Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnline CONFERENCE PAPER ARCHIVE

IFAC PapersOnLine 54-5 (2021) 127–132

# Symbolic Qualitative Control for Stochastic Systems via Finite Parity Games ⋆

**Rupak Majumdar** * **Kaushik Mallik** *
**Anne-Kathrin Schmuck** * **Sadegh Soudjani** **

* *Max Planck Institute for Software Systems (MPI-SWS), Kaiserslautern, Germany*
** *School of Computing, Newcastle University, United Kingdom*

**Abstract:** We consider the controller synthesis problem for stochastic, continuous-state, nonlinear systems against $\omega$-regular specifications. We synthesize a symbolic controller that ensures *almost sure* (qualitative) satisfaction of the specification. The problem reduces, after some automata-theoretic constructions, to computing the *almost sure winning region*—the largest set of states from which a parity condition can be satisfied with probability 1 (on a possibly hybrid state space). While characterizing the exact almost sure winning region is still open for the considered system class, we propose an algorithm for obtaining a tight under-approximation of this set. The heart of our approach is a technique to *symbolically* compute this under-approximation via a *finite-state abstraction* as a $2\frac{1}{2}$*-player parity game*. Our abstraction procedure uses only the support of the probabilistic evolution; it does not use precise numerical transition probabilities. We have implemented our approach and evaluated it on the nonlinear model of the perturbed Dubins vehicle.



*Keywords:* Abstraction-based control design, Approximate model checking, Discrete-time stochastic systems, Finite games, Formal specifications, Policy synthesis

## 1. INTRODUCTION

Controlled Markov processes (CMPs) over continuous state spaces and evolving in discrete time form a general model for temporal decision making under stochastic uncertainty. In recent years, the problem of finding or approximating optimal policies in CMPs for specifications given in temporal logics or automata has received great attention. While there is a steady progression towards more expressive models and properties (Tkachev et al., 2017; Haesaert and Soudjani, 2018; Svoreňová et al., 2017; Majumdar et al., 2020; Dutreix et al., 2020; Laurenti et al., 2020), a satisfactory *symbolic* solution for *nonlinear* models and general *$\omega$-regular specifications* is still open. In this paper, we address a qualitative aspect of this problem (i.e., satisfying the specification with probability 1).

We are interested to compute the largest set of states, called the *almost sure winning region*, from which the given specification can be satisfied *almost surely*, i.e., with probability 1. For *finite-state* Markov decision processes (MDP), one can compute the *almost sure winning region* using graph theoretic techniques, ignoring the actual transition probabilities. Further, for any state in this almost sure winning region, an optimal policy for almost sure satisfaction of the specification can be derived (Baier and Katoen, 2008). For *continuous-state* CMPs, as considered in this paper, computation of the *exact* almost sure winning region is difficult. This is because a characterization of the optimal control policies for CMPs subject to general $\omega$-regular specifications is not available. In particular, it is not known if deterministic or finite-memory policies are sufficient for this challenging control problem.

The main contribution of this paper is a new technique to *under-approximate* the almost sure winning region of a given CMP for a *parity* specification. A parity specification is a canonical representation for all $\omega$-regular properties (Thomas, 1995); thus, our approach provides a way to under-approximate the almost sure winning region for any $\omega$-regular specification.

Our approach uses a finite-state *abstraction* of the given CMP which is qualitative in nature, i.e., the constructed abstraction disregards the exact transition probabilities of the CMP. As a result, we obtain a purely symbolic algorithm for the computation of the almost sure winning region of the CMP subject to a given specification. This *abstraction-based* policy synthesis technique is inspired by the abstraction-based controller design (ABCD) paradigm, that has been proposed for non-stochastic systems (Pola et al., 2008; Nilsson et al., 2017; Reissig et al., 2017). By following the ABCD paradigm, we first build an abstraction of the original CMP using a $2\frac{1}{2}$-player game (Chatterjee and Henzinger, 2012). The abstract $2\frac{1}{2}$-player game models the interplay between the controller (*Player* 0), the precision loss incurred due to the abstraction process

⋆ R. Majumdar and K. Mallik are funded through the DFG (Deutsche Forschungsgemeinschaft or German Science Foundation) project 389792660 TRR 248–CPEC, A.-K. Schmuck is funded through the DFG project SCHM 3541/1-1, and S. Soudjani is funded through the EPSRC-funded CodeCPS project (EP/V043676/1).


10.1016/j.ifacol.2021.08.486

(*Player* 1), and the environmental randomness (the 1/2-player). A solution of this abstract game generates a control strategy, which can then be refined as a controller for the original CMP.

The key insight in our $2\frac{1}{2}$-player game abstraction is that the disturbances need to be handled in a *fair* way—in the long run, all transitions with positive probability will eventually occur. In contrast, for the ABCD with nonstochastic systems, the abstractions need to treat the disturbances in a *worst-case* fashion, making the controller synthesis problem much more difficult.

This provides a conceptually very appealing result of our paper. Using $2\frac{1}{2}$-player games as abstractions of CMPs allows to utilize the symbolic game solving machinery, analogous to ABCD techniques for non-stochastic systems, while capturing the intuitive differences between the problem instances by the use of a random player in the abstract game. Most interestingly, the stochastic nature of the resulting abstract game *eases* the abstract synthesis problem compared to standard ABCD where disturbances are non-stochastic. In conclusion, we obtain a *symbolic algorithm* to compute an under-approximation of the almost sure winning region in a *continuous-state* CMP for all $\omega$-regular specifications. Moreover, similar to the results for finite-state MDPs, this shows that the (approximate) solution of almost sure winning region for CMPs does not need to handle the actual transition probabilities.

**Related Work.** Stochastic nonlinear systems were abstracted to finite-state interval Markov decision processes by Dutreix et al. (2020), where they provide an alternative approach for approximating the almost sure winning region for CMPs by using algorithms for model checking finite interval Markov chains against deterministic Rabin automata. Our method is conceptually very different from the one by Dutreix et al. (2020), where they explicitly compute lower and upper bounds of all involved probabilities and construct winning regions by an enumerative algorithm taking these probability bounds into account. On the other hand, our approach shows that this knowledge is not needed for the almost sure winning case. This allows us to provide a conceptually simpler *symbolic algorithm* approximately solving the qualitative aspect of the synthesis problem via abstract $2\frac{1}{2}$-player games.

$2\frac{1}{2}$-player games were used as abstractions of probabilistic systems, both in the finite case (Kwiatkowska et al., 2020) and for stochastic *linear* systems with *GR1 specifications* (Svoreňová et al., 2017). Our paper subsumes the result of Svoreňová et al. (2017) by showing a computational procedure to abstract a general, nonlinear CMP with a *parity specification* into a finite-state $2\frac{1}{2}$-player game. We also extend the work by Weininger et al. (2019) from finite to continuous spaces. Our paper also extends the recent results of Majumdar et al. (2020) from Büchi specifications to parity specifications. Due to space constrains, the proofs for all technical results are provided in the extended version of this paper (Majumdar et al., 2021).

## 2. PRELIMINARIES

### 2.1 Notation

We consider Borel space $S$ which is assumed to be endowed with a Borel sigma-algebra (i.e., the one generated by the open sets in the topology), which is denoted by $\mathcal{B}(S)$. We say that a map $f : S \to Y$ is measurable whenever it is Borel measurable. A Borel space $(S, \mathcal{B}(S))$ is endowed with a probability measures $P$, which is assumed to be induced by a random variable mapping elements of some underlying probability space to the space $(S, \mathcal{B}(S))$. More details can be found in any standard book on Markov processes (Bertsekas and Shreve, 1996).

Given an alphabet $A$, we use the notation $A^*$ and $A^\omega$ to denote respectively the set of all finite and infinite words formed using the letters of the alphabet $A$, and use $A^\infty$ to denote the set $A^* \cup A^\omega$. Let $X$ be a set and $R \subseteq X \times X$ be a relation. For simplicity, let us assume that $\mathsf{dom}\ R \coloneqq \{x \in X \mid \exists y \in X\ .\ (x, y) \in R\} = X$. For any given $x \in X$, we use the notation $R(x)$ to denote the set $\{y \in X \mid (x, y) \in R\}$. We extend this notation to sets: For any given $Z \subseteq X$, we write $R(Z)$ to denote $\cup_{z \in Z} R(z)$.

A probability distribution over a finite set $A$ is a probability measure on the space $(A, 2^A)$. We use the notation $Dist(A)$ to denote the set of all probability distributions over $A$. Given any distribution $f \in Dist(A)$, we define the support of $f$ as: $\mathsf{supp}\ (f) \coloneqq \{a \in A \mid f(a) > 0\}$.

We denote the set of nonnegative integers by $\mathbb{N} \coloneqq \{0, 1, 2, \ldots\}$ and the set of integers in an interval by $[a; b] \coloneqq \{a+k \mid k \in \mathbb{N},\ k \leq b-a\}$. We also use the symbols "$\in_{even}$" and "$\in_{odd}$" to denote memberships in the set of even and odd integers within a given set of integers: For example, for a given set of natural numbers $M \subseteq \mathbb{N}$, the notation $n \in_{even} M$ is equivalent to $n \in M \cap \{0, 2, 4, \ldots\}$, and the notation $n \in_{odd} M$ is equivalent to $n \in M \cap \{1, 3, 5, \ldots\}$.

### 2.2 Controlled Markov Processes

A *controlled Markov process (CMP)* is a tuple $\mathfrak{S} = (\mathcal{S}, \mathcal{U}, T_\mathfrak{s})$, where $\mathcal{S}$ is a Borel space called the *state space*, $\mathcal{U}$ is a finite set called the *input space*, and $T_\mathfrak{s}$ is a conditional stochastic kernel $T_\mathfrak{s} : \mathcal{B}(\mathcal{S}) \times \mathcal{S} \times \mathcal{U} \to [0, 1]$ with $\mathcal{B}(\mathcal{S})$ being the Borel sigma-algebra on the state space and $(\mathcal{S}, \mathcal{B}(\mathcal{S}))$ being the corresponding measurable space. The kernel $T_\mathfrak{s}$ assigns to any $s \in \mathcal{S}$ and $u \in \mathcal{U}$ a probability measure $T_\mathfrak{s}(\cdot|s, u)$ on the measurable space $(\mathcal{S}, \mathcal{B}(\mathcal{S}))$ so that for any set $A \in \mathcal{B}(\mathcal{S}), P_{s,u}(A) = \int_A T_\mathfrak{s}(ds|s, u)$, where $P_{s,u}$ denotes the conditional probability $P(\cdot|s, u)$.

The evolution of a CMP is as follows. For $k \in \mathbb{N}$, let $X^k$ denote the state at the $k$th time step and $A^k$ the input chosen at that time. If $X^k = s \in \mathcal{S}$ and $A^k = u \in \mathcal{U}$, then the system moves to the next state $X^{k+1}$, according to the probability distribution $P_{s,u}$. Once the transition into the next state has occurred, a new input is chosen, and the process is repeated.

Given a CMP $\mathfrak{S}$, a *finite path* of length $n+1$ is a sequence

$$w^n = (s^0, s^1, \ldots, s^n), \quad n \in \mathbb{N},$$

where $s^i \in \mathcal{S}$ are state coordinates of the path. The space of all paths of length $n + 1$ is denoted $\mathcal{S}^{n+1}$. An *infinite*

*path* of the CMP $\mathfrak{S}$ is the sequence $w = (s^0, s^1, s^2, \ldots)$, where $s^i \in \mathcal{S}$ for all $i \in \mathbb{N}$. The space of all infinite paths is denoted by $\mathcal{S}^\omega$. The spaces $\mathcal{S}^{n+1}$ and $\mathcal{S}^\omega$ are endowed with their respective product topologies and are Borel spaces.

A *stationary control policy* is a universally measurable function $\rho : \mathcal{S} \to \mathcal{U}$ such that at any time step $n \in \mathbb{N}$, the input $u^n$ is taken to be $\rho(s^n) \in \mathcal{U}$. As we only deal with stationary control policies in this paper, we simply refer to them as *policies* for short. We denote the class of all such policies by $\Pi$. The function $\rho$ 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 (Hernández-Lerma and Lasserre, 1996), denoted by $P^\rho_\alpha$ with the expectation $\mathbb{E}^\rho_\alpha$. In the case when the initial probability measure is supported on a single state $s \in \mathcal{S}$, i.e., $\alpha(s) = 1$, we write $P^\rho_s$ and $\mathbb{E}^\rho_s$ in place of $P^\rho_\alpha$ and $\mathbb{E}^\rho_\alpha$, respectively. We denote the set of probability measures on $(\mathcal{S}, \mathcal{B}(\mathcal{S}))$ by $\mathfrak{D}$.

### *2.3 Parity Specifications*

Let $\mathfrak{S} = (\mathcal{S}, \mathcal{U}, T_s)$ be a CMP and suppose $\mathcal{P} = \langle B_0, B_1, \ldots, B_\ell \rangle$ is a partition of $\mathcal{S}$ with measurable sets $B_0, \ldots, B_\ell$; that is, $B_i \cap B_j = \emptyset$ for $i \neq j$ and $\cup_{i=1}^{\ell} B_i = \mathcal{S}$. We allow some $B_i$'s to be empty. For each $B_i$, we call the integer $i$ its *priority*. An infinite path $w \in \mathcal{S}^\omega$ satisfies the *parity specification* if the highest priority set visited infinitely often by $w$ is even. We indicate the set of all infinite paths $w \in \mathcal{S}^\omega$ of a CMP $\mathfrak{S}$ that satisfy the property $\mathit{Parity}(\mathcal{P})$ by $\mathfrak{S} \models \mathit{Parity}(\mathcal{P})$. The event $\mathfrak{S} \models \mathit{Parity}(\mathcal{P})$ is measurable because $\mathfrak{S} \models \mathit{Parity}(\mathcal{P})$ can be written as a Boolean combination of events $\mathfrak{S} \models \Box\Diamond A$, where $A$ is a measurable set, and $\Box\Diamond A$ is a canonical $G_\delta$ set in the Borel hierarchy. Thus, $P^\rho_\alpha(\mathfrak{S} \models \mathit{Parity}(\mathcal{P}))$ denotes the probability of satisfaction of $\mathit{Parity}(\mathcal{P})$ by $\mathfrak{S}$ under the effect of the control policy $\rho$, when the initial probability measure is given by $\alpha$.

It is well-known that every $\omega$-regular specification whose propositions range over measurable subsets of the state space of a CMP can be modeled as a deterministic parity automaton (Gradel and Thomas, 2002, Thm. 1.19). By taking a synchronized product of this parity automaton with the CMP, we can obtain a product CMP and a parity specification over the product state space such that every path satisfying the parity specification also satisfies the original $\omega$-regular specification and vice versa. Moreover, a stationary policy for the parity objective gives a (possibly history-dependent) policy for the original specification. Thus, without loss of generality, we assume that an $\omega$-regular objective is already given as a parity condition using a partition of the state space of the system.

### *2.4 Problem Statement*

We are interested in finding the set of initial states of a CMP $\mathfrak{S}$ from which a given parity specification $\mathit{Parity}(\mathcal{P})$ can be satisfied with probability 1 using a given stationary policy $\rho$. The respective set of states is called the *almost sure winning region*, and is defined as follows:

$$\mathsf{WinDom}(\mathfrak{S}, \rho) \coloneqq \{s \in \mathcal{S} \mid P^\rho_s(\mathfrak{S} \models \mathit{Parity}(\mathcal{P})) = 1\}. \quad (1)$$

We also define the *maximal* almost sure winning region as follows:

$$\mathsf{WinDom}^*(\mathfrak{S}) \coloneqq \{s \in \mathcal{S} \mid \sup_{\rho \in \Pi} P^\rho_s(\mathfrak{S} \models \mathit{Parity}(\mathcal{P})) = 1\}. \quad (2)$$

An *optimal control policy* $\rho^*$ is a policy such that $\mathsf{WinDom}(\mathfrak{S}, \rho^*) = \mathsf{WinDom}^*(\mathfrak{S})$. Note that an optimal control policy might not exist, since the supremum (in the definition of $\mathsf{WinDom}^*$) may not be achievable by any policy. We are unaware of any work characterizing necessary or sufficient conditions for existence of optimal control policies on continuous-space CMPs for parity specifications. Additionally, we restrict our attention to *stationary* policies. While it is possible to define more general classes of policies, that depend on the entire history and use randomization over $\mathcal{U}$, we are again unaware of any work that characterizes the class of policies that are sufficient for optimal control of CMPs for parity specifications. For finite-state systems, stationary policies are sufficient and we restrict our attention to this class of policies.

Since we cannot prove existence or computability of optimal policies, in this paper, we focus on providing a best-effort *under-approximation* procedure to compute a possibly sub-optimal control policy $\rho$ and the corresponding almost sure winning region $\mathsf{WinDom}(\mathfrak{S}, \rho)$:

*Problem 1.* (Approximate Maximal Winning Region). Given $\mathfrak{S}$ and a parity specification $\mathit{Parity}(\mathcal{P})$, find a (sub-optimal) control policy $\rho \in \Pi$, its winning region $\mathsf{WinDom}(\mathfrak{S}, \rho) \neq \emptyset$, and an upper bound on the volume of the set difference $\mathsf{WinDom}^*(\mathfrak{S}) \backslash \mathsf{WinDom}(\mathfrak{S}, \rho)$, which we call the *approximation error*.

We provide a solution for Prob. 1 in Sec. 3, where the sought control policy $\rho$ and the respective winning region $\mathsf{WinDom}(\mathfrak{S}, \rho)$ are obtained through abstraction-based controller design. Besides, we compute an *over-approximation* of $\mathsf{WinDom}^*(\mathfrak{S}, \rho)$, and the volume of the set difference between the over- and the under-approximation gives us the sought bound on the approximation error. Unsurprisingly, when we use a finer discretization of the state space during the abstraction step, we get a tighter (i.e., larger) approximation of $\mathsf{WinDom}^*(\mathfrak{S})$, resulting in a smaller approximation error. This fact is empirically validated using a numerical example in Sec. 4.

## 3. ABSTRACTION-BASED POLICY SYNTHESIS

The main result of our paper is a solution to Prob. 1 via a *symbolic algorithm* over abstract $2\frac{1}{2}$-player games in the spirit of abstraction-based controller design (ABCD). Standard ABCD techniques for *non-stochastic* nonlinear systems use a finite two-player game abstraction of the given system to synthesize a controller (Reissig et al., 2017). We build a $2\frac{1}{2}$-player game (Chatterjee and Henzinger, 2012) abstraction of the given CMP to synthesize the controller. The key insight in our abstract $2\frac{1}{2}$-player game is that the stochastic disturbance can be modeled as a fair random player (the $\frac{1}{2}$-player), which makes the synthesis problem easier compared to a purely nondeterministic adversary. In the abstract game, only the effect of the discretization is handled by *Player* 1 in an adversarial manner. In the rest of the section, we explain our approach.

### 3.1 Preliminaries: $2\frac{1}{2}$-Player Parity Games

A $2\frac{1}{2}$-player *game graph* is a tuple $\mathcal{G} = \langle V, E, \langle V_0, V_1, V_r\rangle\rangle$, where $V$ is a finite set of vertices, $E$ is a set of directed edges $E \subseteq V \times V$, and the sets $V_0, V_1, V_r$ form a partition of the set $V$. A $2\frac{1}{2}$-player *parity game* is a pair $\langle \mathcal{G}, \mathcal{P}\rangle$, where $\mathcal{G}$ is a $2\frac{1}{2}$-player game graph, and $\mathcal{P} = \langle B_0, B_1, \dots, B_\ell\rangle$ is a tuple of $\ell$ disjoint subsets of $V$, some of which can possibly be empty. The tuple $\mathcal{P}$ induces the parity specification $\mathit{Parity}(\mathcal{P})$ over the set of vertices $V$ in the natural way. In order to ensure that $\mathit{Parity}(\mathcal{P})$ is well defined, we impose the restriction that every infinite run must have infinitely many occurrences of vertices from at least one of the sets in $\mathcal{P}$. In other words, we require that every set of vertices $U \subseteq V$ for which there is no $i \in [1; \ell]$ with $U \cap B_i \neq \emptyset$ must be "transient" vertices.

**The players and their strategies.** We assume that there are two players *Player* 0 and *Player* 1, who are playing a game by moving a token along the edges of the game graph $\mathcal{G}$. In every step, if the token is located in a vertex in $V_0$ or $V_1$, *Player* 0 or *Player* 1 respectively moves the token to one of the successors according to the edge relation $E$. On the other hand, if the token is located in a vertex $v \in V_r$, then in the next step the token moves to a vertex $v'$ which is chosen uniformly at random from the set $E(v)$. Strategies of *Player* 0 and *Player* 1 are respectively the functions $\pi_0 \colon V^*V_0 \to \mathit{Dist}(V)$ and $\pi_1 \colon V^*V_1 \to \mathit{Dist}(V)$ such that for all $w \in V^*$, $v_0 \in V_0$ and $v_1 \in V_1$, we have $\mathsf{supp}\ \pi_0(wv_0) \subseteq E(v_0)$ and $\mathsf{supp}\ \pi_1(wv_1) \subseteq E(v_1)$. We use the notation $\Pi_0$ and $\Pi_1$ to denote the set of all strategies of *Player* 0 and *Player* 1 respectively. A strategy $\pi_i$ of *Player* $i$, for $i \in \{0, 1\}$, is *deterministic memoryless* if for every $w_1, w_2 \in V^*$ and for every $v \in V_i$, $\pi_i(w_1 v) \equiv \pi_i(w_2 v)$ holds; we simply write $\pi_i(v)$ in this case. We use the notation $\Pi_i^{\mathrm{DM}}$ to denote the set of all deterministic memoryless strategies of *Player* $i$. Observe that $\Pi_i^{\mathrm{DM}} \subseteq \Pi_i$.

**Runs and winning conditions.** An infinite (finite) run of the game graph $\mathcal{G}$, compatible with the strategies $\pi_0 \in \Pi_0$ and $\pi_1 \in \Pi_1$, is an infinite (a finite) sequence of vertices $r = v^0 v^1 v^2 \dots$ ($r = v^0 \dots v^n$ for some $n \in \mathbb{N}$) such that for every $k \in \mathbb{N}$, (a) $v^k \in V_0$ implies $v^{k+1} \in \mathsf{supp}\ \pi_0(v^0 \dots v^k)$, (b) $v^k \in V_1$ implies $v^{k+1} \in \mathsf{supp}\ \pi_1(v^0 \dots v^k)$, and (c) $v^k \in V_r$ implies $v^{k+1} \in E(v^k)$. Given an initial vertex $v^0$ and a fixed pair of strategies $\pi_0 \in \Pi_0$ and $\pi_1 \in \Pi_1$, we obtain a probability distribution over the set of infinite runs of the system. For a measurable set of runs $R \subseteq V^\omega$, we use the notation $P_{v^0}^{\pi_0,\pi_1}(R)$ to denote the probability of obtaining the set of runs $R$ when the initial vertex is $v^0$ and the strategies of *Player* 0 and *Player* 1 are fixed to respectively $\pi_0$ and $\pi_1$. For an $\omega$-regular specification $\varphi$, defined using a predicate over the set of vertices of $\mathcal{G}$, we write $(\mathcal{G} \models \varphi)$ to denote the set of all infinite runs for all possible strategies of *Player* 0 and *Player* 1 which satisfy $\varphi$. For example, $(\mathcal{G} \models \mathit{Parity}(\mathcal{P}))$ denotes the set of all infinite runs for all possible strategies of *Player* 0 and *Player* 1 which satisfy the parity condition $\mathit{Parity}(\mathcal{P})$. We say that *Player* 0 wins $\mathit{Parity}(\mathcal{P})$ almost surely from a vertex $v \in V$ (or $v$ is almost sure winning for *Player* 0) if *Player* 0 has a strategy $\pi_0 \in \Pi_0$ such that for all $\pi_1 \in \Pi_1$ we have $P_v^{\pi_0,\pi_1}(\mathcal{G} \models \mathit{Parity}(\mathcal{P})) = 1$. We collect all the vertices for which this is true in the almost sure winning region $\mathcal{W}(\mathcal{G} \models \mathit{Parity}(\mathcal{P}))$.

### 3.2 Abstraction: CMPs to $2\frac{1}{2}$-Player Games

Given a CMP $\mathfrak{S} = (\mathcal{S}, \mathcal{U}, T_{\mathfrak{s}})$ and a parity specification $\mathit{Parity}(\mathcal{P})$ for a partition $\mathcal{P}$ of the state space $\mathcal{S}$ we construct an abstract $2\frac{1}{2}$-player game.

**State-space abstraction.** We introduce a finite partition $\widehat{\mathcal{S}} \coloneqq \{\widehat{s}_i\}_{i \in I}$ such that $\cup_{i \in I} \widehat{s}_i = \mathcal{S}$, $\widehat{s}_i \neq \emptyset$ and $\widehat{s}_i \cap \widehat{s}_j = \emptyset$ for every $\widehat{s}_i, \widehat{s}_j \in \widehat{\mathcal{S}}$ with $i \neq j$. Furthermore, we assume that the partition $\widehat{\mathcal{S}}$ is consistent with the given priorities $\mathcal{P}$, i.e., for every partition element $\widehat{s} \in \widehat{\mathcal{S}}$, and for every $x, y \in \widehat{s}$, $x$ and $y$ belong to the same partition element in $\mathcal{P}$ (i.e., $x$ and $y$ are assigned the same priority). We call the set $\widehat{\mathcal{S}}$ the *abstract state space* and each element $\widehat{s} \in \widehat{\mathcal{S}}$ an *abstract state*.

We introduce the abstraction function $Q \colon \mathcal{S} \to \widehat{\mathcal{S}}$ as a mapping from the continuous to the abstract states: For every $s \in \mathcal{S}$, $Q \colon s \mapsto \widehat{s}$ such that $s \in \widehat{s}$. We define the concretization function as the inverse of the abstraction function: $Q^{-1} \colon \widehat{\mathcal{S}} \to 2^{\mathcal{S}}$, $Q^{-1} \colon \widehat{s} \mapsto \{s \in \mathcal{S} \mid s \in \widehat{s}\}$. We generalize the use of $Q$ and $Q^{-1}$ to sets of states: For every $U \subseteq \mathcal{S}$, $Q(U) = \bigcup_{s \in U} Q(s)$, and for every $\widehat{U} \subseteq \widehat{\mathcal{S}}$, $Q^{-1}(\widehat{U}) = \bigcup_{\widehat{s} \in \widehat{U}} Q^{-1}(\widehat{s})$.

**Transition abstraction.** We also introduce an over- and an under-approximation of the probabilistic transitions of the CMP $\mathfrak{S}$ using the non-deterministic abstract transition functions $\overline{F} \colon \widehat{\mathcal{S}} \times \mathcal{U} \to 2^{\widehat{\mathcal{S}}}$ and $\underline{F} \colon \widehat{\mathcal{S}} \times \mathcal{U} \to 2^{\widehat{\mathcal{S}}}$ with the following properties:

$$\overline{F}(\widehat{s}, u) \supseteq \{\widehat{s}' \in \widehat{\mathcal{S}} \mid \exists s \in \widehat{s}\ .\ T_{\mathfrak{s}}(\widehat{s}' \mid s, u) > 0\}, \tag{3a}$$

$$\underline{F}(\widehat{s}, u) \subseteq \{\widehat{s}' \in \widehat{\mathcal{S}} \mid \exists \varepsilon > 0\ .\ \forall s \in \widehat{s}\ .\ T_{\mathfrak{s}}(\widehat{s}' \mid s, u) \geq \varepsilon\}. \tag{3b}$$

Intuitively, given an abstract state $\widehat{s}$ and an input $u$, the set $\overline{F}$ over-approximates the set of abstract states reachable by probabilistic transitions from $\widehat{s}$ on input $u$. On the other hand, $\underline{F}$ under-approximates those abstract states which can be reached by *every* state in $\widehat{s}$ with probability bounded away from zero. Unlike abstractions for control of non-stochastic systems using $\overline{F}$-like transitions, we need both $\overline{F}$ and $\underline{F}$ for stochastic systems.

The parameter $\varepsilon$ states that there is a uniform lower bound on transition probabilities for all states in an abstract state. This ensures that, provided $\widehat{s}$ is visited infinitely often and $u$ is applied infinitely often from $\widehat{s}$, then $\widehat{s}'$ will be reached almost surely from $\widehat{s}$. In the absence of a uniform lower bound, this property need not hold for infinite state systems; for example, if the probability goes to zero, the probability of escaping $\widehat{s}$ can be strictly less than one.

While it is difficult to compute $\overline{F}$ and $\underline{F}$ in general, they can be approximated for the important subclass of stochastic nonlinear systems with *affine* disturbances.

**Abstract $2\frac{1}{2}$-player game graph.** Given the abstract state space $\widehat{\mathcal{S}}$ and the over and under-approximations of the transition functions $\underline{F}$ and $\overline{F}$, we are ready to construct the abstract $2\frac{1}{2}$-player game graph induced by a CMP.

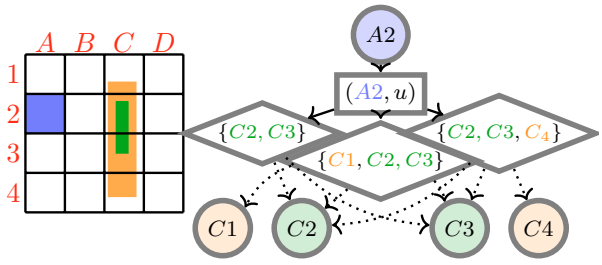


Fig. 1. Construction of the abstract $2^1/_2$-player game (right) from a continuous-state CMP (left). The state space of the CMP is discretized into rectangular abstract states $A1, \ldots, D4$; $\underline{F}(A2, u) = \{C2, C3\}$ (intersecting the green region), and $\overline{F}(A2, u) = \{C1, C2, C3, C4\}$ (intersecting the orange region). $V_0$, $V_1$ and $V_r$ are indicated by circular, rectangular, and diamond-shaped nodes. Random edges are dashed.

*Definition 3.1.* Let $\mathfrak{S}$ be a given CMP. Then its induced abstract $2^1/_2$-player game graph is given by $\mathcal{G} = \langle V, E, \langle V_0, V_1, V_r\rangle\rangle$ such that

- $V_0 = \widehat{\mathcal{S}}$ and $V_1 = \widehat{\mathcal{S}} \times \mathcal{U}$;
- $V_r = \bigcup_{v_1 \in V_1} V_r(v_1)$, where $V_r(v_1) := \{v_r \subseteq \widehat{\mathcal{S}} \mid \underline{F}(v_1) \subseteq v_r \subseteq \overline{F}(v_1), 1 \leq |v_r| \leq |\underline{F}(v_1)|+1\}$;
- and it holds that
  - for all $v_0 \in V_0$, $E(v_0) = \{(v_0, u) \mid u \in \mathcal{U}\}$
  - for all $v_1 \in V_1$, $E(v_1) = V_r(v_1)$, and
  - for all $v_r \in V_r$, $E(v_r) = \{v_0 \in V_0 \mid v_0 \in v_r\}$.

Note that $V_r(v_1)$ contains non-empty subsets of $\widehat{\mathcal{S}}$ that includes all the abstract states in $\underline{F}(v_1)$ and possibly include only one additional element from $\overline{F}(v_1)$. The construction is illustrated in Fig. 1.

In the reduced game, *Player* 0 models the controller, *Player* 1 models the effect of discretization of the state space of $\mathfrak{S}$, and the random edges from the states in $V_r$ model the stochastic nature of the transitions of $\mathfrak{S}$. Intuitively, the game graph in Def. 3.1 captures the following interplay which is illustrated in Fig. 1: At every time step, the control policy for $\mathfrak{S}$ has to choose a control input $u \in \mathcal{U}$ based on the current vertex $\widehat{s}$ of $\mathcal{G}$. Since the control policy is oblivious to the precise continuous state $s \in \mathcal{S}$ of $\mathfrak{S}$, hence $u$ is required to be an optimal choice for *every* continuous state $s \in \widehat{s}$. This requirement is materialized by introducing a fictitious adversary (i.e. *Player* 1) who, given $\widehat{s}$ and $u$, picks a continuous state $s \in \widehat{s}$ from which the control input $u$ is to be applied. Now, we know that no matter what continuous $s$ is chosen by *Player* 1, $T_{\mathfrak{s}}(\underline{F}(\widehat{s}, u) \mid s, u) > \varepsilon$ holds for some $\varepsilon > 0$. This explains why every successor of the $(\widehat{s}, u) \in V_1$ states contains the set of vertices $\underline{F}(\widehat{s}, u)$. Moreover, depending on which exact $s \in \widehat{s}$ *Player* 1 chooses, with positive probability the system might go to some state in $\overline{F}(\widehat{s}, u) \setminus \underline{F}(\widehat{s}, u)$. This is materialized by adding every state in $\overline{F}(\widehat{s}, u) \setminus \underline{F}(\widehat{s}, u)$ at a time to the successors of the states in $V_1$ (see Def. 3.1). Finally, we assume that the successor from every state in $V_r$ is chosen uniformly at random (indicated by dotted edges in Def. 3.1). It can be shown that the exact probability values are never used for obtaining the almost sure winning region, and so we could have chosen any other probability distribution.

**Abstract parity specification.** To conclude the abstraction of a given CMP $\mathfrak{S}$ and its parity specification $\mathcal{P} = \{B_1, \ldots, B_k\}$, we have to formally translate the priority sets $B_i$ defined over subsets of states of the CMP into a partition of the vertices of the abstract $2^1/_2$-player game graph $\mathcal{G}$ induced by $\mathfrak{S}$. To this end, recall that we have assumed that the state space abstraction $\widehat{\mathcal{S}}$ respects the priority set $\mathcal{P}$.

*Definition 3.2.* Let $\mathfrak{S}$ be a CMP with parity specification $Parity(\mathcal{P})$ and $\mathcal{G}$ be the abstract $2^1/_2$-player game graph induced by $\mathfrak{S}$. Then the induced abstract parity specification $\widehat{\mathcal{P}} = \{\widehat{B}_0, \ldots, \widehat{B}_\ell\}$ is defined such that $\widehat{B}_i = \{v_0 \in V_0 \mid Q^{-1}(v_0) \subseteq B_i\}$ for all $i \in [0; \ell]$. We denote the resulting $2^1/_2$-player parity game by the tuple $\langle \mathcal{G}, \widehat{\mathcal{P}}\rangle$.

We note that the choice of the abstract parity set $\widehat{\mathcal{P}}$ does not partition the state space. Indeed, we implicitly assign an "undefined" color "$-$" to all nodes $V_1 \cup V_r$. Thereby, we only interpret the given parity specification over a projection of a run to its player 0 nodes. Formally, a run $r$ over the abstract game graph $\mathcal{G}$ starting from a vertex $s^0 \in V_0$ is of the form $r = s^0, (s^0, u^0), (\{s^{0,0}, \ldots, s^{0,i_0}\}), s^1, (s^1, u^1), (\{s^{1,0}, \ldots, s^{1,i_1}\}), \ldots$, where $s^k \in \{s^{k,0}, \ldots, s^{k,i_k}\}$ for all $k \in \mathbb{N}$. The projection of the run $r$ to the player 0 states is defined as $\mathrm{Proj}_{V_0}(r) = s^0, s^1, \ldots$. Let $\varphi$ be an $\omega$-regular specification defined using a set of predicates over $V_0$. We use the convention that $(\mathcal{G} \models \varphi)$ will denote the set of every infinite run $r$ of $\mathcal{G}$, for any arbitrary pair of strategies of *Player* 0 and *Player* 1, such that $\mathrm{Proj}_{V_0}(r)$ satisfies $\varphi$. This convention is well-defined because every infinite run of $\mathcal{G}$ will have infinitely many occurrences of vertices from $V_0$ in it: This follows from the strict alternation of the vertices in $V_0$, $V_1$, and $V_r$, as per Def. 3.1.

### *3.3 Synthesis and Refinement*

Once the $2^1/_2$-player parity game $\langle \mathcal{G}, \widehat{\mathcal{P}}\rangle$ is constructed from the CMP $\mathfrak{S}$ according to Def. 3.1, one can use existing techniques (Chatterjee et al., 2003) to compute the almost sure winning states of *Player* 0 along with an associated almost sure memoryless *Player* 0 winning strategy $\pi_0$ over $\langle \mathcal{G}, \widehat{\mathcal{P}}\rangle$. Then we can *refine* $\pi_0$ to a policy $\rho$ for the CMP by setting $\rho(s) := u$ for every $s \in \mathcal{S}$, if and only if $s \in \widehat{s} \in \widehat{\mathcal{S}}$ and $\pi_0(\widehat{s}) = (\widehat{s}, u) \in V_1$.

With the completion of this last step of our ABCD method for stochastic nonlinear systems we can finally state our main theorem providing a solution to Problem 1.

*Theorem 3.3.* (Solution of Problem 1). Let $\mathfrak{S}$ be a CMP and $Parity(\mathcal{P})$ be a given parity specification. Let $\langle \mathcal{G}, \widehat{\mathcal{P}}\rangle$ be the abstract $2^1/_2$-player game defined in Def. 3.1. Suppose, a vertex $\widehat{s} \in V_0$ is almost sure winning for *Player* 0 in the game $\langle \mathcal{G}, \widehat{\mathcal{P}}\rangle$, and $\pi_0 \in \Pi^{\mathrm{DM}}$ is the corresponding *Player* 0 winning strategy. Then the refinement $\rho$ of $\pi_0$ ensures that $\widehat{s} \subseteq \mathsf{WinDom}(\mathfrak{S}, \rho)$.

*Remark 1.* An *over*-approximation of the optimal almost sure winning domain $\mathsf{WinDom}^*(\mathfrak{S})$ of $\mathfrak{S}$ w.r.t. $Parity(\mathcal{P})$ can be computed via $\langle \mathcal{G}, \widehat{\mathcal{P}}\rangle$ as well. To obtain an *over*-

Table 1. Performance evaluation: Col. 1 shows the size of the abstract states, Col. 2 shows an upper-bound on the approximation error (obtained by measuring the volume of the set difference between the over- and the under-approximation), and Col. 3, 4, and 5 respectively show the computation times for the $2^1/_2$-player game, the over-approximation, and the under-approximation of the winning region.

| Size of abstract states | Bound on approx. error | Computation time | | |
|---|---|---|---|---|
| | | Abs. | Over-approx. | Under-approx. |
| $0.1 \times 0.1 \times 0.1$ | 6.6 | $< 1m$ | $9m$ | $31m$ |
| $0.08 \times 0.08 \times 0.08$ | 4.8 | $2m$ | $84m$ | $4h$ |
| $0.06 \times 0.06 \times 0.06$ | 4.5 | $7m$ | $102m$ | $9h$ |

approximation, we solve this abstract game *cooperatively*. That is, we let player *Player* 0 choose both its own moves and the moves of player $p_1$ to win almost surely w.r.t. $\mathit{Parity}(\widehat{\mathcal{P}})$. The volume of the set difference between the over- and the under-approximation gives us an upper bound on the approximation error.

## 4. NUMERICAL EXAMPLE

We demonstrate the effectiveness of our controller synthesis approach using a numerical example. We consider the controller synthesis problem for a mobile robot, modeled using the sampled-time version of the perturbed Dubins vehicle with 3 state variables and 1 control input. The state space of the robot is annotated using a set of atomic propositions $A_0$, $A_1$, $G_0$, $G_1$, and Crash; $A_0$ and $A_1$ represent certain events under the environment's influence such as the opening/closing of doors, etc., $G_0$ and $G_1$ represent certain events under the robot's influence such as reaching a target, and Crash represents the event of the robot colliding against any obstacle. The specification for the robot is given as:

$$\begin{aligned} &\square\neg\text{Crash} \\ \wedge\ &(\square\lozenge A_0 \wedge \square\,(A_0 \rightarrow (A_0 \mathcal{U} G_1)) \rightarrow \square\lozenge G_0 \wedge \square\lozenge G_1)\,. \end{aligned} \tag{4}$$

The specification in (4) can be modeled as a 3-color parity automaton. We computed the synchronous product of the parity automaton and the vehicle's dynamics model. We used the infrastructure of Mascot-SDS (Majumdar et al., 2020) to compute a $2^1/_2$-player game and to synthesize an almost sure winning controller for the product system. We performed the experiments on a computer with 3.3GHz Intel Xeon E5 v2 processor and 256 GB RAM. We used three different levels of discretization for the abstract state space for computing the $2^1/_2$-player game. The results are summarized in Tab. 1. We would like to highlight two key facts which came out of the experiments: (a) In all three cases, when we treated the environmental noise in the worst case fashion, the synthesis process failed to provide us any controller, and (b) as we decreased the size of the abstract states (i.e., finer abstraction), the bound on the approximation error got monotonically smaller, which empirically confirms the intuition that the quality of the approximation improves with finer abstraction.

## REFERENCES


Baier, C. and Katoen, J.P. (2008). *Principles of Model Checking*. MIT Press.

Bertsekas, D. and Shreve, S. (1996). *Stochastic Optimal Control: The Discrete-Time Case*. Athena Scientific.

Chatterjee, K. and Henzinger, T.A. (2012). A survey of stochastic $\omega$-regular games. *J. Comput. Syst. Sci.*, 78(2), 394–413.

Chatterjee, K., Jurdzinski, M., and Henzinger, T.A. (2003). Simple stochastic parity games. In *CSL 2003*, volume 2803 of *Lecture Notes in Computer Science*, 100–113. Springer.

Dutreix, M., Huh, J., and Coogan, S. (2020). Abstraction-based synthesis for stochastic systems with omega-regular objectives. *arXiv preprint*. ArXiv:2001.09236.

Gradel, E. and Thomas, W. (2002). *Automata, logics, and infinite games: a guide to current research*, volume 2500. Springer Science & Business Media.

Haesaert, S. and Soudjani, S. (2018). Robust dynamic programming for temporal logic control of stochastic systems. *CoRR*, abs/1811.11445.

Hernández-Lerma, O. and Lasserre, J.B. (1996). *Discrete-time Markov control processes*, volume 30 of *Applications of Mathematics*. Springer.

Kwiatkowska, M., Norman, G., Parker, D., and Santos, G. (2020). PRISM-games 3.0: Stochastic game verification with concurrency, equilibria and time. In *CAV'20*, volume 12225 of *LNCS*, 475–487. Springer.

Laurenti, L., Lahijanian, M., Abate, A., Cardelli, L., and Kwiatkowska, M. (2020). Formal and efficient synthesis for continuous-time linear stochastic hybrid processes. *IEEE TAC*.

Majumdar, R., Mallik, K., Schmuck, A.K., and Soudjani, S. (2021). Symbolic control for stochastic systems via parity games.

Majumdar, R., Mallik, K., and Soudjani, S. (2020). Symbolic controller synthesis for Büchi specifications on stochastic systems. In *HSCC'20*, 1–11.

Nilsson, P., Ozay, N., and Liu, J. (2017). Augmented finite transition systems as abstractions for control synthesis. *DEDS*, 27(2), 301–340.

Pola, G., Girard, A., and Tabuada, P. (2008). Approximately bisimilar symbolic models for nonlinear control systems. *Automatica*, 44(10), 2508–2516.

Reissig, G., Weber, A., and Rungger, M. (2017). Feedback refinement relations for the synthesis of symbolic controllers. *TAC*, 62(4), 1781–1796.

Svoreňová, M., Křetínskỳ, J., Chmelík, M., Chatterjee, K., Černá, I., and Belta, C. (2017). Temporal logic control for stochastic linear systems using abstraction refinement of probabilistic games. *NAHS*, 23, 230–253.

Thomas, W. (1995). On the synthesis of strategies in infinite games. In *STACS '95*, volume 900 of *Lecture Notes in Computer Science*, 1–13. Springer.

Tkachev, I., Mereacre, A., Katoen, J.P., and Abate, A. (2017). Quantitative model-checking of controlled discrete-time Markov processes. *Information and Computation*, 253, 1 – 35.

Weininger, M., Meggendorfer, T., and Křetínskỳ, J. (2019). Satisfiability bounds for $\omega$-regular properties in bounded-parameter markov decision processes. In *CDC'19*, 2284–2291. IEEE.