistic guarantee. For example, with $|U| = 60$, $|U'| = 20$, the number of iMDP transitions is reduced from 39.8 to 15.8 million at negligible loss in probabilistic guarantee. Shrinking the set U' further reduces the iMDP size; however, at the cost of a considerable reduction in probabilistic guarantee, as shown in Fig. 2a.

[7] The correctness of our iMDP abstraction is independent of the horizon. For numerical experiments with an infinite time horizon, we refer to [15].



(a) Single-layer abstraction. (b) Two-layer abstraction.

Fig. 2: Lower bound satisfaction probabilities (obtained from Theorem 3) for the integrator experiment with initial conditions $\bar{x} = (x_1, 0)$ for all $-41 \leq x_1 \leq 41$.

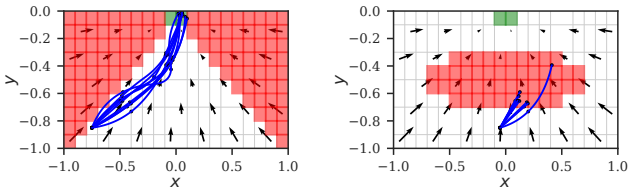
TABLE I: Number of iMDP transitions for the integrator experiment. The highlighted rows are those shown in Fig. 2.

Stabilized?	U	U'	iMDP transitions
No (baseline)	$[-60, 60]^2$	n.a.	39 773 745
No	$[-40, 40]^2$	n.a.	21 289 058
No	$[-20, 20]^2$	n.a.	5 219 518
Yes	$[-60, 60]^2$	$[-30, 30]^2$	25 671 576
Yes	$[-60, 60]^2$	$[-20, 20]^2$	15 757 546
Yes	$[-60, 60]^2$	$[-10, 10]^2$	3 996 029
Yes	$[-40, 40]^2$	$[-20, 20]^2$	11 691 267
Yes	$[-40, 40]^2$	$[-10, 10]^2$	2 895 878
Yes	$[-20, 20]^2$	$[-10, 10]^2$	1 034 996

B. Spacecraft docking

We consider the spacecraft docking problem from [34], with $x \in \mathbb{R}^4$ modeling the position and velocity in two dimensions (see [34] for the full model). We illustrate that our method generally works well if the stabilizing feedback controller is aligned with the reach-avoid property. That is, the stabilizing controller should *steer* the state x towards the goal region $X_G \subset \mathbb{R}^4$, while steering clear from the unsafe states $X_U \subset \mathbb{R}^4$. We consider the two reach-avoid problems shown in Fig. 3 (only the position state variables are shown). As a baseline, we construct the abstraction with a partition into 3 200 elements and an input constraint $U = [-0.1, 0.1]^2$. For the reach-avoid problem in Fig. 3a, the resulting iMDP has 1.6 million transitions and leads to a lower bound satisfaction probability of 0.80 in Theorem 3. Similarly, for the problem in Fig. 3b, the iMDP has 2.1 million transitions and leads to a lower bound satisfaction probability of 0.86.

Now, we apply our method with $U' = [-0.08, 0.08]^2$, resulting in iMDPs with 280 and 330 thousand transitions (reductions of 79% and 85% respectively). For the problem in Fig. 3a, the lower bound on the satisfaction probability is 0.79 (only 0.01 below the baseline). However, for Fig. 3b, the bound drops to 0.0072, i.e., almost zero. To explain this



(a) Aligned property. (b) Disaligned property.

Fig. 3: Simulated trajectories and stabilized vector fields $(A - BK)x$ for both reach-avoid properties considered in the spacecraft problem (goal states in green; unsafe states in red). Case (a) is aligned, while case (b) is not.

severe performance loss, we show simulated trajectories under the refined controller (as per Def. 8) in Fig. 3. Moreover, the arrows show the vector field under the stabilized dynamics, i.e., $(A - BK)x$ for different $x \in \mathbb{R}^n$. In Fig. 3a, the vector field points to the goal region and away from unsafe states, and is thus aligned with the property. By contrast, the vector field in Fig. 3b is not aligned since it steers the system into unsafe states, causing a performance loss of the controller.

VI. Conclusion

In this paper, we have developed a novel formal abstraction method for stochastic linear dynamical systems that exploits stability to generate smaller abstract models. By stabilizing the dynamics with a linear feedback gain first, we have shown that we can reduce the size of abstractions (in terms of the number of edges in the underlying graph) significantly. Our experiments have shown that, when the feedback gain steers the system toward the goal states (and away from the unsafe states), we can reduce the number of transitions by up to 90% with negligible performance loss.

However, if the stabilizing controller is not aligned with the control task (as in Fig. 3b), the controller performance degrades significantly. One possible solution is to use a piecewise affine trajectory-tracking controller, which selects different gains in different regions of the state space. Exploring this latter approach will be the next step of our research.

References

[1] A. Lavaei, S. Soudjani, A. Abate, and M. Zamani, "Automated verification and synthesis of stochastic hybrid systems: A survey," *Autom.*, vol. 146, p. 110617, 2022.

[2] A. Abate, M. Prandini, J. Lygeros, and S. Sastry, "Probabilistic reachability and safety for controlled discrete time stochastic hybrid systems," *Autom.*, vol. 44, no. 11, pp. 2724–2734, 2008.

[3] M. Lahijanian, S. B. Andersson, and C. Belta, "Formal verification and synthesis for discrete-time stochastic systems," *IEEE Trans. Autom. Control.*, vol. 60, no. 8, pp. 2031–2045, 2015.

[4] P. Tabuada, *Verification and Control of Hybrid Systems - A Symbolic Approach*. Springer, 2009.

[5] G. Reissig, A. Weber, and M. Rungger, "Feedback refinement relations for the synthesis of symbolic controllers," *IEEE Trans. Autom. Control.*, vol. 62, no. 4, pp. 1781–1796, 2017.

[6] A. Makdesi, A. Girard, and L. Fribourg, "Efficient data-driven abstraction of monotone systems with disturbances," in *ADHS*, vol. 54 of *IFAC-PapersOnLine*, pp. 49–54, Elsevier, 2021.

[7] R. Coppola, A. Peruffo, and M. Mazo Jr., "Data-driven abstractions for verification of linear systems," *IEEE Control. Syst. Lett.*, vol. 7, pp. 2737–2742, 2023.

[8] A. Lavaei and E. Frazzoli, "Data-driven synthesis of symbolic abstractions with guaranteed confidence," *IEEE Control. Syst. Lett.*, vol. 7, pp. 253–258, 2023.

[9] M. Kazemi, R. Majumdar, M. Salamati, S. Soudjani, and B. Wooding, "Data-driven abstraction-based control synthesis," *Nonlinear Analysis: Hybrid Systems*, vol. 52, p. 101467, 2024.

[10] K. Hashimoto, A. Saoud, M. Kishida, T. Ushio, and D. V. Dimarogonas, "Learning-based symbolic abstractions for nonlinear control systems," *Autom.*, vol. 146, p. 110646, 2022.

[11] S. Sadraddini and C. Belta, "Formal guarantees in data-driven model identification and control synthesis," in *HSCC*, pp. 147–156, ACM, 2018.

[12] A. Devonport, A. Saoud, and M. Arcak, "Symbolic abstractions from data: A PAC learning approach," in *CDC*, pp. 599–604, IEEE, 2021.

[13] A. Banse, L. Romao, A. Abate, and R. M. Jungers, "Data-driven abstractions via adaptive refinements and a Kantorovich metric [extended version]," *CoRR*, vol. abs/2303.17618, 2023.

[14] 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 *AAAI*, pp. 9669–9678, AAAI Press, 2022.

[15] T. S. 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," *J. Artif. Intell. Res.*, vol. 76, pp. 341–391, 2023.

[16] G. F. Franklin, J. D. Powell, and A. Emami-Naeini, *Feedback control of dynamic systems*, vol. 8. Pearson, 2019.

[17] A. Girard, "Approximately bisimilar finite abstractions of stable linear systems," in *HSCC*, vol. 4416 of *LNCS*, pp. 231–244, Springer, 2007.

[18] A. Girard, G. Pola, and P. Tabuada, "Approximately bisimilar symbolic models for incrementally stable switched systems," *IEEE Trans. Autom. Control.*, vol. 55, no. 1, pp. 116–126, 2010.

[19] P. Tabuada, "An approximate simulation approach to symbolic control," *IEEE Trans. Autom. Control.*, vol. 53, no. 6, pp. 1406–1418, 2008.

[20] G. Pola, A. Girard, and P. Tabuada, "Approximately bisimilar symbolic models for nonlinear control systems," *Autom.*, vol. 44, no. 10, pp. 2508–2516, 2008.

[21] M. Zamani, G. Pola, M. Mazo Jr., and P. Tabuada, "Symbolic models for nonlinear control systems without stability assumptions," *IEEE Trans. Autom. Control.*, vol. 57, no. 7, pp. 1804–1809, 2012.

[22] M. C. Campi, A. Carè, and S. Garatti, "The scenario approach: A tool at the service of data-driven decision making," *Annu. Rev. Control.*, vol. 52, pp. 1–17, 2021.

[23] L. Romao, A. Papachristodoulou, and K. Margellos, "On the exact feasibility of convex scenario programs with discarded constraints," *IEEE Trans. Autom. Control.*, vol. 68, no. 4, pp. 1986–2001, 2023.

[24] R. Durrett, *Stochastic Calculus: A Practical Introduction*. CRC Press, 1st ed., 1996.

[25] D. Salamon, *Measure and Integration*. European Mathematical Society, 2016, 2016.

[26] D. P. Bertsekas and S. E. Shreve, *Stochastic Optimal Control: The Discrete-time Case*. Athena Scientific, 1978.

[27] C. Baier and J. Katoen, *Principles of model checking*. MIT Press, 2008.

[28] A. Puggelli, W. Li, A. L. Sangiovanni-Vincentelli, and S. A. Seshia, "Polynomial-time verification of PCTL properties of MDPs with convex uncertainties," in *CAV*, vol. 8044 of *LNCS*, pp. 527–542, Springer, 2013.

[29] H. Hermanns, A. Parma, R. Segala, B. Wachter, and L. Zhang, "Probabilistic logical characterization," *Inf. Comput.*, vol. 209, no. 2, pp. 154–172, 2011.

[30] L. Rickard, T. S. Badings, L. Romao, and A. Abate, "Formal controller synthesis for Markov jump linear systems with uncertain dynamics," in *QEST*, vol. 14287 of *LNCS*, pp. 10–29, Springer, 2023.

[31] C. Belta, B. Yordanov, and E. A. Gol, *Formal methods for discrete-time dynamical systems*, vol. 15. Springer, 2017.

[32] M. C. Campi and S. Garatti, "A sampling-and-discarding approach to chance-constrained optimization: Feasibility and optimality," *J. Optim. Theory Appl.*, vol. 148, no. 2, pp. 257–280, 2011.

[33] M. Z. Kwiatkowska, G. Norman, and D. Parker, "PRISM 4.0: Verification of probabilistic real-time systems," in *CAV*, vol. 6806 of *LNCS*, pp. 585–591, Springer, 2011.

[34] A. P. Vinod, J. D. Gleason, and M. M. K. Oishi, "Sreachtools: a MATLAB stochastic reachability toolbox," in *HSCC*, pp. 33–38, ACM, 2019.