Symbolic Controller Synthesis for Büchi Specifications on Stochastic Systems

Rupak Majumdar
MPI-SWS, Germany.
rupak@mpi-sws.org

Kaushik Mallik
MPI-SWS, Germany.
kmallik@mpi-sws.org

Sadegh Soudjani
Newcastle University, UK.
sadegh.soudjani@ncl.ac.uk

ABSTRACT

We consider the policy synthesis problem for continuous-state controlled Markov processes evolving in discrete time, when the specification is given as a Büchi condition (visit a set of states infinitely often). We decompose computation of the maximal probability of satisfying the Büchi condition into two steps. The first step is to compute the maximal *qualitative winning set*, from where the Büchi condition can be enforced with probability one. The second step is to find the maximal probability of reaching the already computed qualitative winning set. In contrast with finite-state models, we show that such a computation only gives a lower bound on the maximal probability where the gap can be non-zero.

In this paper we focus on approximating the qualitative winning set, while pointing out that the existing approaches for unbounded reachability computation can solve the second step. We provide an abstraction-based technique to approximate the qualitative winning set by simultaneously using an over- and under-approximation of the probabilistic transition relation. Since we are interested in qualitative properties, the abstraction is non-probabilistic; instead, the probabilistic transitions are assumed to be under the control of a (fair) adversary. Thus, we reduce the original policy synthesis problem to a Büchi game under a fairness assumption and characterize upper and lower bounds on winning sets as nested fixed point expressions in the μ-calculus. This characterization immediately provides a symbolic algorithm scheme. Further, a winning strategy computed on the abstract game can be refined to a policy on the controlled Markov process.

We describe a concrete abstraction procedure and demonstrate our algorithm on two case studies. We show that our techniques are able to provide tight approximations to the qualitative winning set for the Van der Pol oscillator and a 3-d Dubins' vehicle.

1 INTRODUCTION

Decision making under stochastic uncertainty has many applications in science, engineering, and economics. Typically, one models a system with uncertainty as a *controlled Markov process* evolving in time. Such a process consists in a (possibly uncountable) set of states and actions. At a given state, an agent picks an action and the state and the action together determine the distribution over the next states. The choice of the control action depends on the history of states seen so far and may be randomized; the decision rule that assigns to each history a distribution of control actions is called a *policy*. Given a temporal specification over trajectories, the goal of the agent is to find an *optimal* policy: one that maximizes the probability that the resulting trajectory of the system satisfies the specification. The control problem asks, given a controlled Markov process and a temporal specification (given, e.g., in linear temporal logic), to design an optimal policy. In the finite-state setting, the control problem can be solved algorithmically based on graph traversal and linear programming [1, 3, 6]. A lot of recent research has focused on extending algorithmic policy synthesis techniques to *continuous-state* systems. The goal for continuous-state systems is to provide approximations to the probability of satisfaction, while providing formal guarantees on convergence of the error.

While synthesis for reachability and safety properties have been studied in this setting both for infinite horizon [9, 33] and for finite horizon [13, 14, 18–20, 26, 34], there are few techniques for synthesis against *Büchi* specifications, which requires the trajectory to visit a given set of states infinitely often. In this paper, we consider the problem of synthesizing controllers for controlled Markov processes for properties specified as Büchi conditions.

The key aspect of the solution in the finite-state case is to separate a synthesis problem into a *qualitative* part (find the set of states from which the agent has a policy to satisfy the property almost surely) and a *quantitative* part (find the policy that maximizes the probability of reaching the qualitatively winning states). Given the qualitative solution, one can iteratively compute the quantitative solution by solving a reachability problem, where the target is the absorbing set given by the qualitative solution.

Our first contribution is to show that a similar decomposition for Büchi properties does not hold for continuous state systems in general: we provide an example of a Markov process over continuous state space for which the qualitative *winning set* (from which there is a policy that ensures the Büchi property holds almost surely) is empty but the maximal probability of satisfying the property has a non-zero solution. Moreover, we show that such a decomposition is able to provide a lower bound on the quantitative part of the problem. Thus, if one can compute (an approximation of) the

winning set, a lower bound on the quantitative solution can be obtained by Bellman iteration or by other techniques for (unbounded) reachability in the continuous-state setting [9, 33].

As our second contribution, we provide *symbolic* algorithms for computing under- and over-approximations of the qualitative winning set. We compute finite-state abstractions of the continuous-state system. Our abstraction uses two transition relations: an over-approximation and an under-approximation of the continuous transitions. For qualitative probabilistic analysis on finite-state systems, one can replace the probabilistic transitions by an adversarial scheduler with a fairness requirement [10]. Accordingly, our abstractions are *non-probabilistic* and only require the knowledge of the support of the stochastic kernel associated to the process. We characterize the qualitative winning states as a nested fixed point expression in the μ-calculus [17]; such an expression naturally gives a symbolic implementation. Since the abstraction is non-probabilistic, the symbolic implementation avoids numerical issues and can use standard encodings based on satisfiability checkers or binary decision diagrams.

Our fixed point characterization is similar to the characterization of qualitative almost-sure winning in concurrent games [7], but the use of two kinds of transitions—acting as upper and lower bounds—is a key distinguishing ingredient in our characterization. We show through examples why both are required. The qualitative winning region in the original continuous-state process is *not* characterized by a similar fixed point: it is well known that unlike the finite-state case, the actual probability values matter in deciding qualitative winning in infinite-state systems [1, pp. 779-780].

We demonstrate our approach on the Van der Pol oscillator and the 3-d Dubin's vehicle, both in the presence of stochastic perturbation. Our computation shows that when the disturbance is treated as a worst case adversary, there exists a deterministic value of the disturbance for which the specification is violated for all initial states. On the contrary, when the disturbance is treated as random, we are able to satisfy the specification almost surely. Moreover, we empirically show that the difference between the over- and under-approximation reduces as we pick finer abstractions.

There are relatively few results on the algorithmic analysis of liveness properties for controlled Markov processes. A theoretical study of the Büchi objective $\square\lozenge B$ is conducted by Tkachev et al. [33] via persistence properties $\lozenge\square\overline{B}$. It is shown that $P_s(\lozenge\square\overline{B})$ can be characterized by two fixed-point equations but no computational method is provided. In particular, their techniques do not provide a way to solve the qualitative Büchi problem that we solve. Our computational approach is similar in nature to results that employ Interval MDP or Interval Markov chains (MC) as abstractions. For example, Dutreix and Coogan [8] use Interval MC for verification of a particular class of systems. The method of Dutreix and Coogan requires numerical computations of lower and upper bounds of the probabilities and provides an enumerative algorithm. Our approach generalizes their construction of the over- and under-approximations to the setting of controlled Markov processes and the synthesis problem. We focus on the winning region of the specification, which allows us to write symbolic algorithms purely on non-probabilistic structures, thus avoiding numerical

optimization procedures. Once the winning region is approximated, a quantitative reachability can be solved using standard techniques.

2 CONTROLLED MARKOV PROCESSES

2.1 Preliminaries

We consider a probability space $(\Omega, \mathcal{F}_\Omega, P_\Omega)$, where Ω is the sample space, $\mathcal{F}_\Omega$ is a sigma-algebra on Ω comprising subsets of Ω as events, and P_Ω is a probability measure that assigns probabilities to events. We assume that random variables introduced in this article are measurable functions of the form $X : (\Omega, \mathcal{F}_\Omega) \to (S_X, \mathcal{F}_X)$. Any random variable X induces a probability measure on its space $(S_X, \mathcal{F}_X)$ as $Prob\{A\} = P_\Omega\{X^{-1}(A)\}$ for any $A \in \mathcal{F}_X$. We often directly discuss the probability measure on $(S_X, \mathcal{F}_X)$ without explicitly mentioning the underlying probability space and the function X itself.

A topological space S is called a Borel space if it is homeomorphic to a Borel subset of a Polish space (i.e., a separable and completely metrizable space). Examples of a Borel space are the Euclidean spaces $\mathbb{R}^n$, its Borel subsets endowed with a subspace topology, as well as hybrid spaces. Any Borel space S is assumed to be endowed with a Borel sigma-algebra, which is denoted by $\mathcal{B}(S)$. We say that a map $f : S \to Y$ is measurable whenever it is Borel measurable.

We denote the set of nonnegative integers by $\mathbb{N} := \{0, 1, 2, \ldots\}$.

2.2 Controlled Markov processes

We consider controlled Markov processes (CMP) in discrete time defined over a general state space, characterized by a tuple $\mathfrak{S} = (S, \mathcal{U}, T_s)$, where S is a Borel space serving as the state space of the process, $\mathcal{U}$ is a finite input space, and T_s is a conditional stochastic kernel $T_s : \mathcal{B}(S) \times S \times \mathcal{U} \to [0, 1]$ with $\mathcal{B}(S)$ being the Borel sigma-algebra on the state space and $(S, \mathcal{B}(S))$ being the corresponding measurable space. The kernel T_s assigns to any $s \in S$ and $u \in \mathcal{U}$ a probability measure $T_s(\cdot|s, u)$ on the measurable space $(S, \mathcal{B}(S))$ so that for any set $A \in \mathcal{B}(S), P_{s,u}(A) = \int_A T_s(ds|s, u)$, where $P_{s,u}$ denotes the conditional probability $P(\cdot|s, u)$.

REMARK 1. *The input space $\mathcal{U}$ in general can be any Borel space and the set of valid inputs can be state dependent. We have considered that $\mathcal{U}$ is a finite set and all elements of this set can be taken at any state. This choice is motivated by the digital implementation of control policies and also facilitates concise presentation of the results.*

2.3 Semantics of controlled Markov processes

The semantics of a CMP are characterized by its *paths* or executions, which reflect both the history of previous states of the system and of implemented control inputs. Paths are used to measure the performance of the system.

Definition 2.1. Given a CMP $\mathfrak{S}$, a *finite path* is a sequence

$$w_n = (s_0, u_0, \ldots, s_{n-1}, u_{n-1}, s_n), \quad n \in \mathbb{N},$$

where $s_i \in S$ are state coordinates and $u_i \in \mathcal{U}$ are control input coordinates of the path. The space of all paths of length n is denoted by $\text{PATH}_n := \mathcal{K}^n \times S$ with $\mathcal{K} := S \times \mathcal{U}$. Further, we denote projections by $w_n[i] := s_i$ and $w_n(i) := u_i$. An *infinite path* of the CMP $\mathfrak{S}$ is the sequence $w = (s_0, u_0, s_1, u_1, \ldots)$, where $s_i \in S$ and $u_i \in \mathcal{U}$ for all $i \in \mathbb{N}$. As above, let us introduce $w[i] := s_i$ and

$w(i) := u_i$. The space of all infinite paths is denoted by $\text{PATH}_\infty := \mathcal{K}^\infty$.

Given an infinite path w or a finite path w_n, we assume below that s_i and u_i are their state and control coordinates respectively, unless otherwise stated. For any infinite path $w \in \text{PATH}_\infty$, its n-prefix (ending in a state) w_n is a finite path of length n, which we also call n-history. We are now ready to introduce the notion of control policy.

Definition 2.2. A *policy* is a sequence $\rho = (\rho_0, \rho_1, \rho_2, \ldots)$ of universally measurable stochastic kernels ρ_n [2], each defined on the input space $\mathcal{U}$ given PATH_n. The set of all policies is denoted by Π.

Given a policy $\rho \in \Pi$ and a finite path $w_n \in \text{PATH}_n$, the distribution of the next control input u_n is given by $\rho_n(\cdot|w_n)$. In this work, we restrict our attention to the class of *stationary* policies.

Definition 2.3. A policy ρ is *stationary* if there is a universally measurable function $C : \mathcal{S} \to \mathcal{U}$ such that at any time epoch $n \in \mathbb{N}$, the input u_n is taken to be $C(s_n) \in \mathcal{U}$. Namely, the stochastic kernel $\rho_n(\cdot|w_n)$, $n \in \mathbb{N}$, in Definition 2.2 is a Dirac delta measure centered at $C(s_n)$ with $s_n = w_n[n]$ being the last element of w_n. We denote the class of stationary policies by $\Pi_S \subset \Pi$ and a stationary policy just by the function $C \in \Pi_S$. The function C is also called *state feedback controller* in control theory.

For a CMP $\mathfrak{S}$, any policy $\rho \in \Pi$ together with an initial probability measure $\alpha : \mathcal{B}(\mathcal{S}) \to [0, 1]$ of the CMP induces a unique probability measure on the canonical sample space of paths [11] denoted by P_α^ρ with the expectation $\mathbb{E}_\alpha^\rho$. In the case when the initial probability measure is supported on a single point, i.e., $\alpha(s) = 1$, we write P_s^ρ and $\mathbb{E}_s^\rho$ in place of P_α^ρ and $\mathbb{E}_\alpha^\rho$, respectively. We denote the set of probability measures on $(\mathcal{S}, \mathcal{B}(\mathcal{S}))$ by $\mathfrak{D}$.

3 PROBLEM DEFINITION

Liveness specification. We consider liveness or repeated reachability specification as the synthesis objective. Given a measurable set of states $B \subseteq \mathcal{S}$, the liveness specification is denoted by $\square\lozenge B$ in linear temporal logic (LTL) notation [1]. An infinite path $w \in \text{PATH}_\infty$ of a CMP $\mathfrak{S}$ satisfies the liveness specification $\square\lozenge B$ if for all $k_0 \in \mathbb{N}$, there exists $k_1 \in \mathbb{N}$ such that $k_1 > k_0$ and $w[k_1] \in B$. This requires that the path w visits the set B infinitely often. We indicate the set of all infinite paths $w \in \text{PATH}_\infty$ of $\mathfrak{S}$ that satisfy the property $\square\lozenge B$ by $\mathfrak{S} \models \square\lozenge B$.

We are interested in the probability that the liveness specification can be satisfied by paths of a CMP $\mathfrak{S}$ under different policies. Given a policy $\rho \in \Pi$ and initial state $s \in \mathcal{S}$, we define the satisfaction probability as

$$f(s, \rho) := P_s^\rho(\mathfrak{S} \models \square\lozenge B), \tag{1}$$

and the supremum satisfaction probability

$$f^*(s) := \sup_{\rho \in \Pi} P_s^\rho(\mathfrak{S} \models \square\lozenge B). \tag{2}$$

Problem 1 (Policy Synthesis). *Given $\mathfrak{S}$ and a set $B \subseteq \mathcal{S}$, find the optimal policy ρ^* along with $f^*(s)$ s.t. $P_s^{\rho^*}(\mathfrak{S} \models \square\lozenge B) = f^*(s)$.*

Measurability of the event $\{\mathfrak{S} \models \square\lozenge B\}$ in the canonical sample space of paths under the probability measure P_s^ρ is proved in [33].

An initial attempt is also made to study the properties of the function $f^*(\cdot)$. For instance, it is shown that $f^* \equiv 1$ if and only if the probability that the path w reaches B is one for all initial states. We anticipate that the sets where $f^*(s) = 1$ plays a crucial role in the computation of f^*.

Definition 3.1 (Almost sure winning region). Given the CMP $\mathfrak{S}$, the policy ρ, and a set $B \subseteq \mathcal{S}$, the state $s \in \mathcal{S}$ wins the specification $\square\lozenge B$ almost surely (a.s. in short) if $f(s, \rho) = 1$. The a.s. *winning region* of the policy ρ is defined as

$$\text{WinDom}(\mathfrak{S}, \rho) := \{s \in \mathcal{S} \mid f(s, \rho) = 1\}. \tag{3}$$

We also define the *maximal a.s. winning region* as

$$\text{WinDom}^*(\mathfrak{S}) := \{s \in \mathcal{S} \mid f^*(s) = 1\}. \tag{4}$$

Proposition 3.2. *The set $\text{WinDom}^*(\mathfrak{S})$ is universally measurable. The set $\text{WinDom}(\mathfrak{S}, C)$ is also universally measurable for any stationary policy $C \in \Pi_S$. The set $W := \text{WinDom}(\mathfrak{S}, C)$ is an absorbing set, i.e., the paths stating from this set will stay in the set with probability one.*

The proof can be found in the appendix.

In the sequel, we restrict our attention to stationary policies $C \in \Pi_S$ and decompose the computation of $P_s^C(\mathfrak{S} \models \square\lozenge B)$ into the computation of the winning set $\text{WinDom}(\mathfrak{S}, C)$ and then computation of reachability probability $P_s^C(\mathfrak{S} \models \lozenge\text{WinDom}(\mathfrak{S}, C))$. This is formalized next.

Assumption 1. *There exists an optimal policy for (2), and in addition that policy is stationary. Formally, there is a stationary policy $C^* \in \Pi_S$ such that $f^*(s) = f(s, C^*)$ for all $s \in \mathcal{S}$.*

It is known, already for CMPs with countable state spaces, that for Prob. 1 an optimal policy from a state $s \in \mathcal{S}$ may not exist even when $f^*(s)$ is non-zero [15]. As stated in the first part of Assump. 1, we restrict the scope of the rest of the paper to problems for which an optimal policy exists. For the second part of Assump. 1, we do not yet know if stationary policies are sufficient for general CMPs even when an optimal policy is guaranteed to exist. We do know that for CMPs with *countable* state spaces, when an optimal policy exists, then it is stationary [16]. If optimal policies need not be stationary, our proposed algorithms in the following sections would still produce sound but sub-optimal approximations of the set $\text{WinDom}^*(\mathfrak{S})$; see Rem. 6. Nevertheless, even if stationary policies are not sufficient for optimality, they are still of practical interest due to their simple structure and ease of implementation. It is worthwhile to note that similar assumption was used by other authors [33, Assump. 2].

Theorem 3.3. *For any policy $C \in \Pi_S$ on CMP $\mathfrak{S}$, and $W := \text{WinDom}(\mathfrak{S}, C)$, we have*

$$\begin{cases} P_s^C(\mathfrak{S} \models \square\lozenge B) = 1 & \text{if } s \in W \\ P_s^C(\mathfrak{S} \models \square\lozenge B) \geq P_s^C(\mathfrak{S} \models \lozenge W) & \text{if } s \notin W, \end{cases} \tag{5}$$

The proof can be found in the appendix.

Computation of the reachability probability has been studied extensively in the literature for both infinite horizon [9, 31–33] and finite horizon [13, 14, 18–20, 27–29, 34] using different abstract models and computational methods. These approaches can be used

to provide a lower bound on the probability of satisfaction of the Büchi condition. So from this point onward, we mostly consider the first half of (5) which is formalized as in the following:

PROBLEM 2 (MAXIMAL WINNING REGION). *Given* $\mathfrak{S}$ *and a set* $B \subseteq S$, *find a stationary policy* $C^* \in \Pi_S$ *such that*

$$\text{WinDom}(\mathfrak{S}, C) \subseteq \text{WinDom}(\mathfrak{S}, C^*) \text{ almost everywhere}$$

for all $C \in \Pi_S$.

The maximal policy C^* defined in Problem 2 is not necessarily unique, but the winning region associated to such maximal policies is unique. A formal treatment of this claim can be found in the appendix in Sec. 9.4. Moreover, when Assump. 1 holds, using properties of the sets $\text{WinDom}(\mathfrak{S}, C)$ for different policies, we have

$$\text{WinDom}(\mathfrak{S}, C^*) = \text{WinDom}^*(\mathfrak{S}).$$

In the following, we mainly focus on the approximate computation of $\text{WinDom}(\mathfrak{S}, C^*)$ with a suitable policy. For the reachability part, it has been shown that linking the infinite-horizon reachability to the finite-horizon one requires knowledge of the *absorbing sets* from which the trajectory cannot escape. So we briefly discuss in Sec. 5.5 how the absorbing sets can be over-approximated to enable linking the reachability in (5) to its finite-horizon version.

A solution outline for Problem 2. Computing the exact maximal policy for $\mathfrak{S}$ is difficult in general. In the following we propose an approximation procedure which is motivated by the three-step abstraction-based controller synthesis methods [22, 24, 30]:

Abstraction. First, the given CMP $\mathfrak{S}$ is approximated using a finite state transition system $\mathcal{A}$, called the *abstraction*, by means of state space discretization. The specification—which is the Büchi condition in our case—is also approximated using the discretized state space of the abstract transition system, and is called the *abstract specification*.

Synthesis. Second, the policy synthesis problem is posed as a zero-sum *game* on $\mathcal{A}$ between the controller and an adversary, where in each step the controller chooses a control input, and the adversary chooses a successor allowed by the transitions in $\mathcal{A}$. The goal of the control player is a.s. satisfaction of the Büchi objective from as many discrete states as possible, whereas the goal of the environment player is the complement of the same. The outcome of the game, when played from the perspective of the control player, is an *abstract controller* $\widehat{C}$.

Controller refinement. Third, the abstract controller $\widehat{C}$ is mapped back to the continuous state space using a process called *controller refinement*. This results in a continuous controller C that can be paired with $\mathfrak{S}$ and a continuous winning domain $\text{WinDom}(\mathfrak{S}, C)$.

4 A GENERIC FINITE STATE ABSTRACTION

Our proposed solution relies on constructing an abstraction $\mathcal{A}$ which uses two transition functions to approximate the transition kernel of $\mathfrak{S}$.

Definition 4.1. A transition system is a tuple $(Q, \Sigma, \delta_1, \delta_2)$ where Q is a finite set of states, Σ is a finite input alphabet, and $\delta_i : Q \times \Sigma \to 2^Q$, $i \in \{1, 2\}$ are two transition functions with the property that $\delta_1(q, u) \supseteq \delta_2(q, u)$ for all $(q, u) \in Q \times \Sigma$.

The abstraction $\mathcal{A}$ is constructed based on a finite partition of the state space. Therefore, we require a bounded state space S. If S is unbounded, we truncate it to a measurable set S', which serves as the working region; the rest is represented by a symbolic *sink* state ϕ. It is true that this truncation could possibly lead to computation of only an under-approximation of the true a.s. winning region. However, from practical standpoint, it is often the case that bounds of the state variables are already in place (e.g. the work space of a robot, or the maximum rated voltage across a capacitor in a voltage converter, etc.), which would serve as the boundary of S'. The state ϕ models all the out-of-domain behaviors of $\mathfrak{S}$, and should be avoided. The new CMP will be $\mathfrak{S}' := (S' \cup \{\phi\}, \mathcal{U}, T_s')$ with

$$T_s'(A \mid s, u) = \begin{cases} T_s(A \mid s, u) & \text{if } s \in S' \\ 0 & \text{if } s = \phi, \end{cases} \tag{6}$$

for any $A \in \mathcal{B}(S')$, and $T_s'(\phi \mid s, u) = 1 - T_s'(S' \mid s, u)$. In order to avoid change of notation, we work in the sequel with $\mathfrak{S} := (S, \mathcal{U}, T_s)$ where the state space S is bounded but may also include a symbolic sink state ϕ. We also assume that $B \subseteq S$.

4.1 The abstraction

We propose a novel abstraction, which will later be used in Sec. 5 to synthesize approximations of the maximal winning region and a winning policy. First, we introduce some notation. Given the state space S of the CMP $\mathfrak{S}$, we define a finite partition of S denoted by $\widehat{S} := \{\widehat{x}_i\}_{i \in I}$ s.t. $S = \bigcup_{i \in I} \widehat{x}_i$ and $\widehat{x}_i \cap \widehat{x}_j = \emptyset$ for all $\widehat{x}_i, \widehat{x}_j \in \widehat{S}$, $i \neq j$. The set $\widehat{S}$ will be called the abstract state space (state space of the abstraction), and each element $\widehat{x}_i$ is an abstract state.

REMARK 2. *For the theory that is going to be presented in this paper, the abstract states need not be of the same size. However, for a practical implementation, partition sets are chosen to be hyper-rectangular of the form $\widehat{x} = [\![a, b]\!]$ where $a, b \in S$ are vectors. The partition sets are uniformly sized and their boundaries are assigned to only one partition element. In our implementation, we have also used a hyper-rectangular state space with an additional symbolic state ϕ that is also an element of the abstract state space.*

Definition 4.2. Let $\mathfrak{S} = (S, \mathcal{U}, T_s)$ be a CMP and $\widehat{S}$ be a finite partition of S. Suppose $\mathcal{A} = (\widehat{S}, \mathcal{U}, \overline{F}, \underline{F})$ is a transition system with two modes of transitions $\overline{F} : \widehat{S} \times \mathcal{U} \to 2^{\widehat{S}}$ and $\underline{F} : \widehat{S} \times \mathcal{U} \to 2^{\widehat{S}}$. The system $\mathcal{A}$ will be called an *abstraction* of $\mathfrak{S}$ if for all $\widehat{x} \in \widehat{S}$ and all $u \in \mathcal{U}$,

$$\overline{F}(\widehat{x}, u) \supseteq \{\widehat{x}' \in \widehat{S} \mid \exists s \in \widehat{x} . T_s(\widehat{x}' \mid s, u) > 0\}, \tag{7}$$

$$\underline{F}(\widehat{x}, u) \subseteq \{\widehat{x}' \in \widehat{S} \mid \exists \varepsilon > 0 . \forall s \in \widehat{x} . T_s(\widehat{x}' \mid s, u) \geq \varepsilon\}. \tag{8}$$

In words, in the presence of the stochastic disturbance and given an abstract state $\widehat{x}$ and a control input $u \in \mathcal{U}$, $\overline{F}(\widehat{x}, u)$ represents an over-approximation of the set of all abstract states which can be reached with positive probability from *some* continuous state in $\widehat{x}$, and $\underline{F}(\widehat{x}, u)$ represents a subset of $\overline{F}(\widehat{x}, u)$ under-approximating the set of states that can be reached with probability bounded away from 0 from *all* the states in $\widehat{x}$. The role of the lower bound ε in (8) will be clear in Sec. 4.2. We defer the actual computation of the abstract transition system until Sec. 6.

REMARK 3. *The abstract transition systems in the usual abstraction based control methods [22, 24, 30] play the role of a game graph for a two-player zerosum game between the controller and an imaginary adversary, where the adversary is an accumulation of the external perturbation and the discretization-induced non-determinism. The rule of the game is that at each discrete step, the controller plays a control input, to which the adversary responds by choosing one of the many non-deterministic successors. A control policy is synthesized for the controller by treating the adversary actions in worst-case fashion.*

In our case, the controller effectively plays simultaneously against two imaginary adversaries who use two different types of actions: The first adversary—called the random adversary—uses the external random noise, while the second adversary—called the non-deterministic adversary—uses the discretization-induced non-determinism. This separation enables the controller to somewhat relax the worst-case treatment of the problem, by assuming that the random adversary is fair in choosing its actions, meaning all the noise values in the support of the distribution will appear always eventually. The non-deterministic adversary is still treated in the worst-case fashion.

Keeping this two-adversary interpretation in perspective, given some control action, one can interpret a $\overline{F}$-transition as a joint colluding move of the two adversaries, while one can interpret a $\underline{F}$-transition as a move of the lone random adversary. The rule of the game in our case is that at each discrete step, the controller plays a control input, to which either the adversaries jointly respond by choosing one of the $\overline{F}$-successors (which is possibly not an $\underline{F}$-successor), or the random adversary independently chooses an $\underline{F}$-successor. Since the random adversary is fair in its moves, hence it will not collude with the non-deterministic adversary all the time, and all the $\underline{F}$-transitions will be chosen at some point in the long run. This additional fairness assumption in the underlying game creates a favorable condition for the controller in many cases, as will be shown in the next section.

4.2 Almost sure progress

The fairness of the random adversary is materialized using the ε in the definition of $\underline{F}$, which guarantees that a trajectory eventually exits from an abstract state $\widehat{x}$ in the long run, even when there is a non-zero probability for a single-step successor of a continuous state $s \in \widehat{x}$ to stay within $\widehat{x}$. This feature is a central element of our synthesis method that will be presented in Sec. 5.

Following is an example of a general continuous state Markov chain which demonstrates that in the absence of the bound ε in the definition of $\underline{F}$, trajectories could get trapped inside $\widehat{x}$ forever.

Example 4.3. For simplicity and a focused exposition of the actual issue, we use a system admitting no control over state trajectories (can be alternatively thought of as a system with a control input space that has a single element). Consider a one dimensional CMP with state space $\mathcal{S} = [0, 2]$, and with the following transition kernel (see Fig. 1):

$$s_k \in [1, 2] \Rightarrow T_\mathbb{s}([\alpha, \beta] \mid s_k) = (\beta - \alpha),$$

$$s_k = 0 \Rightarrow \begin{cases} T_\mathbb{s}(\{0\} \mid s_k) = 0.5 \\ T_\mathbb{s}([\alpha, \beta] \mid s_k) = 0.5(\beta - \alpha), \end{cases} \quad (9)$$

$$s_k \in (0, 1) \Rightarrow \begin{cases} T_\mathbb{s}(\{b(s_k)\} \mid s_k) = 1 - a(s_k) \\ T_\mathbb{s}([\alpha, \beta] \mid s_k) = a(s_k)(\beta - \alpha), \end{cases}$$
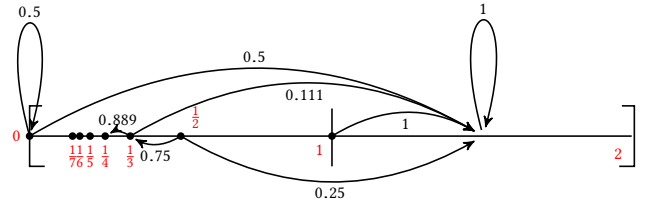

Figure 1: A trajectory could get trapped inside $[0, 1)$ for infinite time if the transition probabilities are allowed to be arbitrarily small. In the figure, the nodes with red labels are continuous states, and the labels on the edges are the probability of the associated transition.

for any $[\alpha, \beta] \subset [1, 2]$ with $\alpha \leq \beta$, and $a : s_k \mapsto [0, 1]$ is some probability assigning function, and $b(s_k) := \frac{s_k}{1+s_k}$. In words, the next state is uniformly distributed over the interval $[1, 2]$ if the current state of the CMP is in the same interval; for current state $s_k = 0$, the next state either stays at zero with probability 0.5 or jumps uniformly to the interval $[1, 2]$; for current state $s_k \in (0, 1)$, the next state jumps to the interval $[1, 2]$ with probability $a(s_k)$ or jumps to a single state $b(s_k)$ with probability $1 - a(s_k)$.

Let us consider two CMPs $\mathfrak{S}_1, \mathfrak{S}_2$ with kernels obtained respectively by $a(s) = s^2$ and $a(s) = 0.5$ for all $s \in (0, 1)$, and compute the probability $P_s(\mathfrak{S}_i \models \Box\Diamond[0, 1))$. For a trajectory starting from an initial state $s_0 \in (0, 1)$, the probability of staying inside $[0, 1)$ is:

$$P_{s_0}(\mathfrak{S}_1 \models \Box[0, 1))$$

$$= \left(1 - s_0^2\right) \cdot \left(1 - \left(\frac{s_0}{1 + s_0}\right)^2\right) \cdot \left(1 - \left(\frac{\frac{s_0}{1+s_0}}{1 + \frac{s_0}{1+s_0}}\right)^2\right) \cdots$$

$$= \left(1 - s_0^2\right) \cdot \left(1 - \frac{s_0^2}{(1 + s_0)^2}\right) \cdot \left(1 - \frac{s_0^2}{(1 + 2s_0)^2}\right) \cdots$$

$$= \prod_{k=0}^{\infty} \left(1 - \frac{s_0^2}{(1 + ks_0)^2}\right)$$

$$= 1 - s_0.$$

Then, for any $s_0 \in (0, 1)$ there is a non-zero probability of staying forever inside $[0, 1)$. For example, for $s_0 = 0.5$, this probability is 0.5. Doing the computations for other states results in

$$P_s(\mathfrak{S}_1 \models \Box\Diamond[0, 1)) = \begin{cases} 1 - s & \text{if } s \in (0, 1) \\ 0 & \text{if } s \in [1, 2] \cup \{0\}. \end{cases}$$

For the second model with $a(s) = 0.5$, we have $P_s(\mathfrak{S}_2 \models \Box\Diamond[0, 1)) = 0$ for all $s \in [0, 2]$. □

Ex. 4.3 clearly shows that unlike discrete MDPs, in case of continuous-space CMPs we cannot ignore the actual value of the probabilities, as otherwise we would have had $P_s(\mathfrak{S}_i \models \Box[0, 1)) = 0$ for both $i \in \{1, 2\}$. So unlike the case of discrete MDPs, we can no longer just use the support of the distribution to find the winning region, and then solve a reachability problem.

Ex. 4.3 also shows that the inequality in (5) can be strict: the winning region is empty but there are states with positive probability

of satisfying the liveness specification. We leave the formulation of conditions under which the equality holds as future work.

The CMP $\mathfrak{S}_1$ in Ex. 4.3 also justifies the use of ε in the definition of $\underline{F}$. Assume that we want to compute an abstraction using the hyper-rectangular cover $\{[0, 1), [1, 2]\}$. The continuous states in the cell $[0, 1)$ have positive transition probability to $[1, 2]$, although there *does not exist* a uniform lower bound $\varepsilon > 0$ of these transition probabilities. As we showed in Ex. 4.3, even then a trajectory can remain trapped inside $[0, 1)$ with a positive probability in the long run. Because of the presence of the ε in the definition of $\underline{F}$, the abstraction will have $\underline{F}([0, 1)) = \emptyset$ (there is no control input, so $\underline{F}$ takes only state as argument).

PROPOSITION 4.4. *Let $\widehat{x} \in \widehat{\mathcal{S}}$ be an abstract state and $u \in U$ be an input s.t. there exists $\widehat{x}' \in \widehat{\mathcal{S}}$, $\widehat{x}' \neq \widehat{x}$, such that $\widehat{x}' \in \underline{F}(\widehat{x}, u)$. Then, there is a policy C such that*

$$P_s^C(\mathfrak{S} \models \Diamond\Box\widehat{x}) = P_s^C(\mathfrak{S} \models \Box\widehat{x}) = 0, \quad \forall s \in \mathcal{S},$$

i.e. the probability of an infinite trajectory getting trapped forever inside $\widehat{x}$ is 0.

PROOF. Since $\widehat{x}' \in \underline{F}(\widehat{x}, u)$, there exists an $\varepsilon > 0$ such that $T_s(\widehat{x}' \mid s, u) \geq \varepsilon$ for all $s \in \widehat{x}$. Define the policy to be the constant one $C(s) = u$ for all $s \in \widehat{x}$ and any other input policy for other states $s \notin \widehat{x}$. We first show that $P_s^C(\mathfrak{S} \models \Box\widehat{x}) = 0$. Note that $P_s^C(\mathfrak{S} \models \Box\widehat{x}) = \lim_{n\to\infty} V_n(s)$, with $V_0(s) = \mathbf{1}_{\widehat{x}}(s)$ and $V_{n+1}(s) = \int_{\widehat{x}} V_n(s') T_s(ds'|s, u)$. It is easy to show inductively that $V_n(s) \leq (1 - \varepsilon)^n$ for all s. By taking the limit as $n \to \infty$ we get the claimed result. For the $\Diamond\Box\widehat{x}$ property, we have $P_s^C(\mathfrak{S} \models \Diamond\Box\widehat{x}) = \lim_{n\to\infty} W_n(s)$ with

$$W_{n+1}(s) = P_s^C(\mathfrak{S} \models \Box\widehat{x}) + \int_{\mathcal{S}\setminus\widehat{x}} W_n(s') T_s(ds'|s, u),$$

which results in $W_n(s) = 0$ for all n. □

Intuitively, Prop. 4.4 says that if there is a $\widehat{x}' \in \underline{F}(\widehat{x}, u)$ with $\widehat{x}' \neq \widehat{x}$, then almost all trajectories reaching $\widehat{x}$ will eventually leave $\widehat{x}$ in finite time with the repeated use of the control input u.

We presented Ex. 4.3 to show that having $T_s(\widehat{x} \mid s, u) > 0$ for all $s \in \widehat{x}$ is inadequate, and we need a uniform lower bound ε for the definition of $\underline{F}$ in (8). In fact, this condition is enough under proper continuity assumptions on the stochastic kernel.

PROPOSITION 4.5. *Suppose the kernel T_s is continuous, i.e., $g(s, u) = \int_{\mathcal{S}} f(s') T_s(ds' \mid s, u)$ is upper (lower) semi-continuous for any upper (lower) semicontinuous function f. Moreover, the partition sets in $\widehat{\mathcal{S}} = \{\widehat{x}_i\}_{i \in I}$ are such that $T_s(\text{int}(\widehat{x}') \mid s, u) = T_s(\text{cl}(\widehat{x}') \mid s, u)$ for all $s \in \text{cl}(\widehat{x})$, $u \in \mathcal{U}$, and $\widehat{x}, \widehat{x}' \in \widehat{\mathcal{S}}$, where $\text{int}(\cdot)$ and $\text{cl}(\cdot)$ indicate respectively the interior and the closure of a set. Then, $\underline{F}(\widehat{x}, u)$ can be defined alternatively as*

$$\underline{F}(\widehat{x}, u) \subseteq \{\widehat{x}' \in \widehat{\mathcal{S}} \mid \forall s \in \text{cl}(\widehat{x}) . T_s(\widehat{x}' \mid s, u) > 0\}. \quad (10)$$

The proof can be found in the appendix.

5 CONTROLLER SYNTHESIS AND REFINEMENT

With the abstraction $\mathcal{A}$ of the given CMP $\mathfrak{S}$ computed in Sec. 4, we now propose algorithms to approximate—from above and below—the maximal a.s. winning region $\text{WinDom}(\mathfrak{S}, C^*)$. As a by-product,

we will also obtain a suitable control policy. We first lift the specification $\Box\Diamond B$ to an abstract specification that can be specified using the states of $\mathcal{A}$. For that we define an under-approximation $\underline{B}$ and an over-approximation $\overline{B}$ of the set $B \subseteq \mathcal{S}$ using the states of $\mathcal{A}$:

$$\underline{B} := \{\widehat{x} \in \widehat{\mathcal{S}} \mid \forall s \in \widehat{x} . B(s) = 1\}$$
$$\overline{B} := \{\widehat{x} \in \widehat{\mathcal{S}} \mid \exists s \in \widehat{x} . B(s) = 1\}.$$

Note that the sink state $\phi \notin \underline{B}$, since we assume that the set B is fully contained in the working region of the CMP. Hence, the satisfaction of $\Box\Diamond\underline{B}$ would ensure that ϕ is always avoided.

To formalize the synthesis process, we introduce four operators:

Controllable predecessor: Define $Cpre_F : 2^{\widehat{\mathcal{S}}} \to 2^{\widehat{\mathcal{S}}}$ for $F \in \{\overline{F}, \underline{F}\}$,

$$Cpre_F : T \mapsto \{\widehat{x} \in \widehat{\mathcal{S}} \mid \exists u \in \mathcal{U} . F(\widehat{x}, u) \subseteq T\}. \quad (11)$$

Cooperative predecessor: Define $Pre_F : 2^{\widehat{\mathcal{S}}} \to 2^{\widehat{\mathcal{S}}}$ for $F \in \{\overline{F}, \underline{F}\}$,

$$Pre_F : T \mapsto \{\widehat{x} \in \widehat{\mathcal{S}} \mid \exists u \in \mathcal{U} . F(\widehat{x}, u) \cap T \neq \emptyset\}. \quad (12)$$

Almost sure predecessor: Define $Apre : 2^{\widehat{\mathcal{S}}} \times 2^{\widehat{\mathcal{S}}} \to 2^{\widehat{\mathcal{S}}}$,

$$Apre : (T, S) \mapsto \{\widehat{x} \in \widehat{\mathcal{S}} \mid \exists u \in \mathcal{U} . \overline{F}(\widehat{x}, u) \subseteq T \wedge \underline{F}(\widehat{x}, u) \cap S \neq \emptyset\}. \quad (13)$$

Uncertain predecessor: Define $Upre : 2^{\widehat{\mathcal{S}}} \times 2^{\widehat{\mathcal{S}}} \to 2^{\widehat{\mathcal{S}}}$,

$$Upre : (T, S) \mapsto \{\widehat{x} \in \widehat{\mathcal{S}} \mid \exists u \in \mathcal{U} . \underline{F}(\widehat{x}, u) \subseteq T \wedge \overline{F}(\widehat{x}, u) \cap S \neq \emptyset\}. \quad (14)$$

5.1 Warm-up: reachability specification

As a warm-up, we first consider under-approximation of the largest winning domain for a.s. satisfaction of the reachability specification $\Diamond B$ in the presence of stochastic noise. For that we take inspiration from the fixed point of a.s. reachability in concurrent two-player games [7], and obtain the following nested fixed-point on the abstract system $\mathcal{A}$:

$$\nu Y . \mu Z . (\underline{B}^c \cap (Apre(Y, Z) \cup Cpre_{\overline{F}}(Z))) \cup \underline{B}, \quad (15)$$

where $\underline{B}^c$ represents the complement of the set $\underline{B}$. Intuitively, the above fixed point computes the largest set Y s.t. from every state $y \in Y \setminus \underline{B}$ there exists control input sequence s.t. either (a) there is a finite sequence of $\underline{F}$-transitions to $\underline{B}$ and no $\overline{F}$-transition outside Y, or (b) there is a bounded finite sequence of $\overline{F}$-transitions all of whose non-deterministic branches reach $\underline{B}$. On the CMP level this means: for all $y \in Y \setminus \underline{B}$ and for all $s \in y$ there exist control input sequence s.t. either (a) there exist paths that enter $\underline{B}$ with positive probability bounded away from zero and all paths stay inside Y with probability 1, or (b) there exist paths that enter $\underline{B}$ with probability 1. It can be shown, by repeated use of Prop. 4.4 on the $\underline{F}$-transitions, that $\underline{B}$ (and hence B) will be reached a.s. by such a control sequence.

The fixed point (15) is in contrast with the usual reachability fixed point for worst case disturbances, which is given as $\mu Z . Cpre_{\overline{F}}(Z) \cup \underline{B}$, where it is required that from every state $z \in Z$, all the non-deterministic branches of $\overline{F}$ reach $\underline{B}$ in at most some finite number of steps. Note that the solution of (15) subsumes the solution of the usual fixed point, and in practice the usual reachability fixed point
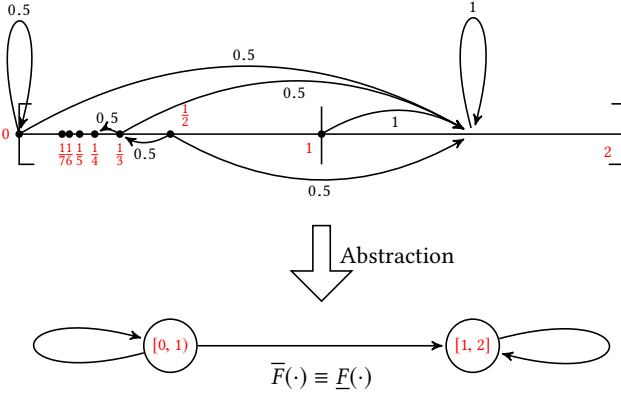
Figure 2: The top figure represents $\mathfrak{S}_2$ and the bottom figure represents both transition relations $\underline{F}([0,1))$ and $\overline{F}([0,1))$ (same in this case) of $\mathcal{A}$. It is not possible to reach $[1,2]$ from $[0,1)$ if the non-determinism in $\underline{F}$ or $\overline{F}$ is treated as a worst-case adversary: the adversary can choose the loop at $[0,1)$ all the time. In the top figure, the labels on the edges are the probabilities of the associated transitions, and in both figures the labels in red are continuous states.

is much stronger than its stochastic counterpart. Following is an illustrative example that captures this intuition.

Example 5.1. Consider the CMP $\mathfrak{S}_2$ (see Fig. 2) defined in Ex. 4.3, with stochastic kernel (9) and $a(s) = 0.5$. The transition probabilities from states in $[0,1)$ to the interval $[1,2]$ have a lower bound 0.5, and so unlike $\mathfrak{S}_1$ of Ex. 4.3, $\underline{F}([0,1)) = \{[0,1),[1,2]\}$. Moreover, since there exist transitions with positive probability from all the state in $[0,1)$ to $[1,2]$, hence $\overline{F}([0,1)) = \{[0,1),[1,2]\}$ as well.

Assume that the abstract reachability specification is given $\Diamond[1,2]$. If we start from some state in $[0,1)$ and treat the adversary (resolving the non-determinism) as worst-case, then by using both $\underline{F}$ or $\overline{F}$, we will forever loop in $[0,1)$ and would violate the specification. Formally, the fixed point for $\Diamond[1,2]$ will converge to the singleton set $\{[1,2]\}$, since $Cpre_{\overline{F}}(\{[1,2]\}) = \{[1,2]\}$.

On the other hand, if we treat the disturbance as stochastic noise, then from Prop. 4.4 we know that if we loop in $[0,1)$ indefinitely long, then in the long run a.s. the system is going to move to $[1,2]$ (recall the interpretation of fair random adversary). So the winning region in this case should be the whole state space $\{[0,1),[1,2]\}$. Indeed, (15) will include the state $[0,1)$ since $[0,1) \in Apre(\{[0,1),[1,2]\},\{[1,2]\})$. □

REMARK 4. *Nilsson et al. [22] introduced augmented transition systems as abstractions of non-stochastic systems. Augmented transition systems embed liveness information in progress groups. If a set of abstract states form a progress group under some control action, then the system eventually leaves the progress group under repeated use of this particular control action. Even though our work deals with an unrelated problem, we note that our fairness assumption on the random adversary allowing the CMP to make progress to escape a given abstract state a.s. (Prop. 4.4) has a similar flavor.*

5.2 Under-approximation of the maximal a.s. winning region

We build up on the intuition of the solution of the a.s. reachability specification, and present the computation of a sound under-approximation of the maximal a.s. winning region $\text{WinDom}(\mathfrak{S}, C^*)$ with a suitable abstract controller $\widehat{C}$. In μ-calculus notation, this under-approximation can be computed as:

$$\underline{W} := \nu Y \,.\, \mu Z \,.\, \Big[(\underline{B}^c \cap (Apre(Y,Z) \cup Cpre_{\overline{F}}(Z)))$$
$$\cup (\underline{B} \cap Cpre_{\overline{F}}(Y)) \Big] \,. \quad (16)$$

Note that the only new term in (16) as compared to (15) is the intersection of $\underline{B}$ with $Cpre_{\overline{F}}(Y)$. This additional term makes sure that each time $\underline{B}$ is reached, the winning region Y is not left in the next step to make sure that $\underline{B}$ can be reached once again.

The fixed point (16) and the associated abstract controller $\widehat{C}$ can be computed as the nested iteration given in Alg. 1. The controller $\widehat{C}$ is a partial function from $\widehat{S}$ to $\mathcal{U}$, and we use the notation dom $\widehat{C}$ to denote the domain of the controller $\widehat{C}$.

Algorithm 1 Computation of $\underline{W}$

Input: $\underline{B} \subseteq \widehat{S}$
Output: $\underline{W}, \widehat{C}$
1: $Y \leftarrow \widehat{S}, Y' \leftarrow \emptyset$
2: **while** $Y \neq Y'$ **do**
3:      $Y' \leftarrow Y$
4:      $Z \leftarrow \emptyset, Z' \leftarrow \widehat{S}$
5:      $\forall \widehat{x} \in \widehat{S} \,.\, \widehat{C} : \widehat{x} \mapsto \emptyset$
6:      **while** $Z \neq Z'$ **do**
7:          $Z' \leftarrow Z$
8:          $Z'' \leftarrow (\underline{B}^c \cap (Apre(Y,Z) \cup Cpre_{\overline{F}}(Z))) \cup (\underline{B} \cap Cpre_{\overline{F}}(Y))$
9:          $\forall \widehat{x} \in Z'' \setminus (\underline{B} \cup \text{dom } \widehat{C}) \,.\, \widehat{C} : \widehat{x} \mapsto u$ s.t. $\overline{F}(\widehat{x}, u) \subseteq Y \wedge \underline{F}(\widehat{x}, u) \cap Z \neq \emptyset$ or $\overline{F}(\widehat{x}, u) \subseteq Z$
10:          $Z \leftarrow Z''$
11:      **end while**
12:      $Y \leftarrow Z$
13: **end while**
14: $\underline{W} \leftarrow Y$
15: $\forall \widehat{x} \in \underline{W} \cap \underline{B} \,.\, \widehat{C} : \widehat{x} \mapsto u$ s.t. $\overline{F}(\widehat{x}, u) \subseteq \underline{W}$
16: **return** $\underline{W}, \widehat{C}$

Note that, the existence of the control input u in Lines 9 and 15 is guaranteed because of the definition of $Cpre$ and $Apre$.

PROPOSITION 5.2. *The set $\underline{W}$ is an under-approximation of the maximal a.s. winning region $\text{WinDom}(\mathfrak{S}, C^*)$.*

PROOF. The goal is to prove $\underline{W} \subseteq \text{WinDom}(\mathfrak{S}, C^*)$. Let $q \in \widehat{S}$ be a state s.t. $q \in \underline{W}$. We show that $q \subseteq \text{WinDom}(\mathfrak{S}, C^*)$. In the last iteration of the outer while loop in Alg. 1, we obtain a growing sequence of states $Z_0 \subset Z_1 \subset \ldots \subset Z_k = Y$, where $Z_0 = \emptyset$ and $Y = \underline{W}$. Since $Apre(Y, \emptyset) = Cpre_{\overline{F}}(\emptyset) = \emptyset$, hence $Z_1 \subseteq \underline{B}$. For all the other states $\widehat{x}$ in Z_i for $i \in (2; k]$, one of the two cases happen: Either (a) by $Cpre_{\overline{F}}(Z_{i-1})$, it is ensured that Z_{i-1} is surely reached

from $\widehat{x}$ in one step, or (b) by $Apre(Y, Z_{i-1})$, it is ensured that Y (same as $\underline{W}$ in the last iteration) is not left from $\widehat{x}$, and additionally (follows from Prop. 4.4) from all the (continuous) states inside $\widehat{x}$, transition to Z_{i-1} happens almost surely in the long run. Thus, for every $q \in \underline{W} \setminus \underline{B}$ and for all $x \in q$, $\underline{B}$ is reached almost surely in the long run.

Moreover, the operator $Cpre_{\overline{F}}(Y)$ ensures that Y—same as $\underline{W}$ in the last iteration—is not left in the one step from Z_1. Hence almost surely $\underline{B}$ is visited infinitely often. □

REMARK 5. *The operators Apre and the fix-point (16) are inspired by how a.s. winning strategies are synthesized for Büchi specification in two-player concurrent games [7]: There the optimal strategy for the protagonist player (who wants to satisfy the Büchi condition) is to play an action that surely keeps the game within the winning region, while making progress towards the target with positive probability.*

At a very high level, we use the same insight to express $\underline{W}$ in (16), though for us the underlying game structure is totally different (see Rem. 3). It turns out that winning the game almost surely in our case means to either stay in the winning region using all $\overline{F}$-successors, while at the same time making progress using some $\underline{F}$-successor.

5.3 Controller refinement

The state space discretization in the abstraction process induces a quantizer map $Q : \mathcal{S} \to \widehat{\mathcal{S}}'$ s.t. $Q : s \mapsto \widehat{x}$ when $s \in \widehat{x}$. Given the abstract controller $\widehat{C} : \widehat{\mathcal{S}} \to \mathcal{U}$, we can obtain a continuous controller $C : \mathcal{S} \to \mathcal{U}$ as $C \equiv \widehat{C} \circ Q$, where "∘" denotes function composition. The following theorem states that C is a sound controller.

THEOREM 5.3. *Consider the control policy $C \equiv \widehat{C} \circ Q$ and the set $\underline{W}$, where $\underline{W}$ and $\widehat{C}$ are the outputs of Alg. 1 and Q is the quantizer map. Then, $P_s^C(\mathfrak{S} \models \Box\Diamond B) = 1$ for all $s \in \underline{W}$.*

The proof of the above theorem directly follows from the proof of Prop. 5.2, and hence is omitted.

5.4 Over-approximation of the maximal a.s. winning region

The over-approximation of $\mathsf{WinDom}(\mathfrak{S}, C^*)$ is given by the following fixed-point

$$\overline{W} := \nu Y \,.\, \mu Z \,.\, \Big[(\overline{B}^c \cap Upre(Y, Z))$$
$$\cup \Big(\overline{B} \cap (Cpre_{\underline{F}}(Y) \cup Pre_{\overline{F} \setminus \underline{F}}(Y)) \Big) \Big]. \quad (17)$$

The expression (17) can be solved in the same way as Alg. 1 by replacing the update in Line 8 with the update in the r.h.s. of (17). Also, Line 9 and 15 are not needed, as the control policy in this case does not serve any useful purpose.

PROPOSITION 5.4. *The set $\overline{W}$ is a superset of the maximal a.s. winning region $\mathsf{WinDom}(\mathfrak{S}, C^*)$.*

PROOF. Let $x^* \in \mathsf{WinDom}(\mathfrak{S}, C^*)$. Since $\widehat{\mathcal{S}}$ creates a partition of the state space $\mathcal{S}$, hence there exists an abstract state $\widehat{x}^* \in \widehat{\mathcal{S}}$ s.t. $x^* \in \widehat{x}^*$. We need to show that $\widehat{x}^* \in \overline{W}$. For the sake of contradiction, assume that $\widehat{x}^* \in \overline{W}^c$. We will show that this cannot happen.

The fixed point computation (17) produces a shrinking sequence of states $\widehat{\mathcal{S}} = Y^0 \supseteq Y^1 \supseteq \ldots \supseteq Y^k = \overline{W}$. Let $i \in \mathbb{N}$ be the round index when $\widehat{x}^*$ was excluded from Y for the first time i.e., $\widehat{x}^* \in Y^{i-1}$ but $\widehat{x}^* \notin Y^i$. Consider the following two possible cases: (a) When $\widehat{x}^* \in \overline{B}$, then this means that for all $u \in \mathcal{U}$, $\underline{F}(\widehat{x}^*, u) \nsubseteq Y^{i-1}$ (*all states in $\widehat{x}^*$ leave Y^{i-1} with positive probability*) and $\left(\overline{F}(\widehat{x}^*, u) \setminus \underline{F}(\widehat{x}^*, u)\right) \cap Y^{i-1} = \emptyset$ (*no state in $\widehat{x}^*$ can stay in Y^{i-1} with positive probability*). (b) When $\widehat{x}^* \notin \overline{B}$, then this means that for all $u \in \mathcal{U}$, either $\underline{F}(\widehat{x}^*, u) \nsubseteq Y^{i-1}$ (*all states in $\widehat{x}^*$ leave Y^{i-1} with positive probability*), or from *all states $x \in \widehat{x}^*$ there does not exist any path to $\overline{B}$.* Both (a) and (b) mean that from *all* the continuous states $x \in \widehat{x}^*$, the specification will be violated with positive probability after i time steps. This is a contradiction to our assumption that $x^* \in \mathsf{WinDom}(\mathfrak{S}, C^*)$, since we know that from $x^* \in \widehat{x}^*$ the specification can be satisfied for infinite duration with probability 1. Hence, it must hold that $\widehat{x}^* \in \overline{W}$. □

5.5 Over-approximation of the minimal a.s. losing region for reachability

Once we have a tight approximation of the a.s. winning region, we can compute a lower-bound of the satisfaction probability for the quantitative version of the $\Box\Diamond B$ through Eqn. (5):

$$P_s^C(\mathfrak{S} \models \Box\Diamond B) \geq P_s^C(\mathfrak{S} \models \Diamond\mathsf{WinDom}(\mathfrak{S}, C^*)) \geq P_s^C(\mathfrak{S} \models \Diamond\underline{W})$$

for all $s \in \underline{W}^c$, where $\underline{W}^c$ is the complement of $\underline{W}$. Efficient computation of maximal reachability requires computation of minimal a.s. losing region, i.e., the set

$$L := \{s \in \mathcal{S} \mid \sup_{C \in \Pi_S} P_s^C(\mathfrak{S} \models \Diamond\underline{W}) = 0\}.$$

The following fixed point over-approximates L:

$$\overline{L} := \left[\mu X \,.\, Pre_{\underline{F}}(X) \cup \underline{W} \right]^c.$$

THEOREM 5.5. *$\overline{L}$ is an over-approximation of L.*

PROOF. We show the contra-positive, i.e. $\overline{L}^c \subseteq L^c$. Consider any abstract state $\widehat{x} \in \overline{L}^c$. By construction, from all the continuous states $x \in \widehat{x}$, $\underline{W}$ is reached with non-zero probability. Hence $\widehat{x} \notin L$. □

Once we have an over-approximation of L, the stochastic kernel of the CMP becomes contractive over $\mathcal{S} \setminus (\overline{L} \cup \underline{W})$ under mild continuity assumptions. Then approximate computational techniques in the literature on finite-horizon reachability can be utilised to find $P_s^C(\mathfrak{S} \models \Diamond\underline{W})$ with tunable error bounds [27, 28, 33].

REMARK 6. *If Assump. 1 does not hold, then $\mathsf{WinDom}(\mathfrak{S}, C^*)$ would be the "largest a.s. winning region achievable using stationary policies" contained in, but not necessarily equal to, the "true a.s. winning region." As a result, $\overline{W}$ would only be an over-approximation of $\mathsf{WinDom}(\mathfrak{S}, C^*)$, and not necessarily an over-approximation of the true a.s. winning region anymore. Nevertheless, the final controller obtained using Alg. 1 would still be a sound a.s. winning controller, but with a smaller domain than the optimal a.s. winning controller. Moreover, $\overline{L}$ would still be an over-approximation—but a more conservative one—of the "true minimal a.s. losing region."*

6 COMPUTATION OF THE ABSTRACTION

The dynamical system. We consider sampled-time continuous state dynamical systems with additive stochastic disturbance. The system is formalized using the tuple $\Sigma = (\mathcal{S}, \mathcal{U}, f, t_w)$, where $\mathcal{S} \subset \mathbb{R}^n$ is the state space, $\mathcal{U} \subset \mathbb{R}^m$ is the finite input space, $f : \mathcal{S} \times \mathcal{U} \to \mathcal{S}$ is the nominal state transition function and $t_w : \mathbb{R}^n \to \mathbb{R}_{\geq 0}$ is the density function of the stochastic disturbance. The state update of Σ is given as:

$$s(k+1) = f(s(k), u(k)) + w(k), \quad k \in \mathbb{N}, \tag{18}$$

where $s(k) \in \mathcal{S}$ and $u(k) \in \mathcal{U}$ are the state and input at the kth time instant, $w(k)$ is a random variable with the density function $t_w(\cdot)$, and $s(k+1)$ is the state at the $(k+1)^{\text{st}}$ time instant.

The random variables $\{w(k)\}_{k \in \mathbb{N}}$ are pairwise independent with the same density function $t_w(\cdot)$. We can write the system as a CMP $\mathfrak{S} = (\mathcal{S}, \mathcal{U}, T_s)$ with the stochastic kernel $T_s(A \mid s, u) = \int_A t_w(s' - f(s, u)) ds'$ for all $A \in \mathcal{B}(\mathcal{S})$. For the construction of the abstraction we assume that $t_w(\cdot)$ is piecewise continuous and $f(\cdot, u)$ is continuous for all $u \in \mathcal{U}$.

The abstraction. We assume that $\mathcal{S} = \mathcal{S}' \cup \{\phi\}$, where $\mathcal{S}'$ is a compact hyper-rectangular working region of the system and ϕ is a sink state representing the complement of $\mathcal{S}'$. The disturbance has a compact support $D \subset \mathbb{R}^n$. Let $\widehat{\mathcal{S}}'$ be a hyper-rectangular partition of $\mathcal{S}'$. The overall abstract state space is $\widehat{\mathcal{S}} = \widehat{\mathcal{S}}' \cup \{\phi\}$. Given an abstract state $\widehat{x} = [\![a, b]\!] \in \widehat{\mathcal{S}}'$ and a control input $u \in \mathcal{U}$, we denote the approximate *nominal* reachable set of $\mathfrak{S}$ by $\Phi(\widehat{x}, u)$ s.t.

$$\Phi(\widehat{x}, u) \supseteq \bigcup_{s \in \text{cl}(\widehat{x})} f(s, u), \tag{19}$$

where $\text{cl}(\widehat{x})$ is the closure of the set $\widehat{x}$. Note that $\Phi(\widehat{x}, u)$ can be computed using any reachability analysis method for deterministic dynamical systems [4, 5].

Define two functions $S_1, S_2 : \widehat{\mathcal{S}} \times \mathcal{U} \to 2^{\mathbb{R}^n}$ s.t.

$$S_1 : (\widehat{x}, u) \mapsto \overline{D} \oplus \Phi(\widehat{x}, u) \quad \text{and} \tag{20}$$

$$S_2 : (\widehat{x}, u) \mapsto \underline{D} \ominus (-\Phi(\widehat{x}, u)), \tag{21}$$

where $\overline{D} \supseteq D$ is any over-approximation of the support of disturbance D and $\underline{D} \subseteq D$ is any compact under-approximation of D over which $t_w(\cdot)$ is strictly positive. The operators $\oplus$ and $\ominus$ are Minkowski sum and Minkowski difference of two sets, respectively, and the minus sign in $(-\Phi(\widehat{x}, u))$ is applied to all elements.
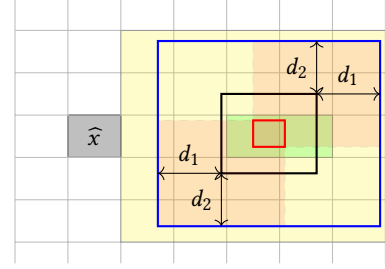
THEOREM 6.1. *Let $\Sigma = (\mathcal{S}, \mathcal{U}, f, t_w)$ be a dynamical system and $\mathfrak{S} = (\mathcal{S}, \mathcal{U}, T_s)$ be the CMP induced by Σ. Define $\mathcal{A} = (\widehat{\mathcal{S}}, \mathcal{U}, \overline{T}, \underline{T})$ s.t.:*

$$\overline{T}(\widehat{x}, u) := \{\widehat{x}' \in \widehat{\mathcal{S}} \mid (\widehat{x}' \neq \phi) \Rightarrow (\widehat{x}' \cap S_1(\widehat{x}, u) \neq \emptyset) \wedge$$
$$(\widehat{x}' = \phi) \Rightarrow (S_1(\widehat{x}, u) \not\subseteq \mathcal{S}')\}, \tag{22}$$

$$\underline{T}(\widehat{x}, u) := \{\widehat{x}' \in \widehat{\mathcal{S}} \mid (\widehat{x}' \neq \phi) \Rightarrow (\lambda(\widehat{x}' \cap S_2(\widehat{x}, u)) > 0) \wedge$$
$$(\widehat{x}' = \phi) \Rightarrow \lambda(S_2(\widehat{x}, u) \backslash \mathcal{S}') > 0\}, \tag{23}$$

where $\lambda(\cdot)$ gives the Lebesgue measure (volume) of a set. Then $\mathcal{A}$ is an abstraction of $\mathfrak{S}$.

PROOF. We show that $\overline{T}$ and $\underline{T}$ satisfy the properties of $\overline{F}$ and $\underline{F}$ as formalized in Def. 4.2. For (7), consider any pair of abstract states

Figure 3: Illustration of abstraction computation: given the abstract state $\widehat{x}$ (filled with grey) and some control input u, first the nominal reachable set is over-approximated (black rectangle). Next, the sets S_1 (blue rectangle) and S_2 (red rectangle) are computed. Finally, the images of the transition functions $\overline{F}$ (filled with yellow) and $\underline{F}$ (filled with green) are the abstract states intersecting with S_1 and S_2 respectively.

$\widehat{x}, \widehat{x}' \in \widehat{\mathcal{S}}'$ and input $u \in U$ and there exists $s \in \widehat{x}$, $T_s(\widehat{x}' \mid s, u) > 0$. We show that $\widehat{x}' \in \overline{T}(\widehat{x}, u)$:

$$\int_{\widehat{x}'} t_w(s' - f(s, u)) ds' > 0 \Rightarrow \int_{\widehat{x}' \ominus \{f(s, u)\}} t_w(w) dw > 0$$
$$\Rightarrow (\widehat{x}' \ominus \{f(s, u)\}) \cap D \neq \emptyset$$
$$\Rightarrow \exists w \in D, \exists s' \in \widehat{x}' \text{ s.t. } w = s' - f(s, u)$$
$$\Rightarrow \exists w \in D, \exists s' \in \widehat{x}' \text{ s.t. } s' = f(s, u) + w.$$

At the same time we know that $f(s, u) \in \Phi(\widehat{x}, u)$ since $s \in \widehat{x}$. Then,

$$s' \in S_1(\widehat{x}, u) \Rightarrow \widehat{x}' \cap S_1(\widehat{x}, u) \neq \emptyset \Rightarrow \widehat{x}' \in \overline{T}(\widehat{x}, u).$$

A similar reasoning holds for the case of $\widehat{x}' = \phi$.

Now we show that $\underline{T}$ satisfies the condition given in (8). Take $\widehat{x} \in \widehat{\mathcal{S}}'$, input $u \in U$, and $\widehat{x}' \in \underline{T}(\widehat{x}, u)$ s.t. $\widehat{x}' \neq \phi$. Then $\lambda(\widehat{x}' \cap S_2(\widehat{x}, u)) > 0$ according to (23). For any $s' \in \widehat{x}' \cap S_2(\widehat{x}, u)$, we have

$$s' \in S_2(\widehat{x}, u) \Rightarrow \{s'\} \oplus (-\Phi(\widehat{x}, u)) \subseteq \underline{D}$$
$$\Rightarrow s' - f(s, u) \subseteq \underline{D} \quad \forall s \in \text{cl}(\widehat{x})$$
$$\Rightarrow T_s(\widehat{x}' \mid s, u) \geq \int_{\widehat{x}' \cap S_2(\widehat{x}, u)} t_w(s' - f(s, u)) ds' > 0$$

The right-hand side is strictly positive since the integrand is strictly positive and the domain of integration has a positive measure. It is also assumed that f is continuous and t_w piecewise continuous. Therefore, we have a positive function over the compact domain $\text{cl}(\widehat{x})$, which will have a positive minimum:

$$\exists \varepsilon > 0 . \forall s \in \text{cl}(\widehat{x}) . T_s(\widehat{x}' \mid s, u) \geq \varepsilon \implies \widehat{x}' \in \underline{F}(\widehat{x}, u).$$

□

The abstraction procedure can be summarized as follows: first compute the approximate nominal reachable set $\Phi(\widehat{x}, u)$ in (19), then take the Minkowski sum and difference for S_1, S_2 in (20)-(21), and finally compute the transition relations (23)-(22). Fig. 3 illustrates the abstraction procedure for a 2-d system and when D is of the form $[-d_1, d_1] \times [-d_2, d_2]$.

6.1 Computation for mixed-monotone systems

If the function $f(\cdot, u)$ is mixed-monotone for every $u \in U$ and the partition sets are hyper-rectangles, then the nominal reachable set $\Phi(\widehat{x}, u)$ can be computed particularly efficiently. We recall the definition of mixed-monotonicity [4].

Definition 6.2. Let $g : \mathcal{S} \to \mathcal{S}$ be a function, and $\leq_{\mathcal{S}}$ be an order relation on $\mathcal{S}$ induced by positive cones. The function g is called mixed-monotone w.r.t. $\leq_{\mathcal{S}}$ (or simply mixed-monotone if $\leq_{\mathcal{S}}$ is obvious from the context) if there exists a function $h : \mathcal{S} \times \mathcal{S} \to \mathcal{S}-$ called the decomposition function—with the following properties:

(1) $\forall x \in \mathcal{S} . h(x, x) = g(x)$,
(2) $\forall x_1, x_2, y \in \mathcal{S} . (x_1 \leq_{\mathcal{S}} x_2) \Rightarrow (h(x_1, y) \leq_{\mathcal{S}} h(x_2, y))$, and
(3) $\forall x, y_1, y_2 \in \mathcal{S} . (y_1 \leq_{\mathcal{S}} y_2) \Rightarrow (h(x, y_2) \leq_{\mathcal{S}} h(x, y_1))$.

Intuitively, a mixed-monotone function can be decomposed into an increasing and a decreasing component. This phenomenon can be seen from the definition of the decomposition function. The following proposition shows a fast over-approximation method of the image of a rectangular set under a mixed-monotone function.

PROPOSITION 6.3 ([4, THM. 1]). *Let g be a mixed-monotone function with the decomposition function h, and $[\![a, b]\!] \subseteq \mathcal{S}$ be any hyper-rectangle. The image of $[\![a, b]\!]$ under g can be over-approximated as $[\![h(a, b), h(b, a)]\!]$.*

For mixed-monotone $f(\cdot, u)$ with decomposition function h_u, the function $\Phi(\widehat{x}, u)$ can be computed using Prop. 6.3 as $\Phi(\widehat{x}, u) = [\![h_u(a, b), h_u(b, a)]\!]$ for $\widehat{x} = [\![a, b]\!]$.

7 EXAMPLES

We implemented the presented algorithms in an open-source tool called Mascot-SDS (https://gitlab.mpi-sws.org/kmallik/mascot-sds.git), which is an extension of the tool Mascot [12], and is built on top of the basic symbolic computation framework of the tool SCOTS [25]. All the experiments were performed on a computer with a 3GHz Intel Xeon E7-8857 v2 processor and 1.5 TB memory.

7.1 Perturbed Van der Pol Oscillator

Our first example considers the computation of the maximal a.s. winning region of an autonomous stochastically perturbed Van der Pol oscillator [21]. The state evolution of the oscillator is given by:

$$x_1(k + 1) = x_1(k) + x_2(k)\tau + w_1(k)$$
$$x_2(k + 1) = x_2(k) + (-x_1(k) + (1 - x_1(k)^2)x_2(k))\tau + w_2(k),$$

where the sampling time τ is set to $0.1s$ and $(w_1(k), w_2(k))$ is a pair of stochastic noise signals at time k drawn from a piece-wise continuous density function with a compact support $D = [-0.02, 0.02] \times [-0.02, 0.02]$. Note that for the computation of the winning set, we do not need the actual density function as discussed in the previous section. We consider a safety specification $\square \mathcal{S}'$ for staying within the working area $\mathcal{S}' = [-0.5, 0.5] \times [-0.5, 0.5]$, as well as a Büchi specification $\square \Diamond B$ for repeatedly reaching the target set $B = [-1.2, -0.9] \times [-2.9, -2]$ (green rectangle in Fig. 4a). Our algorithm is able to compute the under and over-approximation of the set of a.s. winning region. In Fig. 4a, the under-approximation $\underline{W}$ is shown in grey and $\overline{W} \setminus \underline{W}$ is shown in blue.

It turns out that when the noise is treated as worst case, then there exists a deterministic value of the noise for which the oscillator trajectory *never* reaches the target from all the initial states inside the domain, thus violating the specification. So the winning region is empty. A trajectory with a fixed deterministic perturbation that misses the target all the time is shown in black in Fig. 4a.

On the other hand, when the noise is treated as stochastic, then there are initial states from where the perturbed trajectory always eventually reaches the target polytope. Hence, the specification is satisfied. A trajectory with stochastic perturbation and the initial state I is shown in red in Fig. 4a. Table 1 summarizes the abstraction parameters used in our experiment, ratio of the computed volume of $\underline{W}$ to the computed volume of $\overline{W}$, and the computation time.

7.2 Controlled perturbed vehicle

Our second example is a controller synthesis problem for a perturbed sampled-time 3-d Dubins' vehicle [23]. We consider almost sure satisfaction of the Büchi specification while avoiding obstacles in the state space. Although we did not discuss avoidance of obstacle in the theory part, this can be easily handled by redefining the working region $\mathcal{S}'$ of the system by excluding the obstacles. Thus given a hyper-rectangular working region $[0, 2] \times [0, 3] \times [-\pi, \pi]$, and an obstacle $[0.8, 1.2] \times [1, 1.4] \times [-\pi, \pi]$ within that working region, we define $\mathcal{S}' = [0, 2] \times [0, 3] \times [-\pi, \pi] \setminus [0.8, 1.2] \times [1, 1.4] \times [-\pi, \pi]$. The system dynamics is given as: when $u \neq 0$,

$$x_1(k + 1) = x_1(k) + \frac{V}{u} \sin(x_3(k) + u\tau) - \frac{V}{u} \sin(x_3(k)) + w_1(k)$$
$$x_2(k + 1) = x_2(k) - \frac{V}{u} \cos(x_3(k) + u\tau) + \frac{V}{u} \cos(x_3(k)) + w_2(k)$$
$$x_3(k + 1) = x_3(k) + u\tau + w_3(k),$$

and when $u = 0$,

$$x_1(k + 1) = x_1(k) + V \cos(x_3(k))\tau + w_1(k)$$
$$x_2(k + 1) = x_2(k) - V \sin(x_3(k))\tau + w_2(k)$$
$$x_3(k + 1) = x_3(k) + w_3(k),$$

where the sampling time $\tau = 1s$, the constant forward velocity $V = 0.1$ (maintained by e.g. a low level cruise control system), and $(w_1(k), w_2(k), w_3(k))$ is a collection of stochastic noise samples drawn from a piecewise continuous density function with the support $D = [-0.06, 0.06] \times [-0.06, 0.06] \times [-0.06, 0.06]$. It is due to this fixed velocity that the vehicle cannot stay stationary (or near stationary) after reaching the target, which makes the synthesis problem with Büchi specification much more challenging than the same with normal reachability specification.

When the noise is treated as a worst case adversary, the winning region is empty. However, when the noise is treated as stochastic, the approximate winning regions $\underline{W}$ and $\overline{W}$ are non-empty as shown in Fig. 4b. Fig. 4c shows the simulated trajectory of the vehicle using the synthesized controller. It was observed that even though the trajectory moves away from the target from time to time, either due to the external noise or due to the constant velocity, it always returns to the target eventually.

We performed the computation for four different levels of discretization granularity, and the results are summarized in Table 1. It can be observed (from the ratio $\lambda(\underline{W})/\lambda(\overline{W})$) that the gap between

$\overline{W}$ and $\underline{W}$ monotonically shrinks as the abstract states get smaller, which means that the approximations $\overline{W}$ and $\underline{W}$ get progressively better with refinement of the state space partition. However, computation time sharply increases with finer discretization.

7.3 A note on computation time

The computation time for $\underline{W}$ reported in Table 1 is based on a warm-start of Alg. 1 by replacing $Y \leftarrow \widehat{S}$ in Line 1 with $Y \leftarrow \overline{W}$. The intuition is that since it is known upfront that $\underline{W} \subseteq \overline{W}$, hence we do not need to consider the set $\widehat{S} \setminus \overline{W}$ in Alg. 1. In practice, the computation time for $\underline{W}$ would be higher than the numbers reported in Table 1 had we started with $\widehat{S}$.

In general, we observed that the computation of $\underline{W}$ takes much longer than the computation of $\overline{W}$. Our hypothesis is that this is due to the properties of the operators defined in (11)-(14), and how they are used in the computation of $\underline{W}$ and $\overline{W}$. For example, because $\overline{F}(\widehat{x}, u)$ is a superset of $\underline{F}(\widehat{x}, u)$ for all $\widehat{x}, u$, it can be shown that for a given $Y, Z \subseteq \widehat{S}$, $Apre(Y, Z) \subseteq Upre(Y, Z)$. Similarly $Cpre_{\overline{F}}(Z) \subseteq Upre(Y, Z)$ when $Y \supseteq Z$. Moreover, $\underline{B} \cap Cpre_{\overline{F}}(Y) \subseteq \overline{B} \cap (Cpre_{\underline{F}}(Y) \cup Pre_{\overline{F} \setminus \underline{F}}(Y))$. Thus each iteration in the inner "μ" fixed point would add possibly fewer states in case of $\underline{W}$ than in case of $\overline{W}$. Since ultimately the size of $\overline{W}$ and $\underline{W}$ are not very far apart, as shown in Col. 4 of Table 1, hence the iterations for $\underline{W}$ would take many more number of steps than $\overline{W}$.

8 FUTURE WORK

We are working on three different extensions of this work. First, we plan to develop computation techniques for the qualitative winning regions for more general Rabin or parity conditions. Second, we are working on formulating conditions to guarantee convergence of the computations to the actual winning region when the discretization gets finer. Finally, we plan to improve the scalability of the approach using multi-resolution abstractions.

ACKNOWLEDGMENTS

REFERENCES

[1] C. Baier and J.-P. Katoen. *Principles of Model Checking*. MIT Press, 2008.
[2] D. Bertsekas and S. Shreve. *Stochastic Optimal Control: The Discrete-Time Case*. Athena Scientific, 1996.
[3] A. Bianco and L. de Alfaro. Model checking of probabilistic and nondeterministic systems. In P. S. Thiagarajan, editor, *Foundations of Software Technology and Theoretical Computer Science*, pages 499–513, Berlin, Heidelberg, 1995. Springer Berlin Heidelberg.
[4] S. Coogan and M. Arcak. Efficient finite abstraction of mixed monotone systems. In *Proceedings of the 18th International Conference on Hybrid Systems: Computation and Control*, pages 58–67. ACM, 2015.
[5] T. Dang and R. Testylier. Reachability analysis for polynomial dynamical systems using the bernstein expansion. *Reliable Computing*, 17(2):128–152, 2012.
[6] L. de Alfaro. *Formal Verification of Probabilistic Systems*. PhD thesis, Department of Computer Science, Stanford University, 1998.
[7] L. de Alfaro and T. A. Henzinger. Concurrent omega-regular games. In *Proceedings Fifteenth Annual IEEE Symposium on Logic in Computer Science (Cat. No. 99CB36332)*, pages 141–154. IEEE, 2000.
[8] M. Dutreix and S. Coogan. Specification-guided verification and abstraction refinement of mixed-monotone stochastic systems, 2019.
[9] S. Haesaert and S. Soudjani. Robust dynamic programming for temporal logic control of stochastic systems. *CoRR*, abs/1811.11445, 2018.
[10] S. Hart, M. Sharir, and A. Pnueli. Termination of probabilistic concurrent program. *ACM Trans. Program. Lang. Syst.*, 5(3):356–380, July 1983.
[11] O. Hernández-Lerma and J. B. Lasserre. *Discrete-time Markov control processes*, volume 30 of *Applications of Mathematics*. Springer, 1996.
[12] K. Hsu, R. Majumdar, K. Mallik, and A.-K. Schmuck. Multi-layered abstraction-based controller synthesis for continuous-time systems. In *Proceedings of the 21st International Conference on Hybrid Systems: Computation and Control (part of CPS Week)*, pages 120–129, 2018.
[13] P. Jagtap, S. Soudjani, and M. Zamani. Formal synthesis of stochastic systems via control barrier certificates. *arXiv: eess.SY*, abs/1905.04585, 2019.
[14] N. Kariotoglou, M. Kamgarpour, T. H. Summers, and J. Lygeros. The linear programming approach to reach-avoid problems for Markov decision processes. *J. Artif. Int. Res.*, 60(1):263–285, Sept. 2017.
[15] S. Kiefer, R. Mayr, M. Shirmohammadi, and P. Totzke. Büchi objectives in countable MDPs. *arXiv preprint arXiv:1904.11573*, 2019.
[16] S. Kiefer, R. Mayr, M. Shirmohammadi, and D. Wojtczakz. Parity objectives in countable mdps. In *2017 32nd Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, pages 1–11. IEEE, 2017.
[17] D. Kozen. Results on the propositional μ-calculus. *Theoretical Computer Science*, 27(3):333 – 354, 1983. Special Issue Ninth International Colloquium on Automata, Languages and Programming (ICALP) Aarhus, Summer 1982.
[18] M. Lahijanian, S. B. Andersson, and C. Belta. Formal verification and synthesis for discrete-time stochastic systems. *IEEE Transactions on Automatic Control*, 60(8):2031–2045, Aug 2015.
[19] A. Lavaei, S. Soudjani, and M. Zamani. Compositional construction of infinite abstractions for networks of stochastic control systems. *Automatica*, 107:125 – 137, 2019.
[20] K. Lesser and M. Oishi. Approximate safety verification and control of partially observable stochastic hybrid systems. *IEEE Transactions on Automatic Control*, 62(1):81–96, Jan 2017.
[21] H. Leung. Stochastic transient of a noisy van der pol oscillator. *Physica A: Statistical Mechanics and its Applications*, 221(1):340 – 347, 1995.
[22] P. Nilsson, N. Ozay, and J. Liu. Augmented finite transition systems as abstractions for control synthesis. *Discrete Event Dynamic Systems*, 27(2):301–340, 2017.
[23] R. Pepy, A. Lambert, and H. Mounier. Path planning using a dynamic vehicle model. In *2006 2nd International Conference on Information Communication Technologies*, volume 1, pages 781–786, April 2006.
[24] G. Reissig, A. Weber, and M. Rungger. Feedback refinement relations for the synthesis of symbolic controllers. *TAC*, 62(4):1781–1796, 2017.
[25] M. Rungger and M. Zamani. SCOTS: A tool for the synthesis of symbolic controllers. In *Proceedings of the 19th International Conference on Hybrid Systems: Computation and Control*, HSCC '16, pages 99–104, New York, NY, USA, 2016. ACM.
[26] S. Soudjani. *Formal Abstractions for Automated Verification and Synthesis of Stochastic Systems*. PhD thesis, Technische Universiteit Delft, The Netherlands, 2014.
[27] S. Soudjani and A. Abate. Higher-Order Approximations for Verification of Stochastic Hybrid Systems. In S. Chakraborty and M. Mukund, editors, *Automated Technology for Verification and Analysis*, volume 7561 of *Lecture Notes in Computer Science*, pages 416–434. Springer Verlag, Berlin Heidelberg, 2012.
[28] S. Soudjani and A. Abate. Adaptive and sequential gridding procedures for the abstraction and verification of stochastic processes. *SIAM Journal on Applied Dynamical Systems*, 12(2):921–956, 2013.
[29] S. Soudjani, A. Abate, and R. Majumdar. Dynamic Bayesian networks for formal verification of structured stochastic processes. *Acta Informatica*, 54(2):217–242, Mar 2017.
[30] P. Tabuada. *Verification and Control of Hybrid Systems: A Symbolic Approach*. Springer, 2009.
[31] I. Tkachev and A. Abate. Regularization of Bellman equations for infinite-horizon probabilistic properties. In *Proceedings of the 15th ACM international conference on Hybrid Systems: computation and control*, pages 227–236, Beijing, PRC, April 2012.
[32] I. Tkachev and A. Abate. Characterization and computation of infinite-horizon specifications over markov processes. *Theoretical Computer Science*, 515:1–18, 2014.
[33] I. Tkachev, A. Mereacre, J.-P. Katoen, and A. Abate. Quantitative model-checking of controlled discrete-time Markov processes. *Information and Computation*, 253:1 – 35, 2017.
[34] A. P. Vinod and M. M. K. Oishi. Scalable underapproximative verification of stochastic LTI systems using convexity and compactness. In *Proceedings of the 21st International Conference on Hybrid Systems: Computation and Control (Part*

(a)   (b)   (c)
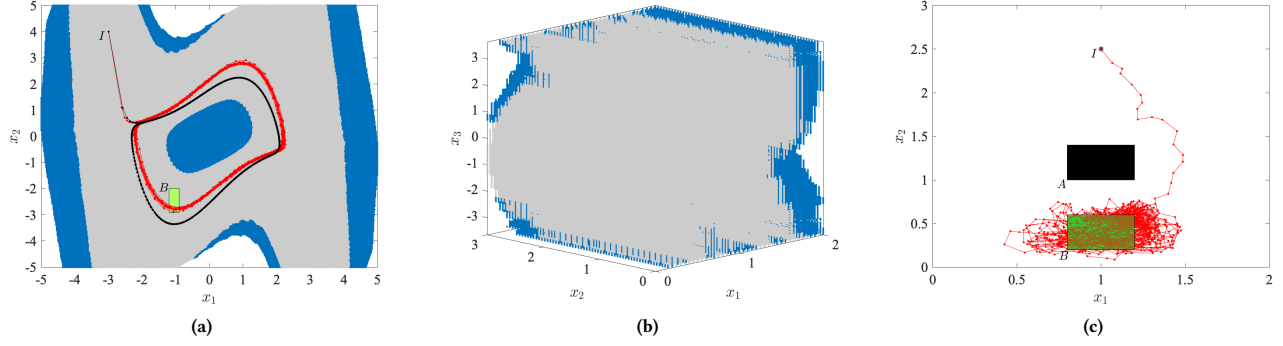
Figure 4: (a) The Van der Pol oscillator example: B **(green box) is the target,** $\underline{W}$ **is in grey,** $\overline{W} \setminus \underline{W}$ **is in blue, and** I **is the initial state for simulation. The trajectory with stochastic perturbation is shown in red, and the trajectory with a fixed deterministic perturbation that always misses the target is shown in black. (b) The approximate a.s. winning domain for the Dubin's vehicle (for abstract state size** $0.1 \times 0.1 \times 0.1$**):** $\underline{W}$ **is in grey and** $\overline{W} \setminus \underline{W}$ **is in blue. (c) Simulation of perturbed trajectory for the Dubin's vehicle projected onto** x_1, x_2 **plane. The black box** A **is the obstacle and the green box** B **is the target.**

System	Size of abstract states	Volume of the worst-case winning region	$\lambda(\underline{W})/\lambda(\overline{W})$	Computation time		
				Abstraction	$\overline{W}$	$\underline{W}$
Van der pol oscillator	0.02×0.02	empty	73.0%	$< 1m$	$12m$	$416m$
Dubins' vehicle	$0.1 \times 0.1 \times 0.1$	empty	80.0%	$1m$	$2m$	$11m$
	$0.07 \times 0.07 \times 0.07$	empty	83.8%	$4m$	$23m$	$35m$
	$0.05 \times 0.05 \times 0.05$	empty	87.2%	$18m$	$124m$	$140m$
	$0.04 \times 0.04 \times 0.04$	empty	88.9%	$37m$	$132m$	$128m$

Table 1: Performance evaluation of our method on the Van der Pol oscillator and the Dubin's vehicle. The 2nd column shows the size of each hyper-rectangular abstract state in the underlying uniform grid, the 3rd column shows the volume of the approximate winning domain when the noise is treated in the usual worst-case sense, the 4th column shows the ratio of the Lebesgue measure (volume) of $\underline{W}$ **to** $\overline{W}$**, and the 5th, 6th, and 7th columns show the computation times of different phases of our algorithm in minutes. Note that in our implementation, the computation of** $\underline{W}$ **was warm started with already computed** $\overline{W}$**. Had** $\underline{W}$ **been computed from scratch, the computation time for** $\underline{W}$ **would be higher than what is shown in the last column.**

of CPS Week), HSCC '18, pages 1–10, New York, NY, USA, 2018. ACM.

9 APPENDIX

9.1 Proof of Prop. 3.2

Following the steps utilized in [33, Theorem 7], we have that $1 - f^*(s)$ is lower semi-analytic. Then $\{s \in \mathcal{S} \mid 1 - f^*(s) < c\}$ is an analytic subset of $\mathcal{S}$ for all $c \in \mathbb{R}$. Take a positive sequence $\{c_n \to 0\}$. The set $\cap_n \{s \in \mathcal{S} \mid f^*(s) > 1 - c_n\} = \{s \in \mathcal{S} \mid f^*(s) = 1\}$ is also analytic. Every analytic set is universally measurable.

9.2 Proof of Thm. 3.3

PROOF OF THM. 3.3. We already know that $P_s^C(\mathfrak{S} \models \Box \Diamond B) = 1$ for all $s \in \text{WinDom}(\mathfrak{S}, C)$ by definition of the winning set. Take any $s \notin W := \text{WinDom}(\mathfrak{S}, C)$. We make the event conditional on τ

which is the first time the path hits a state in W. Then we have

$$P_s^C(\mathfrak{S} \models \Box \Diamond B) = \mathbb{E}_s^C \left[P_s^C(\mathfrak{S} \models \Box \Diamond B \mid s_1, s_2, \ldots, s_n, \tau = n) \right]$$

$$= \sum_{n=0}^{\infty} P_s^C(s_1, s_2, \ldots, s_{n-1} \in \mathcal{S} \setminus W, s_n \in W)$$

$$+ P_s^C(\mathfrak{S} \models \Box \Diamond B \text{ and } \mathfrak{S} \models \Box \mathcal{S} \setminus W).$$

The sum is the reachability probability and the last term is always non-negative. □

9.3 Proof of Prop. 4.5

PROOF. We take an element $\widehat{x}' \in \widehat{\mathcal{S}}$ and $u \in \mathcal{U}$. We show that $\widehat{x}'$ belongs to the right-hand side of (10) if and only if it belongs to the right-hand side of (8). Observe that

$$T_{\mathfrak{s}}(\widehat{x}' \mid s, u) = T_{\mathfrak{s}}(\text{cl}(\widehat{x}') \mid s, u)$$

$$= \int_{\text{cl}(\widehat{x}')} T_{\mathfrak{s}}(ds' \mid s, u) = \int_{\mathcal{S}} \mathbf{1}_{\text{cl}(\widehat{x}')}(s') T_{\mathfrak{s}}(ds' \mid s, u).$$

Since the indicator function of a closed set is upper semi-continuous, $T_s(\widehat{x}' \mid s, u)$ is also upper semi-continuous for all $s \in \mathrm{cl}(\widehat{x})$. Similarly,

$$T_s(\widehat{x}' \mid s, u) = T_s(\mathrm{int}(\widehat{x}') \mid s, u)$$

$$= \int_{\mathrm{int}(\widehat{x}')} T_s(ds' \mid s, u) = \int_{\mathcal{S}} \mathbf{1}_{\mathrm{int}(\widehat{x}')}(s') T_s(ds' \mid s, u).$$

Since the indicator function of an open set is lower semi-continuous, $T_s(\widehat{x}' \mid s, u)$ is also lower semi-continuous for all $s \in \mathrm{cl}(\widehat{x})$. Therefore, $T_s(\widehat{x}' \mid s, u)$ is continuous over the domain $s \in \mathrm{cl}(\widehat{x})$, which means it attains its minimum over $s \in \mathrm{cl}(\widehat{x})$. Suppose $\widehat{x}'$ belongs to the right-hand side of (10) and take $\varepsilon = \min_{s \in \mathrm{cl}(\widehat{x})} T_s(\widehat{x}' \mid s, u)$. Since $T_s(\widehat{x}' \mid s, u)$ is positive, $\varepsilon > 0$ and $\widehat{x}'$ belongs to the right-hand side of (8). Now suppose $\widehat{x}'$ belongs to the right-hand side of (8). Then there is an $\varepsilon > 0$ such that $T_s(\widehat{x}' \mid s, u) \geq \varepsilon$ for all $s \in \widehat{x}$. Since $T_s(\widehat{x}' \mid s, u)$ is continuous over $\mathrm{cl}(\widehat{x})$, we get that $T_s(\widehat{x}' \mid s, u) > 0$ for all $s \in \mathrm{cl}(\widehat{x})$, which means $\widehat{x}'$ belongs to the right-hand side of (10). □

9.4 Properties of the winning region

Proposition 9.1. *For any control policy C, The set $W :=$ $\mathrm{WinDom}(\mathfrak{S}, C)$ is an absorbing set, i.e., the paths starting from this set will stay in the set a.s.:*

$$P_s^C(w[k+1] \in W \mid w[k]) = 1,$$

for all $w[k] \in W$.

Proof. For any $s \in W$, we have

$$P_s^C(\mathfrak{S} \models \square \Diamond B) = \int_{\mathcal{S}} P_{s_1}^C(\mathfrak{S} \models \square \Diamond B) T_s(ds_1 \mid s, C(s))$$

$$= \int_W T_s(ds_1 \mid s, C(s)) + \int_{\mathcal{S} \backslash W} P_{s_1}^C(\mathfrak{S} \models \square \Diamond B) T_s(ds_1 \mid s, C(s)).$$

This means

$$\int_{\mathcal{S} \backslash W} (1 - P_{s_1}^C(\mathfrak{S} \models \square \Diamond B)) T_s(ds_1 \mid s, C(s)) = 0,$$

$$P_s^C \left[(1 - P_{s_1}^C(\mathfrak{S} \models \square \Diamond B)) \mathbf{1}_{\mathcal{S} \backslash W}(s_1) \geq \epsilon \right] \leq \frac{0}{\epsilon} = 0,$$

where the last inequality is a consequence of Markov's inequality for non-negative random variables. By taking the union over a monotone positive sequence $\{\epsilon_n \to 0\}$, we get

$$P_s^C \left[(1 - P_{s_1}^C(\mathfrak{S} \models \square \Diamond B)) \mathbf{1}_{\mathcal{S} \backslash W}(s_1) > 0 \right] = 0,$$

$$P_s^C \left[s_1 \in \mathcal{S} \backslash W \text{ and } P_{s_1}^C(\mathfrak{S} \models \square \Diamond B) < 1 \right] = 0,$$

$$P_s^C \left[s_1 \in \mathcal{S} \backslash W \right] = 0.$$

□

Proposition 9.2. *Given a countable sequence of stationary policies $\{C_1, C_2, \ldots\}$ for the system $\mathfrak{S}$ with winning regions $\{\mathrm{WinDom}(\mathfrak{S}, C_n), n = 1, 2, \ldots\}$, there is a controller C with winning region $\mathrm{WinDom}(\mathfrak{S}, C) = \cup_{n=1}^{\infty} \mathrm{WinDom}(\mathfrak{S}, C_n)$.*

Proof. Define the sets $\{\overline{W}_n, n = 1, 2, \ldots\}$ inductively as $\overline{W}_1 :=$ $\mathrm{WinDom}(\mathfrak{S}, C_1)$ and $\overline{W}_n := \mathrm{WinDom}(\mathfrak{S}, C_n) \backslash \cup_{i=1}^{n-1} \overline{W}_n$ for all $n \in$
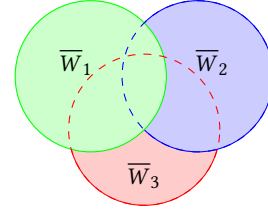
Figure 5: Illustration of construction of $\overline{W}_1$ (green filled part), $\overline{W}_2$ (blue filled part), and $\overline{W}_3$ (red filled part) from $\mathrm{WinDom}(\mathfrak{S}, C_1)$ (green circle), $\mathrm{WinDom}(\mathfrak{S}, C_2)$ (blue circle), and $\mathrm{WinDom}(\mathfrak{S}, C_3)$ (red circle).

$\{2, 3, \ldots\}$. This construction is illustrated in Fig. 5. Also, define the new stationary policy:

$$C(s) := \begin{cases} C_1(s) & \text{if } s \in \overline{W}_1 \\ C_2(s) & \text{if } s \in \overline{W}_2 \\ \vdots \end{cases} \tag{24}$$

for all non-empty sets $\overline{W}_n$. It is easy to show that the sets $\{\overline{W}_i\}$ are non-intersecting and $\cup_{i=1}^n \overline{W}_i = \cup_{i=1}^n \mathrm{WinDom}(\mathfrak{S}, C_i)$. Then for any initial state $s \in \cup_{i=1}^{\infty} \mathrm{WinDom}(\mathfrak{S}, C_i)$, there is some n such that $s \in \overline{W}_n$. Note that all sets $\mathrm{WinDom}(\mathfrak{S}, C_i)$ are absorbing under their respective policy. The path starting from s with C_n either stay in $\mathrm{WinDom}(\mathfrak{S}, C_n)$ or will reach some $\overline{W}_i$ with $i < n$. In the first case, the path satisfies the specification with probability one. The same argument can be applied a finite number of times until reaching the lowest index $i = 1$.

measurability of C. Note that the sets $\{\overline{W}_1, \overline{W}_2, \ldots\}$ are universally measurable and the policies $\{C_1, C_2, \ldots\}$ are universally measurable functions. We also have $C^{-1}(A) = \cup_{i=1}^{\infty} [C_i^{-1}(A) \cap \overline{W}_i]$, which means $C^{-1}(A)$ is universally measurable for any universally measurable A. Therefore, C is a universally measurable function. □