Proceedings of Machine Learning Research vol vvv:1–24, 2025

Temporal Logic Control for Nonlinear Stochastic Systems Under Unknown Disturbances

Ibon Gracia IBON.GRACIA@COLORADO.EDU
University of Colorado Boulder

Luca Laurenti L.LAURENTI@TUDELFT.NL
Delft University of Technology

Manuel Mazo Jr. M.MAZO@TUDELFT.NL
Delft University of Technology

Alessandro Abate ALESSANDRO.ABATE@CS.OX.AC.UK
University of Oxford

Morteza Lahijanian MORTEZA.LAHIJANIAN@COLORADO.EDU
University of Colorado Boulder

Abstract

In this paper, we present a novel framework to synthesize robust strategies for discrete-time nonlinear systems with random disturbances that are unknown, against temporal logic specifications. The proposed framework is data-driven and abstraction-based: leveraging observations of the system, our approach learns a high-confidence abstraction of the system in the form of an uncertain Markov decision process (UMDP). The uncertainty in the resulting UMDP is used to formally account for both the error in abstracting the system and for the uncertainty coming from the data. Critically, we show that for any given state-action pair in the resulting UMDP, the uncertainty in the transition probabilities can be represented as a convex polytope obtained by a two-layer state discretization and concentration inequalities. This allows us to obtain tighter uncertainty estimates compared to existing approaches, and guarantees efficiency, as we tailor a synthesis algorithm exploiting the structure of this UMDP. We empirically validate our approach on several case studies, showing substantially improved performance compared to the state-of-the-art.

Keywords: Data-Driven Control, Strategy Synthesis, Uncertain MDPs, Safe Autonomy

1. Introduction

The synthesis of safe strategies for stochastic systems is critical in ensuring *reliable* and *safe* operations in domains such as robotics, autonomous vehicles, and cyber-physical systems (Belta et al., 2007; Lavaei et al., 2022). A key challenge arises when the system dynamics include *unknown* random disturbances, making it difficult to account for uncertainties while guaranteeing performance against high-level complex specifications. Existing methods often assume known distributions for the disturbances or rely on abstractions with overly conservative uncertainty estimates, limiting their scalability and applicability to complex systems. This paper aims to address these gaps by presenting a novel framework to synthesize optimal strategies for nonlinear stochastic systems with unknown disturbances, ensuring both formal guarantees and computational efficiency.

Our framework employs a data-driven, abstraction-based approach to strategy synthesis for stochastic systems with unknown noise under *linear temporal logic over finite traces* (LTL_f) (De Giacomo and Vardi, 2013) specifications. Starting with data from the system's trajectories, we construct a high-confidence abstraction in the form of an uncertain Markov decision process (UMDP)

(Iyengar, 2005), a flexible model that captures complex uncertainties. Unlike existing methods relying on interval-based abstractions or conservative assumptions, our framework represents transition probability uncertainties as convex polytopes. These sets are derived through a novel two-layer discretization scheme and learning the support of the unknown disturbance. This leads to tighter uncertainty sets and less conservative results compared to existing methods. Exploiting this UMDP structure, we introduce a synthesis algorithm for LTL_f specifications that simplifies the computation, reducing the complexity of standard UMDP linear programming approaches. By incorporating uncertainties from both abstraction errors and data limitations, our framework yields a strategy that is robust. Our empirical evaluations over various types of systems reveals the efficacy of this approach over existing methods, namely in data efficiency, tightness of results, and scalability.

The main contributions of this paper are fourfold: (i) a novel framework for synthesizing strategies for nonlinear stochastic systems under non-additive, unknown disturbances with LTL_f specifications, (ii) a distribution-agnostic, data-driven construction of UMDP abstraction with a specific structure that reduces conservatism of existing abstraction-based techniques, (iii) a tailored synthesis algorithm for this UMDP abstraction that is both efficient and results in tight error bounds, and (iv) a series of case studies and benchmarks that show superiority of the framework over the state-of-the-art, with up to 3 orders of magnitude improvement in sample complexity and an order of magnitude reduction in computation time.

Related Work Abstractions of stochastic systems to finite Markov decision processes (MDPs) are powerful tools for controller synthesis on highly-complex systems under complex logic specifications (Lavaei et al., 2022). In particular, Interval MDPs (IMDPs) (Lahijanian et al., 2015; Givan et al., 2000) abstract systems by presenting uncertain transition probabilities within intervals, capturing the full range of system behaviors. For example, (Cauchi et al., 2019) efficiently abstracts linear systems with additive Gaussian noise, while (Skovbekk et al., 2023) extends this to nonlinear dynamics. Uncertain MDPs (UMDPs) (Iyengar, 2005; El Ghaoui and Nilim, 2005) generalize IMDPs by allowing transition probabilities to belong to more complex sets and have been used for strategy synthesis against specifications such as linear temporal logic (LTL) (Wolff et al., 2012). However, these abstractions typically require system models, which are often unavailable in practice.

To address model uncertainty, various methods leverage Gaussian processes (Jackson et al., 2021b), neural networks (Adams et al., 2022), and ambiguity sets (Gracia et al., 2022), which are then abstracted as IMDPs or UMDPs. Statistical tools like the scenario approach have also been used to abstract stochastic (Badings et al., 2023b,a), non-deterministic (Kazemi et al., 2024), and deterministic systems (Coppola et al., 2022, 2023). Also, techniques such as super-martingales and barrier functions enable safety verification and control synthesis for general dynamics (Lechner et al., 2022; Badings et al., 2024; Mazouz et al., 2024). Nevertheless, all these works assume that the disturbance distribution is known.

When disturbance distributions are uncertain, some works combine IMDP abstractions with statistical tools like the scenario approach (Badings et al., 2023b,a; Schön et al., 2023), while others employ barrier certificates with the scenario approach for safety verification (Mathiesen et al., 2023). Another approach constructs Wasserstein ambiguity sets from data samples to abstract systems as UMDPs, ensuring high-confidence containment of unknown distributions (Gracia et al., 2022). However, these methods typically assume simple dynamics or additive noise. For general dynamics with unknown disturbance distributions, only a few works exist. (Salamati et al., 2021) uses barrier certificates for safety verification, and (Gracia et al., 2024) extends the ambiguity set

approach for LTL control synthesis. Both assume that certain distribution-related properties, such as variance or support, are known—an assumption often unrealistic in practice, and also suffer from high sample complexity, especially under high-confidence requirements. Our work overcomes these limitations by removing assumptions about disturbance distributions and offering a data-efficient and scalable approach suitable for systems with general dynamics.

2. Problem Formulation

In this work the focus is on discrete-time stochastic systems given by

$$\boldsymbol{x}_{k+1} = f(\boldsymbol{x}_k, u_k, \boldsymbol{w}_k), \tag{1}$$

where $\boldsymbol{x}_k \in \mathbb{R}^n$ denotes the state at time $k \in \mathbb{N}_0$, $u_k \in U \subset \mathbb{R}^m$ is the control input chosen from a finite set U, and $\boldsymbol{w}_k \in W \subseteq \mathbb{R}^d$ is the disturbance. The latter is a sequence of independent and identically distributed (i.i.d.) random variables on the probability space $(W, \mathcal{B}(W), P)$, with $\mathcal{B}$ being the Borel σ-algebra on W, and where the support W and probability distribution P of $\boldsymbol{w}_k$ are *unknown*. The vector field (possibly nonlinear) $f : \mathbb{R}^n \times U \times W \to \mathbb{R}^n$ is assumed to be Lipschitz continuous on its third argument, uniformly for all values of its first argument on some set.

Assumption 1 *There exists a set $X \subset \mathbb{R}^n$, such that, for every $u \in U$, there exists a constant $L_u > 0$ such that, for all $x \in X$, $w, w' \in W$, it holds that $\|f(x,u,w) - f(x,u,w')\| \leqslant L_u \|w - w'\|$.*

In lieu of unknown W and P, we assume a dataset on the disturbance is available.

Assumption 2 *A set $\{\hat{\boldsymbol{w}}^{(i)}\}_{i=1}^N$ of N i.i.d. samples from P is available.*

Assumption 2 is commonly made in related work (Gracia et al., 2024) and can be practically satisfied through, e.g., observations of the state and control. The straightforward example is when f is affine in $\boldsymbol{w}_k$; otherwise, it suffices for f to be injective over only a subset of $\mathbb{R}^n$ as discussed in (Gracia et al., 2024). This condition is met by many practical systems, including those in our case studies.

Given $x_0, \ldots, x_K \in \mathbb{R}^n$, $u_0, \ldots, u_{K-1} \in U$, and $K \geqslant 0$, we denote a finite *trajectory* of System (1) by $\omega_x = x_0 \xrightarrow{u_0} \ldots \xrightarrow{u_{K-1}} x_K$. We let $|\omega_x|$ denote the length of ω_x, define Ω_x as the set of all trajectories with $|\omega_x| < \infty$ and denote by $\omega_x(k)$ the state of ω_x at time $k \in \{0, \ldots, |\omega_x|\}$. A *strategy* of System (1) is a function $\sigma_x : \Omega_x \to U$ that assigns a control u to each finite trajectory ω_x. Given $x \in \mathbb{R}^n$, $u \in U$, the *transition kernel* $T : \mathcal{B}(\mathbb{R}^n) \times \mathbb{R}^n \times U \to [0,1]$ of System (1) assigns the probability $T(B \mid x, u) = \int_W \mathbb{1}_B(f(x,u,w))P(dw)$, where the indicator function $\mathbb{1}_B(x) = 1$ if $x \in B$, and 0 otherwise, to each Borel set $B \in \mathcal{B}(\mathbb{R}^n)$. For a strategy σ_x and an initial condition $x_0 \in \mathbb{R}^n$, the transition kernel defines a unique probability measure $P_{x_0}^{\sigma_x}$ over the trajectories of System (1) (Bertsekas and Shreve, 1996). In this way, $P_{x_0}^{\sigma_x}[\boldsymbol{\omega}_x(k) \in X]$ denotes the probability that $\boldsymbol{x}_k$ belongs to the set $X \subseteq \mathbb{R}^n$ when following strategy σ_x from initial state x_0. In this work, we are interested in the temporal behavior of System (1) w.r.t. a bounded (*safe*) set $X \subset \mathbb{R}^n$ and a set of regions of interest R_{int}, with $r \subseteq X$ for all $r \in R_{\text{int}}$. We denote by $r_{\text{unsafe}} = \mathbb{R}^n \backslash X$ the unsafe region. We consider a set $AP := \{\mathfrak{p}_1, \ldots, \mathfrak{p}_{|AP|-1}, \mathfrak{p}_{\text{unsafe}}\}$ of *atomic propositions*, and associate a subset of atomic propositions to each region $r \in R_{\text{int}} \cup \{r_{\text{unsafe}}\}$. We define the *labeling function* $L : \mathbb{R}^n \to 2^{AP}$ as the function that maps each state $x \in \mathbb{R}^n$ to the atomic propositions that are true in the region where x lies, e.g., if we associate $\{\mathfrak{p}_1\}$ to region r_1, we conclude that $\mathfrak{p}_1$ is *true* at x, denoted $\mathfrak{p}_1 \equiv \top$, if $x \in r_1$. In consequence, each trajectory $\omega_x = x_0 \xrightarrow{u_0} \ldots \xrightarrow{u_{K-1}} x_K$ results in the (observation) *trace* $\rho = \rho_0 \ldots \rho_K$, where $\rho_k := L(x_k)$.

In order to formally characterize behaviors of System (1), we use *linear temporal logic over finite traces* (LTL$_f$) (De Giacomo and Vardi, 2013), which generalizes Boolean logic to temporal behaviors. An LTL$_f$ property φ is a logical formula defined over atomic proposition AP using Boolean connectives "negation" ($\neg$) and "conjunction" ($\wedge$), and the temporal operators "until" ($\mathcal{U}$) and "next" ($\bigcirc$). The syntax of formula φ is recursively defined as

$$\varphi := \top \mid \mathfrak{p} \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \bigcirc\varphi \mid \varphi_1 \mathcal{U} \varphi_2,$$

where $\mathfrak{p} \in AP$ and φ_1, φ_2 are also LTL$_f$ formulas. The temporal operators "eventually" ($\Diamond$) and "globally" ($\Box$) are derived from the above syntax as $\Diamond\varphi := \top\mathcal{U}\varphi$ and $\Box\varphi := \neg\Diamond(\neg\varphi)$. LTL$_f$ formulae are semantically interpreted over finite traces (De Giacomo and Vardi, 2013). We say a trajectory ω_x satisfies a formula φ, i.e., $\omega_x \models \varphi$, if some prefix of its trace ρ satisfies φ.

Our goal is to synthesize a strategy for System (1) to ensure satisfaction of a given LTL$_f$ formula φ. However, note that (i) under a given strategy, the satisfaction of φ is probabilistic, and (ii) in our setting, the distribution of the disturbance is unknown. Hence, we aim to leverage data samples to generate a strategy that guarantees System (1) satisfies φ with high probability. Furthermore, note that the synthesized strategy must account for the learning gap due to the lack of knowledge of P.

Problem 1 *Consider stochastic System (1), a set $\{\hat{\boldsymbol{w}}^{(i)}\}_{i=1}^{N}$ of N i.i.d. samples from P, a bounded set $X \subset \mathbb{R}^n$ on which Assumption 1 holds, and an LTL$_f$ formula φ defined over the regions of interest R. Given a confidence level $1-\alpha \in (0,1)$, synthesize a strategy σ_x and a high probability bound function $\underline{p} : X \to [0,1]$ such that, with confidence at least $1-\alpha$, for every initial state $x_0 \in X$, σ_x guarantees that the probability that the paths $\omega_x \in \Omega_x$ satisfy φ while remaining in X is lower bounded by $\underline{p}(x_0)$, i.e.,*

$$P_{x_0}^{\sigma_x}[\omega_x \models \varphi \wedge \Box\neg\mathfrak{p}_{unsafe}] \geqslant \underline{p}(x_0).$$

We emphasize that the noise distribution P is unknown, and no assumptions are imposed on it. Instead, since only samples are available, the probabilistic guarantees for the closed-loop system must hold with a *confidence*. This confidence is related to the probability that the N samples are representative of P and is interpreted in the frequentist sense: if the process of obtaining N samples from P and synthesizing the strategy is repeated infinitely many times, the condition $P_{x_0}^{\sigma_x}[\omega_x \models \varphi \wedge \Box\neg\mathfrak{p}_{\text{unsafe}}] \geqslant \underline{p}(x_0)$ for all $x_0 \in X$ holds in at least $1-\alpha$ of the cases.

Overview of the approach Given the uncountable nature of the state-space of System (1) and the unknown distribution P, solving Problem 1 exactly is infeasible. Therefore, we adopt an abstraction-based approach. This method provides a strategy along with a conservative, high-probability bound for every initial state. The abstraction is an uncertain Markov decision process (UMDP) constructed from a finite discretization of set X. We learn the transition relations between the discrete regions using System (1) and disturbance samples, capturing the system's behavior with confidence $1-\alpha$. Our UMDP construction is specifically designed to tightly capture the learning uncertainty. Then, we devise a strategy synthesis algorithm based on robust dynamic programming to (robustly) maximize the probability of satisfying φ on this UMDP. Next, we refine the obtained strategy to System (1) such that it guarantees the closed-loop system satisfies φ with a probability higher than the one obtained for the abstraction with confidence $1-\alpha$.

3. Preliminaries on Uncertain Markov Decision Processes

An uncertain MDP (UMDP), also known as a robust MDP, is a stochastic system that generalizes the MDP class by allowing its transition probability distributions to be uncertain, taking values from a set (Iyengar, 2005; El Ghaoui and Nilim, 2005; Wiesemann et al., 2013).

Definition 1 (Uncertain MDP) *A labeled uncertain Markov decision process (UMDP) $\mathcal{M}$ is a tuple $\mathcal{M} = (S, A, \Gamma, s_0, AP, L)$, where S and A are finite sets of states and actions, respectively, $s_0 \in S$ is the initial state, S and A are finite sets of states and actions, respectively, $s_0 \in S$ is the initial state, $\Gamma = \{\Gamma_{s,a} \subseteq \mathcal{P}(S) : s \in S, a \in A\}$, where $\mathcal{P}(S)$ is the set of probability distributions over S, and $\Gamma_{s,a}$ is a nonempty set of transition probability distributions for state $s \in S$ and action $a \in A$, AP is a finite set of atomic propositions, and $L : S \to 2^{AP}$ denotes the labeling function.*

A finite *path* of UMDP $\mathcal{M}$ is a sequence $\omega = s_0 \xrightarrow{a_0} s_1 \xrightarrow{a_1} \dots \xrightarrow{a_{K-1}} s_K$ of states $s_k \in S$ and actions $a_k \in A$ such that there exists $\gamma \in \Gamma_{s_k,a_k}$ with $\gamma(s_{k+1}) > 0$ for all $k \in \{0, \dots, K-1\}$. We denote by Ω the set of all finite paths. Given a path $\omega \in \Omega$, $\omega(k) = s_k$ is the state of ω at time $k \in \{0, \dots, K\}$, and we denote its last state by $\text{last}(\omega)$. A *strategy* of a UMDP $\mathcal{M}$ is a function $\sigma : \Omega \to A$ that maps each finite path to the next action. We denote by Σ the set of all strategies of $\mathcal{M}$. Given path $\omega \in \Omega$ and $\sigma \in \Sigma$, the process evolves from $s_k = \text{last}(\omega)$ under $a_k = \sigma(\omega)$ to the next state according to a probability distribution in Γ_{s_k,a_k}. An adversary is a function that chooses this distribution (Givan et al., 2000). Formally, an *adversary* is a function $\xi : S \times A \times \mathbb{N} \to \mathcal{P}(S)$ that maps each state s_k, action a_k, and time step $k \in \mathbb{N}$ to a transition probability distribution $\gamma \in \Gamma_{s,a}$, according to which s_{k+1} is distributed. We denote the set of all adversaries by Ξ. Given an initial condition $s_0 \in S$, a strategy $\sigma \in \Sigma$ and an adversary $\xi \in \Xi$, the UMDP collapses to a Markov chain with a unique probability distribution $Pr_{s_0}^{\sigma,\xi}$ over its paths.

Definition 2 (Interval MDP) *A labeled interval Markov decision process (IMDP) $\mathcal{I}$ is an UMDP whose transition probability distributions are defined by intervals: for all $s \in S$, $a \in A$, $\Gamma_{s,a} := \{\gamma \in \mathcal{P}(S) : \underline{P}(s,a)(s') \leqslant \gamma(s') \leqslant \overline{P}(s,a)(s')\}$, where $\underline{P}(s,a)(\cdot), \overline{P}(s,a)(\cdot) : S \to [0,1]$.*

4. Data-driven UMDP Abstraction

In this section, we introduce a construction of a UMDP $\mathcal{M}$, whose path probabilities are guaranteed to encompass the probabilities of System (1)'s trajectories with confidence $1 - \alpha$. We define the set of states S of $\mathcal{M}$ as follows. Let $R := \{r_1, \dots, r_{|R|}\}$ be a finite partition of the continuous state-space $\mathbb{R}^n$ into non-overlapping, non-empty regions, which respects the regions of interest R_{int} and the safe set X, and such that $r \in \mathcal{B}(\mathbb{R}^n)$ for all $r \in R$. We let region $r_{|R|} := r_{\text{unsafe}}$ represent the unsafe set. We assign each region $r \in R$ to a state $s \in S$ in the abstraction $\mathcal{M}$ through the bijective map $J : R \to S$, which ensures that $J^{-1}(s) = r \in R$ is unique. For simplicity, we abuse the notation and also say $J(x) = s$ if $x \in r$ with $J(r) = s$. We define the action set A of $\mathcal{M}$ to be the finite control set U of System (1). Furthermore, and with a slight abuse of language, we denote by L the labeling function of $\mathcal{M}$, which maps each state $s \in S$ to the atomic propositions that hold at $x \in r = J^{-1}(s)$.

Next, we define the set of transition probability distributions of the abstraction. To that end, we begin by stating the following proposition, whose proof follows from (Badings et al., 2023a,

Eq.12)[1], which gives uniform bounds in the probabilities that System (1) transitions from each point x in some region $r \in R$ to some region $\tilde{r} \subset \mathbb{R}^n$.

Proposition 3 *Given a region $r \in R$, an action $a \in A$ and a realization $w \in W$ of $\boldsymbol{w}$, denote by Reach$(r, a, w) := \{f(x, a, w) : x \in r\}$ the reachable set of r under a and w. Then, the probability of transitioning from each state $x \in r$ to region $\tilde{r} \in \mathcal{B}(\mathbb{R}^n)$ under action $a \in A$ is bounded by*

$$P(\{w \in W : \textit{Reach}(r, a, w) \subseteq \tilde{r}\}) \leqslant T(\tilde{r} \mid x, a) \leqslant P(\{w \in W : \textit{Reach}(r, a, w) \cap \tilde{r} \neq \varnothing\}) \quad (2)$$

Below, we use the samples of $\boldsymbol{w}$ to derive data-driven bounds that contain the ones in (2), and leverage them to define the set of transition probability distributions for $\mathcal{M}$.

4.1. Data-Driven Transition Probability Bounds

We now construct the sets $\Gamma_{s,a}$ of transition probability distributions of the abstraction by leveraging the samples from $\boldsymbol{w}$. Specifically, in our UMDP abstraction, the set $\Gamma_{s,a}$ for each state-action pair (s, a) is defined by: (i) interval bounds on the probability of transitioning to each state $s' \in S$, (ii) interval bounds on the probability of transitioning to a cluster of states in 2^S, and (iii) a bound on the probability of transitioning to states within the reachable set of the learned support of P. Notably, (ii) and (iii) distinguish our construction from prior work, which relies solely on (i). As a result, our UMDP incorporates additional constraints, leading to tighter uncertainty sets. This yields less conservative probabilistic guarantees, as shown in the case studies.

To derive the bounds in steps (i)-(iii), we use Proposition 3, samples from $\boldsymbol{w}$, and two well-known concentration inequalities. The proposition below enables us to compute bounds on transition probabilities between regions, which we later use to obtain bounds in (i)-(ii).

Proposition 4 *Consider the set $\{\hat{\boldsymbol{w}}^{(i)}\}_{i=1}^N$ of i.i.d. samples from $\boldsymbol{w}$. Pick $r \in R$, $a \in A$, $\tilde{r} \in \mathcal{B}(\mathbb{R}^n)$ and $\beta \in (0, 1)$, and let $\epsilon = \sqrt{\log(2/\beta)/(2N)}$. Then, with confidence at least $1 - \beta$ that, for all $x \in r$ we have*

$$T(\tilde{r} \mid x, a) \geqslant \underline{P}(r, a)(\tilde{r}) := \frac{1}{N}\left|\{i \in \{1, \dots, N\} : \textit{Reach}(r, a, \hat{\boldsymbol{w}}^{(i)}) \subseteq \tilde{r}\}\right| - \epsilon \quad (3a)$$

$$T(\tilde{r} \mid x, a) \leqslant \overline{P}(r, a)(\tilde{r}) := \frac{1}{N}\left|\{i \in \{1, \dots, N\} : \textit{Reach}(r, a, \hat{\boldsymbol{w}}^{(i)}) \cap \tilde{r} \neq \varnothing\}\right| + \epsilon. \quad (3b)$$

Proof Consider the lower bound in (2). Denote $E := \{w \in W : \text{Reach}(r, a, w) \subseteq \tilde{r}\}$ and note that $T(\tilde{r} \mid x, a) \geqslant P(E) = \mathbb{E}_P[\mathbb{1}_E(\omega)]$ for all $x \in r$. Therefore applying Hoeffding's inequality to the random variable $\frac{1}{N}\sum_{i=1}^N \mathbb{1}_E(\hat{\boldsymbol{w}}^{(i)})$ yields $P^N[P(E) \geqslant \frac{1}{N}\sum_{i=1}^N \mathbb{1}_E(\hat{\boldsymbol{w}}^{(i)}) - \epsilon] \geqslant 1 - \beta/2$, with $\epsilon = \sqrt{\log(2/\beta)/(2N)}$. Thus, the first expression in (3) holds for all $x \in r$ with confidence $1 - \beta/2$. Employing a similar argument, we obtain that the second expression in (3) also holds for all $x \in r$ with confidence $1 - \beta/2$. Combining both results via the union bound, we obtain the result. ■

Remark 5 *The complexity of computing the bounds in (3) is proportional to N, which is typically high to obtain tight bounds. To reduce this complexity, we cluster the N samples from $\boldsymbol{w}$ into $N_c \ll N$ clusters, each with center c_j and diameter ϕ_j. Substituting the sets Reach$(r, a, \hat{\boldsymbol{w}}^{(j)})$ in*

1. Measurability of the events in (2) is formally proved in Appendix A.

(3) *by* $\{f(x,a,w) \in \mathbb{R}^n : x \in r, \|w - c_j\| \leqslant \phi_j/2)\}$*, it is evident that Proposition 4 still holds, with relaxed bounds. Note that this clustering induces a partition on* W*, allowing to overapproximate the sets* $\{f(x,a,w) \in \mathbb{R}^n : x \in r, \|w - c_j\| \leqslant \phi_j/2)\}$ *as shown by Skovbekk et al. (2023).*

Next, we estimate the support of P in (iii). Including this information into $\mathcal{M}$ tightens the sets $\Gamma_{s,a}$ of transition probability distributions, thus yielding a less conservative abstraction.

Proposition 6 (Confidence Region (Tempo et al., 2013)) *Let* $\hat{c} = \max\{\|\hat{\boldsymbol{w}}^{(1)}\|, \ldots, \|\hat{\boldsymbol{w}}^{(N)}\|\}$*. Then, for any* $\epsilon_c, \beta_c > 0$ *and* $N \geqslant \log(1/\beta_c)/\log(1/(1-\epsilon_c))$*, it holds, with a confidence greater than* $1 - \beta_c$ *with respect to the random choice of* $\{\hat{\boldsymbol{w}}^{(i)}\}_{i=1}^N$*, that* $P(\{w \in W : \|w\| \leqslant \hat{c}\}) \geqslant 1 - \epsilon_c$*.*

We denote the learned confidence region for w by $\widehat{W} := \{w \in W : \|w\| \leqslant \hat{c}\}$, which contains at least $1 - \epsilon_c$ probability mass from P with a confidence greater than $1 - \beta_c$. We also define $\widehat{\text{Post}}(s,a) := \{J(f(x,a,w)) \in S : x \in J^{-1}(s), w \in \widehat{W}\}$ for each $s \in S$, $a \in A$ as the set of states of $\mathcal{M}$ that can be reached from region $J^{-1}(s)$ and for some disturbance $w \in \widehat{W}$.

We now have all the components needed to formally define our abstraction class. Intuitively, the abstraction relies on a two-layer discretization: a fine one represented by S and a coarse one formed by clustering the elements of S (see Figure 1). Let $Q \subseteq 2^S$ represent this clustering, which is non-overlapping, i.e., $\bigcup_{q \in Q} q = S$ and $q \cap q' = \varnothing$ for all $q \neq q' \in Q$. This clustering is crucial for obtaining non-zero lower-bound transition probabilities in (3a), as $\text{Reach}(r,a,\hat{\boldsymbol{w}}^{(i)})$ often cannot be contained within a single small region but can be captured by a cluster of regions (see Figure 1). Additionally, we leverage the learned support $\widehat{W}$ of the disturbance to impose the constraint that the successor state corresponding to a given state-action pair lies on some region with high probability. As described in Section 4.2, this constraint is key to make our approach work in practice. With this intuition, we formally define our abstraction as follows.

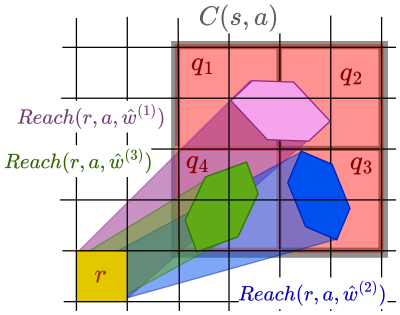


Figure 1: Illustration of the sets in Def. 1. $C(s,a) = \{q_1, q_2, q_3, q_4\}$, and each q_i contains 4 states. The probability that the successor state of $s = J(r)$ under action a will be in $C(s,a)$ is higher than $1 - \epsilon$. Note that the reachable sets corresponding to $\hat{\boldsymbol{w}}^{(2)}$ and $\hat{\boldsymbol{w}}^{(3)}$ are contained in q_3 and q_4, respectively, but no single regions contains them completely.

Definition 7 (UMDP Abstraction) *Let* $Q \subseteq 2^S$ *be a non-overlapping clustering of* S *and* $Q(s,a) \subseteq Q$ *be the subset that covers* $\widehat{Post}(s,a)$*, i.e.,* $\widehat{Post}(s,a)$ *is contained in* $C(s,a) := \bigcup_{q \in Q(s,a)} q$*. We define the UMDP abstraction of System (1) as* $\mathcal{M} = (S, A, s_0, \Gamma, AP, L)$*, with,* $\forall s \in S \backslash \{s_{|S|}\}$ *and* $\forall a \in A$*,*

$$\Gamma_{s,a} := \Big\{\gamma \in \mathcal{P}(S) : \underline{P}(r_s,a)(r_{s'}) \leqslant \gamma(s') \leqslant \overline{P}(r_s,a)(r_{s'}) \qquad \forall s' \in C(s,a),$$
$$\underline{P}(r_s,a)(r_{q'}) \leqslant \sum_{s' \in q'} \gamma(s') \leqslant \overline{P}(r_s,a)(r_{q'}) \;\; \forall q' \in Q(s,a), \sum_{s' \in C(s,a)} \gamma(s') \geqslant 1 - \epsilon_c\Big\}, \quad (4)$$

where $\underline{P}, \overline{P}$ *are defined in (3),* $r_{s'} = J^{-1}(s')$*, and* $r_{q'} = \bigcup_{s' \in q'} J^{-1}(s')$*, and* $\Gamma_{s_{|S|},a} = \{\delta_{s_{|S|}}\}$ *for all* $a \in A$*, where* $\delta_{s_{|S|}}$ *denotes the Dirac measure located at* $s_{|S|}$*.*

In Theorem 8, we establish that the UMDP $\mathcal{M}$ is a sound abstraction of System (1), i.e., that $\mathcal{M}$ captures all 1-step behaviors of System (1).

Theorem 8 (Soundness of UMDP Abstraction) *For all $s \in S$, $a \in A$, $x \in J^{-1}(s)$, define $\gamma_x \in \mathcal{P}(S)$ as $\gamma_x(s') := T(J^{-1}(s') \mid x, a)$ for all $s' \in S$. Then, $\gamma_x \in \Gamma_{s,a}$ with confidence of at least $1-\alpha$, where $\alpha = \beta_c + \big(\sum_{s\in S, a\in A} |C(s,a)| + |Q(s,a)|\big)\beta$.*

Proof By Proposition 4 we have that $\underline{P}(r_s, a)(r_{s'}) \leqslant \gamma_x(s')$ with confidence $1-\beta/2$ pointwisely for all $s' \in C(s,a)$, and similarly for the upper bound $\gamma_x(s') \leqslant \overline{P}(r_s, a)(r_{s'})$. Note that, for all $s' \notin C(s,a)$, no interval constraints are learned. Therefore, the total number of learned intervals for states s' is $\sum_{s\in S, a\in A} |C(s,a)|$. Using again Proposition 4 with $\tilde{r} = \bigcup_{s'\in q'} r_{s'}$ and noting that $T(\tilde{r} \mid x, a) = \sum_{s'\in q'} T(r_{s'} \mid x, a)$ since the regions in the partition R are disjoint, it follows that $\underline{P}(r_s, a)(r_{q'}) \leqslant \sum_{s'\in q'} \gamma_x(s')$ with confidence $1-\beta/2$ pointwisely for all $q' \in Q(s,a)$, and similarly for the upper bound $\sum_{s'\in q'} \gamma_x(s') \leqslant \overline{P}(r_s, a)(r_{q'})$. This makes the total number of learned intervals for clusters q' be $\sum_{s\in S, a\in A} |Q(s,a)|$.

Furthermore, since, by definition, $C(s,a) \supseteq \widehat{\text{Post}}(s,a)$, we also have that $\sum_{s'\in C(s,a)} \gamma_x(s') \geqslant \sum_{s'\in \widehat{\text{Post}}(s,a)} \gamma_x(s') = \sum_{s'\in \widehat{\text{Post}}(s,a)} T(r_{s'} \mid x, a) = T(\bigcup_{s'\in \widehat{\text{Post}}(s,a)} r_{s'} \mid x, a) = P(\{w \in W : f(x,a,w) \in \bigcup_{s'\in \widehat{\text{Post}}(s,a)} r_{s'}\}) = P(\{w \in W : J(f(x,a,w)) \in \widehat{\text{Post}}(s,a)\}) \geqslant P(\widehat{W})$, which is at least $1-\epsilon_c$ with confidence at least $1-\beta_c$. Then, it follows that the last constraint in (4) also holds with the same confidence. Combining this confidence with the $\sum_{s\in S, a\in A} |C(s,a)| + |Q(s,a)|$ learned intervals via the union bound, we get that γ_x fulfills all constraints in the definition of $\Gamma_{s,a}$ with confidence at least $1-\alpha$. ■

Corollary 9 *Given $\alpha \in (0,1)$ and $\epsilon = \epsilon_c \in (0,1)$, the sample complexity of obtaining a UMDP abstraction with confidence at least $1-\alpha$ is $N = \max\big\{\log(\frac{n_{learn}/\alpha}{2\epsilon^2}), \log(\frac{n_{learn}/\alpha}{\log(1/(1-\epsilon_c))})\big\}$, with $n_{learn} = 1 + \sum_{s\in S, a\in A} |C(s,a)| + |Q(s,a)|$.*

4.2. Issues of Naïve Data-Driven IMDP Abstractions: Loose Abstraction

As mentioned above, compared to the abstraction classes frequently used in the literature, namely, interval MDPs (IMDPs) (Lahijanian et al., 2015), our UMDP abstraction differs in that the sets $\Gamma_{s,a}$ of transition probability distributions are defined by more constraints than just the intervals $[\underline{P}(r_s, a)(r_{s'}), \overline{P}(r_s, a)(r_{s'})]$ for all $s' \in S$. Here, we discuss why IMDPs are not a good abstraction choice in our setting because they do not capture the dynamics (1) very tightly, and thus why a more complex Γ is required. Then, in Section 5.1 we explain how the strategy synthesis process often fails to return meaningful results for these baïve abstractions.

Let $\mathcal{I}$ be a UMDP abstraction of System (1), where Γ is defined only by the first row of constraints in (4). Note that the lower bound $\underline{P}(s,a)(s')$ is obtained using Expression (3a), which boils down to checking wether or not the the event $\text{Reach}(r_s, a, \hat{\boldsymbol{w}}^{(i)}) \subseteq r_{s'}$ happens. However, this condition rarely takes place, since it requires the reachable set being smaller than the region $r_{s'}$ (see Figure 1), which is uncommon in a big portion of the state-space unless (i) the system's dynamics are 1-step contractive and (ii) the partition is "aligned" with the dynamics. As a consequence, when s' is not a terminal state, e.g., the goal or unsafe regions, which are typically way bigger than the rest, it very often happens that $\underline{P}(r_s, a)(r_{s'}) = 0$ for all such states s'. However, the upper bound

$\overline{P}(r_s, a)(r_{s'})$ behaves very differently: by (3b), $\overline{P}(r_s, a)(r_{s'}) \geqslant \epsilon$ for all $s' \in S$. As a consequence, the set $\Gamma_{s,a}$ contains many spurious distributions, making $\mathcal{I}$ a very loose abstraction of System (1). In Section 5.1 we describe how this looseness of $\mathcal{I}$ typically translates into poor strategy synthesis results.

5. Strategy Synthesis

Here, we focus on synthesizing a strategy for System (1) and provide a lower bound on the probability that the closed-loop system satisfies the LTL$_f$ formula φ. We first show that standard synthesis procedures for general UMDP abstractions from the literature (Wolff et al., 2012; Gracia et al., 2024, 2025) also apply to our setting and then introduce a novel (tailored) algorithm that leverages the specific structure of our UMDP abstraction to reduce computational complexity.

Strategy synthesis begins by translating φ into its equivalent deterministic finite automaton (DFA) $\mathcal{A}_\varphi$ (De Giacomo and Vardi, 2013) and constructing the product $\mathcal{M}_\varphi = \mathcal{M} \otimes \mathcal{A}_\varphi$. A strategy σ^φ is then synthesized on $\mathcal{M}_\varphi$ via unbounded-horizon *robust dynamic programming* (RDP) with a reachability objective (Wolff et al., 2012), as detailed in (Gracia et al., 2025, Theorems 6.2, 6.6). σ^φ robustly maximizes the probability of satisfying φ under adversarial choices of transition probabilities from the set Γ^φ. Finally, σ^φ is refined into a strategy σ_x for System (1).

We start by defining the DFA $\mathcal{A}_\varphi$.

Definition 10 (DFA) *Let φ be an LTL$_f$ formula defined over a set of atomic propositions AP. The* deterministic finite automaton *(DFA) corresponding to φ is a tuple $\mathcal{A}_\varphi = (Z, 2^{AP}, \delta, z_0, Z_F)$ where Z is a finite set of states, 2^{AP} is a finite set of input symbols, $\delta : Z \times 2^{AP} \to Z$ is the transition function, $z_0 \in Z$ is the initial state, and $Z_F \subseteq Z$ is the set of accepting states.*

Given a trace $\rho = \rho_0\rho_1 \dots \rho_K \in (2^{AP})^*$, a run $z = z_0 z_1 \dots z_{K+1}$ is induced on $\mathcal{A}_\varphi$, where $z_{k+1} = \delta(z_k, \rho_k)$ for all $k \in \{0, \dots, K\}$. By construction of $\mathcal{A}_\varphi$, trace ρ satisfies φ iff $z_{K+1} \in Z_F$ De Giacomo and Vardi (2013). Such a run is called *accepting* for $\mathcal{A}_\varphi$.

Next, we define the product UMDP $\mathcal{M}_\varphi$, which contains information about the (uncertain transition probabilities) of $\mathcal{M}$ and the transition function of $\mathcal{A}_\varphi$.

Definition 11 (Product UMDP) *Given UMDP $\mathcal{M}$ and DFA $\mathcal{A}_\varphi$, the product $\mathcal{M} \otimes \mathcal{A}_\varphi$ is another UMDP $\mathcal{M}_\varphi = (S^\varphi, A^\varphi, \Gamma^\varphi, s_0^\varphi, S_F^\varphi)$, where $S^\varphi = S \times Z$, $A^\varphi = A$, $s_0^\varphi = (s_0, \delta(z_0, L(s_0)))$, $S_F^\varphi = S \times Z_F$, and $\Gamma^\varphi = \{\Gamma^\varphi_{(s,z),a} : (s, z) \in S^\varphi, a \in A^\varphi\}$ with $\Gamma^\varphi_{(s,z),a} := \{\gamma^\varphi \in \mathcal{P}(S^\varphi) : \exists \gamma \in \Gamma_{s,a}$ s.t. $\gamma^\varphi((s', z')) = \gamma(s')$ with $z' = \delta(z, L(s')), \forall s' \in S\}$. We denote a finite path of $\mathcal{M}^\varphi$ by ω^φ and the set of all such paths by Ω^φ. We also let Σ^φ and Ξ^φ denote the sets of strategies and adversaries of $\mathcal{M}^\varphi$, respectively.*

Intuitively, $\mathcal{M}_\varphi$ is a UMDP whose state is the product between the state spaces of $\mathcal{M}$ and $\mathcal{A}_\varphi$, and whose transition probability distributions combine information regarding the transitions of $\mathcal{M}$ and $\mathcal{A}_\varphi$. Specifically, the set $\Gamma^\varphi_{(s,z),a}$ corresponding to a given state-action pair $((s, z), a)$ contains probability distributions γ^φ over the product space S^φ such that their projections (pushforward measure) γ onto the set of probability distributions $\mathcal{P}(S)$ belong to $\Gamma_{s,a}$. Conversely, each $\gamma^\varphi \in \Gamma_{(s,z),a}$ is obtained by taking some $\gamma \in \Gamma_{s,a}$ and lifting it to the space $\mathcal{P}(S^\varphi)$ by taking into account the transition function of $\mathcal{A}_\varphi$.

Having obtained the product UMDP $\mathcal{M}_\varphi$, we synthesize a strategy σ^φ which maximizes the probability of reaching set S_F^φ under an adversarial choice of the transition probabilities of $\mathcal{M}_\varphi$ by

the adversary. It can be proved Wolff et al. (2012) that this strategy, when mapped to a strategy $\sigma \in \Sigma$ of $\mathcal{M}$, also maximizes the worst-case probability of $\mathcal{M}$ satisfying the LTL$_f$ formula φ. Proposition 12 shows how to obtain the reachability probabilities.

Proposition 12 (Robust Dynamic Programming (Gracia et al., 2025, Theorem 6.2)) *Given $s^\varphi \in S^\varphi$, define the optimal robust reachability probability as*

$$\underline{p}(s^\varphi) := \sup_{\sigma^\varphi \in \Sigma^\varphi} \inf_{\xi^\varphi \in \Xi^\varphi} Pr_{s^\varphi}^{\sigma^\varphi, \xi^\varphi}(\{\omega^\varphi \in \Omega^\varphi : \exists k \in \mathbb{N} \cup \{0\} \text{ s.t. } \omega^\varphi(k) \in S_F^\varphi\}). \tag{5}$$

Consider also the recursion

$$\underline{p}^{k+1}(s^\varphi) = \begin{cases} 1 & \text{if } s^\varphi \in S_F^\varphi \\ \max_{a \in A^\varphi} \min_{\gamma \in \Gamma_{s^\varphi, a}} \sum_{s^{\varphi\prime} \in S^\varphi} \gamma(s^{\varphi\prime}) \underline{p}^k(s^{\varphi\prime}) & \text{otherwise,} \end{cases} \tag{6}$$

for all $k \in \mathbb{N} \cup \{0, \infty\}$, with initial condition $\underline{p}^0(s^\varphi) = 1$ for all $s^\varphi \in S_F^\varphi$ and 0 otherwise. Then, recursion (6) converges to $\underline{p}$.

Having obtained the reachability probabilities, we obtain a memoryless strategy σ^φ that attains the probability in (5) for all states by following the procedure in (Gracia et al., 2025, Theorem 6.6), and then map it to a finite-memory strategy σ_x of System (1) as described in (Gracia et al., 2025, Section 6.4). The following theoremensures that satisfaction probability bounds are preserved under this procedure, thus solving Problem 1.

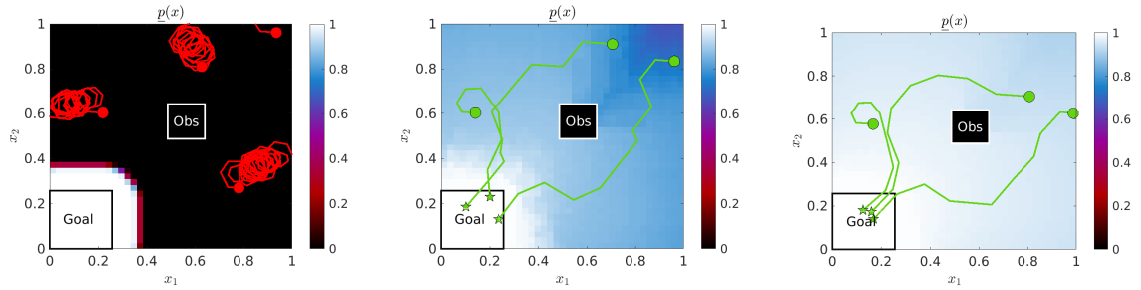
Theorem 13 (Strategy Synthesis through Product UMDP) *Let σ^φ and $\underline{p}^\varphi$ be respectively the optimal strategy and the lower bound in the probability of satisfying φ obtained via RDP on $\mathcal{M}_\varphi$. Furthermore, let σ_x be the strategy obtained by refining σ^φ to System (1). Then, with confidence $1 - \alpha$, $Pr_x^{\sigma_x}[\omega_x \models \varphi \wedge \mathcal{G}\neg \mathfrak{p}_u] \geqslant \underline{p}^\varphi((s, z))$ for all $x \in X$, where $s = J(x)$, $z = \delta(z_0, L(s))$.*

Proof Assume $\mathcal{M}$ is a correct abstraction of System (1). Then, it follows that (Jackson et al., 2021a, Theorem 2) $Pr_x^{\sigma_x}[\omega_x \models \varphi \wedge \mathcal{G}\neg \mathfrak{p}_u] \geqslant \underline{p}^\varphi((s, z))$, for all $x \in X$, where $s = J(x)$, $z = \delta(z_0, L(s))$. Since by Theorem 8 this assumption holds with confidence not smaller than $1 - \alpha$, the implication also holds with the same confidence, which concludes the proof. ■

5.1. Issues of Naïve Data-Driven IMDP Abstractions: Overly Conservative and Uninformed Solution to Problem 1

In this subsection we show that strategy synthesis often yields poor results if the abstraction is a naïve IMDP obtained as described in Section 4.2.

Consider the IMDP abstraction $\mathcal{I}$ of Section 4.2, and the strategy synthesis process for a simple reachability specification, which is carried out by applying Proposition 12 on $\mathcal{I}$ with goal set S_{goal}. At iteration k, denote by S_k^0 the set of states with zero value function. Since Recursion (6) starts with an initial value function that is zero for all $s \notin S_{\text{goal}}$, it easy to check that $|S_k^0| \approx |S|$ during the first iterations. By the structure of $\mathcal{I}$, the adversary ξ is allowed to pick a distribution γ over S that assigns probability at least ϵ of transitioning to each $s' \in S_k^0$, and that, if S_k^0 is big enough, the total probability of transitioning to these states adds up to 1, and thus $s' \in S_k^0$ with probability one. As a result, the value function at the iteration $k+1$ is again zero for all $s \in S_k^0$, and therefore RDP yields



(a) Naïve IMDP (b) UMDP with learned support (c) UMDP (7)

Figure 2: Empirical demonstration of the issues of naïve IMDP abstractions considered in Sections 4.2 and 5.1 The figures show the probabilistic guarantees that the unicycle system in case study #4 in Table 1 satisfies the reach-avoid specification φ_1 as a function of the initial position and the choice of the abstraction, for the most favorable heading angle. The abstraction choices are: (a) a Naïve IMDP, (b) a UMDP without the constraints involving clusters, (c) a UMDP as in Definition 7. In all cases, the number of samples used to construct the abstraction was of $N = 10^6$. Furthermore, increasing the number of samples to 10^7 sowed almost identical results when using a naïve IMDP abstraction. The green and red lines are trajectories of the unicycle system in closed loop with the synthesized strategy: in green, the ones that satisfy the specification. In red the ones that do not.

Algorithm 1 2-layer O-maximization

Require: $\mathcal{M}, s \in S, a \in A, \underline{p}^k$
Ensure: γ
Sort Post(s,a) according to $\{\underline{p}^k(s')\}_{s' \in \text{Post}(s,a)}$ in increasing order
$\gamma(s') \leftarrow \underline{P}(s,a)(s')$ for all $s' \in \text{Post}(s,a)$
$\gamma(s') \leftarrow 0$ for all $s' \notin \text{Post}(s,a)$
$\gamma(q') \leftarrow \sum_{s' \in q'} \gamma(s')$ for all $q' \in Q(s,a)$
$M \leftarrow 1 - \sum_{s' \in S} \gamma(s')$
for $q' \in Q(s,a)$ **do**
  $m \leftarrow \underline{P}(s,a)(q') - \gamma(q')$
  if $m > 0$ **then**
    for $s' \in q'$ **do**
      $g \leftarrow \min\{m, \overline{P}(s,a)(s') - \gamma(s')\}$
      $\gamma(s') \leftarrow \gamma(s') + g, \gamma(q') \leftarrow \gamma(q') + g$
      $m \leftarrow m - g, M \leftarrow M - g$
for $s' \in \text{Post}(s,a)$ **do**
  $q' \leftarrow$ get cluster q' such that $s' \in q'$
  if $q' \neq \varnothing$ **then**
    $m \leftarrow \min\{M, \overline{P}(s,a)(q') - \gamma(q'), \overline{P}(s,a)(s') - \gamma(s')\}$
    $\gamma(q') \leftarrow \gamma(q') + m$
  else
    $m \leftarrow \min\{M, \overline{P}(s,a)(s') - \gamma(s')\}$
  $\gamma(s') \leftarrow \gamma(s') + m, M \leftarrow M - m$

a vacuous satisfaction probability on a big region of the state space. We empirically demonstrate this issue on case study #4 in Table 1, and plot this results in Figure 2.

Note that irrespective of how small ϵ is, refining the abstraction eventually leads to this issue, as it increases the number of states of the abstraction. Furthermore, given a constant discretization granularity, the size of S_k^0 is exponential in the dimension of System (1), which implies that the value of ϵ required to avoid this issue typically requires an impractical number N of samples from $\boldsymbol{w}$.

5.2. Tailored Synthesis Algorithm

We introduce a synthesis algorithm tailored for UMDPs as such in Definition 7, which exploits their structure for greater efficiency. The algorithm draws inspirations from IMDP value iteration (Givan et al., 2000) to speed up the computation of the optimal adversaries in RDP, i.e., the inner minimization problem in Equation (6), which is typically formulated a linear program in standard UMDPs. We note that this approach is applicable to our product UMDP $\mathcal{M}_\varphi$ because it retains the same structure of $\mathcal{M}$In consequence and, for simplicity, we then describe the algorithm in the context of a reachability problem on $\mathcal{M}$ rather than on $\mathcal{M}_\varphi$.

Proposition 14 proves that $\mathcal{M}$ and $\mathcal{M}_\varphi$ have the same structure.

Proposition 14 *For each $s^\varphi := (s,z) \in S^\varphi$, $a \in A^\varphi$, $q' \in Q(s,a)$, let $q^{\varphi\prime} := \{(s',z') \in q' \times Z : z' = \delta(z, L(s'))\}$, and define $Q^\varphi(s^\varphi, a)$ as the set of all these $q^{\varphi\prime}$. Then, $\Gamma^\varphi_{s^\varphi,a}$ is equivalently expressed as*

$$\Gamma^\varphi_{s^\varphi,a} := \Big\{\gamma^\varphi \in \mathcal{P}(S^\varphi) : \begin{aligned} &\underline{P}(s^\varphi,a)((s',z')) \leqslant \gamma^\varphi((s',z')) \leqslant \overline{P}(s^\varphi,a)((s',z')) \quad \forall s' \in C(s,a), z' \in Z, \\ &\underline{P}(s^\varphi,a)(q^{\varphi\prime}) \leqslant \sum_{s^{\varphi\prime} \in q^{\varphi\prime}} \gamma^\varphi(s^{\varphi\prime}) \leqslant \overline{P}(s^\varphi,a)(q^{\varphi\prime}) \qquad \forall q^{\varphi\prime} \in Q^\varphi(s^\varphi,a), \\ &\sum_{s^{\varphi\prime} \in C(s,a)\times Z} \gamma^\varphi(s^{\varphi\prime}) \geqslant 1-\epsilon_c \Big\}, \end{aligned} \tag{7}$$

with $\underline{P}(s^\varphi,a)((s',z')) := \underline{P}(s,a)(r_{s'})$, $\overline{P}(s^\varphi,a)((s',z')) := \overline{P}(s,a)(r_{s'})$ if $z' = \delta(z,L(s'))$ and 0 otherwise, and $\underline{P}(s^\varphi,a)(q^{\varphi\prime}) := \underline{P}(s,a)(q')$, $\overline{P}(s^\varphi,a)(q^{\varphi\prime}) := \overline{P}(s,a)(q')$, where $q' \in Q(s,a)$ is the projection of $q^{\varphi\prime}$ onto S. for all $q^{\varphi\prime} \in Q^\varphi(s^\varphi,a)$.

Proof Pick arbitrary $s^\varphi = (s,z) \in S^\varphi$, $a \in A^\varphi$ and $\gamma^\varphi \in \mathcal{P}(S^\varphi)$ that satisfies the bounds in Expression (7). By the interval bounds in said expression for each $(s',z') \in C(s,a) \times Z$, we have that $\gamma^\varphi((s',z')) = 0$ for all $z' \neq \delta(z,L(s'))$. Define $\gamma \in \mathcal{P}(S)$ as $\gamma(s') := \gamma^\varphi(s', \delta(z,L(s')))$ for all $s' \in S$, noting that γ is a probability over S because $\sum_{s'\in S}\gamma(s') = \sum_{s'\in S}\gamma^\varphi((s',\delta(z,L(s')))) = \sum_{(s',z')\in S^\varphi}\gamma^\varphi((s',z')) = 1$. Since for each $(s',z') \in S^\varphi$ we have that $\underline{P}(s^\varphi,a)((s',z')) \leqslant \gamma^\varphi((s',z')) \leqslant \overline{P}(s^\varphi,a)((s',z'))$, it is easy to see that $\underline{P}(s,a)(s') \leqslant \gamma(s') \leqslant \overline{P}(s,a)(s')$ for all $s' \in S$. Next, pick $q' \in Q(s,a)$ and let $q^{\varphi\prime}$ be the set in $Q^\varphi(s^\varphi,a)$ such that q' is its projection onto S. Because of this relationship we obtain that $\sum_{s'\in q'}\gamma(s') = \sum_{s'\in q'}\gamma^\varphi((s',\delta(z,L(s')))) = \sum_{(s',z')\in q^{\varphi\prime}}\gamma^\varphi((s',z')) \in [\underline{P}((s,z),a)(q^{\varphi\prime}), \overline{P}((s,z),a)(q^{\varphi\prime})] = [\underline{P}(s,a)(q'), \overline{P}(s,a)(q')]$. Finally, since γ^φ satisfies the last condition in (7), it follows that

$$\sum_{s'\in C(s,a)}\gamma(s') = \sum_{s'\in C(s,a)}\gamma^\varphi((s',\delta(z,L(s')))) = \sum_{s^{\varphi\prime}\in C(s,a)\times Z}\gamma^\varphi(s^{\varphi\prime}) \geqslant 1-\epsilon_c.$$

Therefore, $\gamma \in \Gamma_{s,a}$, which implies that $\gamma^\varphi \in \Gamma^\varphi_{s^\varphi,a}$. ■

Since the set Γ in (7) possesses the same structure as the set Γ in (4), we now consider simply a reachability problem on $\mathcal{M}$ instead of $\mathcal{M}_\varphi$. Next, we simplify the sets $\Gamma_{s,a}$ of $\mathcal{M}$ by discarding the last constraint in (4) and adjusting the transition probability bounds accordingly. Lemma 15 states that performing RDP on this simplified UMDP yields a lower bound on the reachability probability in (5).

Lemma 15 *Consider the UMDP $\mathcal{M}$ as in Definition 7, and let $\widetilde{\mathcal{M}}$ be the UMDP that differs from the latter only in the sets $\widetilde{\Gamma}_{s,a}$, defined as*

$$\widetilde{\Gamma}_{s,a} := \big\{\gamma \in \mathcal{P}(S) : \underline{P}(s,a)(s') \leqslant \gamma(s') \leqslant \overline{P}(s,a)(s') \qquad \forall s' \in S,$$
$$\underline{P}(s,a)(q') \leqslant \sum_{s' \in q'} \gamma(s') \leqslant \overline{P}(s,a)(q') \;\; \forall q' \in Q(s,a)\big\}, \tag{8}$$

with

$$\underline{P}(s,a)(s') = \begin{cases} \underline{P}(r_s,a)(r_{s'}) & \forall s' \in C(s,a) \\ 0 & \forall s' \notin C(s,a) \end{cases}$$

$$\overline{P}(s,a)(s') = \begin{cases} \overline{P}(r_s,a)(r_{s'}) & \forall s' \in C(s,a) \backslash \{s_{|S|}\} \\ 0 & \forall s' \notin C(s,a) \bigcup \{s_{|S|}\} \\ \overline{P}(r_s,a)(r_{s'}) + \epsilon_c & \text{if } s' = s_{|S|} \text{ and } s' \in C(s,a) \\ \epsilon_c & \text{if } s' = s_{|S|} \text{ and } s' \notin C(s,a) \end{cases}$$

$$\underline{P}(s,a)(q') = \underline{P}(r_s,a)(r_{q'}) \qquad \text{for all } \; q' \in Q(s,a)$$

$$\overline{P}(s,a)(q') = \begin{cases} \overline{P}(r_s,a)(r_{q'}) & \text{if } q' \in Q(s,a) \text{ and } \; s_{|S|} \notin q' \\ \overline{P}(r_s,a)(r_{q'}) + \epsilon_c & \text{if } q' \in Q(s,a) \text{ and } \; s_{|S|} \in q' \end{cases}$$

for all $s \in S$, $a \in A$. Let $\underline{p}_{\widetilde{\mathcal{M}}}$ and $\underline{p}$ be the value functions returned by running the RDP recursion in (6) on $\mathcal{M}$ and $\mathcal{M}_\varphi$, respectively. Then, $\underline{p}_{\widetilde{\mathcal{M}}}(s) \leqslant \underline{p}(s)$ for all $s \in S$.

Proof Consider the value functions $\underline{p}^k$ and $\underline{p}^k_{\widetilde{\mathcal{M}}}$ obtained after k iterations of robust dynamic programming on $\mathcal{M}$ and on $\widetilde{\mathcal{M}}$, respectively, and let $s \in S$ and $a \in A$. It is easy to observe that a minimizer $\gamma \in \Gamma_{s,a}$ of the inner problem in Expression (6) is any γ that assigns as much probability mass as possible to the states $s' \in S$ with the smallest $\underline{p}^k(s')$. Note that, by the last constraint in the definition of $\Gamma_{s,a}$, at least $1 - \epsilon_c$ mass must be allocated to the states in $C(s,a)$, which allows us to define $\gamma \in \mathcal{P}(S)$ with $\tilde{\gamma}(s') := \gamma(s')$ for all $s' \in C(s,a) \backslash \{s_{|S|}\}$, $\tilde{\gamma}(s_{|S|}) := \gamma(s_{|S|}) + \epsilon_c$ and $\tilde{\gamma}(s') := 0$ everywhere else. Then, it is easy to observe that $\tilde{\gamma} \in \widetilde{\Gamma}_{s,a}$. Furthermore, since $\underline{p}^k(s') \geqslant 0$ for all $s' \in S$ $\underline{p}^k(s_{|S|}) = 0$, $\sum_{s' \in S} \gamma(s') p^k(s') \geqslant \sum_{s' \in S} \tilde{\gamma}(s') p^k(s')$, thus implying that

$$\min_{\gamma \in \Gamma_{s,a}} \sum_{s' \in S} \gamma(s') p^k(s') = \sum_{s' \in S} \gamma(s') p^k(s') \geqslant \sum_{s' \in S} \tilde{\gamma}(s') p^k(s') \geqslant \min_{\gamma \in \widetilde{\Gamma}_{s,a}} \sum_{s' \in S} \gamma(s') p^k(s')$$

Maximizing over the actions on both sides yields $\underline{p}^{k+1}_{\widetilde{\mathcal{M}}}(s) \leqslant \underline{p}^{k+1}(s)$, which holds for all $s \in S$. Using this result in an induction argument and letting $k \to \infty$ we obtain the desired result. ■

Intuitively, in the modified UMDP, the adversary is always allowed to pick a higher probability of transitioning to the unsafe state $s_{|S|}$, even when the last constraint in (4) is omitted. As a result, RDP returns a lower bound on the probability (5). During the reminder of this section we consider that $\mathcal{M}$ has the structure described in Lemma 15.

Then, on the modified $\mathcal{M}$, Alg. 1 computes the optimal adversary for each state-action pair (s,a) by extending the O-maximizing algorithm devised for IMDP value iteration (Givan et al., 2000) to $\mathcal{M}$: via a 2-layer O-maximizing logic, the algorithm efficiently allocates probability mass

to states of $\mathcal{M}$ with the lowest value function while respecting the constraints in (4). It begins by ensuring that the lower bounds $\underline{P}(s,a)(r_{s'})$ are satisfied for all states s' (Lines 2-3), then proceeds to allocate mass to each cluster $q' \in Q(s,a)$ to meet the required lower bounds (Lines 4-12). The algorithm ensures that the total probability mass remains feasible by maintaining the constraints $\gamma(s') \leqslant \overline{P}(s,a)(r_{s'})$, $\sum_{s' \in q'} \gamma(s') \leqslant \overline{P}(s,a)(r_{q'})$ throughout the allocation (Lines 13-20). This allocation process guarantees that as much mass as possible is assigned to states with the smallest value function while ensuring $\gamma \in \Gamma_{s,a}$. Alg. 1 terminates once all mass is allocated. Note that RDP algorithm (Gracia et al., 2025) calls Alg. 1 for every (s,a) and in every iteration until termination. The following theorem proves its correctness and runtime complexity.

Theorem 16 (Correctness of Algorithm 1) *Let $\underline{p}^k \in \mathbb{R}^{|S|}$ be as defined in Proposition 12, $k \in \mathbb{N} \cup \{0\}$, $s \in S$ and $a \in A$. Define Post$(s,a) := \{s' \in S : \overline{P}(s,a)(s') > 0\}$. Then, for all $s \in S$, $a \in A$, the output γ of Algorithm 1 satisfies $\gamma \in \arg\min_{\gamma \in \Gamma_{s,a}} \sum_{s' \in S} \gamma(s') \underline{p}^k(s')$, and it has a computational complexity of $\mathcal{O}(|\text{Post}(s,a)| \log(|\text{Post}(s,a)|))$.*

Note that the computational complexity of solving the linear program in the theorem statement using a standard Simplex algorithm is of $\mathcal{O}(|\text{Post}(s,a)|^3)$, highlighting the computational advantage of using Alg. 1.

Proof We equivalently prove that any γ generated by the algorithm belongs to $\Gamma_{s,a}$, and that γ assigns the most probability to states $s' \in \text{Post}(s,a)$ with the smallest $\underline{p}^k(s')$. First we prove that $\gamma \in \Gamma_{s,a}$. From lines 2 and 3, we know that $\gamma(s') \geqslant \underline{P}(s,a)(r_{s'})$ for all $s' \in S$. Furthermore, by construction of $\mathcal{M}$ (see Proposition 4), it is easy to verify that $\underline{P}(s,a)(r_{q'}) \leqslant \sum_{s' \in q'} \underline{P}(s,a)(r_{s'})$ for all, $q' \in Q(s,a)$, and therefore lines $4-14$ guarantee that each $q' \in Q(s,a)$ receives exactly $\underline{P}(s,a)(r_{q'})$ probability mass. Since, also by construction, $\gamma(q') \geqslant \underline{P}(s,a)(r_{q'})$ and that $\underline{P}(s,a)(r_{q'}) \leqslant \sum_{s' \in q'} \overline{P}(s,a)(r_{s'})$ for all q', this allocation of probability mass is always feasible, and line 15 is reached with $m = 0$. Additionally, by the logic of lines $10-11$, $\gamma(s') \leqslant \overline{P}(s,a)(r_{s'})$ for all s' so far. In lines $15-24$ we allocate the remaining mass from M to the states $s' \in \text{Post}(s,a)$ while respecting $\gamma(s') \leqslant \overline{P}(s,a)(r_{s'})$ and $\gamma(q') \leqslant \overline{P}(s,a)(r_{q'})$. Since $\Gamma_{s,a} \neq \varnothing$, $\sum_{s' \in \text{Post}(s,a)} \overline{P}(s,a)(r_{s'}) \geqslant 1$, line 24 is reached with $M = 0$. Furthermore, by construction of $\mathcal{M}$, the constraint $\gamma(q') \leqslant \overline{P}(s,a)(r_{q'})$ never prevents all mass from M to be allocated. Therefore, $\gamma \in \Gamma_{s,a}$. Furthermore, since 1) the algorithm assigns first the least amount of mass that guarantees satisfaction of the lower bounds $\underline{P}(s,a)(r_{s'}), \underline{P}(s,a)(r_{q'})$ for all s', q', 2) when mass is assigned to $\gamma(s')$ (lines $2, 3, 11, 23$), the states with smaller $\underline{p}^k(s')$ are considered first, and 3) the mass allocated to such states is the maximum amount that the upper bounds $\overline{P}(s,a)(r_{s'}), \overline{P}(s,a)(r_{q'})$ allow. It follows that γ minimizes the expression in Theorem 16. Finally we prove the statement regarding the computational complexity. Note that the for loops in lines 15 and 17 are not nested. Additionally, since the sets $q' \in Q(s,a)$ are disjoint, the for loops in lines 6 and 9 are equivalent to a single for loop on $s' \in S$. Therefore lines 2 through 23 have complexity $\mathcal{O}(|\text{Post}(s,a)|)$, which is negligible in comparison with the complexity of sorting in line 1, which is $\mathcal{O}(|\text{Post}(s,a)| \log(|\text{Post}(s,a)|))$. Therefore the algorithm has complexity $\mathcal{O}(|\text{Post}(s,a)| \log(|\text{Post}(s,a)|))$, which concludes the proof. ■

Remark 17 *Note that classical IMDP abstractions use only the first constraint of Γ in (4), giving the adversary freedom to choose worse γ than if the abstraction were like our UMDPs. Consequently, IMDPs yield lower probabilistic guarantees, further motivating our abstraction choice.*

Table 1: Benchmark results. "Approach" indicates the abstraction used: UMDP (Definition 1), "IMDP (Learn Support)" (a UMDP with only the first and third constraints in (4)), and "Naïve IMDP" (traditional IMDP). e_{avg} represents the average difference in satisfaction probabilities between the lower and upper bounds across all states. Time for abstraction and synthesis is given in minutes, and N_{cluster} denotes the number of noise samples after clustering.

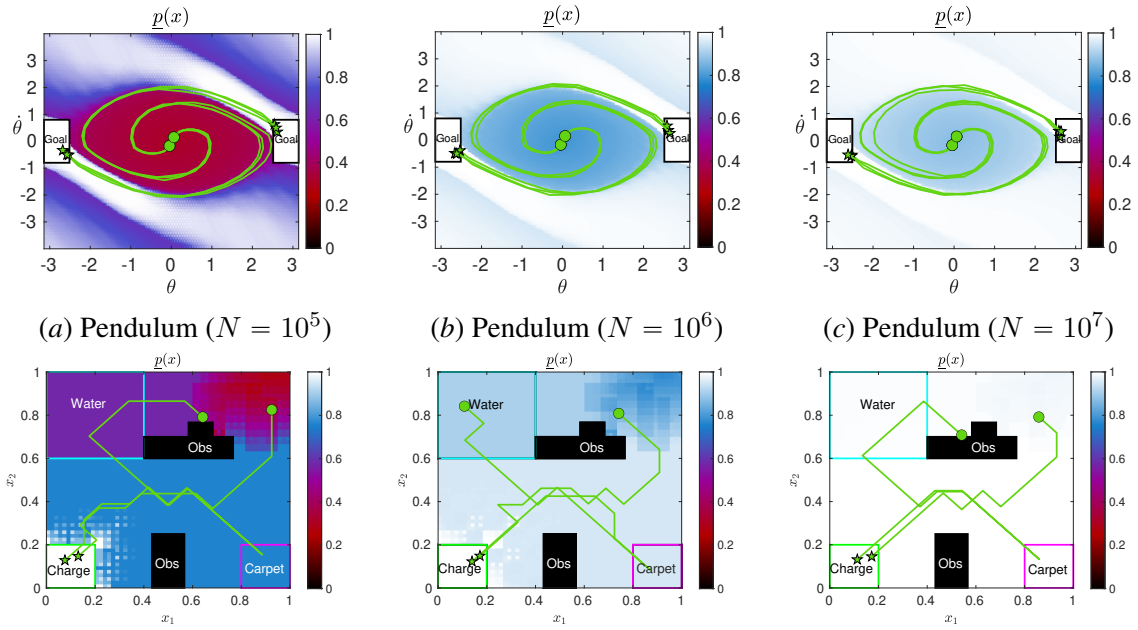
System (Spec.)	Approach	$\lvert Q\rvert$	$\lvert A\rvert$	N	N_{cluster}	e_{avg}	Abstr. Time	Synth. Time
Pendulum (φ_1)	UMDP	10^4	5	5×10^3	47	0.552	1.796	3.643
Gracia et al. (2024)	UMDP	10^4	5	10^4	44	0.007	1.857	6.679
	UMDP	10^4	5	10^5	47	0.003	1.797	2.515
	Gracia et al. (2024)	4×10^4	5	10^6	49	0	5.273	61.167
Pendulum (φ_1)	UMDP	1.225×10^5	5	5×10^4	320	0.136	138.705	187.653
(Torque-Limited)	UMDP	6.25×10^4	5	10^5	175	0.082	40.602	70.773
	UMDP	4×10^4	5	10^5	172	0.279	25.751	42.321
	UMDP	4×10^4	5	10^6	179	0.058	26.694	39.667
	UMDP	4×10^4	5	10^7	191	0.033	27.852	27.066
	IMDP (Learn Support)	4×10^4	5	10^7	191	0.172	14.474	20.281
$3D$ Unicycle (φ_1)	UMDP	5.932×10^4	10	10^4	235	0.579	34.689	33.156
Gracia et al. (2024)	UMDP	5.932×10^4	10	10^5	325	0.267	45.610	36.18
	UMDP	5.932×10^4	10	10^6	358	0.156	52.733	33.751
	Gracia et al. (2024)	6.4×10^4	10	5×10^8	8869	0.447	457.431	43.342
$3D$ Unicycle (φ_1)	UMDP	7.401×10^4	10	10^5	269	0.404	48.057	75.164
(difficult)	UMDP	5.932×10^4	10	10^6	589	0.312	87.753	56.221
	UMDP	7.401×10^4	10	10^6	295	0.249	53.371	69.565
	IMDP (Learn Support)	7.401×10^4	10	10^6	295	0.6923	25.109	38.680
	Naïve IMDP	5.932×10^4	10	10^7	340	0.870	45.865	3.68
	UMDP	5.932×10^4	10	10^7	357	0.199	54.505	47.365
	UMDP	7.401×10^4	10	10^7	345	0.1848	62.510	63.437
Multiplicative	UMDP	3.6×10^3	1	4.652×10^3	231	0.316	0.294	0.302
noise (φ_1)	IMDP (Learn Support)	3.6×10^3	1	4.371×10^3	235	0.437	0.412	0.086
(Skovbekk et al., 2023)	UMDP	3.6×10^3	1	4.68×10^4	257	0.251	0.339	0.034
	UMDP	9.6×10^3	1	4.66×10^5	279	0.223	1.150	0.146
	IMDP	9.6×10^3	1	4.66×10^5	276	0.307	0.228	0.034
	Gracia et al. (2024)	10^4	1	4.66×10^5	1066	0.323	13.149	3.863
$2D$ Unicycle (φ_2)	UMDP	3.6×10^3	8	10^3	32	0.288	0.302	3.304
Gracia et al. (2024)	IMDP (Learn Support)	3.6×10^3	8	5×10^3	31	0.387	0.291	1.174
	UMDP	3.6×10^3	8	5×10^3	34	0.095	0.318	2.818
	UMDP	3.6×10^3	8	10^4	37	0.062	0.341	3.103
	UMDP	3.6×10^3	8	10^5	41	0.017	0.375	3.075
	UMDP	3.6×10^3	8	10^6	43	0.003	0.390	2.767
	UMDP	3.6×10^3	8	10^7	45	0.001	0.408	2.667
	Gracia et al. (2024)	3.6×10^3	8	10^7	46	0.03	0.106	3.043
4-Room Heating (φ_3)	UMDP	2.074×10^4	16	5×10^4	620	0.086	188.506	6.296
Abate et al. (2010)	UMDP	2.074×10^4	16	10^6	818	0.044	282.854	7.156
	IMDP (Learn Support)	2.074×10^4	16	10^6	818	0.069	204.963	3.296

6. Case Studies

We now demonstrate empirically the effectiveness of our approach through 7 case studies. These include a nonlinear pendulum with non-additive disturbances, kinematic unicycle models with 2- and 3-D state-spaces and under nonlinear coulomb friction, a 2-D linear system with multiplicative noise, and a 4-D thermal regulation benchmark with multiplicative uncertainty. The considered specifications are reach-avoid (φ_1), the LTL$_f$ specification from (Gracia et al., 2024) (φ_2) and a 15-step safety specification (φ_3). For details of these setups, see Appendix B.

We compare our approach against (Gracia et al., 2024), the only related work addressing the same problem, in Case Studies 1, 3, 5-6. 2 is a more challenging version of 1, where $\boldsymbol{w}$ is unbounded and the pendulum cannot swing up in one go due to control saturation. Similarly, 4 extends 3 with

unbounded noise of larger variance and a smaller goal set. Note that (Gracia et al., 2024) relies on ambiguity set learning and cannot handle an unbounded $\boldsymbol{w}$. We also show results obtained using a naïve IMDP abstraction with and without learned support, to show tightness of our approach. Additionally, we show the results of our approach for case studies #2 and #6 in Figure 3, and for case study #4 in Figure 2. Table 1 summarizes our results, highlighting the clear advantages of our approach over (Gracia et al., 2024). Our method significantly reduces sample complexity, often by orders of magnitude, and allows for smaller abstractions while achieving similar or tighter results, which also reduces abstraction time in most cases. For abstractions of the same size, our synthesis time is typically smaller, sometimes by an order of magnitude. Additionally, the table demonstrates that our approach produces tighter results (less error hence higher probabilistic guarantees) than using a naïve IMDP, showcasing the benefits of incorporating additional information into the definition of Γ, albeit with higher computational effort. Furthermore, Table 1 also shows how the probabilistic satisfaction guarantees become tighter with the number of samples, which can also be observed in Figure 3, and as the size of the abstraction increases. We also compared the performance of Alg. 1 against the linear programming solver *Linprog* on an abstraction with $|S| = 1600$ and $|A| = 8$, achieving the same guarantees but reducing the total synthesis time from 2880s to 60s, a reduction of $48\times$.



(*a*) Pendulum ($N = 10^5$) (*b*) Pendulum ($N = 10^6$) (*c*) Pendulum ($N = 10^7$)

(*d*) 2D Unicycle ($N = 10^3$) (*e*) 2D Unicycle ($N = 10^4$) (*f*) 2D Unicycle ($N = 10^6$)

Figure 3: Synthesis results obtained in case studies #3 (Figures (a-c)) and #6 (Figures (d-f)). The figures show the lower bound in the probability of satisfying φ_1 and φ_2, respectively, along with trajectories of the system in closed loop with the synthesized strategy. It can be observed that increasing the number of samples raises this lower bound, and that all trajectories satisfy the respective specification.

7. Conclusion

We propose an approach to synthesize strategies for nonlinear stochastic systems with unknown disturbances via abstractions to UMDPs. We also identify pitfalls in the use of naïve abstractions for nonlinear systems and present a synthesis algorithm tailored to our UMDP class. Our extensive case studies show the efficacy and advantages of our framework w.r.t. existing works. In future research we plan to increase the tightness of our results by including additional information into the UMDP and investigate overlapping clusters.

Acknowledgments

I. Gracia and M. Lahijanian are supported in part by National Science Foundation (NSF) under grant number 2039062 and Air Force Research Lab (AFRL) under agreement number FA9453-22-2-0050. L. Laurenti is partially supported by the NWO(grant OCENW.M.22.056).

Appendix A. Measurability of the events in (2)

Let $r \in R$, and $a \in A$, and consider the set-valued function $w \mapsto g(w) := \text{Reach}(r, a, w)$, for all $w \in W$,. Denote also by $B(0, \epsilon) \subset \mathbb{R}^n$ the ϵ-ball centered at the origin. We begin by stating the following technical lemma:

Lemma 18 *For all $w \in W$, $\epsilon > 0$, there exists a $\delta > 0$ such that for all $w' \in W$ with $\|w - w'\| < \delta$, $g(w') \subseteq g(w) \oplus B(0, \epsilon)$ holds.*

Proof Pick $w \in W$, $\epsilon > 0$, and let $\delta = \epsilon / L(r, a)$. We want to show that for all $w' \in W$ with $\|w - w'\| < \delta$ and $y' \in g(w')$ there exists a $y \in g(w)$ that is only ϵ-apart. Then the statement in the proposition follows. Define x such that $f(x, a, w') = y'$, which is possible by definition of $g(w')$, and let $y = f(x, a, w)$, which implies $y \in g(w)$. By Lipschitz continuity of f, we obtain that $\|y - y'\| = \|f(x, a, w) - f(x, a, w')\| \leqslant L(r, a)\|w - w'\| < \epsilon$. Thus the proof is concluded. ■

Proposition 19 *The events in Expression (2) are measurable.*

Proof Denote by $\mathcal{F}$, $\mathcal{G}$ and $\mathcal{K}$ respectively the sets of all closed, open and compact subsets of $\mathbb{R}^n$ with respect to the usual topology. Consider the Fell topology $\mathcal{T}(\mathcal{F})$ on $\mathcal{F}$, i.e. the one generated by the sets

$$\mathcal{F}^K := \left\{F \in \mathcal{F}(\mathbb{R}^n) : F \bigcap K = \varnothing\right\} \qquad \text{and} \qquad \mathcal{F}_G := \left\{F \in \mathcal{F}(\mathbb{R}^n) : F \bigcap G \neq \varnothing\right\},$$

for all $G \in \mathcal{G}$ and $K \in \mathcal{K}$. Denote by $\mathcal{B}(\mathcal{F})$ the Borel σ-algebra on $(\mathcal{F}, \mathcal{T})$. Denote also by $\mathcal{T}(W)$ the usual topology on W and by $\mathcal{B}(W)$ the Borel σ-algebra on W. For simplicity, assume that $\text{Reach}(r, a, w)$ is closed and note that if it is not, we can always consider its closure without compromising the soundness of the approach. First, we establish that the function g is $(W, \mathcal{B}(W))$-$(\mathcal{F}, \mathcal{B}(\mathcal{F}))$ measurable by proving a sufficient condition, namely, upper semi-continuity of g: g is upper semi-continuous if $g^-(K) := \{w \in W : g(w) \bigcap K \neq \varnothing\}$ is closed on $(W, \mathcal{T}(W))$ for all $K \in \mathcal{K}$. We proceed by contradiction: assume that $g^-(K)$ is not closed for some $K \in \mathcal{K}$. Then, there must exist a $w_\infty \in W \backslash g^-(K)$ and a sequence $\{w_n\}_{n \in \mathbb{N}} \subset g^-(K)$ that converges to w_∞. Since K and $g(w_\infty)$ are closed and non-intersecting, there exists $\epsilon > 0$ such that $(g(w_\infty) \oplus B(0, \epsilon)) \bigcap K = \varnothing$. By convergence of $\{w_n\}_{n \in \mathbb{N}}$, for every $\delta > 0$ $\exists N \in \mathbb{N}$ such that $\|w_\infty - w_N\| < \delta$. This implies, by Lemma 18, that we can choose N large enough so that we ensure, $g(w_N) \subseteq g(w_\infty) \oplus B(0, \epsilon)$. It follows that $g(w_N) \bigcap K \subseteq (g(w_\infty) \oplus B(0, \epsilon)) \bigcap K = \varnothing$, which proves that $w_N \notin g^-(K)$, being this result a contradiction. Therefore $g^-(K)$ is closed in $(W, \mathcal{T}(W))$ for all $K \in \mathcal{K}$, making g upper-semicontinuous and measurable, i.e., $w \mapsto \text{Reach}(r, a, w)$ is a well-defined random set Molchanov and Molchanov (2005). Finally, since $\tilde{r}$ can be written as the union of countably many closed sets, the events $\{F \in \mathcal{F} : F \bigcap \tilde{r} \neq \varnothing\}$ and $\{F \in \mathcal{F} : F \subseteq \tilde{r}\}$ are measurable, i.e., they belong to $\mathcal{B}(\mathcal{F})$, which implies measurability of $\{\omega \in \Omega : \text{Reach}(r, a, w) \bigcap \tilde{r} \neq \varnothing\}$ and $\{\omega \in \Omega : \text{Reach}(r, a, w) \subseteq \tilde{r}\}$, thus concluding the proof. ■

Appendix B. Details of the Case Studies

In this section we describe in detail our case-studies, and provide precise values of all system and approach-related parameters.

We consider 3 different specifications:

- Reach-avoid $\varphi_1 := \Box(\neg \mathfrak{p}_{\text{unsafe}}) \wedge \Diamond \text{goal}$,
- Complex LTL$_f$ specification $\varphi_2 := \Box(\neg \mathfrak{p}_{\text{unsafe}}) \wedge \Box(\text{water} \rightarrow (\neg \text{charge}\, \mathcal{U} \text{carpet})) \wedge \Diamond(\text{charge})$ considered in (Vazquez-Chanlatte et al., 2018) and representing the task of reaching a charge station while remaining safe and, if the system goes through a region with water, then first drying in a carpet before charging,
- 15-step safety specification $\varphi_3 := \Box^{\leqslant 15}(\neg \mathfrak{p}_{\text{unsafe}})$ considered in Abate et al. (2010),

where $\Box^{\leqslant 15}$ is the *bounded globally* operator, which is interpreted as "something happens at all time steps less or equal than 15".

B.1. Pendulum

Case studies #1 and #2 both consider the problem of swinging up a pendulum, starting from the downward orientation, without ever exceeding some limits in its angular velocity. The state of the system is 2-dimensional, composed by the angle $\boldsymbol{\theta}_k$ w.r.t. the vertical (downward) position, and the angular velocity $\dot{\boldsymbol{\theta}}_k$, whereas the control input u_k is the torque applied at the joint. The disturbance $\boldsymbol{w}_k$ corresponds to a horizontal wind disturbance, which generates an aerodynamic drag force that is quadratic in $\boldsymbol{w}_k$ and depends on the state in a nonlinear fashion. The system dynamics are

$$\begin{bmatrix} \boldsymbol{\theta}_{k+1} \\ \dot{\boldsymbol{\theta}}_{k+1} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\theta}_k + \Delta t \dot{\boldsymbol{\theta}}_k \\ \dot{\boldsymbol{\theta}}_k + \Delta t\big(- c_d \operatorname{sign}(l\dot{\boldsymbol{\theta}}_k - \boldsymbol{w}_k \cos(\boldsymbol{\theta}_k))(l\dot{\boldsymbol{\theta}}_k - \boldsymbol{w}_k \cos(\boldsymbol{\theta}_k))^2 - \sin(\boldsymbol{\theta}_k) + u_k\big) \end{bmatrix}.$$

In the first case study in Table 1, we let $\Delta t = 0.25$, $c_d = 0.3$ and $\boldsymbol{w}_k$ is distributed according to a zero-mean Gaussian distribution with covariance 0.04, truncated in the interval $[-1, 1]$. We also let X be the set of all states $[\theta, \dot{\theta}]$ such that $|\dot{\theta}| \leqslant 3$, and define U by uniformly discretizing the interval $[-0.8, 0.8]$ into 5 values. Furthermore, the goal set is defined as the set of states $[\theta, \dot{\theta}]$ such that θ is within 0.628 of the upward position and $|\dot{\theta}| \leqslant 0.6$. On the other hand, in the second case study in Table 1 we let $\Delta t = 0.3$, $c_d = 0.2$ and $\boldsymbol{w}_k \sim \mathcal{N}(0, 0.0625)$. We also let X contain all states with an angular velocity $|\dot{\theta}| \leqslant 4$, and define U by discretizing the interval $[-0.415, 0.415]$ as explained before. Note that the limit in the maximum control input in case study #2 makes it impossible to swing up the pendulum in one go, which can only be achieved by swinging it back and forth. This is the reason why we refer to case study #2 as "torque limited". Furthermore, we let the goal set contain all states $[\theta, \dot{\theta}]$ such that θ is within 0.628 of the upward position and $|\dot{\theta}| \leqslant 0.6$. In both case studies we partition the state-space via a uniform grid. Finally, in case study #2 we use $\epsilon_c = 0.001$. When constructing our UMDP abstractions, we define the coarse clusters q in such a way that each contains 4 states of the abstraction, i.e., each corresponds to 4 regions of the state-space partition (see Figure 1).

B.2. 3D Unicycle

The state of the system is 3-dimensional, composed by the 2D components $[\boldsymbol{x}_k, \boldsymbol{y}_k, \boldsymbol{\theta}_k]$ of the position in the plane and the unicycle's heading angle. The control inputs are the linear velocity and the yaw rate. The disturbance $\boldsymbol{w}_k$ represents *Coulomb friction*, which generates a force in the

opposite direction of the linear velocity. Its dynamics are the following:

$$\begin{bmatrix} \boldsymbol{x}_{k+1} \\ \boldsymbol{y}_{k+1} \\ \boldsymbol{\theta}_{k+1} \end{bmatrix} = \begin{bmatrix} \boldsymbol{x}_k + \Delta t(u_k^{(1)} - c_d^{(1)}\boldsymbol{w}_k^{(1)})\cos(\boldsymbol{\theta}_k) \\ \boldsymbol{y}_k + \Delta t(u_k^{(1)} - c_d^{(1)}\boldsymbol{w}_k^{(1)})\sin(\boldsymbol{\theta}_k) \\ \boldsymbol{\theta}_k + \Delta t\, u_k^{(2)} + c_d^{(2)}\boldsymbol{w}_k^{(2)} \end{bmatrix} \tag{9}$$

Note that the effect of the disturbance is nonlinear in the state. In the third and fourth case study in Table 1, we let $\Delta t = 0.5$, $c_d = [0.1, 0.05]$. We also let X be the set of all states whose position is in $[0,1]^2$, except for a rectangular obstacle in the center of X whose edges are 0.154 long (see Figure 2). In case study $\#3$, $\boldsymbol{w}_k$ is distributed according to a Gaussian distribution with mean $[0.4, 0]^T$ and covariance $\mathrm{diag}(0.067^2, 0.067^2)$, truncated in a ball of radius 1 centered on the mean. We also define $U := \{0.21, 0.3\} \times \{-2, -1, 0, 1, 2\}$. Furthermore, the goal set is defined as the set of states whose position components lie inside the box $[0, 0.359]^2$. On the other hand, in case study $\#4$, $\boldsymbol{w}_k$ is distributed according to a non-truncated Gaussian distribution with the same mean as in case study $\#3$, but a significantly larger covariance of $\mathrm{diag}(0.2^2, 0.2^2)$. We also let $U := \{0.15, 0.3\} \times \{-2, -1, 0, 1, 2\}$, and define the goal set as the smaller box $[0, 0.256]^2$. Given the smaller goal set and the bigger (and unbounded) noise, we denote case study $\#4$ as "3D unicycle (difficult)". In both case studies we partition the state-space via a uniform grid. Finally, in case study $\#4$ we use $\epsilon_c = 0.01$. When constructing our UMDP abstractions, we define the coarse clusters q in such a way that each contains 27 states of the abstraction, i.e., each corresponds to 27 regions of the state-space partition (see Figure 1).

B.3. Multiplicative Noise

Case study $\#5$ is a 2-dimensional system with multiplicative noise taken from Skovbekk et al. (2023). When constructing our UMDP abstractions, we define the coarse clusters q in such a way that each contains 9 states of the abstraction, i.e., each corresponds to 9 regions of the state-space partition (see Figure 1).

B.4. 2D Unicycle

Case study $\#6$ is a 2-dimensional unicycle model obtained from (9) by fixing the first component of the input to 0.3 and by considering the heading angle θ_k as the input. The latter takes values in the set U, obtained via a uniform discretization of the interval $[-\pi, \pi]$ into 8 values. We consider $c_d = 0.2$, $\Delta t = 0.5$, and let $\boldsymbol{w}_k$ be distributed according to a Gaussian with mean 0.4 and variance 0.067^2, truncated in the interval $[-0.6, 1.4]$. The size and position of the regions of interest is depicted in Figure 3 (d-f). We discretize the state space uniformly, and when constructing our UMDP abstractions, we define the coarse clusters q in such a way that each contains 4 states of the abstraction, i.e., each corresponds to 4 regions of the state-space partition (see Figure 1).

B.5. 4-Room Heating

Case study $\#7$ is a 4-dimensional system whose state is composed by the temperatures of 4 rooms, and its dynamics are taken from Abate et al. (2010) and modified to make the noise multiplicative:

$$\boldsymbol{x}_{k+1} = \mathrm{diag}(1 + \boldsymbol{w}_k)A\boldsymbol{x}_t + b + b_u u_k,$$

with

$$A := \begin{bmatrix} 0.901 & 0.0625 & 0 & 0 & \\ 0.0625 & 0.839 & 0.0625 & 0 & 0 \\ 0 & 0.0625 & 0.839 & 0.0625 & \\ & 0 & 0 & 0.0625 & 0.901 \end{bmatrix}, \quad b := \begin{bmatrix} 0.219 \\ 0.219 \\ 0.219 \\ 0.219 \end{bmatrix},$$

and $b_u = \text{diag}(0.7, 0.7, 0.7, 0.7)$. We let each component of $\boldsymbol{w}_k$ be Gaussianly distributed with mean 0 and covariance 1.11×10^{-5}, and let all components be independent from each other. We define the safe set as $X := [18.5, 23.5]^4$. Note that even though the covariance of the disturbance is small, its effect on the dynamics is way bigger because of the multiplicative nature of $\boldsymbol{w}_k$. We define the set U as the set of all binary-valued vectors in $\mathbb{R}^n$, meaning that each control input $u \in U$ is such that $u^{(i)} = 1$ if the radiator of the i-th room is on and 0 if it is off, for all $i \in \{1, 2, 3, 4\}$. We partition the state-space uniformly to obtain the abstraction, and use $\epsilon_c = 0.001$.

When constructing our UMDP abstractions, we define the coarse clusters q in such a way that each contains 81 states of the abstraction, i.e., each corresponds to 81 regions of the state-space partition (see Figure 1).

References

Alessandro Abate, Joost-Pieter Katoen, John Lygeros, and Maria Prandini. Approximate model checking of stochastic hybrid systems. *European Journal of Control*, 16(6):624–641, 2010.

Steven Adams, Morteza Lahijanian, and Luca Laurenti. Formal control synthesis for stochastic neural network dynamic models. *IEEE Control Systems Letters*, 6:2858–2863, 2022.

Thom Badings, Licio Romao, Alessandro Abate, and Nils Jansen. Probabilities are not enough: Formal controller synthesis for stochastic dynamical models with epistemic uncertainty. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 14701–14710, 2023a.

Thom Badings, Licio Romao, Alessandro Abate, David Parker, Hasan A Poonawala, Marielle Stoelinga, and Nils Jansen. Robust control for dynamical systems with non-gaussian noise via formal abstractions. *Journal of Artificial Intelligence Research*, 76:341–391, 2023b.

Thom Badings, Wietze Koops, Sebastian Junges, and Nils Jansen. Learning-based verification of stochastic dynamical systems with neural network policies. *arXiv preprint arXiv:2406.00826*, 2024.

Calin Belta, Antonio Bicchi, Magnus Egerstedt, Emilio Frazzoli, Eric Klavins, and George J Pappas. Symbolic planning and control of robot motion [grand challenges of robotics]. *IEEE Robotics & Automation Magazine*, 14(1):61–70, 2007.

Dimitri Bertsekas and Steven E Shreve. *Stochastic optimal control: the discrete-time case*, volume 5. Athena Scientific, 1996.

Nathalie Cauchi, Luca Laurenti, Morteza Lahijanian, Alessandro Abate, Marta Kwiatkowska, and Luca Cardelli. Efficiency through uncertainty: Scalable formal synthesis for stochastic hybrid systems. In *Proceedings of the 22nd ACM international conference on hybrid systems: computation and control*, pages 240–251, 2019.

Rudi Coppola, Andrea Peruffo, and Manuel Mazo Jr. Data-driven abstractions for verification of deterministic systems. *arXiv preprint arXiv:2211.01793*, 2022.

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

Giuseppe De Giacomo and Moshe Y Vardi. Linear temporal logic and linear dynamic logic on finite traces. In *IJCAI'13 Proceedings of the Twenty-Third international joint conference on Artificial Intelligence*, pages 854–860. Association for Computing Machinery, 2013.

Laurent El Ghaoui and Arnab Nilim. Robust solutions to markov decision problems with uncertain transition matrices. *Operations Research*, 53(5):780–798, 2005.

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

Ibon Gracia, Dimitris Boskos, Morteza Lahijanian, Luca Laurenti, and Manuel Mazo Jr. Distributionally robust strategy synthesis for switched stochastic systems. *arXiv preprint arXiv:2212.14260*, 2022.

Ibon Gracia, Dimitris Boskos, Luca Laurenti, and Morteza Lahijanian. Data-driven strategy synthesis for stochastic systems with unknown nonlinear disturbances. *arXiv preprint arXiv:2406.09704*, 2024.

Ibon Gracia, Dimitris Boskos, Morteza Lahijanian, Luca Laurenti, and Manuel Mazo Jr. Efficient strategy synthesis for switched stochastic systems with distributional uncertainty. *Nonlinear Analysis: Hybrid Systems*, 55:101554, 2025.

Garud N Iyengar. Robust dynamic programming. *Mathematics of Operations Research*, 30(2): 257–280, 2005.

John Jackson, Luca Laurenti, Eric Frew, and Morteza Lahijanian. Formal verification of unknown dynamical systems via gaussian process regression. *arXiv preprint arXiv:2201.00655*, 2021a.

John Jackson, Luca Laurenti, Eric Frew, and Morteza Lahijanian. Strategy synthesis for partially-known switched stochastic systems. In *Proceedings of the 24th International Conference on Hybrid Systems: Computation and Control*, pages 1–11, 2021b.

Milad Kazemi, Rupak Majumdar, Mahmoud Salamati, Sadegh Soudjani, and Ben Wooding. Data-driven abstraction-based control synthesis. *Nonlinear Analysis: Hybrid Systems*, 52:101467, 2024.

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

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

Mathias Lechner, D. Žikelić, Krishnendu Chatterjee, and Thomas A Henzinger. Stability verification in stochastic control systems via neural network supermartingales. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 7326–7336, 2022.

Frederik Baymler Mathiesen, Licio Romao, Simeon C Calvert, Alessandro Abate, and Luca Laurenti. Inner approximations of stochastic programs for data-driven stochastic barrier function design. In *2023 62nd IEEE Conference on Decision and Control (CDC)*, pages 3073–3080. IEEE, 2023.

Rayan Mazouz, Frederik Baymler Mathiesen, Luca Laurenti, and Morteza Lahijanian. Piecewise stochastic barrier functions. *arXiv preprint arXiv:2404.16986*, 2024.

Ilya Molchanov and Ilya S Molchanov. *Theory of random sets*, volume 19. Springer, 2005.

Ali Salamati, Abolfazl Lavaei, Sadegh Soudjani, and Majid Zamani. Data-driven verification and synthesis of stochastic systems through barrier certificates. *arXiv preprint arXiv:2111.10330*, 2021.

Oliver Schön, Birgit van Huijgevoort, Sofie Haesaert, and Sadegh Soudjani. Bayesian approach to temporal logic control of uncertain systems. *arXiv preprint arXiv:2304.07428*, 2023.

John Skovbekk, Luca Laurenti, Eric Frew, and Morteza Lahijanian. Formal abstraction of general stochastic systems via noise partitioning. *IEEE Control Systems Letters*, 2023.

Roberto Tempo, Giuseppe Calafiore, Fabrizio Dabbene, et al. *Randomized algorithms for analysis and control of uncertain systems: with applications*, volume 7. Springer, 2013.

Marcell Vazquez-Chanlatte, Susmit Jha, Ashish Tiwari, Mark K Ho, and Sanjit Seshia. Learning task specifications from demonstrations. *Advances in neural information processing systems*, 31, 2018.

Wolfram Wiesemann, Daniel Kuhn, and Berç Rustem. Robust markov decision processes. *Mathematics of Operations Research*, 38(1):153–183, 2013.

Eric M Wolff, Ufuk Topcu, and Richard M Murray. Robust control of uncertain markov decision processes with temporal logic specifications. In *2012 IEEE 51st IEEE Conference on decision and control (CDC)*, pages 3372–3379. IEEE, 2012.