Formal Verification and Synthesis for Discrete-Time Stochastic Systems

Morteza Lahijanian, *Member, IEEE*, Sean B. Andersson, *Senior Member, IEEE*, and Calin Belta, *Senior Member, IEEE*

Abstract—**Formal methods are increasingly being used for control and verification of dynamic systems against complex specifications. In general, these methods rely on a relatively simple system model, such as a transition graph, Markov chain, or Markov decision process, and require abstraction of the original continuous-state dynamics. It can be difficult or impossible, however, to find a perfectly equivalent abstraction, particularly when the original system is stochastic. Here we develop an abstraction procedure that maps a discrete-time stochastic system to an Interval-valued Markov Chain (IMC) and a switched discrete-time stochastic system to a Bounded-parameter Markov Decision Process (BMDP). We construct model checking algorithms for these models against Probabilistic Computation Tree Logic (PCTL) formulas and a synthesis procedure for BMDPs. Finally, we develop an efficient refinement algorithm that reduces the uncertainty in the abstraction. The technique is illustrated through simulation.**

Index Terms—**Finite abstraction, formal synthesis, formal verification, Markov abstraction, model checking, PCTL, stochastic systems, temporal logics.**

I. Introduction

In classical analysis and control problems, "complex" models, such as systems of differential equations, are usually checked against "simple" specifications. Examples include the stability of an equilibrium, the invariance of a set, and properties such as controllability and observability. There is growing interest, however, in using formal methods [2] to check the behavior of a complex model against "rich" specifications that include notions of safety (i.e., something bad never happens) and liveness (i.e., something good eventually happens). The mathematical foundations of these techniques contribute to the reliability and robustness of a design. They have been used successfully in practice to verify industrial designs (e.g., Rule-Base at IBM), and companies are beginning to market commercial model checkers (e.g., Motorola VeriState-SM and I-Logix Statemate MAGNUM). In addition, these approaches have found application in the synthesis of controls for dynamic systems to ensure a given specification is met (e.g., [3], [4]).

In general, formal methods rely on a relatively simple input model such as a (finite) transition graph. In order to apply these techniques to continuous-domain dynamical systems, these must be abstracted to appropriate finite models. By establishing an equivalence between the system and its abstraction, the satisfaction of the specification by the abstraction guarantees the satisfaction by the original system [5], [6]. In general, however, it can be difficult or impossible to find a perfectly equivalent abstraction, especially when the original system has stochastic dynamics.

In this paper, we consider the problem of formal verification and synthesis of switching strategies for continuous-domain discrete-time stochastic systems evolving in polytopic domains with noise bounds given by polyhedral sets. We are interested in specifications given as Probabilistic Computation Tree Logic (PCTL) [7] formulas. We approach this problem by first constructing an Interval-valued Markov Chain (IMC) [8], [9] or Bounded-parameter Markov Decision Process (BMDP) [10] abstraction of the continuous-domain stochastic system. We interpret the abstraction as a generalized Markov Decision Process (MDP), and model check it using an algorithm similar to MDP model checking algorithms. Lastly, we employ a refinement algorithm to iteratively reduce the uncertainty introduced by the abstraction process.

There are four main contributions of this work. First, we develop a method of creating a finite abstraction of a stochastic dynamic system in a partitioned domain to an IMC or BMDP. Second, we construct a model checking algorithm for IMCs and BMDPs to find the sets of initial states that definitely, possibly, and never satisfy a given specification. Third, we develop an algorithm for BMDPs that synthesizes a policy that maximizes the probability of satisfying a specification. Fourth, we generate an adaptive refinement algorithm that exploits the dynamics of the system and the geometry of the partition to increase the precision of the solution. A preliminary version of this work considering only abstractions to Markov chains appeared in [1].

The remainder of the paper is organized as follows. Section I-A contains related work. In Section II, we formulate the verification and synthesis problems and outline our approach. In Section III, background material on IMCs, BMDPs, and polyhedral operations is given. The abstraction procedure is discussed in Section IV while our model checking algorithms for IMCs and BMDPs are presented in Section V. Section VI

Manuscript received February 27, 2014; revised September 19, 2014; accepted January 18, 2015. Date of publication February 2, 2015; date of current version July 24, 2015. This work was supported in part by NSF under grants CNS-0834260 and CMMI-0928776, the ARO under W911NF-09-1-0088, the AFOSR under FA9550-09-1-0209, and the ONR under ONR MURI N00014-09-1051. This paper appeared in part in the IEEE Conference on Decision and Control, Maui, HI, December 2012. Recommended by Associate Editor H. S. Chang.

M. Lahijanian was with the Department of Mechanical Engineering, Boston University, Boston, MA 02215 USA. He is now with the Department of Computer Science, Rice University, Houston, TX 77005 USA (e-mail: morteza@rice.edu).

S. B. Andersson and C. Belta are with the Department of Mechanical Engineering and the Division of Systems Engineering, Boston University, Boston, MA 02215 USA (e-mail: sanderss@bu.edu; cbelta@bu.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TAC.2015.2398883

discusses the adaptive refinement procedure designed to increase the precision of the solution. In Section VII, we introduce our synthesis algorithms for BMDPs and discuss their conservativeness. Performance of the proposed framework is illustrated through case studies in Section VIII. We conclude with final remarks in Section IX.

A. Related Work

Much work has been done in the areas of abstraction, verification, and synthesis for stochastic systems from temporal logic specifications. Existing methods are generally based on Markov models such as Markov Chains (MCs) and MDPs in which the transition probability distributions are assumed to be known exactly (e.g., [11], [12]). An abstraction is typically obtained using Monte Carlo simulation techniques to determine, at least approximately, the transition probabilities between the states and the state space is augmented if needed to achieve Markovianity in the transitions [13], [14]. Recent works [15], [16] developed a different approach in which stochastic hybrid systems were abstracted to Markov chains with approximation error, also known as Markov set-chains [15]. In [15]–[17], a bound on that error was determined using a Lipschitz continuity condition on the stochastic kernels of the underlying hybrid system. An adaptive grid-based algorithm was employed to reduce this error to any desired level. The technique, however, used a conservative bound that in general leads to a higher cardinality of the abstraction than is necessary to achieve the desired error level. In our previous work, we followed a similar method to construct a Markov chain abstraction for a stochastic linear system with bounded noise [1]. The computational framework allowed for the calculation of exact bounds on the approximation error but not for uncertainty in the abstraction itself.

To establish the equivalence relation between the original stochastic system and its abstraction, the notions of exact and approximate bisimulations have been developed for some classes of stochastic hybrid systems [18]–[22]. The work in [18] uses concepts from category theory to develop a notion of exact bisimulation for general stochastic hybrid systems, while in [19] the notion of exact bisimulation is generated for communicating piecewise deterministic hybrid systems. In [20], [21], the authors develop the notion of approximate bisimulation for labeled Markov processes and probabilistic transition systems by using a Hutchinson-like metric. This metric measures the distance between two distributions of the transition probability. The work in [22] generates a theory of approximate bisimulation for a class of stochastic hybrid automata by constructing a Lyapunov-like stochastic bisimulation function. The relation between the continuous-domain stochastic system and the abstraction model that we construct for it in this paper is closer to the approximate bisimulation concept introduced in [20], [21]. The abstraction model in our study is either an IMC or a BMDP, which include a set of distributions with well-defined bounds. The true distribution of the continuous-domain stochastic system lies in this set. Thus, an upper bound can be defined for the distance between the distribution of the system and the boundaries of the set.

There are a variety of algorithms available for verification of MCs and MDPs against temporal logics such as Linear Temporal Logic (LTL) [2] and PCTL specifications. These algorithms are implemented in software tools such as PRISM [23] and MRMC [24]. Formal synthesis tools for MDPs have also been developed from PCTL [12] and Probabilistic LTL [25] specifications. These verification and synthesis tools have been successfully applied to numerous fields including robotics, systems biology, hardware circuits, and software reliability analysis.

More recently, verification and synthesis methods have been developed for uncertain systems with continuous state spaces. The work in [6] constructed a framework for LTL verification of affine systems under parameter uncertainty. The authors in [11] introduced a PCTL model checking approach to discrete-time Stochastic Hybrid Systems (DTSHS) through approximate MC abstraction. They also used the same abstraction technique for the verification of DTSHS against probabilistic linear-time objectives by constructing the synchronous product of the approximate MC of the DTSHS and the automaton representing the specification and then computing the probability of the satisfying paths of the product automaton [26], [27]. One of the benefits of these techniques is their use of existing model checking and synthesis tools. However, there are a number of disadvantages including the state explosion problem, which is partly due to the current MC and MDP abstraction methods.

To overcome the state explosion problem, a number of works have been developed in the field of model checking that are based on a coarse and uncertain abstract of probabilistic systems. The work in [28] introduces an IMC abstraction technique for discrete-time MCs and proposes a model checking algorithm for IMCs. The verification algorithm, however, suffers from high computational cost for large systems. The authors in [29], [30] extend the IMC abstraction framework to continuous-time MCs and present an IMC model checking algorithm with polynomial complexity. Nevertheless, the considered logics only support bounded-time temporal operators. The complexity analysis of verifying IMCs against different properties are studied in [31]–[33]. The work in [34] introduces a game-based approach to generate coarse abstractions for MDPs. For safety verification of continuous stochastic systems, works [35], [36] propose similar abstraction techniques.

In this paper, we propose a complete formal verification and synthesis framework that avoids the state explosion problem. We use IMCs and BMDPs as abstraction models for stochastic systems and quantify the exact bounds on the transition probabilities by using the distribution of the stochastic system. We then introduce computationally efficient verification and synthesis algorithms for these models. Our algorithms compute the probability bounds of satisfying the specification for each state of the IMC or BMDP. Finally, we introduce a guided refinement algorithm that increases the accuracy of the solution while limiting the growth in the size of the state space by focusing on those states with the largest satisfying probability intervals.

II. Problem Formulation

We consider the following continuous-domain, discrete-time, switched stochastic system:

$$x_{k+1} = \mathcal{F}_a(x_k, w_k), \quad x_k \in \mathrm{P} \subset \mathbb{R}^n,\ w_k \in W \subset \mathbb{R}^n \quad (1)$$

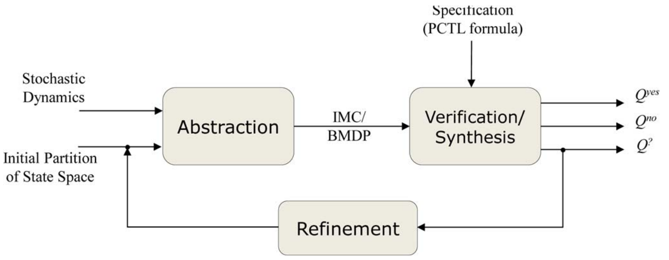


Fig. 1. Block diagram representation of the approach to Problems 1 and 2. Q^{yes}, $Q^{?}$, and Q^{no} are the sets of states that definitely, possibly, and never satisfy the specification, respectively.

where $a \in \mathrm{I}_a = \{1, 2, \cdots, m_a\}$, $\mathcal{F}_a : \mathrm{P} \times W \to \mathrm{P}$ is a possibly nonlinear function, P is a full dimensional polytope in $\mathbb{R}^n$, $k \in \mathbb{N}^0$, and w_k is a sample from a given probability distribution over a polyhedral subset W of $\mathbb{R}^n$. The index set I_a labels the available dynamics of the system. Let Π be a set of arbitrary strict linear predicates (linear inequalities) over the state x. We are interested in developing a theoretical framework and a computational tool for formal verification and formal synthesis of switching policies for system (1) from temporal logic specifications over Π.

We focus on specifications given as PCTL formulas (formally defined in Section III-B) for their high expressivity and low complexity in their verification algorithms. Some examples of the system specifications that can be expressed in PCTL are as follows.

> Is the probability of reaching a *bad* state in less than ten steps greater than 0.01?
>
> Find a set of initial states that with probability 0.95 or greater will eventually reach a *goal* state without visiting a *bad* state.
>
> Find a control policy that maximizes the probability of reaching *goal* through the regions from which the probability of hitting an *obstacle* is less than 0.05.

In this study, we consider the following two distinct but related problems.

Problem 1 (Verification): Given a stochastic system of the form (1) and a PCTL formula ϕ over Π, find a set of initial states that satisfy ϕ under all switching policies.

Problem 2 (Synthesis): Given a stochastic system of the form (1) and a PCTL formula ϕ over Π, find a switching policy that maximizes the probability of satisfying ϕ.

In these problems, the switching policy for system (1) is also referred to as a control policy.

Our general approach to the above two problems includes a finite abstraction process followed by verification or synthesis. To create the abstraction, we model system (1) as a finite-state discrete-time Markov model and use this model for analysis and synthesis. As made clear below, this method introduces conservatism. We reduce this conservatism and increase the precision of the result through a refinement stage. The block diagram of this approach is shown in Fig. 1.

In Problem 1 (verification), we assume that the number of available dynamics for system (1) is greater than or equal to one (i.e., $m_a \geq 1$) while, in Problem 2 (synthesis), it is strictly greater than one (i.e., $m_a > 1$). When $m_a = 1$, only one switching policy exists, and the abstraction is in the form of an IMC. Informally, IMCs are MC models where the exact transition probability for each state transition is known only to lie within a given interval [8], [9]. When $m_a > 1$, system (1) is abstracted to a BMDP. A BMDP allows for inclusion of actions (choice of dynamics) at each state in addition to uncertainties over the transition probabilities [10].

To solve the problems defined above, we perform either model checking (Problem 1) or synthesis (Problem 2) on the finite abstract model for formula ϕ. The approaches to model checking and synthesis for IMCs and BMDPs as well as the final results depend on the interpretation of these abstract models. There are two semantic interpretations for IMCs, each with its own notion of paths and probability measures. The first views an IMC as a family of (possibly uncountably many) MCs [37], where each member of the family has transition probabilities within the interval ranges defined by the IMC. The second views IMCs (and similarly BMDPs) as generalized MDPs in which the uncertainty over the transition probabilities is resolved through non-determinism. That is, a transition probability distribution that respects the bounds is chosen every time a state is visited and then the next transition is chosen from that distribution.

The model checking and synthesis techniques that we develop are based on the MDP view of IMCs and BMDPs. Our algorithms compute exact bounds of the probabilities of satisfaction for each state of the IMC and BMDP at low computational costs. If larger than desired, the distance between the bounds is then reduced by a refinement process that specifically targets the regions which cause large distances.

III. Preliminaries

A. Markov Models

For a finite set Q, we use $|Q|$ and 2^Q to denote its cardinality and power set, respectively.

Definition 1 (MDP): A Markov Decision Process (MDP) is a tuple $\mathcal{D} = (Q, q_0, Act, Steps, \Pi, L)$, where:

- Q is a finite set of states;
- $q_0 \in Q$ is the initial state;
- Act is a finite set of actions;
- $Steps : Q \to 2^{Act \times \Sigma(Q)}$ is a transition probability function, where $\Sigma(Q)$ is the set of all discrete probability distributions over the set Q;
- Π is a finite set of atomic propositions;
- $L : Q \to 2^{\Pi}$ is a labeling function assigning to each state possibly several elements of Π.

The set of actions available at $q \in Q$ is denoted by $\mathcal{A}(q)$. The function $Steps$ is often represented as a matrix with $|Q|$ columns and $\sum_{i=0}^{|Q|-1} |\mathcal{A}(q_i)|$ rows. We denote the probability of transitioning from state q_i to state q_j under action $a \in \mathcal{A}(q_i)$ as $\sigma_a^{q_i}(q_j)$ and the corresponding probability distribution as $\sigma_a^{q_i}$. A path ω through an MDP is a sequence of states $\omega = q_0 \xrightarrow{(a_0, \sigma_{a_0}^{q_0}(q_1))} q_1 \xrightarrow{(a_1, \sigma_{a_1}^{q_1}(q_2))} \ldots q_i \xrightarrow{(a_i, \sigma_{a_i}^{q_i}(q_{i+1}))} q_{i+1} \ldots$, where each transition is induced by a choice of action at the current step $i \geq 0$. We denote the ith state of a path ω by $\omega(i)$ and the set of all finite and infinite paths by $Path^{fin}$ and $Path$, respectively.

A control policy defines a choice of action at each state of MDP. Its formal definition follows.

Definition 2 (Control Policy): A control policy μ of an MDP model $\mathcal{D}$ is a function mapping a finite path $\omega^{fin} = q_0q_1q_2 \ldots q_n$ of $\mathcal{D}$ onto an action in $\mathcal{A}(q_n)$. In other words, a policy is a function $\mu : Path^{fin} \to Act$ that specifies for every finite path, the next action to be applied. If a control policy depends only on the last state of ω^{fin}, it is called a stationary policy.

For each policy μ, a probability measure $Prob_\mu$ over the set of all paths (under μ) $Path_\mu$ is induced by the resulting MC.

Definition 3 (IMC): An IMC is a tuple $\mathcal{I} = (Q, q_0, \check{P}, \hat{P}, \Pi, L)$, where Q, q_0, Π, and L are as in Def. 1, and

- $\check{P} : Q \times Q \to [0,1]$ is a function, where $\check{P}(q,q')$ defines the lower bound of the transition probability from state q to state q';
- $\hat{P} : Q \times Q \to [0,1]$ is a function, where $\hat{P}(q,q')$ defines the upper bound of the transition probability from state q to state q';

For all $q, q' \in Q$, $\check{P}$ and $\hat{P}$ satisfy $\check{P}(q,q') \leq \hat{P}(q,q')$, and, for all states with outgoing transitions, $\sum_{q' \in Q} \check{P}(q,q') \leq 1 \leq \sum_{q' \in Q} \hat{P}(q,q')$. An IMC becomes a Markov chain (MC) when $\check{P} = \hat{P} = P$. In this case, $P(q,\cdot)$ is the precise transition probability distribution of state q over Q.

With a small abuse of notation, we define the set $Steps(q)$ of probability distributions over Q as

$$Steps(q) = \left\{ \sigma^q : Q \to \mathbb{R}^{\geq 0} | \sum_{q' \in Q} \sigma^q(q') = 1 \; \& \right.$$
$$\left. \check{P}(q,q') \leq \sigma^q(q') \leq \hat{P}(q,q') \forall q' \in Q \right\}.$$

In the MDP interpretation of an IMC, at every state $q \in Q$, a probability distribution σ^q is chosen nondeterministically from the set $Steps(q)$. A successor state q' is then chosen according to σ^q over Q. We assume that the nondeterministic choice of σ^q is made by an *adversary*.

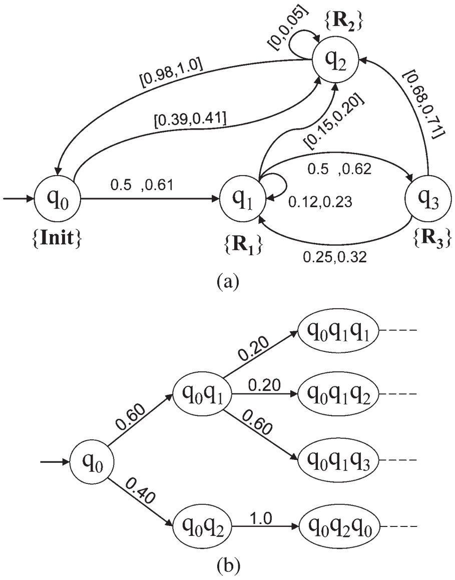
Definition 4 (Adversary): An adversary ν of an IMC model $\mathcal{I}$ is a function mapping a finite path $\omega^{fin} = q_0q_1q_2 \ldots q_n$ of $\mathcal{I}$ onto an element of $Steps(q_n)$. That is, an adversary is a function $\nu : Path^{fin} \to Steps$ that specifies for every finite path, the next distribution to be applied.

From a game-theoretic point of view, an adversary is the opponent's strategy while a control policy is ours. Since there is no choice of control in IMCs, an adversary ν induces a probability measure $Prob_\nu$ over the set of all paths (under ν) $Path_\nu$.

Example 1: A simple IMC is shown in Fig. 2(a) with $Q = \{q_0, q_1, q_2, q_3\}$. The labels are $L(q_0) = \{\mathbf{Init}\}$, $L(q_1) = \{\mathbf{R_1}\}$, $L(q_2) = \{\mathbf{R_2}\}$, and $L(q_3) = \{\mathbf{R_3}\}$. The transition probability bounds are

$$\check{P} = \begin{pmatrix} 0 & 0.58 & 0.39 & 0 \\ 0 & 0.12 & 0.15 & 0.57 \\ 0.98 & 0 & 0 & 0 \\ 0 & 0.25 & 0.68 & 0 \end{pmatrix}$$

$$\hat{P} = \begin{pmatrix} 0 & 0.61 & 0.41 & 0 \\ 0 & 0.23 & 0.20 & 0.62 \\ 1.0 & 0 & 0.05 & 0 \\ 0 & 0.32 & 0.71 & 0 \end{pmatrix}.$$



(a)

(b)

Fig. 2. (a) A four-state IMC $\mathcal{I}$ with the range of the transition probabilities shown in brackets. (b) Fragment of infinite-state Markov chain $\mathcal{I}_{\nu_1}$ for adversary ν_1. (a) IMC $\mathcal{I}$; (b) Fragment of MC $\mathcal{I}_{\nu_1}$.

An example of an adversary for this IMC is ν_1, defined by the mapping

$$\nu_1(\cdots q_0) = (0, 0.60, 0.40, 0)^T$$
$$\nu_1(\cdots q_1) = (0, 0.20, 0.20, 0.60)^T$$
$$\nu_1(\cdots q_2) = (1.0, 0, 0, 0)^T$$
$$\nu_1(\cdots q_3) = (0, 0.30, 0.70, 0)^T$$

where $\cdots q_i$ denotes any finite path terminating in q_i. The initial fragment of the resulting MC is shown in Fig. 2(b). From this fragment, it is easy to see that the probability of the finite path $q_0q_1q_2$ is $Prob_{\nu_1}(q_0q_1q_1) = 0.12$.

An IMC that allows the inclusion of a finite set of actions at each state is called a Bounded-parameter Markov Decision Process (BMDP) [10]. Like MDPs, BMDPs provide a modeling framework for decision making, and, like IMCs, they include uncertainty over the transition probabilities. We formally define a BMDP as follows.

Definition 5 (BMDP): A BMDP is a tuple $\mathcal{B} = (Q, q_0, Act, \widecheck{Steps}, \widehat{Steps}, \Pi, L)$, where Q, q_0, Act, Π, and L are as in Def. 1, and

- $\widecheck{Steps} : Q \times Act \times Q \to [0,1]$ is a function where $\widecheck{Steps}(q,a,q')$ gives the lower bound of the transition probability from state q to state q' under action $a \in \mathcal{A}(q)$;
- $\widehat{Steps} : Q \times Act \times Q \to [0,1]$ is a function where $\widehat{Steps}(q,a,q')$ gives the upper bound of the transition probability from state q to state q' under action $a \in \mathcal{A}(q)$.

The functions $\widecheck{Steps}$ and $\widehat{Steps}$ can be represented as matrices with $|Q|$ columns and $\sum_{i=0}^{|Q|-1} |\mathcal{A}(q_i)|$ rows. Moreover, for all $q, q' \in Q$ and any $a \in \mathcal{A}(q)$, they satisfy $\widecheck{Steps}(q,a,q') \leq \widehat{Steps}(q,a,q')$, and $\sum_{q' \in Q} \widecheck{Steps}(q,a,q') \leq 1 \leq \sum_{q' \in Q} \widehat{Steps}(q,a,q')$. A BMDP becomes an MDP if $\widecheck{Steps} = \widehat{Steps} = Steps$.

To trace a path through a BMDP, a policy and an adversary are needed since they determine the action and the corresponding probability distribution at every state, respectively. Let $\eta = (\mu, \nu)$ denote a given policy μ and adversary ν. Analogous to MDPs and IMCs, a probability measure $Prob_\eta$ is induced over the set of all paths of a BMDP under η.

B. Probabilistic Computation Tree Logic (PCTL)

We use PCTL [7] to write specifications of IMCs and BMDPs.

Definition 6 (Syntax of PCTL): Formulas in PCTL can be recursively defined as follows:

$$\phi ::= true|\pi|\neg\phi|\phi \wedge \phi|\mathcal{P}_{\bowtie p}[\psi] \qquad \text{(state formulas)}$$

$$\psi ::= X\phi|\phi\,\mathcal{U}^{\leq k}\phi|\phi\,\mathcal{U}\phi \qquad \text{(path formulas)}$$

where $\pi \in \Pi$ is an atomic proposition, $\neg$ (negation) and $\wedge$ (conjunction) are Boolean operators, $\bowtie \in \{\leq, <, \geq, >\}$, $p \in [0, 1]$, $k \in \mathbb{N}$, and X ("next"), $\mathcal{U}^{\leq k}$ ("bounded until"), and $\mathcal{U}$ ("until") are temporal operators.

State formulas ϕ are evaluated over the states of a BMDP while path formulas ψ are assessed over paths and only allowed inside the $\mathcal{P}_{\bowtie p}[\psi]$ operator. Intuitively, a state q satisfies $\mathcal{P}_{\bowtie p}[\psi]$ if the probability of all paths from q satisfying ψ is in the range $\bowtie p$.

Definition 7 (Semantics of PCTL): For any state $q \in Q$, the satisfaction relation $\models$ is defined inductively as follows.

- $q \models true$ for all $q \in Q$;
- $q \models \pi \Leftrightarrow \pi \in L(q)$;
- $q \models (\phi_1 \wedge \phi_2) \Leftrightarrow (q \models \phi_1) \wedge (q \models \phi_2)$;
- $q \models \neg\phi \Leftrightarrow q \not\models \phi$;
- $q \models \mathcal{P}_{\bowtie p}[\psi] \Leftrightarrow p^q_\eta \bowtie p$, where p^q_η is the probability of all the infinite paths that start from q and satisfy ψ under all η, i.e., under all pairs of policies and adversaries.

Moreover, for any path $\omega \in Path$:

- $\omega \models X\phi \Leftrightarrow \omega(1) \models \phi$;
- $\omega \models \phi_1 \mathcal{U}^{\leq k} \phi_2 \Leftrightarrow \exists\, i \leq k, \;\; \omega(i) \models \phi_2 \wedge \omega(j) \models \phi_1 \forall j \in [0, i)$;
- $\omega \models \phi_1 \mathcal{U} \phi_2 \Leftrightarrow \exists\, i \geq 0, \qquad \omega(i) \models \phi_2 \wedge \omega(j) \models \phi_1 \forall j \in [0, i)$.

We define the path formula operators $\Diamond^{\leq k}$ ("bounded eventually") and $\Diamond$ ("eventually") as

$$\mathcal{P}_{\bowtie p}[\Diamond^{\leq k}\phi] \equiv \mathcal{P}_{\bowtie p}[true\, \mathcal{U}^{\leq k}\phi],\; \mathcal{P}_{\bowtie p}[\Diamond\phi] \equiv \mathcal{P}_{\bowtie p}[true\, \mathcal{U}\phi]$$

respectively. Intuitively, $\Diamond^{\leq k}\phi$ means that ϕ is satisfied within k time units. Similarly, $\Diamond\phi$ means that ϕ is satisfied at some point in the future.

C. Polytopes and Their Pre and Post Images

Let $n \in \mathbb{N}$ and consider the n-dimensional Euclidean space $\mathbb{R}^n$. A full dimensional *polytope* P is defined as the convex hull of at least $n + 1$ affinely independent points in $\mathbb{R}^n$. The *set of vertices* of P is the set of points $v^{\mathrm{P}}_1, \ldots, v^{\mathrm{P}}_{m_{\mathrm{P}}} \in \mathbb{R}^n$, $m_{\mathrm{P}} \geq n + 1$, whose convex hull gives P and with the property that, for any $i = 1, \ldots, m_{\mathrm{P}}$, point v^{P}_i is not in the convex hull of the remaining points $v^{\mathrm{P}}_1, \ldots, v^{\mathrm{P}}_{i-1}, v^{\mathrm{P}}_{i+1}, \ldots, v^{\mathrm{P}}_{m_{\mathrm{P}}}$. A polytope is completely described by its set of vertices

$$\mathrm{P} = conv\left(v^{\mathrm{P}}_1, \ldots, v^{\mathrm{P}}_{m_{\mathrm{P}}}\right) \tag{2}$$

where $conv$ denotes the convex hull. Alternatively, P can be described as the intersection of at least $n + 1$ closed half spaces. In other words, there exists a $t \geq n + 1$, $h_i \in \mathbb{R}^n$, and $l_i \in \mathbb{R}$, $i = 1, \ldots, t$ such that

$$\mathrm{P} = \left\{x \in \mathbb{R}^n | h_i^T x \leq l_i, i = 1, \ldots, t\right\}. \tag{3}$$

The above definition can be written as the matrix inequality $Hx \leq L$, where $H \in \mathbb{R}^{t \times n}$ and $L \in \mathbb{R}^t$. Forms (2) and (3) are referred to as V- and H-representations of the polytope, respectively. Below, we formally define polyhedral operators PRE, PRER, and POST that are used in our refinement algorithm in Section VI.

Given stochastic system (1) in polytope P with a choice of dynamics given by $a \in \mathrm{I}_a$, we define $\mathrm{PRE}(\mathrm{P}|a)$ as the set of all points that make a transition to P under dynamics a for *some* values of w in one time step, i.e., all points that have a non-zero transition probability to P

$$\mathrm{PRE}(\mathrm{P}|a) = \{x \in \mathrm{P} | \mathcal{F}_a(x, w) \in \mathrm{P} \text{ for some } w \in W\}. \tag{4}$$

Similarly, we define $\mathrm{PRER}(\mathrm{P}|a)$ (robust PRE) to be the set of all points that make a transition to P under dynamics a in one step for *all* possible values of w, that is, all those points that make a transition to P with probability 1

$$\mathrm{PRER}(\mathrm{P}|a) = \{x \in \mathrm{P} | \mathcal{F}_a(x, w) \in \mathrm{P} \text{ for all } w \in W\}. \tag{5}$$

We define $\mathrm{POST}(\mathrm{P}|a)$ as the set of points that can be reached from P under dynamics a in one step for some value of w

$$\mathrm{POST}(\mathrm{P}|a)=\{x \in \mathbb{R}^n | x = \mathcal{F}_a(x', w) \text{ for some } x' \in \mathrm{P}, w \in W\}. \tag{6}$$

As an example, consider a linear system of the form $x_{k+1} = Ax_k + w_k$, with $A \in \mathbb{R}^{n \times n}$ invertible. For polytope $\mathrm{P} = conv(v^{\mathrm{P}}_1, \ldots, v^{\mathrm{P}}_{m_{\mathrm{P}}})$ with matrix form $Hx \leq L$ and polytope $W = conv(v^W_1, \ldots, v^W_{m_W})$, the above sets are convex polytopes that can be computed as [38]

$$\begin{aligned}
\mathrm{PRE}(\mathrm{P}|a) &= conv\left(\left\{A^{-1}\left(v^{\mathrm{P}}_i - v^W_j\right), 1 \leq i \leq m_{\mathrm{P}}, 1 \leq j \leq m_W\right\}\right)\\
\mathrm{POST}(\mathrm{P}|a) &= conv\left(\left\{Av^{\mathrm{P}}_i + v^W_j, 1 \leq i \leq m_{\mathrm{P}}, 1 \leq j \leq m_W\right\}\right)\\
\mathrm{PRER}(\mathrm{P}|a) &= \{x \in \mathbb{R}^n | H_R x \leq L_{R_i}, i = 1, \ldots, m_W\}
\end{aligned} \tag{7}$$

where $H_R = HA$ and $L_{R_i} = L - Hv^W_i$. Therefore, the complexities of these operations on linear systems are polynomial in time, i.e., $\mathcal{O}(t^3 m^3_W)$ in the worst case.

IV. Abstraction

In this section, we describe our method of generating an IMC (when $m_a = 1$) or a BMDP (when $m_a > 1$) for the evolution of system (1) in the polytopic domain P with a partition given by a set of linear predicates. The abstraction method is inspired by [11], [26], but instead of using approximate Markov models and quantifying the error of abstraction conservatively by a Lipschitz condition, we compute exact bounds for the transition probabilities between the regions defined by the partition and employ an IMC or a BMDP to incorporate the uncertainty in the abstraction itself. The IMC and BMDP models capture all the possible probability distributions between the regions, and thus the true distribution of the original system is a member of the distribution set represented by the abstraction model. As a result, if a specification is satisfied by all the distributions of the IMC or BMDP, it is also satisfied by the distribution of the underlying stochastic system. Therefore, with these abstractions, the problem of verification (and synthesis) of the infinite-state system (1) with respect to a given specification is reduced to verification (and synthesis) of the finite-state IMC (or BMDP).

A. IMC Abstraction

Here, we consider the abstraction of system (1) with only one dynamics ($m_a = 1$) into an IMC. We start by defining a polytopic partition $Q = \{q_1, q_2, \ldots, q_{m_Q}\}$, $m_Q \in \mathbb{N}^+$, induced by the linear predicates from the set Π. The set Q is the set of states of the IMC. Through a small abuse of notation, we use q_i to describe both the polytope and the symbol labeling that polytope so that Q represents also a set of discrete states for the IMC. The exact meaning of q_i should be clear from the context.

To compute the transition probability bounds for the IMC, we assume that the stochastic system evolves in P with a Borel-measurable stochastic kernel given by $T : \mathbb{B}(\mathrm{P}) \times \mathrm{P} \to [0, 1]$, where $\mathbb{B}(\mathrm{P})$ is a class of Borel sets in P. The stochastic kernel assigns to each point $x \in \mathrm{P}$ a probability measure $T(\cdot|x)$ on the Borel space $(\mathrm{P}, \mathbb{B}(\mathrm{P}))$ and is determined by (1).

We denote the one-step transition probability from point $x \in q_i$ to polytope q_j by $pr_x(q_j|q_i)$. This transition probability depends on the kernel T and can be obtained by marginalizing over each polytope in the set Q

$$pr_x(q_j|q_i) = Prob(x' \in q_j|x \in q_i) = \int_{q_j} T(dx'|x \in q_i) \quad (8)$$

where $x, x' \in \mathrm{P}$ are related by the system dynamics. Since the probability depends on the initial point $x \in q_i$, there exist possibly infinitely many transition probabilities from region q_i to region q_j. Their values are bounded within the interval $[\check{P}(q_i, q_j), \hat{P}(q_i, q_j)]$, where

$$\check{P}(q_i, q_j) = \min_{x \in q_i} \int_{q_j} T(dx'|x \in q_i) \quad (9)$$

$$\hat{P}(q_i, q_j) = \max_{x \in q_i} \int_{q_j} T(dx'|x \in q_i). \quad (10)$$

It should be noted that the bounds $\check{P}(q_i, q_j)$ and $\hat{P}(q_i, q_j)$ are achievable, that is, there exist states which exactly have these transition probabilities. We use (9) and (10) to construct the lower and upper bound transition probability matrices $\check{P}$ and $\hat{P}$, respectively, of the IMC $\mathcal{I}$.

The computability of the above probabilities depends on the dynamics of the system, the distribution of noise, and the geometry of regions q_i and q_j. For particular system dynamics and noise distributions, closed-form solutions can be derived for general convex regions. For instance, analytical solutions can be derived in a straightforward manner for systems with linear dynamics and unbounded normal noise distribution. However, for linear systems with bounded noise distributions, the above probabilities become highly sensitive on the geometries of q_i, q_j, and W. In the worst case, though, the stochastic kernels can be computed using approximate methods such as particle filters, and the integrals can be evaluated by numerical methods.

B. BMDP Abstraction

We now consider the abstraction of system (1) when there are multiple available dynamics ($m_a > 1$). As in the previous case, we use Q as the set of states for the BMDP. The set of dynamics labels I_a becomes the set of actions of the BMDP. The transition probability bounds under each action are constructed as follows. For each $x \in \mathrm{P}$ and action a_l, the stochastic kernel $T(\cdot|x, a_l)$ is defined from the dynamics of system (1) as in the single action case above. The one-step transition probability from point $x \in q_i$ to region q_j under action a_l is given by

$$\begin{aligned} pr_x(q_j|q_i, a_l) &= Prob(x' \in q_j|x \in q_i, a = a_l) \\ &= \int_{q_j} T(dx'|x \in q_i, a_l) \end{aligned} \quad (11)$$

where again $x, x' \in \mathrm{P}$ are related by the system dynamics. The range of transition probabilities from q_i to q_j is defined by

$$\widetilde{Steps}(q_i, a_l, q_j) = \min_{x \in q_i} \int_{q_j} T(dx'|x \in q_i, a_l)$$

$$\widehat{Steps}(q_i, a_l, q_j) = \max_{x \in q_i} \int_{q_j} T(dx'|x \in q_i, a_l).$$

V. Verification

As discussed above, the true distribution of the continuous-domain stochastic system from any initial state x_0 is contained within the corresponding interval defined by the IMC or BMDP abstraction. Thus, if a specification is satisfied by the distribution bounds of the abstraction model, it is also satisfied by the distribution of the underlying stochastic system. In this section we consider the model checking problem of the abstraction model as formally stated below.

Problem 3 (IMC Model Checking): Given an IMC $\mathcal{I}$ and a PCTL formula $\phi = \mathcal{P}_{\bowtie p}[\psi]$, find the sets of states that definitely (denoted as Q^{yes}), possibly ($Q^?$), and never (Q^{no}) satisfy ϕ.

Problem 4 (BMDP Model Checking): Given a BMDP $\mathcal{B}$ and a PCTL formula ϕ, find the sets of states that definitely, possibly, and never satisfy ϕ for all control policies.

The satisfaction of a path formula ψ from a state of $\mathcal{I}$ is necessarily given as a probability range due to the transition probability intervals. In model checking of $\mathcal{I}$, we need to check whether this range satisfies the bound in the formula; that is whether the probability $p^{q_i}(\psi)$ of all the paths that start from a given state q_i and satisfy the path formula ψ satisfies the bound $\bowtie p$ for all adversaries. To do this, one needs to compute the probability of satisfaction for the extreme scenarios. Thus, to solve Problem 3, one needs to find the adversaries that give rise to the minimum and maximum probability of satisfaction of path formula ψ from state $q_i \in Q$, denoted by $\check{p}^{q_i}(\psi)$ and $\hat{p}^{q_i}(\psi)$, respectively.

To solve Problem 4, one needs to compute the optimal control policies in addition to the adversaries. Recall that under a given control policy, a BMDP becomes an IMC. Thus, the probability interval for satisfying ϕ depends on both the choice of control policy and the adversary under the policy. For model checking purposes, we need to verify that the range of the probabilities of satisfaction of ϕ from each state of $\mathcal{B}$ respects the bound in the formula under all policies and adversaries (i.e., $p_\eta^{q_i}(\psi) \bowtie p$ for all η). Therefore, we seek the minimum lower-bound ($\check{p}_{\min}^{q_i}$ denoting $p_\eta^{q_i}$ with $\eta = (\mu_{\min}, \nu_{\min})$) and the maximum upper-bound ($\hat{p}_{\max}^{q_i}$ denoting $p_\eta^{q_i}$ with $\eta = (\mu_{\max}, \nu_{\max})$) of these probabilities to determine Q^{yes}, $Q^{?}$, and Q^{no} with respect to the relational operator $\bowtie p$.

In the following subsections, we first describe a method that can be used to find the minimizing and maximizing adversaries. This method includes the computation of the true transition probability functions for the states of $\mathcal{I}$ (state-action pairs of $\mathcal{B}$) that is maximizing with respect to an ordering of the states while respecting the probability bounds. Then, for each PCTL temporal operator, we describe a model checking algorithm which employs the order-maximizing transition probability functions to compute the adversaries and control policies that give rise to the minimum lower bound and maximum upper bound of the probability of satisfaction of a path formula. As in MDPs, model checking of IMCs and BMDPs against nested formulas with multiple temporal operators is performed through recursive calls of the corresponding algorithms. For such formulas, satisfaction from the states in $Q^?$ is unclear, and, for reasons described in Section VI, we treat these states as not satisfying to provide a guarantee of satisfaction at the user-specified degree of conservatism.

A. O-Maximizing Transition Probability Function

The minimizing and maximizing adversaries can be obtained iteratively through an ordering of the states of $\mathcal{I}$ [10], [39]. Recall that the number of true distributions captured by $\mathcal{I}$ is in general uncountable. Let O denote an ordering of all the states in Q. The works in [10], [39] showed that there is a unique transition probability density that maximizes, for every state $q \in Q$, the expected "position in the ordering" of the state reached from q. Conceptually, the true transition probability function that for every q allocates as much probability mass as possible to the states early in the ordering O is maximizing with respect to O. This implies that by choosing O to be descending with respect to the probability of satisfaction of a path formula ψ, the transition probability function that is maximizing with respect to satisfaction of ψ can be computed. Similarly, if O is ascending with respect to the probability of satisfaction of ψ, the transition probability function that is "ψ-minimizing" can be obtained.

Let $O = o_1, o_2, \ldots, o_{|Q|}$, where $o_i \in Q$, and r_O be the index $1 \leq r_O \leq |Q|$ which maximizes the following expression without letting it exceed one:

$$\sum_{i=1}^{r_O-1} \hat{P}(q, o_i) + \sum_{i=r_O}^{n} \check{P}(q, o_i).$$

Then, the O-maximizing transition probability function is given by

$$P_O(q, o_i) = \begin{cases} \hat{P}(q, o_i) & \text{if } i < r_O \\ \check{P}(q, o_i) & \text{if } i > r_O \end{cases}$$

$$P_O\left(q, o_{r_O}\right) = 1 - \sum_{j=1, j\neq r_O}^{n} P_O(q, o_j). \tag{12}$$

Algorithmically, the O-maximizing distribution $P_O(q, \cdot)$ can be computed iteratively over O. In each iteration, first the remaining probability mass to be assigned, denoted by $\bar{p}_{rem}$, is determined according to $\bar{p}_{rem} = 1 - (\sum_{i=1}^{\alpha-1} P_O(q, o_i) + \sum_{i=\alpha}^{|Q|} \check{P}(q, o_i))$, where α is the iteration counter. Then, $P_O(q, o_\alpha)$ is set to $\check{P}(q, o_\alpha) + \bar{p}_{rem}$. In the case that $\check{P}(q, o_\alpha) + \bar{p}_{rem}$ exceeds the upper bound of the transition probability, then $P_O(q, o_\alpha) = \hat{P}(q, o_\alpha)$.

Example 2: To demonstrate this concept, consider the IMC in Fig. 2(a) and the ordering $O = q_0, q_1, q_2, q_3$. The O-maximizing transition probability distribution at q_0 is $P_O(q_0, \cdot) = (0\ 0.61\ 0.39\ 0)^T$.

As mentioned above, if O is an ascending ordering of the states of Q with respect to the probability of satisfaction of ψ, P_O becomes minimizing with respect to the probability of achieving ψ. We denote this transition probability function by $P^\downarrow$. Similarly, $P^\uparrow$ denotes the transition probability function that is ψ-maximizing. In the case of BMDPs, O-maximizing transition probability functions can be computed for each state-action pair. This can be achieved with the same algorithm described above. We denote the ψ-minimizing and ψ-maximizing transition probability functions of BMDPs by $Steps^\downarrow$ and $Steps^\uparrow$, respectively. Note that, these transition probability functions, as with their counterparts (e.g., P in MCs and $Steps$ in MDPs), can be represented as matrices, and the complexity of their computation is polynomial in the size of the IMC/BMDP, i.e., $\mathcal{O}(|Q|^2)$ for IMCs and $\mathcal{O}(|Q| \sum_{i=0}^{|Q|-1} |\mathcal{A}(q_i)|)$ for BMDPs.

B. Model Checking Algorithms

In this section, we present BMDP model checking algorithms for each temporal operator. These algorithms can be also used for model checking of IMCs since an IMC is a BMDP with one action per state.

1) Next Operator—$\phi = \mathcal{P}_{\bowtie p}[X\phi_1]$: For this operator, we need to compute the minimum lower bound and maximum upper bound of the probabilities of satisfying $\psi = X\phi_1$ at each state. Let $\check{p}_{\min}^{q_i}(\psi)$ and $\hat{p}_{\max}^{q_i}(\psi)$ denote these satisfying probability bounds from $q_i \in Q$. For $\psi = X\phi_1$, we can compute these probabilities by considering only the one-step transitions at each state-action pair. That is

$$\check{p}_{min}^{q_i}(\psi) = \min_{a\in\mathcal{A}(q_i)} \min_{\sigma_a^{q_i}\in Steps(q_i,a)} \sum_{q_j\in Sat(\phi_1)} \sigma_a^{q_i}(q_j)$$

$$\hat{p}_{max}^{q_i}(\psi) = \max_{a\in\mathcal{A}(q_i)} \max_{\sigma_a^{q_i}\in Steps(q_i,a)} \sum_{q_j\in Sat(\phi_1)} \sigma_a^{q_i}(q_j)$$

where $Sat(\phi_1)$ is the set of states that satisfy ϕ_1. The inner optimization problems can be solved by using the ψ-minimizing and ψ-maximizing transition probability functions $Steps^{\downarrow}$ and $Steps^{\uparrow}$, respectively. Then, through a minimization/maximization step over all actions, the outer optimization problems can be decided.

Let us define a state-indexed vector $\overline{\phi}_1$ with entries $\overline{\phi}_1(q_i)$ equal to 1 if $q_i \models \phi_1$ and 0 otherwise. To compute $\check{p}^{q_i}_{\min}(\psi)$, we first cast $Steps^{\downarrow}(q_i, \cdot)$ through an ascending state ordering O with respect to the values of $\overline{\phi}_1$ for each $q_i \in Q$. Then, we multiply $Steps^{\downarrow}$ by $\overline{\phi}_1$. The result is a vector whose entries are the lower-bound probabilities of satisfying $X\phi_1$ where each row corresponds to a state-action pair. We select the minimum probabilities at each state and save the corresponding action. Similarly, we compute the upper bound for the probabilities of satisfying $X\phi_1$ by multiplying the matrix $Steps^{\uparrow}$ by $\overline{\phi}_1$. Then, we perform a maximization step for the probabilities corresponding to each state. We determine Q^{yes}, $Q^{?}$, and Q^{no} by comparing the optimal bounds with $\bowtie p$. If $\check{p}^{q_i}_{\min}(\psi) \le p \le \hat{p}^{q_i}_{\max}(\psi)$, then $q_i \in Q^{?}$, indicating that there may be a subregion of polytope q_i that satisfies the formula. If $p \notin [\check{p}^{q_i}_{\min}(\psi), \hat{p}^{q_i}_{\max}(\psi)]$, then q_i belongs either to Q^{yes} or Q^{no}. The complexity of this algorithm is two matrix-vector multiplications followed by a one dimensional search in addition to the polynomial computations of $Steps^{\downarrow}$ and $Steps^{\uparrow}$. Therefore, the total computational complexity is polynomial for this algorithm.

Example 3: To demonstrate this method, consider a four-state BMDP with

$$\widetilde{Steps} = \begin{array}{c} \\ q_0;a_1 \\ \hline q_1;a_1 \\ q_1;a_2 \\ \hline q_2;a_1 \\ q_2;a_2 \\ \hline q_3;a_1 \end{array} \left(\begin{array}{cccc} 0 & 0.05 & 0 & 0 \\ \hline 0 & 0.12 & 0.15 & 0.57 \\ 0 & 0 & 0.50 & 0.44 \\ \hline 0 & 0 & 1 & 0 \\ 0.98 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 1 \end{array}\right)$$

$$\widehat{Steps} = \begin{array}{c} q_0;a_1 \\ \hline q_1;a_1 \\ q_1;a_2 \\ \hline q_2;a_1 \\ q_2;a_2 \\ \hline q_3;a_1 \end{array} \left(\begin{array}{cccc} 0.05 & 1 & 0 & 0 \\ \hline 0 & 0.23 & 0.20 & 0.62 \\ 0 & 0 & 0.56 & 0.50 \\ \hline 0 & 0 & 1 & 0 \\ 1 & 0 & 0.05 & 0 \\ \hline 0 & 0 & 0 & 1 \end{array}\right)$$

and labels $L(q_0) = \{\mathbf{Init}\}$, $L(q_1) = \emptyset$, $L(q_2) = \{\mathbf{R_2}\}$, and $L(q_3) = \{\mathbf{R_3}\}$. Let $\phi = \mathcal{P}_{\le 0.40}[X\mathbf{R_2}]$. The property $\mathbf{R_2}$ is satisfied at state q_2; thus, $\overline{\mathbf{R_2}} = (0\ 0\ 1\ 0)^T$. Then, $Steps^{\downarrow} \cdot \overline{\mathbf{R_2}}$ and $Steps^{\uparrow} \cdot \overline{\mathbf{R_2}}$ are, respectively, equal to

$$Steps^{\downarrow} \cdot \overline{\mathbf{R_2}} = \left(\begin{array}{c} 0 \\ \hline 0.15 \\ 0.50 \\ \hline 1 \\ 0 \\ \hline 0 \end{array}\right), \quad Steps^{\uparrow} \cdot \overline{\mathbf{R_2}} = \left(\begin{array}{c} 0 \\ \hline 0.20 \\ 0.56 \\ \hline 1 \\ 0.02 \\ \hline 0 \end{array}\right).$$

By performing a minimization on $Steps^{\downarrow} \cdot \overline{\mathbf{R_2}}$ and a maximization on $Steps^{\uparrow} \cdot \overline{\mathbf{R_2}}$, we find the approximate probability interval of satisfying $X\mathbf{R_2}$ from q_0, q_1, q_2, and q_3 to be [0, 0], [0.15, 0.56], [0, 1], and [0, 0], respectively. The comparison of these values with probability bound $\le$0.40 indicated in ϕ results in $Q^{yes} = \{q_0, q_3\}$, $Q^{?} = \{q_1, q_2\}$, and $Q^{no} = \emptyset$.

2) Bounded Until Operator—$\phi = \mathcal{P}_{\bowtie p}[\phi_1 \mathcal{U}^{\le k} \phi_2]$: Here, we are interested in probabilities $\check{p}^{q_i}_{\min}(\phi_1 \mathcal{U}^{\le k} \phi_2)$ and $\hat{p}^{q_i}_{\max}(\phi_1 \mathcal{U}^{\le k} \phi_2)$. To do this, we first group the BMDP states into three subsets: states that always satisfy the specification, $Q^1 = Sat(\phi_2)$, states that never satisfy the specification $Q^0 = Q \setminus (Sat(\phi_1) \cup Sat(\phi_2))$, and the remaining states $Q^{\dagger} = Q \setminus (Q^1 \cup Q^0)$. Trivially, the probabilities of the states in Q^1 and in Q^0 are 1 and 0 respectively. The lower bound probabilities of the remaining states $q_i \in Q^{\dagger}$ are defined recursively as following:

$$\check{p}^{q_i}_{\min}(\phi_1 \mathcal{U}^{\le k} \phi_2) = \begin{cases} 1 & \text{if } q_i \in Q^1 \\ 0 & \text{if } q_i \in Q^0 \\ 0 & \text{if } q_i \in Q^{\dagger} \,\&\, k = 0 \\ \check{z}^k_{\min}(q_i) & \text{if } q_i \in Q^{\dagger} \,\&\, k > 0 \end{cases}$$

where

$$\check{z}^k_{\min}(q_i) = \min_{a \in \mathcal{A}(q_i)} \min_{\sigma^{q_i}_a \in Steps(q_i, a)} \left(\sum_{q_j \in Q^{\dagger}} \sigma^{q_i}_a(q_j) \right.$$
$$\left. \times \check{p}^{q_j}_{\min}(\phi_1 \mathcal{U}^{\le k-1} \phi_2) + \sum_{q_j \in Q^1} \sigma^{q_i}_a(q_j) \right).$$

The upper bound probabilities can be formulated through a similar system of equations where all the minimum operators are replaced by maximum.

We calculate $\check{p}^{q}_{\min}(\phi_1 \mathcal{U}^{\le k} \phi_2)$ and $\hat{p}^{q}_{\max}(\phi_1 \mathcal{U}^{\le k} \phi_2)$ by k matrix-vector multiplications, each similar to the algorithm described for the Next operator (Section V-B1). After each multiplication step, we replace the resultant vector entries corresponding to the states in Q^1 and Q^0 with 1 and 0, respectively. Moreover, in each iteration, the transition probabilities in $Steps^{\downarrow}$ and $Steps^{\uparrow}$ need to be updated according to the new state ordering induced by the satisfying probability bounds computed in the previous step. With each optimization step, we also obtain the optimal action at each state. It should be noted that multiple actions might be attained for a state, each corresponding to a unique time step. We discuss this in more detail below and in Section VII. The computational complexity of this algorithm is also polynomial.

Example 4: To illustrate this algorithm, again consider the BMDP in Example 3 and the formula $\phi = \mathcal{P}_{>0.50}$ $[\neg\mathbf{R_3}\mathcal{U}^{\le 2}\mathbf{R_2}]$. By inspection, we have $Q^1 = \{q_2\}$, $Q^0 = \{q_3\}$, and $Q^{\dagger} = \{q_0, q_1\}$. The first iteration of the above algorithm results in the following: $\check{p}_{\min}(\neg\mathbf{R_3}\mathcal{U}^{\le 1}\mathbf{R_2}) = (0\ 0.15\ 1\ 0)^T$, $\hat{p}_{\max}(\neg\mathbf{R_3}\mathcal{U}^{\le 1}\mathbf{R_2}) = (0\ \ 0.56\ \ 1\ \ 0)^T$. The second iteration yields: $\check{p}_{\min}(\neg\mathbf{R_3}\mathcal{U}^{\le 2}\mathbf{R_2}) = (0.1425\ \ 0.1845\ \ 1\ \ 0)^T$, $\hat{p}_{\max}(\neg\mathbf{R_3}\mathcal{U}^{\le 2}\mathbf{R_2}) = (0.56\ 0.56\ 1\ 0)^T$. Only one state satisfies the formula in two steps or less, $Q^{yes} = \{q_2\}$. The results for q_0 and q_1 are inconclusive and therefore $Q^{?} = \{q_0, q_1\}$ and $Q^{no} = \{q_3\}$.

3) Until Operator—$\phi = \mathcal{P}_{\bowtie p}[\phi_1 \mathcal{U} \phi_2]$: Here, we are interested in finding probabilities $\check{p}^{q_i}_{\min}(\phi_1 \mathcal{U} \phi_2)$ and $\hat{p}^{q_i}_{\max}(\phi_1 \mathcal{U} \phi_2)$. The problem formulation here is the same as the bounded until problem formulation where $k \to \infty$. As in Section V-B2, we approach this problem by grouping the states into three subsets Q^1, Q^0, and $Q^\dagger$. The computation of the probability bounds for the states in $Q^\dagger$ is in fact the maximal reachability probability problem and can be achieved by value iteration [39]. This implies that the probability bounds are guaranteed to converge to fixed values in a finite number of iterations, i.e., there exist $k_1, k_2 < \infty$ such that for all $k'_1 \geq k_1$ and $k'_2 \geq k_2$

$$\check{p}^{q_i}_{\min}(\phi_1 \mathcal{U} \phi_2) = \check{p}^{q_i}_{\min}(\phi_1 \mathcal{U}^{\leq k_1} \phi_2) = \check{p}^{q_i}_{\min}\left(\phi_1 \mathcal{U}^{\leq k'_1} \phi_2\right)$$
$$\hat{p}^{q_i}_{\max}(\phi_1 \mathcal{U} \phi_2) = \hat{p}^{q_i}_{\max}(\phi_1 \mathcal{U}^{\leq k_2} \phi_2) = \hat{p}^{q_i}_{\max}\left(\phi_1 \mathcal{U}^{\leq k'_2} \phi_2\right)$$

Thus, the algorithm presented for bounded until in Section V-B2 can be used to calculate the values of $\check{p}^{q_i}_{\min}(\phi_1 \mathcal{U} \phi_2)$ and $\hat{p}^{q_i}_{\max}(\phi_1 \mathcal{U} \phi_2)$ and their corresponding optimal policies. Note that the terminal condition of this algorithm is now the convergence of the probabilities instead of the number of iterations. The complexity of this algorithm is also polynomial [39].

VI. Reducing Conservatism Through Refinement

We complete our solution to Problem 1 by reducing the conservatism of the model checking result by reducing the uncertainty in the abstraction model through a refinement process driven by the specification (see Fig. 1). This refinement scheme is inspired by the algorithm in [6], in which the state space of the continuous-domain system is refined locally. As a result, the refinement is performed on specific regions, and only the corresponding portion of the abstraction is updated instead of recomputing the whole abstraction model.

A. Refinement for Systems With $m_a = 1$

We focus on reducing the uncertainty by reducing the size of the regions corresponding to $Q^?$, finding subregions that either definitely or never satisfy the formula. This in turn reduces the size of the intervals of the probability of satisfaction. We first determine the regions that have the most effect on the interval size of the probability of satisfaction from state $q_i \in Q^?$. Then, as described in Section VI-A1, we partition those regions into smaller subregions according to a refinement algorithm that exploits the dynamics of the system and the geometry of the regions. This refinement affects not only the transition probability of the target polytope but also those to which it has a transition. After refinement, the transition probability bounds are recomputed. In Section VI-A2, we discuss the relationship between the transition probability intervals and the probability of satisfaction interval and introduce two methods of targeting regions for refinement.

The refinement strategy introduced here is similar to [40], and the algorithm is based on the one in [6], which is designed for systems with uncertain parameters whose ranges are polytopic. That algorithm efficiently constructs a discrete abstraction for such systems while maintaining the equivalence (bisimulation) property through the use of the PRE operator. The stochastic systems considered in this paper can be viewed as such uncertain systems with the extra knowledge of the distribution over the polytopic domain of the uncertain parameter. Therefore, by adapting that algorithm, we inherit the same properties and efficiency. However, since here the goal of the refinement step is to reduce the transition probability intervals in the abstraction model, we modify the algorithm in [6] so that it first searches for the subregions that have transition probability of one (interval of zero) as advocated by [40]. The work in [40], however considers only discrete probabilistic (transition) systems. Here, we provide the method of refinement of an abstraction state that represents a continuous region using polytopic operations on continuous-domain stochastic systems. For further discussion on the properties of the algorithm see [6].

1) Refinement Algorithm: Our algorithm begins by selecting a target region for refinement and then identifying subregions within it that have probability one transitions to adjacent regions. The selection of the target region is decided by the model checking results and transition probability interval size as explained in Section VI-A2. To find the subregions with probability one transitions, we perform the PRER operation [see (5)] on the adjacent regions. The resulting subregions in the selected target all have probability one transitions and are included in the refinement. The POST [see (6)] of the new subregions are then found and used to refine the adjacent regions; this ensures all probability one transitions are maintained. Finally, the remaining portions of the original regions are convexified. If there are no subregions with probability one transitions (i.e., PRER of the adjacent regions returns an empty set), then the algorithm calculates the PRE [see (4)] of the adjacent regions and uses those to refine the target. Finally, if the PRE operation of the adjacent regions returns the target itself, the target can be refined by any refinement strategy (e.g., simple triangulation). We employed Delaunay triangulation in the implementations.

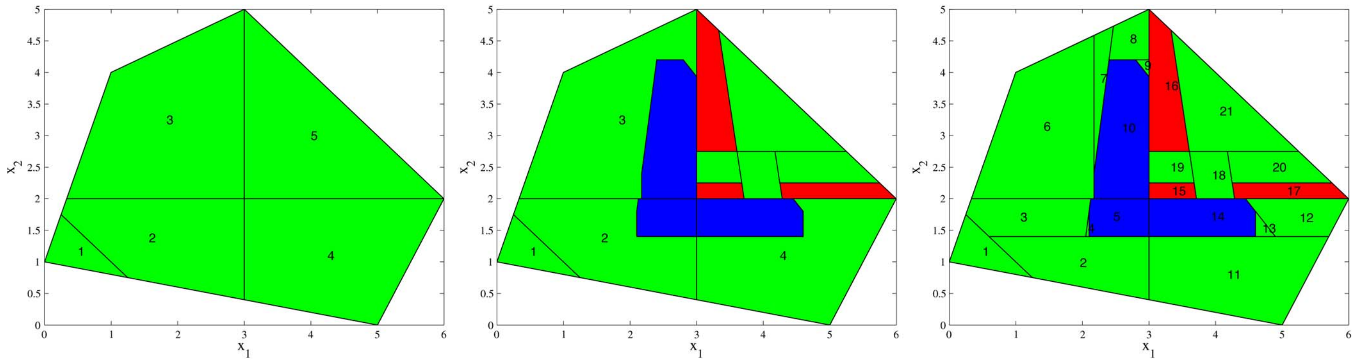
Example 5: An illustration of this refinement method is shown in Fig. 3. Fig. 3(a) shows the domain polytope P partitioned into five subpolytopes. Subpolytope 5 is set as the target region. Applying PRER to subpolytopes 2, 3 and 4 and intersecting the results with the target leads to the red regions in Fig. 3(b). Applying the POST operation to these red regions yields the blue regions. Fig. 3(c) shows the final result after a convexification operation is performed to obtain convex subpolytopes in regions 2, 3, and 4.

2) Interval Size: To close the abstraction and refinement loop (see Fig. 1), we first establish the relationship between the transition probability bounds and the size of the probability of satisfaction interval. Then, we use this relationship together with the refinement algorithm described in Section VI-A1 to design algorithms to reduce the size of the probability of satisfaction intervals for the states in $Q^?$.

Let us denote the largest transition probability interval of the states in Q by

$$\epsilon = \max_{q \in Q} \left\{ \max_{q' \in Q} \left(\hat{P}(q, q') - \check{P}(q, q') \right) \right\}. \tag{13}$$

We also use ψ_k to denote the path formula whose computation of the probability of satisfaction takes k iterations of the corresponding model checking algorithm (e.g., $\psi_k = \phi_1 \mathcal{U}^{\leq k} \phi_2$).



(a) (b) (c)

Fig. 3. Refinement example. (a) The initial partition with Region 5 the target for refinement. (b) Applying the PreR operator yields the subregions of Region 5 with probability one transitions (red regions). The regions these transition to are found using Post (blue regions). (c) Convexification of the regions ensures the final refinement contains only convex subpolytopes.

We define the largest probability interval of satisfaction of path formula ψ_k from the states in Q by

$$I_k = \max_{q \in Q} \left\{ \hat{p}^q(\psi_k) - \check{p}^q(\psi_k) \right\}. \tag{14}$$

The following theorem determines an upper-bound for the growth of I_k over time.

Theorem 1: Consider an IMC $\mathcal{I}$ and a PCTL path formula ψ_k. The k-th step interval size of the probability of satisfaction of the path formula from the states of $\mathcal{I}$ is upper-bounded by

$$I_k \leq \epsilon \sum_{i=1}^{k} N^i \tag{15}$$

where N is the maximum number of outgoing transitions from the states in Q.

Proof: We prove this theorem by induction. For $k = 1$

$$\begin{aligned} I_1 &= \max_{q \in Q} \left\{ \hat{p}^q(\psi_1) - \check{p}^q(\psi_1) \right\} \\ &\leq \max_{q \in Q} \sum_{q' \in Q^1} \left[\hat{P}(q, q') - \check{P}(q, q') \right] \\ &\leq \max_{q \in Q} \left\{ \max_{q' \in Q^1} \left[\hat{P}(q, q') - \check{P}(q, q') \right] N^q(Q^1) \right\} \end{aligned}$$

where Q^1 is the set of states all of whose outgoing paths satisfy the path formula ψ (see Section V-B2), and $N^q(Q^1)$ is the number of outgoing transitions from q to the states in Q^1. Thus, $I_1 \leq \epsilon N$. Next, we assume (15) and show that it holds for $k + 1$. Let $q \in Sat(\psi_k)$ denote that there exists a path from q that satisfies ψ_k. From (14), it follows that:

$$\begin{aligned} I_{k+1} &= \max_{q \in Q} \left\{ \hat{p}^q(\psi_{k+1}) - \check{p}^q(\psi_{k+1}) \right\} \\ &\leq \max_{q \in Q} \sum_{q' \in Sat(\psi_k)} \left[\hat{P}(q, q') \hat{p}^{q'}(\psi_k) - \check{P}(q, q') \check{p}^{q'}(\psi_k) \right] \\ &= \max_{q \in Q} \sum_{q' \in Sat(\psi_k)} \left[\left(\hat{P}(q, q') - \check{P}(q, q') \right) \hat{p}^{q'}(\psi_k) \right. \\ &\qquad \left. + \check{P}(q, q') \left(\hat{p}^{q'}(\psi_k) - \check{p}^{q'}(\psi_k) \right) \right] \\ &\leq \max_{q \in Q} \left\{ \epsilon N^q \left(Sat(\psi_k) \right) + \epsilon N^q \left(Sat(\psi_k) \right) \sum_{i=1}^{k} N^i \right\} \\ &\leq \epsilon \sum_{i=1}^{k+1} N^i. \end{aligned}$$

■

For a given specification ϕ, define the size of the interval of satisfaction for a state q_i to be $I^{q_i} = \hat{p}^{q_i}(\phi) - \check{p}^{q_i}(\phi)$. In order to reduce this size to a desired level I_d, we use Theorem 1 to develop two heuristic approaches.

a) Method 1: In this method, we first obtain the sets Q^{yes}, $Q^?$, and Q^{no} by performing the abstraction and model checking steps on the initial system. Then, we repeatedly select and refine the state with the largest transition interval in $Q^{yes} \cup Q^?$ until

$$\epsilon \sum_{i=1}^{k} N^i \leq I_d \tag{16}$$

where ϵ is the largest transition interval size and N is the largest number of outgoing transitions from the states in $Q^{yes} \cup Q^?$. Once this condition is met, Theorem 1 guarantees that $I^{q_i}(\phi) \leq I_d$ for all $q_i \in Q^?$. A model checking step is then performed on the refined regions to find the modified sets Q^{yes}, $Q^?$, and Q^{no}.

The advantage of this method is that model checking is performed only twice, once on the initial system and once on the end-refined system. Moreover, the refinement and abstraction steps are performed on the target and affected regions only, not on the entire domain (i.e., the states in Q^{no} do not need to be considered for refinement since they contain no subregion that satisfies the specification). Since (15) is an over-approximation of the upper-bound of the satisfying probability interval size, Condition (16), however, might cause excessive refinements.

As a heuristic, the method is not guaranteed to work for every temporal operator since the over-approximation grows exponentially with the number of time steps k. Thus, it is possible that (16) will not be satisfied. However, for the path formulas with one time step (i.e., $\mathcal{X}$ and $\mathcal{U}^{\leq 1}$) this method can be used because the over-approximated bound is at its tightest value. In this case, N remains the same while ϵ monotonically decreases during refinement. The second approach that is described below avoids this problem at the cost of requiring model checking after each refinement step.

b) Method 2: This method is motivated by the fact that as the size of the transition probability intervals of the states that are on the satisfying paths from q_i decreases, $I^{q_i}(\phi)$ also decreases. The algorithm proceeds as follows. First, obtain Q^{yes}, $Q^?$, and Q^{no} by performing the abstraction and model checking steps on the initial system. Then, select $q_i \in Q^?$ with the largest $I^{q_i}(\phi)$. By a simple graph search, find the set of states in $Q^{yes} \cup Q^?$ that can be reached from q_i in k steps with probability greater than zero. From this set, the state q_j with the largest transition probability interval is refined using the algorithm in Section VI-A1. Abstraction and model checking steps are then performed to find the modified sets of Q^{yes}, $Q^?$, and Q^{no} and their intervals of satisfying probabilities. This loop is repeated until $I^{q_i}(\phi) \leq I_d$ for all $q_i \in Q^?$.

This method guarantees that $I^{q_i}(\phi) \leq I_d$ for all $q_i \in Q^?$ is achieved in finite number of iterations. Since the refinement algorithm monotonically decreases the transition probability intervals, the reductions in the interval size of the probability of satisfaction are monotonic. Therefore, for any nonzero I_d, there is an iteration number such that $I^{q_i}(\phi) \leq I_d$. Even though arbitrary small I_d can be theoretically attained, realistically this method could suffer from the *curse of dimensionality* as $I_d \to 0$. Moreover, to achieve small I_d, a large number of refinement steps is required, escalating the size of the abstraction model. As a result, the computation cost of model checking also increases.

Note that in our method of verification for system (1), the value of $1 - I_d$ can be viewed as the desired confidence of the model checking results for the states in $Q^?$. Since the user provides I_d, it is both reasonable and conservative to treat $Q^?$ as unsatisfying states with confidence $1 - I_d$ once the verification algorithm terminates. This view enables us to perform verification of system (1) against nested formulas. Nevertheless, we acknowledge this method is conservative, and we leave the full treatment of nested formulas for future work.

B. Refinement for Systems With $m_a > 1$

Recall that in the case of a BMDP, the model checking algorithm yields three sets of states, Q^{yes}, $Q^?$, and Q^{no}, their corresponding probabilities of satisfaction, and two optimal policies (one corresponding to $\check{p}^{q_i}_{\min}$ and the other to $\hat{p}^{q_i}_{\max}$). As in the single dynamics case, we are interested in reducing the size of the satisfying probability intervals for the states in $Q^?$ to values less than or equal to I_d. To do this we first select an optimal policy (described further below) such that each region has a unique action assigned to it, converting the BMDP to an IMC. We then use either of the two methods described in Section VI-A to ensure $I^{q_i}(\phi) \leq I_d$.

The choice of the optimal policy depends on the relational operator $\bowtie$ in the formula $\phi = \mathcal{P}_{\bowtie p}[\psi]$, with the one corresponding to $\hat{p}^{q_i}_{\max}$ selected if $\bowtie \in \{\leq, <\}$ and the one corresponding to $\check{p}^{q_i}_{\min}$ selected if $\bowtie \in \{\geq, >\}$. If the selected policy is history dependent, that is if the action to select at a given state is a function of the time step, we assign and fix the action that becomes available first in the model checking process to each region. That is because this action is optimal with respect to the satisfaction of ψ in the one-step transition.

VII. Synthesis

In this section we focus on Problem 2, namely the synthesis of a control policy that maximizes the probability of satisfying a given specification $\mathcal{P}_{\max=?}[\phi]$. (The procedure for $\mathcal{P}_{\min=?}[\phi]$ is essentially the same as the one described below with the obvious adjustments for minimization). By the nature of the synthesis problem, we assume system (1) has multiple dynamics. Since our approach produces an interval of the probability of satisfaction of the specification from each state, the goal of synthesis is to find a policy that maximizes the lower bounds of those intervals.

Synthesis is performed by using the algorithms for verification developed in this paper and storing the optimal policy they calculate. While these algorithms only include single next, bounded-until, and until operators, more complex formulas with nested specifications can be handled by techniques developed in our previous work [12], extended in a straightforward manner to account for the interval-based description of the transition probabilities. This synthesis algorithm returns the optimal policy and the approximate maximum lower-bound probability of satisfaction.

Once that policy is determined, the upper bound probabilities of satisfaction for each state are calculated by model checking the IMC induced by the policy. If the size of the probability interval of satisfaction from a region is larger than desired, we employ Method 2 of the refinement procedure introduced in Section VI-A to reduce it. Note that Method 2 (Section VI-A2b) is preferred because after each iteration of the algorithm the optimal control policy is updated. The end result is a set of initial states, their intervals of optimal probability of satisfaction (with size less than a desired value), and the corresponding control policy.

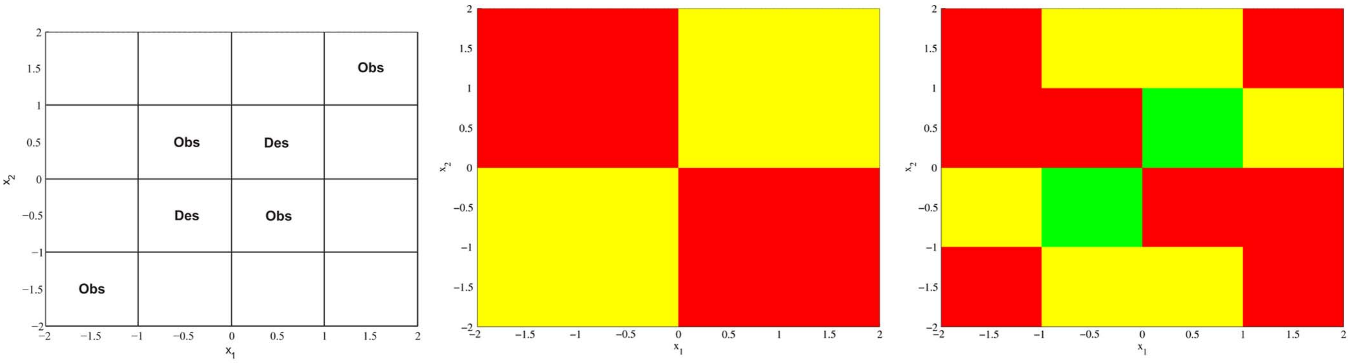
Example 6: To illustrate this synthesis algorithm, consider the BMDP in Example 3 and the PCTL formula $\phi = \mathcal{P}_{\max=?}[\neg \mathbf{R_3} \mathcal{U}^{\leq 2} \mathbf{R_2}]$. By inspection, $Q^1 = \{q_2\}$, $Q^0 = \{q_3\}$, and $Q^\dagger = \{q_0, q_1\}$. We are interested in $\check{p}^{q_i}_{\max}$ for all $q_i \in Q$ and the corresponding policy, defined by the choice of action in q_1. To find these values, we compute the following:

$$Steps^{\downarrow} \cdot \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} = \begin{array}{c} q_0; a_1 \\ \hline q_1; a_1 \\ q_1; a_2 \\ \hline q_2; a_1 \\ q_2; a_2 \\ \hline q_3; a_1 \end{array} \left(\begin{array}{c} 0 \\ \hline 0.15 \\ 0.50 \\ \hline 1 \\ 0 \\ \hline 0 \end{array} \right).$$

Thus, $\check{p}_{\max}(\neg \mathbf{R_3} \mathcal{U}^{\leq 1} \mathbf{R_2}) = (0\, 0.50\, 1\, 0)^T$ and $\mu^1_{\max}(q_1) = a_2$. Similarly, to find $\check{p}_{\max}(\neg \mathbf{R_3} \mathcal{U}^{\leq 2} \mathbf{R_2})$, we compute the product of $Steps^{\downarrow} \cdot \check{p}_{\max}(\neg \mathbf{R_3} \mathcal{U}^{\leq 1} \mathbf{R_2})$ followed by a max step. We obtain, $\check{p}_{\max}(\neg \mathbf{R_3} \mathcal{U}^{\leq 2} \mathbf{R_2}) = (0.48\ 0.50\ 1\ 0)^T$, and $\mu^2_{\max}(q_1) = a_2$. To calculate the upper bound probability values under this policy, we first construct the following lower-bound and upper-bound transition probability matrices from $\widecheck{Steps}$ and $\widehat{Steps}$, respectively, using policy $\mu_{\max}$:

$$\check{P}_{\mu_{\max}} = \begin{pmatrix} 0 & 0.05 & 0 & 0 \\ 0 & 0 & 0.50 & 0.44 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\hat{P}_{\mu_{\max}} = \begin{pmatrix} 0.05 & 1 & 0 & 0 \\ 0 & 0 & 0.56 & 0.50 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$



(a) Domain Polytope P (b) Satisfaction of ϕ_1. (c) Satisfaction of ϕ_2.

Fig. 4. Formal verification of the stochastic linear system presented in Section VIII-A with the partitioned polytopic domain P shown in (a) against (b) ϕ_1 and (c) ϕ_2 before uncertainty reduction. The initial states that definitely, possibly, and never satisfy the specification are shown in green, yellow, and red, respectively.

The upper bound probabilities are then

$$\begin{aligned}\hat{p}_{\mu_{\max}}(\neg \mathbf{R_3}\mathcal{U}^{\le 1}\mathbf{R_2}) &= P^{\uparrow}_{\mu_{\max}}(0\;0\;1\;0)^T \\ &= (0\;0.56\;1\;0)^T, \\ \hat{p}_{\mu_{\max}}(\neg \mathbf{R_3}\mathcal{U}^{\le 2}\mathbf{R_2}) &= P^{\uparrow}_{\mu_{\max}}\hat{p}_{\mu_{\max}}(\neg \mathbf{R_3}\mathcal{U}^{\le 1}\mathbf{R_2}) \\ &= (0.56\;0.56\;1\;0)^T.\end{aligned}$$

The probability of satisfaction intervals under the optimal policy for q_0 and q_1 are [0.48, 0.56] and [0.50,0.56], respectively. The probability of satisfaction for q_2 is 1 and for q_3 is 0.

VIII. Case Studies

A. Case Study 1: Verification

To demonstrate our solution to Problem 1, which includes the execution of the abstraction, model checking, and refinement algorithms presented in Sections IV–VI, we considered the square domain P shown in Fig. 4(a). The region has a length of four units per side, is centered at the origin, and is partitioned by the six linear predicates $x_1 \ge 1$, $x_1 \ge 0$, $x_1 \ge -1$, $x_2 \ge 1$, $x_2 \ge 0$, and $x_2 \ge -1$, resulting in 16 subpolytopes. The properties of interest in these regions were defined to be *Obstacles* (**Obs**) and *Destinations* (**Des**).

The stochastic system was taken to be $x_{k+1} = A_1 x_k + w_k$, $w_k \in W$, where

$$A_1 = \begin{pmatrix} 0.4 & 0.1 \\ 0 & 0.5 \end{pmatrix}$$

$$W = conv\left(\begin{bmatrix} 0.4 \\ 0.4 \end{bmatrix}, \begin{bmatrix} 0.4 \\ -0.4 \end{bmatrix}, \begin{bmatrix} -0.4 \\ -0.4 \end{bmatrix}, \begin{bmatrix} -0.4 \\ 0.4 \end{bmatrix}\right).$$

The random variable w_k was defined by the truncated normal density function

$$g(y; W, 0, 0.04\mathbb{I}) = \begin{cases} \frac{f(y;0,0.09\mathbb{I})}{\int_W f(z;0,0.09\mathbb{I})dz} & \text{if } y \in W \\ 0 & \text{Otherwise} \end{cases}$$

where $\mathbb{I}$ is the identity matrix. Here $f(\cdot; 0, 0.09\mathbb{I})$ is a zero mean Gaussian distribution with covariance $0.09\mathbb{I}$, yielding a zero mean truncated distribution g with covariance of $0.04\mathbb{I}$.

We considered the following specifications:

Specification 1: "Find a set of initial states from which the probability of converging to a region with *Obstacle* is less than 0.05."

Specification 2: "Find a set of initial states that with probability 0.90 or greater will reach *Destination* without colliding with an *Obstacle*."

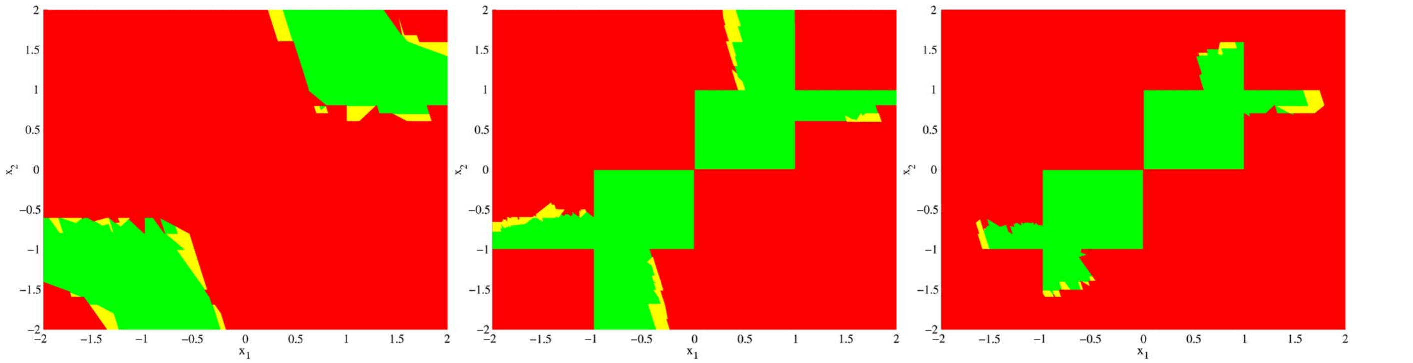
Specification 3: "Find a set of initial states that with probability 0.90 or greater will reach *Destination* through the regions that are not *Obstacles* and that have a probability of less than 0.05 to converge to a region with an *Obstacle*."

These specifications translate naturally to the PCTL formulas ϕ_1, ϕ_2, and ϕ_3 where

$$\begin{aligned}\phi_1 &= \mathcal{P}_{<0.05}[X\,\mathbf{Obs}] \\ \phi_2 &= \mathcal{P}_{\ge 0.90}[\neg\mathbf{Obs}\,\mathcal{U}\,\mathbf{Des}] \\ \phi_3 &= \mathcal{P}_{\ge 0.90}\left[(\neg\mathbf{Obs} \wedge \mathcal{P}_{<0.05}[X\,\mathbf{Obs}])\,\mathcal{U}\mathbf{Des}\right].\end{aligned}$$

An IMC abstraction of the system was generated by setting the states to be the initial partition of P. The transition probability bounds were computed by discretizing each subpolytope to a set of points with 0.01 unit length distance and numerical evaluation of (9) and (10). The stochastic kernel for this system is $T(\cdot|x) = g(\cdot; W, Ax, 0.04\mathbb{I})$. The set of properties $\Pi = \{\mathbf{Obs}, \mathbf{Des}\}$ of the IMC were assigned to the states according to the labels of the corresponding regions. The maximum allowable size of the interval of satisfying probabilities for the possibly satisfying states was set to $I_d = 0.05$.

We performed a model checking step on the states of the IMC to find Q^{yes}, $Q^{?}$, Q^{no}, and their satisfying probability intervals for each formula ϕ_1 and ϕ_2, shown in Fig. 4(b) and (c) where green, yellow, and red correspond to the states in Q^{yes}, $Q^{?}$, and Q^{no}, respectively. Then, for each specification, we performed refinement according to Method 2. The loop of refinement-abstraction-model checking was executed until all the satisfying interval sizes of the states in $Q^{?}$ were less than I_d. The final results for ϕ_1, ϕ_2 are shown in Fig. 5(a) and (b), respectively. For ϕ_3, we began with the final partition generated for ϕ_2. Applying the algorithm showed that this partition already led to the maximum size of the probability of satisfaction interval to be less than I_d. The results are shown in Fig. 5(c). The final IMCs corresponding to ϕ_1 had 310 states



(a) Final partition for ϕ_1.

(b) Final partition for ϕ_2.

(c) Final partition for ϕ_3.

Fig. 5. Formal verification of the stochastic linear system presented in Section VIII-A against (a) ϕ_1, (b) ϕ_2, and (c) ϕ_3 after uncertainty reduction. The initial states that definitely, possibly, and never satisfy the specification are shown in green, yellow, and red, respectively.

and was obtained by 88 refinement iterations while those of ϕ_2 and ϕ_3 had 1,163 states resulted from 371 iterations.

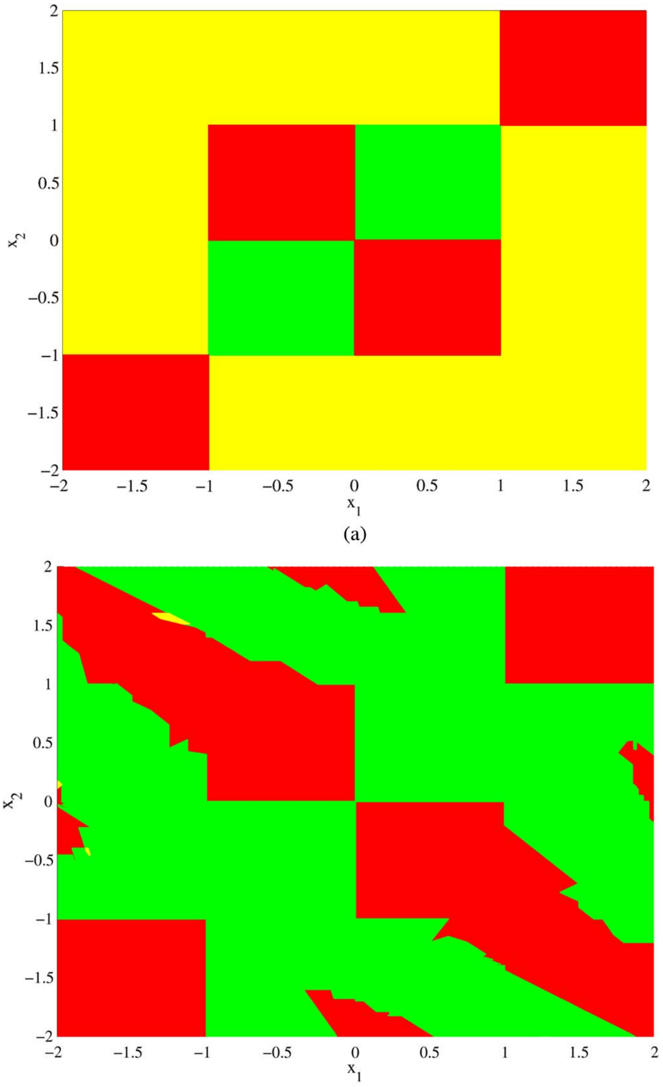
All the computations for this case study were performed in MATLAB on an Ubuntu 12.04 machine with AMD FX 4100 Quad-Core Processor and 16 GB of memory. The total computation time for generating final results for ϕ_1 was 4.8 hours, while it took 51.4 hours for ϕ_2 and ϕ_3. Most of these times were spent on the numerical evaluation of the integrals in (9) and (10).

B. *Case Study 2: Synthesis*

For the synthesis case study, we considered the switched stochastic system $x_{k+1} = A_i x_k + w_k$, $i = \{1, 2, 3\}$, where A_1, P, W, and Π were taken as in case study 1. The additional dynamics were given by

$$A_2 = \begin{pmatrix} 0.4 & 0.5 \\ 0 & 0.5 \end{pmatrix}, \quad A_3 = \begin{pmatrix} 0.4 & 0 \\ 0.5 & 0.5 \end{pmatrix}.$$

Here, w_k was sampled uniformly from W. We performed our BMDP abstraction and then synthesis algorithms to find the policy that maximizes the lower probability bound of satisfaction of ϕ_2 for the above system. The resulting Q^{yes}, $Q^?$, and Q^{no} are shown in Fig. 6(a) with green, yellow, and red, respectively. Then, we performed refinement according to Method 2 to reduce the uncertainty in $Q^?$ to less than $I_d = 0.05$. The results are shown in Fig. 6(b). The final BMDP was obtained after 609 refinement iterations and had 4,489 states. The total computation times for abstraction and refinement were 114.8 hours and 252.6 hours, respectively, while it was only 2.2 hours for synthesis for the BMDP.



(a)

(b)

Fig. 6. Formal synthesis of the policy that maximizes the lower-bound probability of satisfaction of ϕ_2 for the switched stochastic system presented in Section VIII-B (a) before and (b) after uncertainty reduction. Initial states that definitely, possibly, and never satisfy ϕ_2 are shown in green, yellow, and red, respectively. (a) Initial partition for ϕ_2-Synthesis. (b) Final partition for ϕ_2-Synthesis.

IX. Conclusion

We presented a computational framework for formal verification and formal synthesis for discrete-time stochastic systems with polyhedral noise domains in a full-dimensional convex polytope from PCTL specifications. In this framework, we first abstract the evolution of the stochastic system in its polytopic domain to either an IMC or to a BMDP. Next, we model check this IMC or BMDP using algorithms similar to standard Markov chain model checking algorithms. For synthesis, we developed an algorithm for BMDPs that was inspired by our MDP synthesis algorithm presented in [12]. Lastly, we introduced two methods for refinement of the model to reduce the uncertainty in the solution to a desired value.

The main contributions of this work are a Markovian abstraction method which finds the exact bounds for the transition probabilities, model checking algorithms for IMCs and BMDPs with low computational cost, a synthesis algorithm for BMDPs, and an expression for the probability interval size growth over time. For linear dynamics and linear predicates, we also introduced a refinement algorithm that exploits the dynamics of the system and the geometry of the partition.

REFERENCES

[1] M. Lahijanian, S. B. Andersson, and C. Belta, "Approximate Markovian abstractions for linear stochastic systems," in *Proc. IEEE Conf. Decision Control*, Maui, HI, USA, Dec. 2012, pp. 5966–5971.

[2] E. M. Clarke, O. Grumberg, and D. Peled, *Model Checking*. Cambridge, MA, USA: MIT Press, 1999.

[3] P. Tabuada and G. J. Pappas, "Linear time logic control of discrete-time linear systems," *IEEE Trans. Autom. Control*, vol. 51, no. 12, pp. 1862–1877, Dec. 2006.

[4] M. Kloetzer and C. Belta, "A fully automated framework for control of linear systems from temporal logic specifications," *IEEE Trans. Autom. Control*, vol. 53, no. 1, pp. 287–297, Feb. 2008.

[5] R. Alur, T. A. Henzinger, G. Lafferriere, George, and J. Pappas, "Discrete abstractions of hybrid systems," in *Proc. IEEE*, 2000, vol. 88, pp. 971–984.

[6] B. Yordanov and C. Belta, "Formal analysis of discrete-time piecewise affine systems," *IEEE Trans. Autom. Control*, vol. 55, no. 12, pp. 2834–2840, Dec. 2010.

[7] H. Hansson and B. Jonsson, "A logic for reasoning about time and reliability," *Formal Aspects of Computing*, vol. 6, pp. 102–111, 1994.

[8] L. V. Utkin and I. Kozine, "Computing system reliability given interval-valued characteristics of the components," *Reliable Computing*, vol. 11, no. 1, pp. 19–34, 2005.

[9] D. Škulj (2009, Sep.). Discrete time Markov chains with interval probabilities. *Int. J. Approximate Reasoning* [Online]. *50(8)*, pp. 1314–1329. Available: http://dx.doi.org/10.1016/j.ijar.2009.06.007

[10] R. Givan, S. Leach, and T. Dean, "Bounded-parameter Markov decision processes," *Artif. Intell.*, vol. 122, pp. 71–109, 2000.

[11] A. Abate, J. Katoen, J. Lygeros, and M. Prandini, "Approximate model checking of stochastic hybrid systems," *Eur. J. Control*, vol. 16, no. 6, pp. 624–641, 2010.

[12] M. Lahijanian, S. B. Andersson, and C. Belta, "Temporal logic motion planning and control with probabilistic satisfaction guarantees," *IEEE Trans. Robotics*, vol. 28, no. 2, pp. 396–409, Apr. 2012.

[13] M. Lahijanian, S. B. Andersson, and C. Belta, "A probabilistic approach for control of a stochastic system from LTL specifications," in *Proc. 48th IEEE Conf. Decision Control*, Shanghai, China, 2009.

[14] E. Vanden-Eijndena and M. Venturoli, "Markovian milestoning with Voronoi tessellations," *J. Chem. Phys.*, vol. 130, no. 19, pp. 194 101-1–194 101-13, May 2009.

[15] A. Abate, A. D'Innocenzo, M. D. Benedetto, and S. Sastry, "Markov set-chains as abstractions of stochastic hybrid systems," in *Hybrid Syst.: Comput. Control*. Berlin, Germany: Springer Verlag, 2008, pp. 1–15.

[16] A. Abate, A. D'Innocenzo, and M. D. Benedetto, "Approximate abstractions of stochastic hybrid systems," *IEEE Trans. Autom. Control*, vol. 56, no. 10, 2011.

[17] S. E. Z. Soudjani and A. Abate, "Adaptive gridding for abstraction and verification of stochastic hybrid systems," in *Proc. Int. Conf. Quantitative Eval. Syst.*, 2011, pp. 59–69.

[18] M. L. Bujorianu, J. Lygeros, and M. C. Bujorianu, "Bisimulation for general stochastic hybrid systems," in *Hybrid Systems: Computation and Control*. New York, NY, USA: Springer, 2005, pp. 198–214.

[19] S. Strubbe and A. Van Der Schaft, "Bisimulation for communicating piecewise deterministic Markov processes (CPDPs)," in *Hybrid Systems: Computation and Control*. New York, NY, USA: Springer, 2005, pp. 623–639.

[20] J. Desharnais, V. Gupta, R. Jagadeesan, and P. Panangaden, "Metrics for labelled Markov processes," *Theor. Comp. Sci.*, vol. 318, no. 3, pp. 323–354, 2004.

[21] F. van Breugel and J. Worrell, "An algorithm for quantitative verification of probabilistic transition systems," in *Concurrency Theory*. New York, NY, USA: Springer, 2001, pp. 336–350.

[22] A. Julius and G. J. Pappas, "Approximations of stochastic hybrid systems," *IEEE Trans. Autom. Control*, vol. 54, no. 6, pp. 1193–1203, Jun. 2009.

[23] M. Kwiatkowska, G. Norman, and D. Parker, "Probabilistic symbolic model checking with PRISM: A hybrid approach," *Int. J. Softw. Tools Technol. Transfer (STTT)*, vol. 6, no. 2, pp. 128–142, 2004.

[24] J.-P. Katoen, I. S. Zapreev, E. M. Hahn, H. Hermanns, and D. N. Jansen, "The ins and outs of the probabilistic model checker MRMC," in *Proc. Int. Conf. Quantitative Eval. Syst.*, 2009, pp. 167–176.

[25] X. Ding, S. Smith, C. Belta, and D. Rus, "MDP optimal control under temporal logic constraints," in *Proc. IEEE Conf. Decision Control & Eur. Control Conf. (CDC-ECC'11)*, 2011, pp. 532–538.

[26] A. Abate, J.-P. Katoen, and A. Mereacre, "Quantitative automata model checking of autonomous stochastic hybrid systems," in *Proc. Int. Conf. Hybrid Syst.: Comput. Control*, 2011, pp. 83–92.

[27] I. Tkachev and A. Abate, "Formula-free finite abstractions for linear temporal verification of stochastic hybrid systems," in *Proc. Int. Conf. Hybrid Syst.: Comput. Control*, 2013, pp. 283–292.

[28] H. Fecher, M. Leucker, and V. Wolf, "Don't know in probabilistic systems," in *Model Checking Software*. New York, NY, USA: Springer, 2006, pp. 71–88.

[29] J.-P. Katoen, D. Klink, M. Leucker, and V. Wolf, "Three-valued abstraction for continuous-time Markov chains," in *Computer Aided Verification*. New York, NY, USA: Springer, 2007, pp. 311–324.

[30] J.-P. Katoen, D. Klink, M. Leucker, and V. Wolf, "Three-valued abstraction for probabilistic systems," *J. Logic Algebraic Programming*, vol. 81, no. 4, pp. 356–389, May 2012.

[31] K. Sen, M. Viswanathan, and G. Agha, "Model-checking Markov chains in the presence of uncertainties," in *Proc. Int. Conf. Tools Alg. Const. Anal. Sys.*, 2006, pp. 394–410.

[32] K. Chatterjee, K. Sen, and T. A. Henzinger, "Model-checking ω-regular properties of interval Markov chains," in *Proc. Theor. Prac. Soft., Conf. Found. Soft. Sci. Compu. Str.*, 2008, pp. 302–317.

[33] T. Chen, T. Han, and M. Kwiatkowska, "On the complexity of model checking interval-valued discrete time markov chains," *Inform. Processing Lett.*, vol. 113, no. 7, pp. 210–216, Apr. 2013.

[34] M. Kattenbelt, M. Kwiatkowska, G. Norman, and D. Parker, "A game-based abstraction-refinement framework for Markov decision processes," *Formal Methods Syst. Design*, vol. 36, no. 3, pp. 246–280, Sep. 2010.

[35] E. M. Hahn, G. Norman, D. Parker, B. Wachter, and L. Zhang, "Game-based abstraction and controller synthesis for probabilistic hybrid systems," in *Proc. Int. Conf. Quantitative Eval. Syst.*, 2011, pp. 69–78.

[36] L. Zhang, Z. She, S. Ratschan, H. Hermanns, and E. M. Hahn, "Safety verification for probabilistic hybrid systems," *Eur. J. Control*, vol. 18, no. 6, pp. 572–587, Jan. 2012.

[37] B. Jonsson and K. G. Larsen, "Specification and refinement of probabilistic processes," in *Proc. LICS'91*, 1991, pp. 266–277.

[38] B. Yordanov and C. Belta, "Formal analysis of piecewise affine systems under parameter uncertainty with application to gene networks," in *Proc. Amer. Control Conf. (ACC)*, 11–13, 2008, pp. 2767–2772.

[39] D. Wu and X. Koutsoukos, "Reachability analysis of uncertain systems using bounded-parameter Markov decision processes," *Artif. Intell.*, vol. 172, no. 8, pp. 945–954, May 2008.

[40] P. R. D'Argenio, B. Jeannet, H. E. Jensen, and K. G. Larsen, "Reachability analysis of probabilistic systems by successive refinements," in *Process Algebra and Probabilistic Methods. Performance Modelling and Verification*, vol. 2165. New York, NY, USA: Springer, 2001, ser. Lecture Notes in Computer Science, pp. 39–56.

Morteza Lahijanian (M'12) received the B.S. degree in bioengineering from the University of California, Berkeley, CA, USA in 2004 and the Ph.D. degree in mechanical engineering from Boston University, Boston, MA, USA, in 2012.

He is currently a Postdoctoral Researcher in the Department of Computer Science, Rice University, Houston, TX, USA. His research interests include dynamics, control theory, systems, and formal methods with applications in robotics and systems biology, particularly, motion planning, finite abstraction, formal synthesis, and hybrid systems.

Sean B. Andersson (SM'13) received the B.S. degree in engineering and applied physics from Cornell University, Ithaca, NY, USA, in 1994, the M.S. degree in mechanical engineering from Stanford University, Stanford, CA, USA, in 1995, and the Ph.D. degree in electrical and computer engineering from the University of Maryland, College Park, MD, USA, in 2003.

He has worked at AlliedSignal Aerospace and Aerovironment, Inc. and is currently an Associate Professor of mechanical engineering and of systems engineering with Boston University, Boston, MA, USA. His research interests include systems and control theory with applications in scanning probe microscopy, dynamics in molecular systems, and robotics.

Calin Belta (SM'11) received the B.Sc. and M.Sc. degrees in control engineering from the Technical University of Iasi, Splai Bahlui, Iasi, Romania and the M.Sc. and Ph.D. degrees in mechanical engineering from the University of Pennsylvania, Philadelphia, PA, USA.

He is currently a Professor in the Department of Mechanical Engineering, Department of Electrical and Computer Engineering, and the Division of Systems Engineering, Boston University, Boston, MA, USA, where he is also affiliated with the Center for Information and Systems Engineering (CISE) and the Bioinformatics Program. His research focuses on dynamics and control theory, with particular emphasis on hybrid and cyber-physical systems, formal synthesis and verification, and applications in robotics and systems biology.

Dr. Belta is an Associate Editor for the IEEE TRANSACTIONS OF AUTOMATIC CONTROL. He received the Air Force Office of Scientific Research Young Investigator Award and the National Science Foundation CAREER Award.