Adaptive and Sequential Gridding Procedures for the Abstraction and Verification of Stochastic Processes[*]

Sadegh Esmaeil Zadeh Soudjani[†] and Alessandro Abate[†]

Abstract. This work is concerned with the generation of finite abstractions of general state-space processes to be employed in the formal verification of probabilistic properties by means of automatic techniques such as probabilistic model checkers. The work employs an abstraction procedure based on the partitioning of the state-space, which generates a Markov chain as an approximation of the original process. A novel adaptive and sequential gridding algorithm is presented and is expected to conform to the underlying dynamics of the model and thus to mitigate the curse of dimensionality unavoidably related to the partitioning procedure. The results are also extended to the general modeling framework known as stochastic hybrid systems. While the technique is applicable to a wide arena of probabilistic properties, with focus on the study of a particular specification (probabilistic safety, or invariance, over a finite horizon), the proposed adaptive algorithm is first benchmarked against a uniform gridding approach taken from the literature and finally tested on an applicative case study in Biology.

Key words. general state-space processes, Markov chains, stochastic hybrid systems, abstractions, approximations, formal verification, safety and invariance, properties and specifications

AMS subject classifications. 93E03, 93E25, 60J20

DOI. 10.1137/120871456

1. Introduction. This work studies the problem of computing probabilistic properties for discrete time Markov processes evolving over continuous (uncountable) state-spaces. We interpret the analysis of a given property as the formal verification of a related specification expressed in a probabilistic modal logic [4]. Theoretically, the connection between the computation of a class of dynamical properties and the verification of related specifications in probabilistic computation tree logic (PCTL) has been investigated in [26] and extended in [2]. To keep the work focused, this contribution zooms in on the fundamental problem of probabilistic invariance, or safety, and on its related specification. This problem has been recently investigated in [3], which has characterized this concept and put forward an algorithm to compute this quantity.

From a computational perspective, [1] has looked at the numerical evaluation of specifications discussed in [3], among which is probabilistic invariance. This evaluation is made possible by developing a formal abstraction that is based on the partitioning of the state-

[†]Delft Center for Systems & Control, TU Delft - Delft University of Technology, Delft, The Netherlands (S.EsmaeilZadehSoudjani@TUDelft.nl, A.Abate@TUDelft.nl).

space of the original continuous space model, which originates a discrete time, finite space Markov chain (MC) from the original process. The approach is formal in that it allows for the computation of explicit bounds on the error associated with the abstraction. This technique enables considering classes of probabilistic specifications [2, 1] over continuous space models and computing them over MC abstractions via available probabilistic model checkers [17, 18], with explicit bounds on the errors introduced with the abstraction procedure.

This work looks at extending the applicability of the technique developed in [1] by addressing its known bottleneck, the issue of dimensional scalability of the abstraction, which is limited by the "curse of dimensionality" related to the partitioning procedure and subsequent dynamic programming recursions. This new procedure is expected to adapt to the underlying dynamics of the model, which is characterized by (a set of) stochastic kernels. In contrast to the abstraction proposed in [1], which has leveraged a batch algorithm performing uniform partitioning based on the quantification of a global error, this work puts forward an adaptive and sequential procedure that exploits the knowledge of local quantities and performs the normalization of dynamics operating on multiple spatial scales. Furthermore, this work looks at the practical implementation of the adaptive procedure, which hinges on the choice of the shape of partition sets (making up the states of the MC), the execution of the refinement step in the adaptive generation of the grid, and the generation of transition probabilities for the MC over the partition sets (which involves a marginalization procedure). Additionally, the issue of ill-conditioned dynamics (namely, widely separated dynamics operating over slow and fast scales) is tackled by considering a further refinement of the obtained errors based on state-space rescaling.

Owing to the explicit computation of the error bounds related to a given property, this work provides an approach to abstraction that is effectively property-dependent. Furthermore, given the generality of the concepts of reachability and (dually) of invariance and due to their connections to more general properties [2], this abstraction technique allows a general approach to the study of these properties.

Most of the reviewed literature on the subject of formal verification of stochastic processes presents a penchant for models known as stochastic hybrid systems (SHS), which are general dynamical models with interleaved discrete, continuous, and probabilistic dynamics. Fostered by their application in a number of diverse domains [7, 10], the study of SHS has recently flourished and has witnessed interesting advances at the intersection of the fields of systems and control [11] and of formal verification [4]. In this work we develop results over abstract state-spaces and tailor them to SHS at a later stage (see section 4 for the theory and section 6.2 for a case study).

From a different perspective and over classes of continuous time probabilistic hybrid models, [8] has formalized the notion of probabilistic reachability, [24] has put forward a computational technique based on convex optimization, and [13] has developed an approach based on satisfiability modulo theory to attain the verification of similar probabilistic properties. Over models with similar semantics, [21, 25] have quantified the concept of probabilistic reachability as the solution of partial differential equations over the state-space and presented approximation techniques for its computation, which also leverage the use of discrete time MC [22]; however, neither approach provides a quantification of the error made in the approximation step, which is a distinguishing factor of this work.

The article is structured as follows. Sections 2.1 and 2.2 introduce the model and the problem statement (computation of probabilistic invariance). Section 3.1 proposes an abstraction algorithm to relate a general state-space model to an MC. Furthermore, with a focus on the probabilistic invariance problem, the quantification of the error in the abstraction procedure is presented in section 3.2, whereas section 3.3 presents refinements of the error computation based on local properties and state-space rescaling. Section 4 adapts the results to the SHS model framework. Section 5 deals with the algorithmic generation of the abstraction and elaborates on a number of choices leading to a sequential and adaptive scheme. Finally, section 6 develops two numerical studies: a benchmark compares the adaptive and sequential approach versus the uniform procedure known from the literature [1] and tests the scalability of the adaptive approach. Also, section 6 presents a case study drawn from systems biology; in particular, section 6.2 elucidates the results on an SHS model. Section 7 completes the work with conclusions and extensions.

2. Preliminaries.

2.1. Model. We consider a discrete time Markov process $s(k)$, $k \in \mathbb{N} \cup \{0\}$, defined over a general state-space. The model is denoted by $\mathfrak{S} = (\mathcal{S}, T_s)$ and characterized by the following pair:

1. $\mathcal{S}$ is a continuous state-space, which we assume to be endowed with a metric and to be Borel measurable. We denote by $(\mathcal{S}, \mathcal{B}(\mathcal{S}), P)$ the probability structure on $\mathcal{S}$, with $\mathcal{B}(\mathcal{S})$ the associated sigma algebra and P a probability measure to be characterized shortly.

2. T_s is a conditional stochastic kernel that assigns to each point $s \in \mathcal{S}$ a probability measure $T_s(\cdot|s)$, so that for any set $A \in \mathcal{B}(\mathcal{S}), P(s(1) \in A|s(0) = s_0) = \int_A T_s(ds|s_0)$.

The initial condition $s(0)$ for the model is sampled from $\pi : \mathcal{B}(\mathcal{S}) \to [0, 1]$, a probability measure on $\mathcal{S}$. Over a finite horizon $[0, N]$, a Markov process $s(k)$, $k \in [0, N]$, evolves over the product space $\Omega = (\mathcal{S})^{N+1}$, which is also endowed with a sigma algebra and thus allows computing the probability of events related to trajectories; we will again use P to denote such probability. Usually the state-space is taken to be a finite-dimensional Euclidean domain, $\mathcal{S} = \mathbb{R}^n, n < \infty$. In section 4, we tailor this setup to a specific "hybrid" state-space, thus introducing a modeling framework known as SHS (see also section 6.2 for a case study based on an SHS model).

2.2. Problem statement. The problem of finite-horizon probabilistic invariance (alternatively referred to as probabilistic safety) can be formalized as follows: consider a bounded Borel set $A \in \mathcal{B}(\mathcal{S})$, representing a set of safe states. Characterize and compute the probability that an execution of $\mathfrak{S}$, associated with an initial condition $s_0 \in \mathcal{S}$ (sampled from π), remains within set A during the finite time horizon $[0, N]$:

$$p_{s_0}(A) := P\{s(k) \in A \ \forall k \in [0, N] | s(0) = s_0\}.$$

This quantity allows us to extend the result to a general initial probability distribution π as $p_\pi(A) := P\{s(k) \in A \ \forall k \in [0, N]\} = \int p_{s_0}(A)\pi(ds_0)$. The following theorem provides a theoretical framework to study the probabilistic invariance problem.

Theorem 2.1 (see [3]). *Consider value functions* $V_k : \mathcal{S} \to [0, 1]$, $k = 0, 1, \ldots, N$, *computed by the following backward recursion:*

$$V_k(s) = \mathbf{1}_A(s) \int_{\mathcal{S}} V_{k+1}(\bar{s}) T_s(d\bar{s}|s), \quad s \in \mathcal{S},$$

and initialized with

$$V_N(s) = \mathbf{1}_A(s) = \begin{cases} 1 & \text{if } s \in A, \\ 0 & \text{else.} \end{cases}$$

Then $p_{s_0}(A) = V_0(s_0)$.

This result characterizes the finite-horizon probabilistic invariance quantity as the solution of a dynamic programming problem. However, since its explicit solution is in general not available, the actual computation of the quantity $p_{s_0}(A)$ requires N numerical integrations over the whole set A. This is usually performed with techniques based on state-space discretization [6], which leads to two major questions:

1. whether the numerical output can be precisely related to the actual solution; and

2. whether the approach is dimensionally scalable (e.g., as a function of n if $\mathcal{S} = \mathbb{R}^n$), particularly in comparison with alternative known approaches in the literature [1].

The goal of this work is to address these two issues. In the next section we answer the first question by introducing an abstraction of the original model via a numerical approximation, and by explicitly quantifying the error related to the computation of the finite-horizon probabilistic invariance with the abstraction. Furthermore, by focusing on the algorithmic implementation of the abstraction, in the remainder of this work we investigate the scalability properties of the proposed approach (computational complexity and memory usage), thus addressing the second question.

The overall approach, here presented over the problem of probabilistic invariance, can be directly extended to more general properties expressed in PCTL [26], as well as over specifications characterized as certain labeled automata [2]. Both extensions can be reduced to computations of value functions related to that in Theorem 2.1 characterizing probabilistic invariance.

3. Model abstraction.

3.1. Algorithmic abstraction as a finite state MC.

We recall a procedure presented in [1] to approximate a model $\mathfrak{S} = (\mathcal{S}, T_s)$ by a finite state MC $\mathfrak{P} = (\mathcal{P}, T_p)$. Here $\mathcal{P} = \{z_1, z_2, \ldots, z_p\}$ is a finite set of states and $T_p : \mathcal{P} \times \mathcal{P} \to [0, 1]$ is a transition probability matrix, such that $T_p(z, z') = P(z'|z)$ characterizes the probability of transitioning from state z to state z' and thus induces a conditional discrete probability distribution over the finite space $\mathcal{P}$.

Consider the bounded safe set $A \in \mathcal{B}(\mathcal{S})$. Algorithm 1 provides a procedure to abstract model $\mathfrak{S}$ by a finite state MC $\mathfrak{P}$. In Algorithm 1, $\Xi : A_p \to 2^A$ represents a set-valued map that associates to any point $z_i \in A_p$ the corresponding partition set $A_i \subset A$. Furthermore, the map $\xi : A \to A_p$ associates to any point $s \in A$ of $\mathfrak{S}$ the corresponding discrete state in A_p. Additionally, notice that the absorbing set ϕ is added to the definition of the MC $\mathfrak{P}$ in order to render the transition probability matrix T_p stochastic.

Algorithm 1 Abstraction of model $\mathfrak{S}$ by MC $\mathfrak{P}$

Require: input model $\mathfrak{S}$, set A
  1: Select a finite m-dimensional partition of set A as $A = \cup_{i=1}^{m} A_i$ (A_i are nonoverlapping)
  2: For each A_i, select a single representative point $z_i \in A_i, \{z_i\} = \xi(A_i)$
  3: Define $A_p = \{z_i, i = 1, \ldots, m\}$ and take $\mathcal{P} = A_p \cup \{\phi\}$ as the finite state-space of the MC $\mathfrak{P}$ (ϕ being a dummy variable as explained in the text)
  4: Compute the transition probability matrix T_p for $\mathfrak{P}$ as

$$T_p(z, z') = \begin{cases} T_s(\Xi(z')|z), & z' \in A_p, z \in A_p, \\ 1 - \sum_{\bar{z} \in A_p} T_s(\Xi(\bar{z})|z), & z' = \phi, z \in A_p, \\ 1, & z' = z = \phi, \\ 0, & z' \in A_p, z = \phi, \end{cases}$$

Ensure: output MC $\mathfrak{P}$

Remark 1. *Notice that Algorithm 1 can be applied to abstract a general model by a finite state MC, regardless of the specifics of the probabilistic invariance problem studied in this work (that is, regardless of the given safe set A) by assuming that $A = \mathcal{S}$. The quantification of the abstraction error, to be carried out in section 3.2, will, however, require that the set A (thus, as needed, the state-space $\mathcal{S}$) be bounded.*

Given a finite state, discrete time MC $\mathfrak{P} = (\mathcal{P}, T_p)$ and considering a safe set $A_p \subset \mathcal{P}$, the probabilistic invariance problem evaluates the probability that a finite execution associated with the initial condition $p_0 \in \mathcal{P}$ remains within the discrete safe set A_p during the finite time horizon $[0, N]$ and can be stated as follows:

$$p_{p_0}(A_p) := P\{p(k) \in A_p \;\forall k \in [0, N]|p(0) = p_0\}.$$

We now formulate the discrete version of Theorem 2.1.

Theorem 3.1 (see [1]). *Consider value functions $V_k^p : \mathcal{P} \to [0, 1], k = 0, 1, \ldots, N$, computed by the backward recursion,*

$$V_k^p(z) = \mathbf{1}_{A_p}(z) \sum_{\bar{z} \in \mathcal{P}} V_{k+1}^p(\bar{z}) T_p(z, \bar{z}), \quad z \in \mathcal{P},$$

and initialized with

$$V_N^p(z) = \mathbf{1}_{A_p}(z) = \begin{cases} 1 & if \; z \in A_p, \\ 0 & if \; z = \phi. \end{cases}$$

Then $p_{p_0}(A_p) = V_0^p(p_0)$.

It is of interest to provide a quantitative comparison between the discrete outcome obtained by Theorem 3.1 and the continuous solution that results from Theorem 2.1. The following section accomplishes this goal.

3.2. Quantification of the abstraction error. We first introduce a bound, inspired by [1, Theorem 1], on the distance between evaluations of the function $V_k, k = 0, 1, \ldots, N$, in

Theorem 2.1. *Consider a safe set $A \in \mathcal{B}(\mathcal{S})$. For any pair of points $s, s' \in A$ and $k = 0, 1, \ldots, N - 1$, notice that*

$$(3.1) \qquad \left| V_k(s) - V_k(s') \right| = \left| \int_A V_{k+1}(\bar{s}) T_s(d\bar{s}|s) - \int_A V_{k+1}(\bar{s}) T_s(d\bar{s}|s') \right|$$
$$\leq \int_A \left| T_s(d\bar{s}|s) - T_s(d\bar{s}|s') \right|,$$

since the value functions V_k are upper-bounded by the unity. Furthermore, for $k = N$ it holds trivially that $V_N(s) = V_N(s') = 1 \Rightarrow |V_N(s) - V_N(s')| = 0$.

The following Lipschitz continuity condition restricts the generality of the kernel T_s characterizing the dynamics of model $\mathfrak{S}$.

Assumption 1. *Assume that the kernel T_s admits density t_s and that the following holds for a finite positive h:*

$$\left| t_s(\bar{s}|s) - t_s(\bar{s}|s') \right| \leq h \left\| s - s' \right\| \quad \forall \bar{s}, s, s' \in A.$$

Assumption 1 allows us to derive the following bound on the abstraction error (notice the emphasis of the result on the time instance $k = 0$).

Theorem 3.2 (see [1, Theorem 2]). *Under Assumption 1, the invariance probability $p_{s_0}(A)$ for the model $\mathfrak{S}$ initialized at $s_0 \in A$ satisfies*

$$(3.2) \qquad |p_{s_0}(A) - p_{p_0}(A_p)| \leq \gamma \delta,$$

where $p_{p_0}(A_p)$ is the invariance probability for the MC $\mathfrak{P}$ obtained by Algorithm 1, and initialized at the discrete state $p_0 = \xi(s_0) \in A_p$. The constant γ is

$$\gamma = N\mathcal{K}, \qquad where \qquad \mathcal{K} = h\,\mathscr{L}(A),$$

and where δ is the largest diameter of the partition sets $A_i \subset A$: $\delta = \max\{\|s - s'\| \,|\, s, s' \in A_i, i = 1, \ldots, m\}$, h comes from Assumption 1, and $\mathscr{L}(B)$ denotes the Lebesgue measure of any set $B \in \mathcal{B}(\mathcal{S})$.

The result in Theorem 3.2 allows for the synthesis of finite abstractions of continuous-space models with explicit, finite error bounds. The quality of the bounds is key in obtaining useful abstractions (that is, we are interested in bounds that are at least smaller than the unity). Furthermore, if a specific error is the objective of the study, then the quality of the error directly affects the cardinality (m) of the abstraction space, as well as the computational effort to obtain the abstraction; we shall explore this tradeoff later in this work.

In the next section we refine the abstraction error of Theorem 3.2 in three different ways: first, by computing a local version of the error; second, by leveraging continuity requirements that go beyond the Lipschitz condition raised in Assumption 1; and, finally, by normalizing possibly ill-conditioned dynamics operating on multiple spatial scales.

3.3. Refinement of the abstraction error.

3.3.1. Local computation of abstraction error. We relax Assumption 1 as follows.

Assumption 2. *Assume that the kernel T_s admits density t_s and that the following continuity assumption is valid:*

$$\left|t_s(\bar{s}|s) - t_s(\bar{s}|s')\right| \leq h(i,j)\left\|s - s'\right\| \quad \forall \bar{s} \in A_j, \forall s, s' \in A_i,$$

where $i, j \in \{1, \ldots, m\}$, the sets A_i form a partition of A (as obtained, for instance, in Algorithm 1), and $h(\cdot, \cdot)$ are finite and positive constants.

Clearly, the global Lipschitz constant h in Assumption 1 represents an upper bound for the quantities $h(i,j)$ above. Equation (3.1) can be tailored to Assumption 2, which leads to the following result.

Theorem 3.3. *Suppose that the stochastic kernels of the model $\mathfrak{S}$ satisfy Assumption 2. Then the value functions $V_k : \mathcal{S} \to [0, 1]$, characterizing the probabilistic invariance problem for $\mathfrak{S}$ over $A \in \mathcal{B}(\mathcal{S})$, satisfy the following Lipschitz continuity, $k \in [0, N]$:*

$$\left|V_k(s) - V_k(s')\right| \leq \mathcal{K}_i \left\|s - s'\right\|$$

$\forall s, s' \in A_i, i \in \{1, \ldots, m\}$, and where the constant $\mathcal{K}_i$ is given by

$$\mathcal{K}_i = \sum_{j=1}^m h(i,j)\mathscr{L}(A_j).$$

Proof. Using (3.1) together with the inequality in Assumption 2 leads directly to the following:

$$\left|V_k(s) - V_k(s')\right| \leq \int_A \left|T_s(d\bar{s}|s) - T_s(d\bar{s}|s')\right| = \sum_{j=1}^m \int_{A_j} \left|T_s(d\bar{s}|s) - T_s(d\bar{s}|s')\right|$$

$$\leq \sum_{j=1}^m h(i,j)\left\|s - s'\right\| \mathscr{L}(A_j) = \mathcal{K}_i \left\|s - s'\right\|.$$

Notice that the bound provided in this theorem improves that derived from (3.1) and Assumption 1 since $h \geq \max\{h(i,j)|i, j = 1, \ldots, m\}$. ∎

The result in Theorem 3.3 can be employed to quantify the error between the value $p_{p_0}(A_p)$ and $p_{s_0}(A)$, which leads to a refinement of Theorem 3.2.

Theorem 3.4. *Assume that Assumption 2 holds. Then the invariance probability $p_{s_0}(A)$ for model $\mathfrak{S}$, initialized at $s_0 \in A$, satisfies*

(3.3) $$\left|p_{s_0}(A) - p_{p_0}(A_p)\right| \leq \max\{\gamma_i \delta_i | i = 1, \ldots, m\},$$

where $p_{p_0}(A_p)$ is the invariance probability for the MC $\mathfrak{P}$, initialized at the discrete state $p_0 = \xi(s_0) \in A_p$, where δ_i is the diameter of the set $A_i \subset A$, namely,

$$\delta_i = \max\{\left\|s - s'\right\| \mid s, s' \in A_i\},$$

and the constants γ_i are specified as $\gamma_i = N\mathcal{K}_i$, as per Theorem 3.3.

Proof. Let us recall that the function $\xi : A \to A_p$ maps any point $s \in A$ to the corresponding discrete state $z \in A_p$ via a representative point $\xi(s)$, and that $\Xi : A_p \to 2^A$ associates a continuous partition set to a discrete (representative) point in A_p. Let us define a piecewise constant function $\hat{V}_k^p : A \to [0,1]$ with $\hat{V}_k^p(s) = V_k^p(\xi(s))\ \forall s \in A$.

Next we show that

(3.4) $$|V_k(s) - \hat{V}_k^p(s)| \le (N-k)\max\{\mathcal{K}_i\delta_i|i = 1,\ldots,m\}.$$

For $k = N$ the inequality is trivial, since $|V_k(s) - \hat{V}_k^p(s)| = |V_N(s) - \hat{V}_N^p(s)| = 1 - 1 = 0$. Suppose now that the inequality holds for $k + 1$; then at time step k we have

$$|V_k(s) - \hat{V}_k^p(s)| = |V_k(s) - \hat{V}_k^p(\xi(s))| \le |V_k(s) - V_k(\xi(s))| + \left|V_k(\xi(s)) - \hat{V}_k^p(\xi(s))\right|$$

$$\le \mathcal{K}_{\underline{i}}\delta_{\underline{i}} + \left|V_k(\xi(s)) - \hat{V}_k^p(\xi(s))\right|,$$

where the index $\underline{i} \in \{1,\ldots,m\}$ corresponds to the set $A_{\underline{i}} = \Xi(\xi(s))$. On the other hand, by exploiting the discrete feature of the function $\hat{V}_k^p$ evaluated at $\xi(s)$ and its piecewise constant structure, we can observe that

$$\hat{V}_k^p(\xi(s)) = \sum_{z\in A_p}\hat{V}_{k+1}^p(z)T_p(\xi(s),z) = \sum_{z\in A_p}\hat{V}_{k+1}^p(z)\int_{\Xi(z)}T_s(d\omega|\xi(s)) = \int_A \hat{V}_{k+1}^p(\omega)T_s(d\omega|\xi(s)),$$

which results in the following inequality:

$$\left|V_k(\xi(s)) - \hat{V}_k^p(\xi(s))\right| = \left|\int_A V_{k+1}(\omega)T_s(d\omega|\xi(s)) - \sum_{z\in A_p}\hat{V}_{k+1}^p(z)T_p(\xi(s),z)\right|$$

$$= \left|\int_A V_{k+1}(\omega)T_s(d\omega|\xi(s)) - \int_A \hat{V}_{k+1}^p(\omega)T_s(d\omega|\xi(s))\right|$$

$$\le \int_A \left|V_{k+1}(\omega) - \hat{V}_{k+1}^p(\omega)\right|T_s(d\omega|\xi(s)).$$

We then obtain

$$|V_k(s) - \hat{V}_k^p(s)| \le \mathcal{K}_{\underline{i}}\delta_{\underline{i}} + \int_A \left|V_{k+1}(\omega) - \hat{V}_{k+1}^p(\omega)\right|T_s(d\omega|\xi(s))$$

$$\le \mathcal{K}_{\underline{i}}\delta_{\underline{i}} + (N-k-1)\max_i\{\mathcal{K}_i\delta_i\}\underbrace{\int_A T_s(d\omega|\xi(s))}_{\le 1} \le (N-k)\max_i\{\mathcal{K}_i\delta_i\}.$$

The results in Theorems 2.1 and 3.1 and inequality (3.4) applied at $k = 0$ yield the following bound:

$$|p_{s_0}(A) - p_{p_0}(A_p)| = |V_0(s_0) - V_0^p(\xi(s_0))| = |V_0(s_0) - \hat{V}_0^p(s_0)|$$

$$\le N\max\{\mathcal{K}_i\delta_i|i = 1,\ldots,m\} = \max\{\gamma_i\delta_i|i = 1,\ldots,m\},$$

which concludes the proof of the statement. ■

Notice that often in practice the (global or local) Lipschitz constants need to be numerically computed or overapproximated, which relates to a computational cost. This leads us to propose a simplification of Assumption 2 and an adaptation of Theorems 3.3 and 3.4 accordingly. The new requirement is computationally less demanding; however, as expected, the related error bounds will be more conservative (less tight).

Assumption 3. *Assume that the kernel* T_s *admits density* t_s *and that the following holds for a choice of a finite positive* $h(\cdot)$:

$$\left| t_s(\bar{s}|s) - t_s(\bar{s}|s') \right| \le h(i) \left\| s - s' \right\| \quad \forall \bar{s} \in A, \forall s, s' \in A_i,$$

where $i = 1, \ldots, m$ *and* A_i *form a partition of* A *(as obtained, for instance, from Algorithm 1).*

Theorem 3.5. *Suppose the stochastic kernel of the model* $\mathfrak{S}$ *satisfies Assumption 3. Then the value functions* $V_k : \mathcal{S} \to [0,1]$, *characterizing the probabilistic invariance problem for the model* $\mathfrak{S}$ *over* $A \in \mathcal{B}(\mathcal{S})$, *satisfy the following Lipschitz continuity,* $k \in [0, N]$:

$$\left| V_k(s) - V_k(s') \right| \le \mathcal{K}_i \left\| s - s' \right\|$$

$\forall s, s' \in A_i, i \in \{1, \ldots, m\}$, *where the constant* $\mathcal{K}_i$ *is given by*

$$\mathcal{K}_i = h(i)\mathscr{L}(A),$$

and where $\mathscr{L}(B)$ *denotes the Lebesgue measure of any set* $B \in \mathcal{B}(\mathcal{S})$.

Proof. The proof can be directly adapted from that of Theorem 3.3, in particular noticing that $|V_k(s) - V_k(s')| \le \int_A |T_s(d\bar{s}|s) - T_s(d\bar{s}|s')| \le h(i) \left\| s - s' \right\| \mathscr{L}(A)$. ■

Theorem 3.6. *Under Assumption 3 the invariance probability* $p_{s_0}(A)$ *for the model* $\mathfrak{S}$, *initialized at* $s_0 \in A$, *satisfies*

$$(3.5) \qquad \qquad \left| p_{s_0}(A) - p_{p_0}(A_p) \right| \le \max\{\gamma_i \delta_i | i = 1, \ldots, m\},$$

where $p_{p_0}(A_p)$ *is the invariance probability for the MC* $\mathfrak{P}$ *initialized at the discrete state* $p_0 = \xi(s_0) \in A_p$, *the constants* $\gamma_i = N\mathcal{K}_i$, *as per Theorem 3.5, and where* δ_i *is the diameter of the set* $A_i \subset A$: $\delta_i = \max\{\left\| s - s' \right\| | s, s' \in A_i\}$.

Proof. The proof can be directly adapted from that of Theorem 3.4. ■

3.3.2. Variable rescaling and direct computation. We are interested in the application of the abstraction bounds on models with kernels that present ill-conditioned dynamics, that is, dynamics operating on multiple spatial scales or characterized by both slow and fast variables. This goal will be further clarified in light of the algorithmic procedures discussed in section 5. We start by investigating whether a rescaling of the dynamics affects the abstracted MC and the associated computation of the local error, according to Assumption 2 (as needed the results can be easily tailored to the other two presented assumptions). Let us consider a stochastic kernel endowed with a density function $t_s(\bar{s}|s)$, and let us transform the state-space by applying a linear map $s = Pr$, where P is an invertible matrix.[1]

---

[1]We leave to the reader the extension to an affine transformation, namely, $s = Pr + Q$, where Q is properly sized. It is possible to verify that the properties discussed below are shift invariant and to adapt them to the affine case accordingly.

A generic set $A_i \in \mathcal{B}(\mathcal{S})$ is mapped into $\tilde{A}_i = \{r \in \mathcal{S} : s = Pr, s \in A_i\}$, which is such that $\mathcal{L}(\tilde{A}_i) = |\det(P^{-1})|\mathcal{L}(A_i)$. Furthermore, the new density function $t_r(\bar{r}|r)$ is related to the original one $t_s(\bar{s}|s)$ by the equality

$$(3.6) \qquad t_r(\bar{r}|r) = |J(r)| \, t_s \left(P\bar{r}|Pr \right),$$

where $|J(r)|$ denotes absolute value of the determinant of the Jacobian

$$J(r) = \left| \frac{\partial(s_1, \ldots, s_n)}{\partial(r_1, \ldots, r_n)} \right| = \det \begin{bmatrix} \frac{\partial s_1}{\partial r_1} & \cdots & \frac{\partial s_1}{\partial r_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial s_n}{\partial r_1} & \cdots & \frac{\partial s_n}{\partial r_n} \end{bmatrix} = \det(P).$$

Suppose that the representative points $z_i, i = 1, \ldots, m$, of the abstracted MC are also mapped to points $v_i : z_i = Pv_i$, which leads to the entries of a new transition probability matrix T_p specified, for any $j = 1, \ldots, m$, by

$$T_p(v_i, \Xi(v_j)) = \int_{\tilde{A}_j} t_r(\bar{r}|v_i)d\bar{r} = \int_{\tilde{A}_j} |\det(P)| \, t_s \left(P\bar{r}|Pv_i \right) d\bar{r} = \int_{A_j} t_s(\bar{s}|z_i)d\bar{s} = T_s(z_i, \Xi(z_j)).$$

This shows that the MCs obtained from the original and from the rescaled Markov processes are equivalent.

With a focus on Assumption 2, we compute the local Lipschitz constants of the new conditional distribution. Notice that the Lipschitz constant of a function is not uniquely defined, since any finite upper bound is also a legitimate Lipschitz constant. As a result, the abstraction error depends on the method employed to compute local Lipschitz constants. A common method for computation of the Lipschitz constant is maximization of the Euclidean norm of the function gradient. Recall from Assumption 2 that $|t_s(\bar{s}|s) - t_s(\bar{s}|s')| \leq h(i, j)\|s - s'\|$, and assume that the following method is used to compute $h(i, j)$:[2]

$$h(i, j) = \max_{s \in A_i, \bar{s} \in A_j} \left\| \frac{\partial t_s}{\partial s}(\bar{s}|s) \right\|.$$

Then, in the new coordinates, we have that $|t_r(\bar{r}|r) - t_r(\bar{r}|r')| \leq \tilde{h}(i, j)\|r - r'\|$, with the following Lipschitz constant:

$$\tilde{h}(i, j) = \max_{r \in \tilde{A}_i, \bar{r} \in \tilde{A}_j} \left\| \frac{\partial t_r}{\partial r}(\bar{r}|r) \right\|.$$

Let us relate these two Lipschitz constants using (3.6) and applying the chain rule in the computation of partial derivatives:

$$\tilde{h}(i, j) = \max_{r \in \tilde{A}_i, \bar{r} \in \tilde{A}_j} \left\| \frac{\partial t_r}{\partial r}(\bar{r}|r) \right\| = |\det(P)| \max_{r \in \tilde{A}_i, \bar{r} \in \tilde{A}_j} \left\| \frac{\partial}{\partial r} t_s \left(P\bar{r}|Pr \right) \right\|$$

$$= |\det(P)| \max_{r \in \tilde{A}_i, \bar{r} \in \tilde{A}_j} \left\| \frac{\partial t_s}{\partial s} \left(P\bar{r}|Pr \right) P \right\| = |\det(P)| \max_{s \in A_i, \bar{s} \in A_j} \left\| \frac{\partial t_s}{\partial s} \left(\bar{s}|s \right) P \right\|.$$

Then $\tilde{h}$ differs from h over two terms:

---

[2]In the following, we assume that all the optimization problems have been computed over the closure of the corresponding optimization domain. However, for the sake of notation, we simply refer to the optimization domains as they are given.

- The constant term $|\det(P)|$. This constant has no effect on the computation of the abstraction error (cf. terms $\mathcal{K}_i$ in Theorem 3.4), since $\mathcal{L}(\tilde{A}_j) = |\det(P^{-1})|\mathcal{L}(A_j)$. Without loss of generality we can then restrict our attention to matrices with determinant that is equal to one.
- The matrix P within the norm. It provides a weighted sum of the partial derivatives. We can exploit this matrix in order to balance the partial derivatives over different directions. In particular, this scaling matrix can be useful in the presence of ill-conditioned dynamics.

With the above discussion we have argued that the Lipschitz constant depends on the coordinates where the distribution function is defined. Since we are interested in the value of the Lipschitz constant as part of the approximation error formula (as per Theorem 3.4), rescaling provides a degree of freedom in the error computation. This is discussed in the following theorem, which emphasizes improvements of the approximation error bounds, again focusing on Assumption 2.

Theorem 3.7. *Consider the conditional distribution t_s of $\mathfrak{S}$, any set $A \in \mathcal{B}(\mathcal{S})$, a partition $\cup_{i=1}^m A_i$ of A, and a properly sized, square invertible matrix P. Then for all points $s, s' \in A_i, \bar{s} \in A_j$, it holds that*

$$(3.7) \qquad \max_{s \in A_i, \bar{s} \in A_j} \left\| \frac{\partial t_s}{\partial s}(\bar{s}|s) \right\| \max_{s,s' \in A_i} \|s - s'\|$$

$$(3.8) \qquad \geq \min_P \left(\max_{s \in A_i, \bar{s} \in A_j} \left\| \frac{\partial t_s}{\partial s}(\bar{s}|s)P \right\| \max_{s,s' \in A_i} \|P^{-1}(s - s')\| \right)$$

$$(3.9) \qquad \geq \max_{\bar{s} \in A_j} \max_{s,s',\zeta \in A_i} \left| \frac{\partial t_s}{\partial s}(\bar{s}|\zeta)(s - s') \right|$$

$$(3.10) \qquad \geq |t_s(\bar{s}|s) - t_s(\bar{s}|s')|.$$

Proof. The inequality $(3.7) \geq (3.10)$ is employed in the bound discussed in Theorem 3.4 and is based on the maximum norm of the partial derivatives. The expression in $(3.8) \geq (3.10)$ is based on the idea of rescaling the state-space as follows: transform the inequality $|t_r(\bar{r}|r) - t_r(\bar{r}|r')| \leq \tilde{h}(i,j)\|r - r'\|$ into

$$|\det(P)| \, |t_s\left(P\bar{r}|Pr\right) - t_s\left(P\bar{r}|Pr'\right)| \leq \tilde{h}(i,j)\|r - r'\|,$$

which leads to

$$|t_s\left(\bar{s}|s\right) - t_s\left(\bar{s}|s'\right)| \leq \frac{\tilde{h}(i,j)}{|\det(P)|}\|r - r'\| = \max_{s \in A_i, \bar{s} \in A_j} \left\| \frac{\partial t_s}{\partial s}\left(\bar{s}|s\right)P \right\| \|P^{-1}(s - s')\|.$$

Selecting the matrix P to be equal to the identity matrix leads to the inequality $(3.7) \geq (3.8)$. The mean value theorem for scalar fields implies the last inequality, namely, $(3.9) \geq (3.10)$.

Let us finally relate (3.8) to (3.9) by using the Cauchy–Schwarz inequality:

$$\min_P \left(\max_{s \in A_i, \bar{s} \in A_j} \left\| \frac{\partial t_s}{\partial s}(\bar{s}|s)P \right\| \max_{s,s' \in A_i} \left\| P^{-1}(s-s') \right\| \right)$$

$$= \min_P \left(\max_{s,s',\zeta \in A_i, \bar{s} \in A_j} \left\| \frac{\partial t_s}{\partial s}(\bar{s}|\zeta)P \right\| \left\| P^{-1}(s-s') \right\| \right)$$

$$\geq \max_{s,s',\zeta \in A_i, \bar{s} \in A_j} \left(\min_P \left\| \frac{\partial t_s}{\partial s}(\bar{s}|\zeta)P \right\| \left\| P^{-1}(s-s') \right\| \right)$$

$$\geq \max_{s,s',\zeta \in A_i, \bar{s} \in A_j} \left(\min_P \left| \frac{\partial t_s}{\partial s}(\bar{s}|\zeta)PP^{-1}(s-s') \right| \right)$$

$$= \max_{s,s',\zeta \in A_i, \bar{s} \in A_j} \left| \frac{\partial t_s}{\partial s}(\bar{s}|\zeta)(s-s') \right|.$$

This concludes the proof. ∎

The above theorem does not pose any restriction on the choice of the invertible matrix P. Notice that the bound in (3.8) is invariant under constant multiplications of matrix P: we can then reduce the optimization domain to the set of square matrices with $|\det(P)| = 1$. As an alternative to the above bounds, which hinge on the computation of quantities related to the Lipschitz constant, we present the next result.

Corollary 3.8. *Consider the conditional distribution t_s of $\mathfrak{S}$, any set $A \in \mathcal{B}(\mathcal{S})$, and a partition $\cup_{i=1}^m A_i$ of A. The continuous conditional distribution $t_s(\bar{s}|s)$ satisfies the inequality*

$$(3.11) \qquad |t_s(\bar{s}|s) - t_s(\bar{s}|s')| \leq \max_{\bar{s} \in A_j} \left[\max_{s \in A_i} t_s(\bar{s}|s) - \min_{s \in A_i} t_s(\bar{s}|s) \right] \qquad \forall s, s' \in A_i, \forall \bar{s} \in A_j.$$

Proof. The distribution is assumed to be continuous over the closure of A_i; hence it admits finite maximum and minimum, which leads to the following:

$$\max_{s,s' \in A_i, \bar{s} \in A_j} |t_s(\bar{s}|s) - t_s(\bar{s}|s')| = \max_{\bar{s} \in A_j} \left[\max_{s,s' \in A_i} |t_s(\bar{s}|s) - t_s(\bar{s}|s')| \right]$$

$$= \max_{\bar{s} \in A_j} \left[\max_{s \in A_i} t_s(\bar{s}|s) - \min_{s \in A_i} t_s(\bar{s}|s) \right]. \qquad ∎$$

Notice that the quantity in (3.11) provides the optimal (lowest) upper bound over (3.7)–(3.9), since (3.11) represents a particular instantiation of (3.10) and we have shown that (3.10) ≤ (3.9).

Owing to the emphasis of this work on numerics, let us focus on the overhead associated to the computation of the presented bounds. Assume that we are given a Cartesian partition of the safe set A, which will be the underlying assumption for the algorithms developed in section 5. This enables an analytic expression of the distance between points in (3.7)–(3.9). Therefore, the upper bounds (3.7) and (3.9) are clearly related to the same computational cost (maximization over the variables appearing in the partial derivatives). With regard to the bound based on (3.8), the cost is also the same if a specific matrix P is selected; on the contrary, the optimization over this matrix increases the computational overhead. In general,

matrix P in (3.8) can be treated either as an optimization variable or, as discussed, as a transformation matrix for improving the effect of widely separated dynamics. The additional bound in (3.11), which does not depend on the computation of the Lipschitz constant, requires an optimization over three variables, and as such it is computationally heavier than (3.7) and (3.9); however, it can be matched to their complexity—at the expense of losing tightness—by the following simplification:

$$(3.12) \qquad \max_{\bar{s} \in A_j} \left[\max_{s \in A_i} t_s(\bar{s}|s) - \min_{s \in A_i} t_s(\bar{s}|s) \right] \leq \max_{s \in A_i, \bar{s} \in A_j} t_s(\bar{s}|s) - \min_{s \in A_i, \bar{s} \in A_j} t_s(\bar{s}|s).$$

Finally, notice that, while the quantities defined in (3.7)–(3.9) are proportional to the size of the partition sets, that in (3.11) is not. In the following, either of the bounds in (3.7)–(3.9) will be used to construct an adaptive partition, thereafter employing the improved bound in (3.11) as an a posteriori analysis of the abstraction error.

Let us summarize the results in Theorem 3.7 and Corollary 3.8: we have provided four different methods for computing a local upper bound; call it $k(i,j)$, as $|t_s(\bar{s}|s) - t_s(\bar{s}|s')| \leq k(i,j)$, for any $s, s' \in A_i, \bar{s} \in A_j, i, j = 1, \ldots, m$. The upper bound $k(i,j)$, in form (3.7)–(3.9) or (3.11), can be directly used to quantify the abstraction errors in Theorems 3.3 and 3.4 as

$$(3.13) \qquad E = \max_{i=1,\ldots,m} \left\{ \sum_{j=1}^{m} N k(i,j) \mathcal{L}(A_j) \right\}.$$

Notice the difference between the bound $k(i,j)$ and the quantity $h(i,j)$ (Lipschitz constant) as used in Theorems 3.3 and 3.4. Similar to $h(i,j)$, which can be relaxed to $h(i)$ as discussed in Assumption 3, the formulas for $k(i,j)$ can also be relaxed: for instance, the inequality (3.10) $\leq$ (3.9) would become

$$|t_s(\bar{s}|s) - t_s(\bar{s}|s')| \leq \max_{\bar{s} \in A} \max_{s, s', \zeta \in A_i} \left| \frac{\partial t_s}{\partial s}(\bar{s}|\zeta)(s - s') \right|.$$

A similar adaptation can be applied over global bounds based on Assumption 1.

Let us remark that Assumptions 2 and 3, based on local Lipschitz continuity, yield error bounds that are not only tighter than their global counterpart but also practically less conservative. Discontinuous density functions are in fact not globally Lipschitz continuous and thus do not satisfy Assumption 1; however, they can satisfy Assumptions 2 and 3 if the discontinuity points lie on the boundaries of the partition sets: this requirement can then be satisfied by a proper selection of these sets. While we do not focus further on discontinuous kernels in the rest of the manuscript, this discussion hints at the application of the abstraction procedure to a wider range of models—for instance, models endowed with kernels derived from data. Of course this comes at the expense of a more elaborate (and likely slower) partitioning procedure. Along these lines, the Lipschitz continuity assumptions over the densities can be generalized by looking at Lipschitz continuity over the kernels instead: more precisely, in the case of Assumption 3 we would obtain

$$(3.14) \qquad \int_A |t_s(\bar{s}|s) - t_s(\bar{s}|s')| d\bar{s} \leq H(i) \quad \forall s, s' \in A_i, \, i = 1, \ldots, m.$$

The global error becomes then $E = \max_{i=1,\ldots,m} NH(i)$. This assumption is practically less conservative since it allows dealing with discontinuous conditional density functions, regardless of the chosen partitioning procedure. In contrast to (3.13), the error bound based on (3.14) does not explicitly depend on the Lebesgue measure of the partition sets and then provides a tighter upper bound for the error. On the other hand, the computation of the parameters $H(i)$ in (3.14) requires an increased effort: the maximization needs to be performed over two variables (s, s'), and each function evaluation requires a numerical integration. As we shall see in the experiments of section 6.2, the numerical integration makes the computation much more time consuming than the other methods developed above. In conclusion, the bound in (3.14) provides a tighter error bound and can lead to memory savings but also to more time consuming algorithms.

4. Application to stochastic hybrid systems (SHS). In this section we tailor the presented results on error bounds for the abstraction around models endowed with a particular state-space that is "hybrid" in nature [3]; namely, we select

$$\mathcal{S} = \cup_{q \in \mathcal{Q}} \{q\} \times \mathbb{R}^{n(q)}$$

to be the disjoint union of continuous domains over a finite, discrete set of locations (or modes) $\mathcal{Q} = \{q_1, q_2, \ldots, q_\mathfrak{m}\}$. The continuous domains have a dimension $n(q)$ that is mode dependent and characterized by a bounded function $n : \mathcal{Q} \to \mathbb{N}$.

Given a hybrid point $s = (q, x) \in \mathcal{S}$ and a Borel measurable set $A = \cup_{q \in \mathcal{Q}} \{q\} \times A_q, A \in \mathcal{B}(\mathcal{S})$, the stochastic kernel T_s is further specified as follows [3]:

$$(4.1) \qquad T_s(\{\bar{q}\} \times A_{\bar{q}}|(q,x)) = T_q(\bar{q}|(q,x)) \times \begin{cases} T_x(A_{\bar{q}}|(q,x)) & \text{if } \bar{q} = q, \\ T_r(A_{\bar{q}}|(q,x), \bar{q}) & \text{if } \bar{q} \neq q. \end{cases}$$

Here T_s is made up of three distinct conditional kernels. $T_q : \mathcal{Q} \times \mathcal{S} \to [0, 1]$ assigns to each $s \in \mathcal{S}$ a discrete probability distribution $T_q(\cdot|s)$ over $\mathcal{Q}$. Based on a sample of T_q, if the selected location $\bar{q}$ coincides with the current mode q, then $T_x : \mathcal{B}(\mathbb{R}^{n(\cdot)}) \times \mathcal{S} \to [0, 1]$ assigns to each $s \in \mathcal{S}$ a probability measure $T_x(\cdot|s)$ over the continuous domain associated with $q \in \mathcal{Q}$. On the other hand, if $\bar{q} \neq q$, then $T_r : \mathcal{B}(\mathbb{R}^{n(\cdot)}) \times \mathcal{S} \times \mathcal{Q} \to [0, 1]$ assigns to each $s \in S$ and $\bar{q} \in \mathcal{Q}$ a probability measure $T_r(\cdot|s, \bar{q})$ over the continuous domain associated with $\bar{q} \in \mathcal{Q}$.

We shall denote such a discrete time stochastic hybrid model $\mathfrak{S} = (\mathcal{Q}, n, T_q, T_x, T_r)$, and we refer the reader to [3] for technical details on its topological and measurability properties and for an algorithmic definition of its execution. Section 6.2 develops a case study based on an SHS model.

4.1. Abstraction and error computation. The abstraction of an SHS as an MC follows the same reasoning as in section 3.1. Consider the hybrid safe set $A \in \mathcal{B}(\mathcal{S}), A = \cup_{q \in \mathcal{Q}} \{q\} \times A_q$, with $A_q \in \mathcal{B}(\mathbb{R}^{n(q)})$. For all $q \in \mathcal{Q}$, select a finite (m_q-dimensional) partition of the local set A_q as $A_q = \cup_{i=1}^{m_q} A_{q,i}$ ($A_{q,i}$ are nonoverlapping). For each $A_{q,i}$, select a single representative point $(q, z_{q,i}) \in A_{q,i}$, and redefine $A_p = \{(q, z_{q,i})|i \in [1, m_q], q \in \mathcal{Q}\}$. Focusing on bounds based on the Lipschitz constant of densities, the following is an extension of Assumption 2 to the SHS framework.

Assumption 4. *Assume that the kernels T_x, T_r admit densities t_x, t_r, respectively, and that the following continuity assumptions are valid:*

$$\left| T_q(\bar{q}|(q,x)) - T_q(\bar{q}|(q,x')) \right| \le h_q(q,\bar{q},i) \left\| x - x' \right\| \quad \forall x, x' \in A_{q,i},$$
$$\left| t_x(\bar{x}|(q,x)) - t_x(\bar{x}|(q,x')) \right| \le h_x(q,i,j) \left\| x - x' \right\| \quad \forall \bar{x} \in A_{q,j}, \forall x, x' \in A_{q,i},$$
$$\left| t_r(\bar{x}|(q,x),\bar{q}) - t_r(\bar{x}|(q,x'),\bar{q}) \right| \le h_r(q,\bar{q},i,k) \left\| x - x' \right\|, \quad \bar{q} \ne q, \forall \bar{x} \in A_{\bar{q},k}, \forall x, x' \in A_{q,i},$$

where $q, \bar{q} \in \mathcal{Q}$; $i, j \in 1, \dots, m_q$; $k \in 1, \dots, m_{\bar{q}}$; and $h_q(\cdot), h_x(\cdot), h_r(\cdot)$ are finite positive constants.

Let us consider an SHS model, a (hybrid) invariant set $A \in \mathcal{B}(\mathcal{S})$, a finite time horizon $[0, N]$, a point $s_0 \in \mathcal{S}$, and an abstraction procedure over $\mathfrak{S}$. The error between the value $p_{p_0}(A_p)$ for the MC from $p_{s_0}(A)$ for the SHS can be quantified as follows.

Theorem 4.1. *Assume that Assumption 4 holds. Then the invariance probability $p_{s_0}(A)$ for the SHS $\mathfrak{S}$, initialized at $s_0 \in A$, satisfies*

$$(4.2) \qquad |p_{s_0}(A) - p_{p_0}(A_p)| \le \max\{\gamma_{q,i}\delta_{q,i} | i = 1, \dots, m_q, q \in \mathcal{Q}\},$$

where $p_{p_0}(A_p)$ is the invariance probability for the MC $\mathfrak{P}$, initialized at the discrete state $p_0 = \xi(s_0) \in A_p$, where $\delta_{q,i}$ is the diameter of the set $A_{q,i} \subset A_q$, namely,

$$\delta_{q,i} = \max\{\left\| x - x' \right\| \,|\, x, x' \in A_{q,i}\},$$

and the constants $\gamma_{q,i}$ are specified as $\gamma_{q,i} = N\mathcal{K}_{q,i}$, where

$$\mathcal{K}_{q,i} = \sum_{j=1}^{m_q} h_x(q,i,j)\mathscr{L}(A_{q,j}) + \sum_{\bar{q} \in Q} h_q(q,\bar{q},i) + \sum_{\bar{q} \ne q} \sum_{k=1}^{m_{\bar{q}}} h_r(q,\bar{q},i,k)\mathscr{L}(A_{\bar{q},k}),$$

and where $\mathscr{L}(B)$ denotes the Lebesgue measure of any set $B \in \mathcal{B}(\mathcal{S})$.

Proof. Inequality (3.1) corresponds to the following:

$$\left| V_k(q,x) - V_k(q,x') \right| \le \int_{A_q} \left| T_q(q|(q,x))t_x(\bar{x}|(q,x)) - T_q(q|(q,x'))t_x(\bar{x}|(q,x')) \right| d\bar{x}$$
$$+ \sum_{\bar{q} \ne q} \int_{A_{\bar{q}}} \left| T_q(\bar{q}|(q,x))t_r(\bar{x}|(q,x),\bar{q}) - T_q(\bar{q}|(q,x'))t_r(\bar{x}|(q,x'),\bar{q}) \right| d\bar{x}.$$

As in Theorem 3.3, the local Lipschitz continuity of the value functions is established by

Assumption 4: for all $x, x' \in A_{q,i}$,

$$
\begin{aligned}
\left| V_k(q, x) - V_k(q, x') \right| &\leq \int_{A_q} T_q(q|(q, x)) \left| t_x(\bar{x}|(q, x)) - t_x(\bar{x}|(q, x')) \right| d\bar{x} \\
&+ \int_{A_q} t_x(\bar{x}|(q, x')) \left| T_q(q|(q, x)) - T_q(q|(q, x')) \right| d\bar{x} \\
&+ \sum_{\bar{q} \neq q} \int_{A_{\bar{q}}} T_q(\bar{q}|(q, x)) \left| t_r(\bar{x}|(q, x), \bar{q}) - t_r(\bar{x}|(q, x'), \bar{q}) \right| d\bar{x} \\
&+ \sum_{\bar{q} \neq q} \int_{A_{\bar{q}}} t_r(\bar{x}|(q, x'), \bar{q}) \left| T_q(\bar{q}|(q, x)) - T_q(\bar{q}|(q, x')) \right| d\bar{x} \\
&\leq \sum_{j=1}^{m_q} h_x(q, i, j) \mathcal{L}(A_{q,j}) + h_q(q, q, i) \\
&+ \sum_{\bar{q} \neq q} \sum_{k=1}^{m_{\bar{q}}} h_r(q, \bar{q}, i, k) \mathscr{L}(A_{\bar{q},k}) + \sum_{\bar{q} \neq q} h_q(q, \bar{q}, i) = \mathcal{K}_{q,i}.
\end{aligned}
$$

The rest of the proof follows the same reasoning as the proof of Theorem 3.4. ■

5. Algorithms for abstraction. In the previous sections we considered arbitrary partitions of the state-space, and, with a focus on the problem of finite time probabilistic invariance over a given set A, we derived bounds between the exact value $p_{s_0}(A)$ and the approximation $p_{p_0}(A_p)$, based, respectively, on the model $\mathfrak{S}$ and on its MC abstraction $\mathfrak{P}$. In this section we focus on a few alternative techniques for the generation of the abstraction $\mathfrak{P} = (\mathcal{P}, T_p)$ from $\mathfrak{S} = (\mathcal{S}, T_s)$. We explicitly exploit the knowledge of the (local) error to adapt the abstraction to the underlying dynamics of $\mathfrak{S}$, as well as to the invariance problem of interest. Since the approach can be extended to more general specifications, expressed as formulas in a particular probabilistic modal logic [2], the approach effectively allows for a formula-based abstraction of stochastic models.

In order to maintain focus and keep the notation simple, we present the procedures in the case where no rescaling of the state-space has been performed. The abstraction procedure consists of two main steps (see Algorithm 1):

    1. grid generation, namely, the partitioning of $\mathcal{S}$ that yields $\mathcal{P}$; and

    2. marginalization of T_s, which leads to T_p.

We proceed with the analysis of these two successive items.

5.1. Grid generation. Let us first focus on the state-space partitioning, which involves the generation of a grid. The grid can be either uniform and generated instantaneously [1] or variable and generated adaptively. More precisely, for the problem at hand the generation of a uniform grid leverages the explicit knowledge of the global error of Theorem 3.2 and is thus instantaneous. On the other hand, the adaptive partitioning requires knowledge of errors that are local to the existing partition sets (see Theorems 3.4 and 3.6) and proceeds via a progressive refinement of the grid. We will thus sequentially perform adaptive gridding under either Assumption 2 or Assumption 3 (which gives errors that are less tight compared to Assumption 2) over the existing partition sets, whereas Assumption 1 will be associated

to the generation of a uniform gridding [1]. Comparing Assumption 2 against Assumption 3, we will argue that the first ensures tighter error bounds (which leads to smaller cardinality of the partition) but requires error updates for possibly all of the cells during each refinement step (whereas the second will perform just local updates) and is thus computationally more complex.

Let us discuss a few details about adaptive grid generation. Consider for the sake of discussion an n-dimensional model. There are two main options in the shape of the cells of a grid [23, 29]: n-dimensional simplices and Cartesian hyperrectangles. The first option leads to the known Kuhn triangulation [29] and is widely used in numerical solution of partial differential equations. The second approach generates hyperrectangular cells aligned with the main axes; this option, for our problem, appears to be advantageous. Cartesian cells, in fact, better accommodate the subsequent step that involves the marginalization of probability laws, which generates the transition probability matrix T_s. Marginalization over general convex polygons (in particular simplices) is known to be a computationally expensive problem [28].

With a focus on the refinement step, consider a single Cartesian cell. We are again presented with two options for its further partitioning: to replace the cell with 2^n smaller cells by splitting it along its centroid; or to replace the cell with two smaller cells by partitioning along one axis. The second approach is also known as the variable resolution approach [23]. While the first approach decreases the error (which depends on the cell diameter; see Theorems 3.4 and 3.6) faster than the second, it is also associated with the generation of partitions with larger cardinality. Since we aim at economizing the memory usage, we opt for the second option. Based on this choice, the convergence speed of the procedure is optimized by selecting the longest axis for the partitioning. This leads to the following result.

Proposition 5.1. *For an n-dimensional model, the convergence rate of the computed error bound for a partitioning procedure based on a Cartesian grid that proceeds by splitting the longest axis is lower bounded by the factor $\sqrt{1 - \frac{3}{4n}}$.*

Proof. Consider an n-dimensional hyperrectangle, characterized by edges of length $a_i > 0, i = 1, \ldots, n$, which is to be split along (any) the longest axis $a_k = \max_{1 \leq i \leq n} a_i$. The refinement step affects its diameter $\delta = \sqrt{a_1^2 + a_2^2 + \cdots + a_k^2 + \cdots + a_n^2}$ as follows:

$$\delta' = \sqrt{a_1^2 + a_2^2 + \cdots + \left(\frac{a_k}{2}\right)^2 + \cdots + a_n^2}.$$

Introduce the quantity $A = \left(a_1^2 + a_2^2 + \cdots + a_{k-1}^2 + a_{k+1}^2 \cdots + a_n^2\right)$; then

$$\frac{\delta'}{\delta} = \sqrt{\frac{A^2 + (\frac{a_k}{2})^2}{A^2 + a_k^2}} = \sqrt{\frac{\frac{1}{4} + \left(\frac{A}{a_k}\right)^2}{1 + \left(\frac{A}{a_k}\right)^2}} = \sqrt{\frac{\frac{1}{4} + x}{1 + x}},$$

where $x = \left(\frac{A}{a_k}\right)^2 = \left(\frac{a_1}{a_k}\right)^2 + \cdots + \left(\frac{a_n}{a_k}\right)^2$, which leads to $0 < x \leq n - 1$. Since the function $\sqrt{\frac{\frac{1}{4} + x}{1 + x}}$ is monotonically increasing in the interval $(0, n - 1)$, we obtain

$$\frac{\delta'}{\delta} \leq \sqrt{\frac{\frac{1}{4} + n - 1}{1 + n - 1}} = \sqrt{1 - \frac{3}{4n}},$$

which concludes the proof. ∎

The grid generation procedures are formally presented in Algorithm 2 for the uniform error, and in Algorithms 3 and 4 for the local errors. In the first case, the union of the partitioning sets is supposed to include the space $\mathcal{S}$. In the latter case, the initial partition can be any partitioning of the space $\mathcal{S}$, and in particular it can coincide with the state-space $\mathcal{S}$. Furthermore, notice the differences in step 4, which leads us to conclude that Algorithm 3 is geared toward an abstraction with the least number of states, whereas Algorithm 4 aims at faster generation time. More precisely, note than when we split a cell A_i along its main axis the related local error is reduced first because of the decrease in its diameter δ_i and second due to the possible reduction in the local Lipschitz constants $h(i, j)$ (other local errors may also be decreased because of the update of local Lipschitz constants). Hence, if we split a group of cells, as suggested in Algorithm 4, we possibly obtain a larger decrease of the error bound. The actual computation of the errors in the algorithms can be performed based on any of the bounds in section 3.3.

Algorithm 2 Generation of the uniform grid

Require: model $\mathfrak{S} = (\mathcal{S}, T_s)$ under Assumption 1; error threshold ϵ
1: pick a partition diameter δ based on bound (3.2) in Theorem 3.2 and on the threshold ϵ
2: perform partitioning of $\mathcal{S}$ with uniformly packed hypercubes
Ensure: $\mathcal{P}$, error $E = \epsilon$

Algorithm 3 Generation of the adaptive grid

Require: model $\mathfrak{S} = (\mathcal{S}, T_s)$ under Assumption 2 over initial partition; error threshold ϵ
1: set initial partition over the hybrid state-space $\mathcal{S}$
2: compute the error E according to (3.3) in Theorem 3.4
3: **if** $E > \epsilon$ **then**
4:     refine the partition by splitting the single cell with maximum local error along its main axis
5:     go to step 2
6: **end if**
Ensure: $\mathcal{P}$, error $E \leq \epsilon$

5.2. Marginalization. The generation of a grid and the choice of representative points for each of the resulting partition sets (let us recall that the choice of representative points is arbitrary) fully characterize the state-space $\mathcal{P}$ of the MC $\mathfrak{P}$. The second step in the generation of the abstraction involves the computation of the transition probability matrix T_p. This computation necessitates the marginalization of the stochastic kernel T_s, evaluated at the representative points, over the partition sets. While the complexity of the procedure highly depends on the shape of the kernels T_s, we have attempted to alleviate it by (1) working with hyperrectangular partitions, (2) exploiting vectorial representations of the quantities of interest, and (3) leveraging as much as possible the sparsity of the manipulated matrices.

The sparsity of the generated transition probability matrix (number of its nonzero entries) depends on the kernels underlying T_s, particularly on their variance terms. Intuitively, a higher

---

Algorithm 4 Generation of the adaptive grid

Require: model $\mathfrak{S} = (\mathcal{S}, T_s)$ under Assumption 3 over initial partition; error threshold ϵ
  1: set initial partition over the hybrid state-space $\mathcal{S}$
  2: compute the error E according to (3.5) in Theorem 3.6
  3: **if** $E > \epsilon$ **then**
  4:     refine the partition by splitting all the cells with error greater than threshold ϵ along the main axis
  5:     go to step 2
  6: **end if**
Ensure: $\mathcal{P}$, error $E \leq \epsilon$

---

variance relates to a less sparse matrix, since the related probability law is more "spread out." More interestingly, there is a tradeoff between the sparsity of the transition probability matrix and its size, as a function of the variance terms in the underlying dynamics: indeed, both are increased by small variance terms, which are related both to dynamics that are spatially "concentrated" (and thus sparser) and to higher error bounds via the Lipschitz constants.

It is possible to use and tune a tolerance threshold in the marginalization step, below which the transition probabilities are approximated with zero terms. As a last remark, notice that in the uniform partitioning case the marginalization procedure is greatly simplified, given the regular arrangement of the partition cells.

6. Experiments. This section develops a numerical computational benchmark to compare the presented algorithms for abstraction, with a particular focus on grid generation and marginalization steps. Additionally, a case study selects an SHS model and reflects on the choice of the error bounds and on the role of rescaling (cf. section 3.3).

6.1. Computational benchmark. Let us consider an n-dimensional linear, controlled stochastic difference equation

$$x(k+1) = Ax(k) + Bu(k) + w(k), \quad k \in \mathbb{N},$$

where $w(k), k \geq 0$, is the process noise, taken to be normal independent and identically distributed (i.i.d.) with zero mean and covariance W: $w(k) \sim \mathcal{N}(0, W)$. The initial condition $x(0)$ is independent of $w(k), k \geq 0$, and is normal with zero mean and covariance $X : x(0) \sim \mathcal{N}(0, X)$. The input $u(k) \in \mathbb{R}^m, k \geq 0$, is designed according to a state feedback law minimizing the following quadratic cost function of the state and of the input:

$$J = \lim_{N \to \infty} \frac{1}{N} \mathbb{E} \left(\sum_{k=0}^{N-1} \left(x^T(k) Q x(k) + u^T(k) R u(k) \right) \right),$$

with properly sized, positive (semi)definite weighting matrices $Q \succeq 0$ and $R \succ 0$. The optimal control law for this stochastic control problem (also known as the stochastic linear quadratic regulator) is given as a stationary linear state feedback $u(k) = Kx(k)$, where K represents the steady-state feedback gain matrix $K = -\left(R + B^T P_s B \right)^{-1} B^T P_s A$, and P_s is the solution

of the following matrix equation:

$$P_s = Q + A^T P_s A - A^T P_s B \left(R + B^T P_s B\right)^{-1} B^T P_s A.$$

The closed-loop system can be represented as

$$x(k+1) = (A + BK)x(k) + w(k), \quad k \in \mathbb{N},$$

which is a stochastic difference equation evolving over $\mathbb{R}^n$. Given any point $x \in \mathbb{R}^n$ at any time, the distribution at the next time can be characterized by a transition probability kernel $T_x(\cdot|x) \sim \mathcal{N}((A + BK)x, W)$. The computation of the Lipschitz constant of this kernel can be adapted from [1] and involves the calculation of partial derivatives of the density.

With a focus on the closed-loop model, let us consider the probabilistic invariance problem on a safe set defined as $A = [-1, 1]^n$, namely, on a hypercube pointed at the origin, and over a time horizon $[0, N]$. For the cost function, we have selected the weighting matrices $Q = \mathbb{I}_{n \times n}, R = \mathbb{I}_{m \times m}$ (henceforth, $\mathbb{I}_{l \times l}, l \in \mathbb{N}$, will denote the l-dimensional identity matrix). The control dimension has been chosen to be $m = 1$, and the time horizon has been fixed to $N = 10$. The state and control matrices A and B have been randomly generated for each experiment, and A has been further scaled so that $\max_{i=1,\ldots,n} |\lambda_i(A)| = 1$, where $\lambda_i(A)$ denotes the ith eigenvalue of matrix A. The variance of the initial condition has been selected to be $X = 10 \, \mathbb{I}_{n \times n}$.

6.1.1. Grid generation. Let us select a noise variance $W = 0.5 \, \mathbb{I}_{n \times n}$. Figure 1 compares the partition size (i.e., the number of grid cells) generated by Algorithm 3 for the adaptive gridding, and by Algorithm 2 for the uniform one, *given an (upper bound on the) abstraction error ϵ* for all the methods. The horizontal axis represents the threshold ϵ. The error is based on, respectively, (3.3) in Theorem 3.4 and (3.2) in Theorem 3.2. The local Lipschitz constants are computed based on (3.7) in Theorem 3.7. This batch of computations is performed for dimensions $n = 2, 3, 4$. As expected, for the adaptive algorithm the number of generated cells is always less than that for the uniform procedure. Furthermore, the number of cells becomes larger for smaller threshold values ϵ.

Figure 1 also plots the time required to generate the grid according to Algorithm 3 for the adaptive partitioning. The horizontal axis represents again the threshold ϵ on the error. This batch of computations is performed for dimensions $n = 2, 3, 4$ and the results are averaged over 30 runs. The discontinuities discernible in the plots are intrinsic to the implemented refinement algorithm for the adaptive partitioning. Notice that, as expected, the time is larger for smaller thresholds. Recall that for the uniform gridding the grid generation is a one-shot procedure and, as such, independent of the choice of ϵ.

Figure 2 compares the error obtained by generating the adaptive gridding with Algorithm 3 (see Theorem 3.4) against that obtained by generating the uniform gridding of Algorithm 2 (see Theorem 3.2), *given a fixed number of cells* for both methods (these values are represented on the horizontal axis). The experiments are again performed for dimensions $n = 2, 3, 4$. The local Lipschitz constants are computed based on (3.7). It is easily observed that the error associated to the uniform gridding approach is always higher than that associated to the adaptive method. (Notice that, for the probabilistic invariance problem under study, an error greater than one as obtained in the uniform case is not practically useful.)
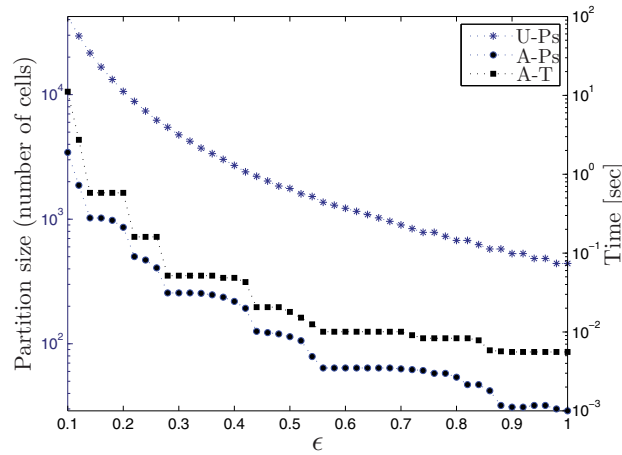
(a) $n = 2$
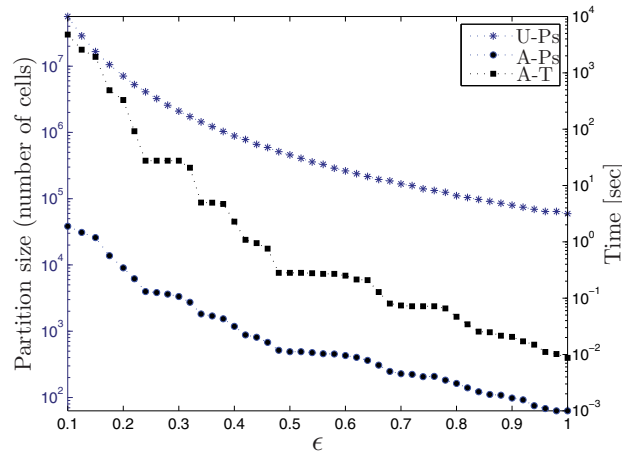
(b) $n = 3$
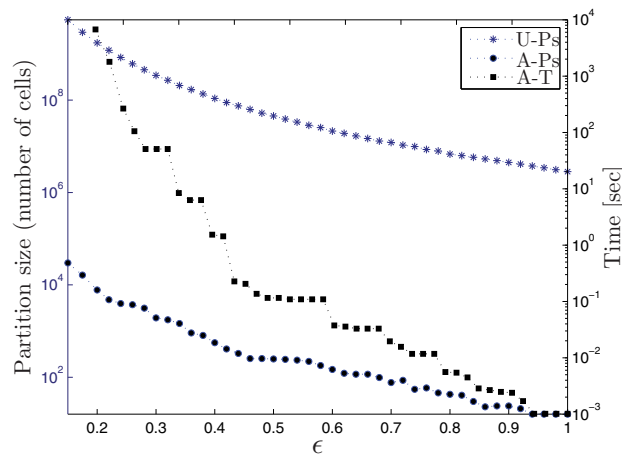
(c) $n = 4$

Figure 1. *Numerical benchmark. For dimensions $n = 2$ (a), $n = 3$ (b), and $n = 4$ (c) and for different levels of the error threshold ϵ (horizontal axis), the plots display partition size (number of cells) generated by adaptive (Algorithm 3, labeled A-Ps) versus uniform gridding (Algorithm 2, labeled U-Ps), as well as time required to generate the adaptive partitioning (Algorithm 3, labeled A-T). The results represent an average over 30 independent runs.*
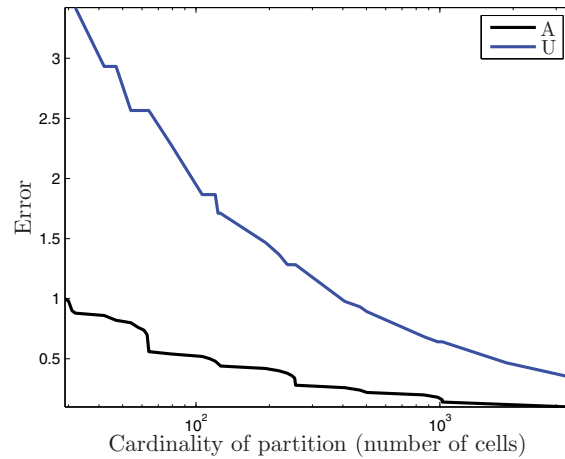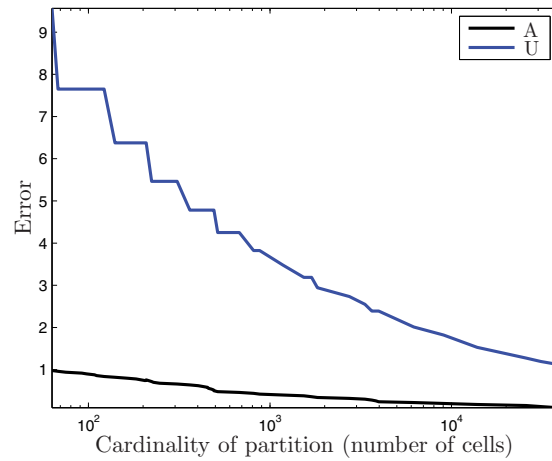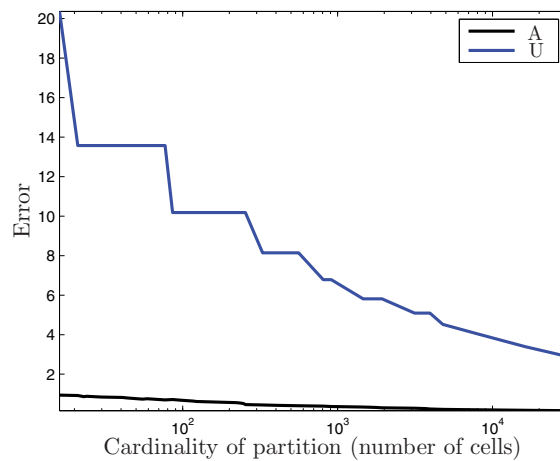
(a) $n = 2$

(b) $n = 3$

(c) $n = 4$

Figure 2. *Numerical benchmark. Errors obtained selecting the same number of cells (same partition size), for dimensions $n = 2$ (a), $n = 3$ (b), $n = 4$ (c), for the adaptive gridding of Algorithm 3 (labeled A) versus the uniform gridding of Algorithm 2 (labeled U).*
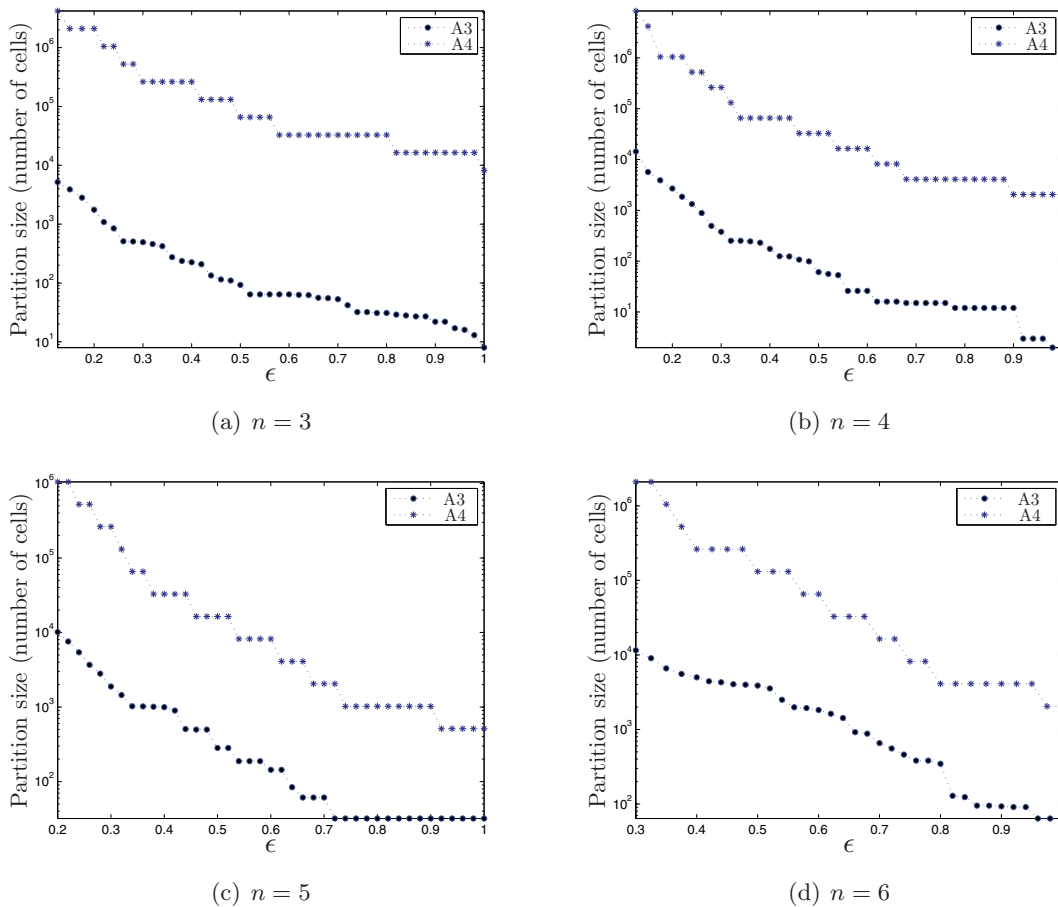
(a) $n = 3$

(b) $n = 4$

(c) $n = 5$

(d) $n = 6$

Figure 3. *Numerical benchmark. Partition size (number of cells), for dimensions $n = 3$ (a), $n = 4$ (b), $n = 5$ (c), and $n = 6$ (d), generated by the adaptive gridding of Algorithm 3 (labeled A3) versus the adaptive gridding of Algorithm 4 (labeled A4) for different levels of error threshold ϵ.*

Let us now select a noise variance $W = \mathbb{I}_{n \times n}$ and benchmark the two adaptive gridding approaches. Figure 3 compares the number of cells generated by the adaptive gridding of Algorithm 3 versus the adaptive gridding of Algorithm 4. This batch of experiments is performed for dimensions $n = 3, 4, 5, 6$. Similarly, Figure 4 compares the runtime required for generating the adaptive partitioning of Algorithm 3 and the adaptive gridding of Algorithm 4. The outcomes of this batch of experiments are averages over 30 runs. Figure 3 confirms that, since the continuity bounds related to Assumption 3 are less tight, Algorithm 4 ends up requiring a larger number of cells, given any threshold ϵ. However (cf. Figure 4), Algorithm 4 works faster than Algorithm 3 in the partition refinement step, since it requires a local error update for the partitions with error greater than the given threshold, whereas Algorithm 3 requires in the worst case a global update of the error of each cell based on the largest obtained error. Thus, for smaller accuracy threshold ϵ and larger dimensions (and large number of generated cells) the method based on Algorithm 4 ends up being faster (Figure 4). Algorithm 3 can

(a) $n = 3$

(b) $n = 4$
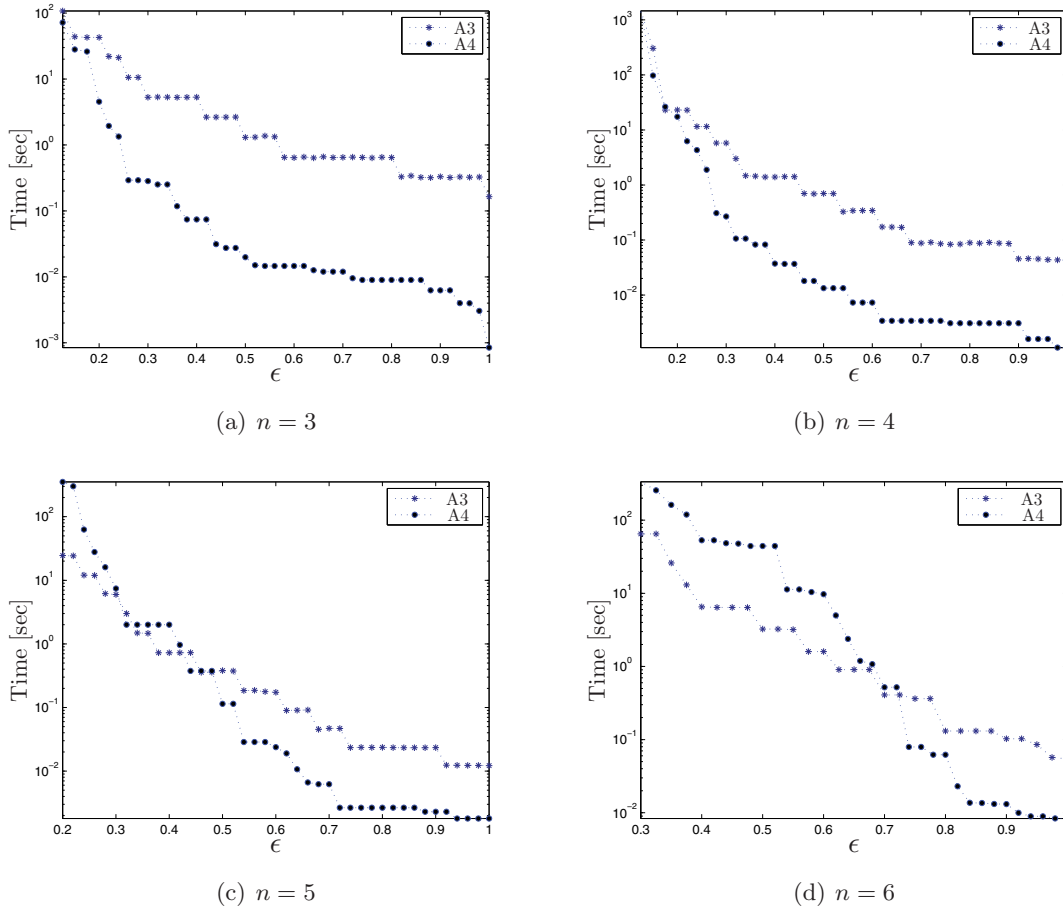
(c) $n = 5$

(d) $n = 6$

Figure 4. *Numerical benchmark. Computation time, for dimensions $n = 3$ (a), $n = 4$ (b), $n = 5$ (c), and $n = 6$ (d), required to generate the adaptive partitioning of Algorithm 3 (labeled A3) and the adaptive gridding of Algorithm 4 (labeled A4) for different levels of error threshold ϵ. The outcomes are obtained as the average over 30 independent runs.*

alternatively be made faster by substituting its refinement step (4) with that of Algorithm 4 (notice that this, however, will not mitigate the possible global update of the error).

6.1.2. Marginalization. The time requirements for the marginalization procedure are recapitulated by the data in Figure 5. These figures are obtained by taking the average over 100 independent runs and display the marginalization time in relative terms versus the time required for the partitioning procedure, which has been discussed in the previous section. More specifically, we have focused on the adaptive gridding obtained according to Algorithm 3 and compared the time spent generating the grid to that needed for performing the marginalization step. The data display that the marginalization step requires more time, relative to the partitioning procedure, as the error level ϵ decreases (that is, as the abstraction precision increases). This trend is consistent regardless of the model size (n).
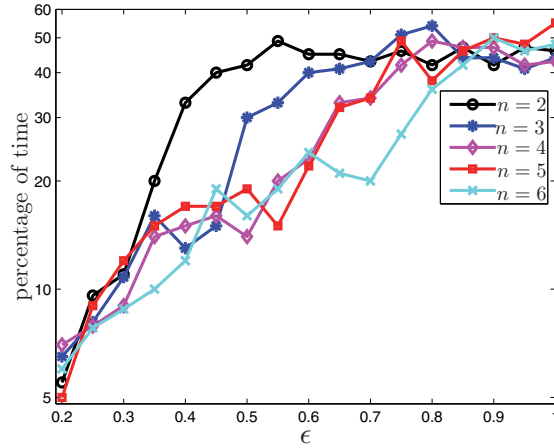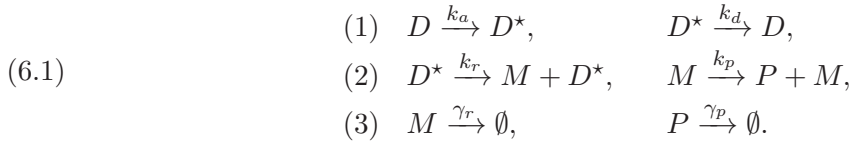
Figure 5. *Time spent on the gridding procedure (according to Algorithm 3) as a percentage of the total time needed for grid generation and marginalization. The figure presents the results for various dimensions ($n = 2, \ldots, 6$) as a function of threshold ϵ for the error. The required data are obtained by running averages over 100 independent runs.*

6.2. Case study. This section applies the abstraction approach developed in this work to the study of a probabilistic invariance problem over a template chemical reaction network. We introduce a model for eukaryotic gene regulation. The stoichiometry (set of chemical reactions) underlying the system is the following:

$$
\begin{aligned}
(1) \quad & D \xrightarrow{k_a} D^\star, & & D^\star \xrightarrow{k_d} D, \\
(2) \quad & D^\star \xrightarrow{k_r} M + D^\star, & & M \xrightarrow{k_p} P + M, \\
(3) \quad & M \xrightarrow{\gamma_r} \emptyset, & & P \xrightarrow{\gamma_p} \emptyset.
\end{aligned}
$$

(6.1)

The reactants represent, respectively, the number of an inactive and active gene (D and $D^\star$ respectively), of m-RNA (M), and of a protein (P). There are three kinds of reactions: (1) conversion (between inactive and active states of the gene), (2) catalytic production (transcription of m-RNA and translation into a protein), and (3) degradation (of m-RNA and protein). The reaction and degradation rates (appearing above the arrows) are directly taken from [9] and are summarized in Table 1.

The dynamics of chemically reacting environments can be described by the general chemical master equation (CME) [15], which seldom has an analytical solution and is usually quite hard to integrate. Alternatively, species dynamics in time are studied via the stochastic simulation algorithm (SSA) [14, 15], a computational scheme that has recently attracted much research [5, 27]. Among the various approaches employed to approximate the SSA and thus expedite its running time, the works in [16, 19] have investigated a technique based on the use of second-order approximations, which assigns probabilistic dynamics (stochastic differential equations) to species concentrations. We leverage this latter approach below.

Parameter	$k_a = k_d$	k_r	γ_r	k_p	γ_p
Value	0.001	0.0078	0.0039	$b\gamma_r, b = 11$	0.0007

6.2.1. Global stochastic approximation. Let us introduce the state-space vector

$$x = \begin{bmatrix} D & D^\star & M & P \end{bmatrix}^T,$$

which describes the concentration of the reactants present in (6.1). Since the new variables are indeed concentrations, they are nonnegative reals rather than natural numbers as in (6.1). We can associate to this state continuous dynamics in time, which can be characterized by a stochastic differential equation of the form [16, 19]

$$dx = f(x)dt + \sigma(x)dW.$$

Time is discretized with constant sampling interval Δ, according to an Euler–Maruyama, first-order scheme [20], obtaining

$$x(k+1) = x(k) + f(x(k))\Delta + \sigma(x(k))\sqrt{\Delta}W(k),$$

where $f(x) = Ax$ and

$$A = \begin{bmatrix} -k_a & k_d & 0 & 0 \\ k_a & -k_d & 0 & 0 \\ 0 & k_r & -\gamma_r & 0 \\ 0 & 0 & k_p & -\gamma_p \end{bmatrix},$$

and

$$\sigma(x) = \begin{bmatrix} \sqrt{k_aD + k_dD^\star} & 0 & 0 \\ -\sqrt{k_aD + k_dD^\star} & 0 & 0 \\ 0 & \sqrt{k_rD^\star + \gamma_rM} & 0 \\ 0 & 0 & \sqrt{k_pM + \gamma_pP} \end{bmatrix}.$$

The noise term is given by $W(k) = [W_1(k), W_2(k), W_3(k)]^T$, where $W_i(k)$, $i = 1, 2, 3$, $k \in \mathbb{N} \cup \{0\}$, are independent standard normal random variables which are also independent of the initial condition of the process. The steady-state values for the dynamics are directly computed as in [19]:

- $P_{ss} = 65\,[nM]$,
- $M_{ss} = \frac{\gamma_p}{k_p}P_{ss} = 1.0606\,[nM]$,
- $D_{ss} = D_{ss}^\star = \frac{\gamma_r}{k_r}M_{ss} = \frac{\gamma_r}{k_r}\frac{\gamma_p}{k_p}P_{ss} = \frac{\gamma_p}{bk_r}P_{ss} = 0.5303\,[nM]$.

Notice that, given the parameter choices in Table 1, the steady-state concentrations assume values that span different dimensions. This will later motivate the use of state-space rescaling.

Since the dynamics of D and $D^\star$ are coupled, it is possible to eliminate either of the variables; here we remove D. The equality $k_a = k_d$ leads to the following discrete time probabilistic dynamical system:

$$(6.2) \quad \begin{cases} x_1(k+1) = (1 - 2k_d\Delta)x_1(k) + 2k_d\Delta D_{ss}^\star + \sqrt{2k_d\Delta D_{ss}^\star}W_1(k), \\ x_2(k+1) = k_r\Delta x_1(k) + (1 - \gamma_r\Delta)x_2(k) + \sqrt{k_r\Delta x_1(k) + \gamma_r\Delta x_2(k)}W_2(k), \\ x_3(k+1) = k_p\Delta x_2(k) + (1 - \gamma_p\Delta)x_3(k) + \sqrt{k_p\Delta x_2(k) + \gamma_p\Delta x_3(k)}W_3(k), \end{cases}$$

where we have denoted
$$\begin{bmatrix} x_1 & x_2 & x_3 \end{bmatrix}^T = \begin{bmatrix} D^\star & M & P \end{bmatrix}^T.$$

Based on the recursive expression in (6.2), the associated conditional probability density function can be defined as

(6.3)
$$t_x(\bar{x}|x) = t_x(\bar{x}_1|x_1)t_x(\bar{x}_2|x_1, x_2)t_x(\bar{x}_3|x_2, x_3),$$

where

$$t_x(\bar{x}_1|x_1) \sim \mathcal{N}(\mu_1(x_1), \sigma_1^2),$$
$$t_x(\bar{x}_2|x_1, x_2) \sim \mathcal{N}(\mu_2(x_1, x_2), \sigma_2^2(x_1, x_2)),$$
$$t_x(\bar{x}_3|x_2, x_3) \sim \mathcal{N}(\mu_3(x_2, x_3), \sigma_3^2(x_2, x_3)),$$

and

$$\mu_1(x_1) = (1 - 2k_d\Delta)x_1 + 2k_d\Delta D_{ss}^\star, \qquad \sigma_1^2 = 2k_d\Delta D_{ss}^\star,$$
$$\mu_2(x_1, x_2) = k_r\Delta x_1 + (1 - \gamma_r\Delta)x_2, \qquad \sigma_2^2(x_1, x_2) = k_r\Delta x_1 + \gamma_r\Delta x_2,$$
$$\mu_3(x_2, x_3) = k_p\Delta x_2 + (1 - \gamma_p\Delta)x_3, \qquad \sigma_3^2(x_2, x_3) = k_p\Delta x_2 + \gamma_p\Delta x_3.$$

It can be observed that, due to differences in variables and parameter ranges, the domain of the density function in (6.3) is compact along the first two variables and stretched along the third. Such an asymmetric shape of the probability density calls for the use of a rescaling by coordinate transformation. The quantities in (6.2)–(6.3) characterize the model of reference for the remainder of the case study.

Probabilistic invariance for global stochastic approximation. In order to introduce a probabilistic invariance problem for the model of interest, we select a hyperbox around the above steady-state values for the variables x_1, x_2, x_3. We plan to assess the probabilistic invariance of the process therein over a finite time horizon. The hyperbox is parameterized by the quantities r_1, r_2, and r_3:

$$\left|\frac{x_1 - D_{ss}^\star}{D_{ss}^\star}\right| \leq r_1, \quad \left|\frac{x_2 - M_{ss}}{M_{ss}}\right| \leq r_2, \quad \left|\frac{x_3 - P_{ss}}{P_{ss}}\right| \leq r_3.$$

This box is set to show 10% variations around the steady-state values, i.e., $r_i = 0.1, i = 1, 2, 3$.

Let us define the abstraction errors computed based on global and local versions of (3.13) under Assumptions 1 and 2, respectively. More precisely, let $E_i, i = 1, 2, 3$, represent the abstraction error using upper bounds (3.7)–(3.9), respectively. Furthermore, let us consider the upper bounds E_4, E_5 obtained via (3.11), (3.14). In order to provide a comparison for these different bounds, two sets of experiments have been set up. Both employ uniform Cartesian gridding based on hyperrectangular partition sets. In the first set of experiments the edges of the partition sets have been selected to be proportional to the length of the edges of the safe set, whereas in the second set of experiments the partition sets have been chosen to be close to cubic cells.

The upper bounds in (3.7)–(3.9) are tailored to Cartesian partitions as follows. Suppose the uniform grid is made up of partition sets characterized by a vector $\boldsymbol{\delta}$ containing its edges.

The grid size δ equals to the Euclidean norm of $\boldsymbol{\delta}$. Assuming that P is a diagonal matrix, whose elements are proportional to the entries of $\boldsymbol{\delta}$, the upper bounds expressed in (3.7)–(3.9) are simplified for the local form (Assumption 2) as

(6.4)
$$\begin{cases} k_1(i,j) = \delta \max_{s \in A_i, \bar{s} \in A_j} \left\| \frac{\partial t_s}{\partial s}(\bar{s}|s) \right\|, \\ k_2(i,j) = \left\| P^{-1}\boldsymbol{\delta} \right\| \max_{s \in A_i, \bar{s} \in A_j} \left\| \frac{\partial t_s}{\partial s}(\bar{s}|s)P \right\|, \\ k_3(i,j) = \max_{s \in A_i, \bar{s} \in A_j} \left| \frac{\partial t_s}{\partial s}(\bar{s}|s) \right| \boldsymbol{\delta} \end{cases}$$

and for the global form (Assumption 1) as

(6.5)
$$\begin{cases} k_1 = \delta \max_{s, \bar{s} \in A} \left\| \frac{\partial t_s}{\partial s}(\bar{s}|s) \right\|, \\ k_2 = \left\| P^{-1}\boldsymbol{\delta} \right\| \max_{s, \bar{s} \in A} \left\| \frac{\partial t_s}{\partial s}(\bar{s}|s)P \right\|, \\ k_3 = \max_{s, \bar{s} \in A} \left| \frac{\partial t_s}{\partial s}(\bar{s}|s) \right| \boldsymbol{\delta}. \end{cases}$$

In order to elucidate the outcomes in Table 2 and what follows, let us discuss the computational overhead related to the different bounds. For the bounds (3.7)–(3.9), the number of optimizations in the local form of (6.4) and in the global form of (6.5) are m^2 and one, respectively. The computation of abstraction error (3.11) in local form can be simplified by using (3.12). Both (3.11) and (3.12) have the same complexity order as the first three upper bounds. Finally, the computation of E_5 based on the local and global forms of (3.14) requires respectively m optimizations and a single one; however, it also needs an integration step. This will lead to higher computational times compared to the errors $E_1 - E_4$.

The time horizon N has been set to be equal to 10 for both experiments. The first set of runs (edges of the partition sets are proportional to the length of the edges of the safe set) is performed with the following specifications:

- length of edges of the (three-dimensional) safe set: $(0.1061, 0.2121, 13)$;
- length of edges of the partition cells: $\boldsymbol{\delta} = (0.0265, 0.0530, 3.25)$;
- resulting number of bins per dimension: $(4, 4, 4)$;
- resulting total number of cells: $m = 4^3 = 64$;
- resulting partitions diameter: $\delta = 3.2505$.

Table 2 summarizes abstraction errors for this set of parameters. The number of optimization steps, as well as the optimization time, has been reported as a measure of the complexity in the error computation.

The second set of runs (partition sets are close to cubic cells) is performed with the following parameters:

- length of edges of the (three-dimensional) safe set: $(0.1061, 0.2121, 13)$;
- length of edges of the partition cells: $\boldsymbol{\delta} = (0.1061, 0.2101, 0.3171)$;
- resulting number of bins per dimension: $(1, 1, 41)$;
- resulting total number of cells: $m = 41$;
- resulting partitions size: $\delta = 0.3949$.

Table 3 summarizes abstraction errors for this second set of parameters. Note that in this second case we have used a lower number of cells, and at the same time we obtained lower abstraction errors and shorter run times. This is due to the underlying ill-conditioned dynamics and to a safe set that is stretched along one axis. This outcome shows the importance of rescaling and of the selection of cubic partition cells for the uniform grid.

Table 2

Global stochastic approximation: Error comparison for the first set of experiments, using a uniform grid with partition sets that are proportional to the length of the edges of the safe set. The computational overhead for the bounds in local form is also reported.

Error bound	E_1	E_2	E_3	E_4	E_5
Inequality used for bound	(3.7)	(3.8)	(3.9)	(3.11)	(3.14)
Global form (Assumption 1)	7095.1	1376.5	799.3	230.9	13.6
Local form (Assumption 2)	1577.3	283.4	167.3	48.6	13.5
Number of optimizations (local form)	4096	4096	4096	8192	64
Computation time	3 (m)	3 (m)	3 (m)	6 (m)	17.9 (h)

Table 3

Global stochastic approximation: Error comparison for the second set of experiments, using a uniform grid with partition sets that are chosen to be cubic. The computational overhead for the bounds in local form is also reported.

Error bound	E_1	E_2	E_3	E_4	E_5
Global form (Assumption 1)	864.3	402	300	219.2	13.6
Local form (Assumption 2)	90.3	42.4	33.5	24.1	7.8
Number of optimizations (local form)	1681	1681	1681	3362	41
Computation time	1.5 (m)	1.5 (m)	1.5 (m)	3 (m)	16.7 (h)

For both batches of experiments we have considered relatively coarse partitions in order to clearly highlight differences in the computed error bounds. As such, the errors are not practically useful since, being larger than 1, they cannot be used in the approximation of probabilistic quantities. Of course, since they monotonically converge to zero as the partition size δ goes to zero, the error bounds can be simply reduced by considering finer partitions, at the expense of longer optimization times.

We now test the adaptive partitioning approach under rescaling. The direct implementation of Algorithms 3 and 4 leads to some computational issues: if the algorithms are initialized over a uniform grid as in the first setup above, they proceed to split the partitions along the longest edge in order to try obtaining cubic-shaped cells. On the other hand, if initialization of the algorithm is set over cubic partition cells, the system dynamics along the shortest direction tend to be lost. To cope with these difficulties we have performed a rescaling of the state-space, so that all the dynamics evolve in a comparable range. More precisely, consider the affine map $x = Py + Q$ with matrices

$$P = \begin{bmatrix} r_1 & 0 & 0 \\ 0 & r_2 & 0 \\ 0 & 0 & r_3 \end{bmatrix}, \qquad Q = \begin{bmatrix} D_{ss}^\star \\ M_{ss} \\ P_{ss} \end{bmatrix},$$

which projects the safe set A to the cube $[-1, 1]^3$. The dynamics in the new state-space become

$$\begin{cases} y_1(k+1) = (1 - 2k_d\Delta)y_1(k) + \frac{1}{r_1}\sqrt{2k_d\Delta D_{ss}^\star}W_1(k), \\ y_2(k+1) = \frac{r_1}{r_2}\gamma_r\Delta y_1(k) + (1 - \gamma_r\Delta)y_2(k) + \frac{1}{r_2}\sqrt{k_r\Delta r_1 y_1 + \gamma_r\Delta r_2 y_2 + 2\gamma_r\Delta M_{ss}}W_2(k), \\ y_3(k+1) = \frac{r_2}{r_3}\gamma_p\Delta y_2(k) + (1 - \gamma_p\Delta)y_3(k) + \frac{1}{r_3}\sqrt{k_p\Delta r_2 y_2 + \gamma_p\Delta r_3 y_3 + 2\gamma_p\Delta P_{ss}}W_3(k). \end{cases}$$

Over the new coordinates we have implemented Algorithm 3, which hinges on Assumption 2. As an outcome of this algorithm, Figures 6(a), 6(b), and 6(c) present the three level sets $p_{x_0}(A) = 0.0015$, $p_{x_0}(A) = 0.0013$, and $p_{x_0}(A) = 0.0011$, respectively, for any $x_0 \in A$ (these figures are in accordance with the uniform results obtained above). The obtained number of cells is 15236 for an error $E_1 = 7.68$ (based on (3.7)), which is as expected lower than that in Tables 2 and 3. An a posteriori computation of the error bound based on (3.11) results in the quantity $E_4 = 1.94$.

6.2.2. Stochastic hybrid approximation. We next present a simplification of the probabilistic dynamics in (6.2)–(6.3) as a stochastic hybrid model, as defined in section 4. Recall that the conditional density of x_1 is normal as

$$t_x(\bar{x}_1|x_1) \sim \mathcal{N}(\mu_1(x_1), \sigma_1^2),$$

where the mean is an affine function exclusively of the conditional variable x_1, and the variance is a constant, namely,

$$\mu_1(x_1) = (1 - 2k_d\Delta)x_1 + 2k_d\Delta D_{ss}^\star, \quad \sigma_1^2 = 2k_d\Delta D_{ss}^\star.$$

As such, the conditional density t_x is independent of x_2, x_3. This suggests performing a simplification of the dynamics over the sole variable x_1. Let us introduce two sets that partition $\mathbb{R}$, the domain of x_1:

$$q_1 = \{x_1 \leq D_{ss}^\star\}, \quad q_2 = \{x_1 > D_{ss}^\star\}.$$

The first set q_1 indicates that the concentration of active genes is lower than its steady-state (we call this the "inactive" mode), whereas the second set q_2 refers to a concentration of active genes that is higher than its steady-state (this is named the "active" mode). These two occurrences make up the discrete modes as $\mathcal{Q} = \{q_1, q_2\}$. Let us additionally select two generic points, one for each of the two modes $l_1 \in q_1, l_2 \in q_2$. We associate to both modes the continuous domain $\mathbb{R}^2$, as needed for the dynamics of the two variables x_2, x_3.

We characterize the discrete probability matrix for the transitions between modes, namely, $T_q(q_j|(q_i, x)), x = (x_2, x_3)$, as follows:

$$(6.6) \qquad T_q(q_1|(q_i, x)) = \mathbb{P}\{x_1 \leq D_{ss}^\star|q_i\} = \Phi\left(\frac{D_{ss}^\star - \mu_i(l_i)}{\sigma_i}\right).$$

We explicitly obtain

$$T_q(q_1|(q_1, x)) = \Phi\left(\frac{D_{ss}^\star - \mu_1(l_1)}{\sigma_1}\right) = \Phi\left((1 - 2k_d\Delta)(D_{ss}^\star - l_1)/\sigma_1\right) \doteq p_1,$$

$$T_q(q_1|(q_2, x)) = \Phi\left(\frac{D_{ss}^\star - \mu_2(l_2)}{\sigma_2}\right) = \Phi\left((1 - 2k_d\Delta)(D_{ss}^\star - l_2)/\sigma_2\right) \doteq p_2,$$

$$T_q(q_2|(q_1, x)) = \mathbb{P}\{x_1 > D_{ss}^\star|q_1\} = 1 - p_1,$$

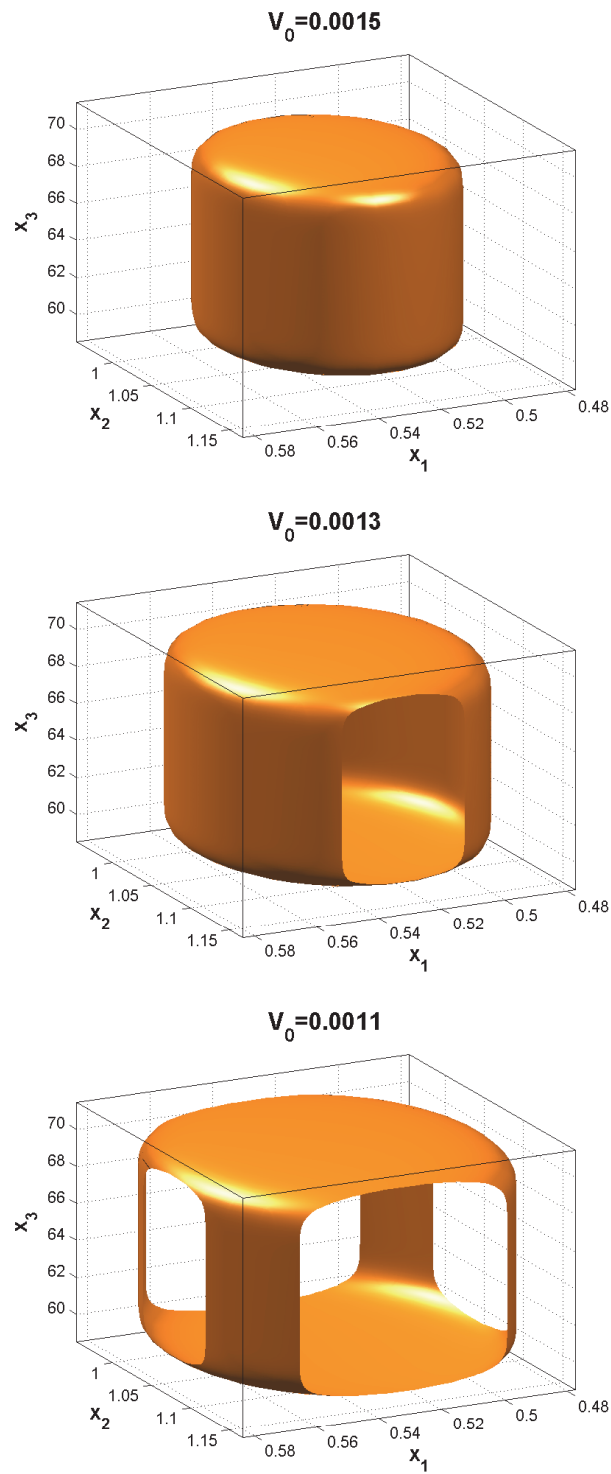$$T_q(q_2|(q_2, x)) = \mathbb{P}\{x_1 > D_{ss}^\star|q_2\} = 1 - p_2.$$

Figure 6. *Level set* $p_{x_0}(A) = 0.0015$, *$p_{x_0}(A) = 0.0013$, and $p_{x_0}(A) = 0.0011$ for original (nonhybrid) model, performed after rescaling under affine mapping, and with an adaptive approach based on Algorithm* 3.

Notice that the probabilities in the above equations (see in particular the second equality in (6.6)) depend on the arbitrary choice of the points $l_i \in q_i, i = 1, 2$. We select such points l_1, l_2 so that the variable x_1 has the same conditional expectation before and after the introduction of the partition sets q_1, q_2. More precisely, with reference to (6.2)–(6.3), given an $x_1 \in \mathbb{R}$, for any $\bar{x}_1 \in \mathbb{R}$,

$$E\{\bar{x}_1 | x_1\} = \mu(x_1) = (1 - 2k_d \Delta)x_1 + 2k_d \Delta D^\star_{ss},$$

whereas for any $\bar{q} \in \mathcal{Q}$, and given the selected $l_i \in q_i$,

$$E\{\bar{q} | q_1\} = E\{\bar{q} | l_1\} = p_1 l_1 + (1 - p_1)l_2,$$
$$E\{\bar{q} | q_2\} = E\{\bar{q} | l_2\} = p_2 l_1 + (1 - p_2)l_2.$$

This leads to the following two equations, which are nonlinear since p_1, p_2 are nonlinear functions of l_1, l_2:

$$\begin{cases} p_1 l_1 + (1 - p_1)l_2 = (1 - 2k_d \Delta)l_1 + 2k_d \Delta D^\star_{ss}, \\ p_2 l_1 + (1 - p_2)l_2 = (1 - 2k_d \Delta)l_2 + 2k_d \Delta D^\star_{ss}. \end{cases}$$

Applying the variable transformation

$$\begin{cases} \ell_1 = (1 - 2k_d \Delta)(D^\star_{ss} - l_1)/\sigma, \\ \ell_2 = (1 - 2k_d \Delta)(D^\star_{ss} - l_2)/\sigma, \end{cases}$$

we obtain

$$\begin{cases} \ell_1 \Phi(\ell_1) + \ell_2(1 - \Phi(\ell_1)) = (1 - 2k_d \Delta)\ell_1, \\ \\ \ell_1 \Phi(\ell_2) + \ell_2(1 - \Phi(\ell_2)) = (1 - 2k_d \Delta)\ell_2. \end{cases}$$

This set of nonlinear equations has a trivial solution $\ell_1 = \ell_2 = 0$, which is not interesting. Its second solution can be computed by the following recursive scheme:

$$\begin{bmatrix} \ell_1^{k+1} \\ \\ \ell_2^{k+1} \end{bmatrix} = (1 - 2k_d \Delta) \begin{bmatrix} \Phi(\ell_1^k) & 1 - \Phi(\ell_1^k) \\ \\ \Phi(\ell_2^k) & 1 - \Phi(\ell_2^k) \end{bmatrix}^{-1} \begin{bmatrix} \ell_1^k \\ \\ \ell_2^k \end{bmatrix},$$

which, for the above nominal values, leads to the quantities $\ell_1 = 3.0902, \ell_2 = -3.0902$. These correspond to the points $l_1 = 0.4295, l_2 = 0.6311$ and finally to the probabilities $p_1 = 0.9990, p_2 = 0.0010$. These values fully characterize the discrete kernel T_q. Further, the continuous kernels $T_x = T_r$ can be directly derived, as done for (6.2)–(6.3), from the following system of stochastic difference equations:

$$\begin{cases} x_2(k + 1) = k_r \Delta q(k) + (1 - \gamma_r \Delta)x_2(k) + \sqrt{k_r \Delta q(k) + \gamma_r \Delta x_2(k)}W_2(k), \\ x_3(k + 1) = k_p \Delta x_2(k) + (1 - \gamma_p \Delta)x_3(k) + \sqrt{k_p \Delta x_2(k) + \gamma_p \Delta x_3(k)}W_3(k). \end{cases}$$

Probabilistic invariance for stochastic hybrid approximation. We implement a uniform gridding with partition cells that are proportional to the edges of the safe set. Note that the safe set in the hybrid state-space is made up of two identical sets A_{q_1}, A_{q_2} for modes q_1, q_2, respectively. As discussed before, both sets coincide over $\mathbb{R}^2$ and are defined as rectangles spanning a 10% variation from the steady-state values of the variables x_2, x_3. The analysis run is performed with the following parameters, defined for each of the two modes q_1, q_2:

<div align="center">**Table 4**</div>

Stochastic hybrid approximation: Error comparison for the first set of experiments, based on a uniform grid. The computational overhead for the bounds in local form is also reported.

Error bound	E_1	E_2	E_3	E_4	E_5
Global form (Assumption 1)	3352	1420.1	1003.8	364.7	15.59
Local form (Assumption 2)	614.89	250.18	178.95	66.04	15.58
Computation time (sec.)	37	38	42	89	478

<div align="center">**Table 5**</div>

Stochastic hybrid approximation: Error comparison for the second set of experiments, which again use a uniform grid but with higher precision. The computational overhead for the bounds in local form is also reported.

Error bound	E_1	E_2	E_3	E_4	E_5
Global form (Assumption 1)	558.67	236.68	167.30	111.56	15.59
Local form (Assumption 2)	53.67	28.58	20.69	20.04	8.36
Computation time (hours)	5.84	6.42	8.93	15.98	16.73

- length of edges of the (two-dimensional) safe set: $(0.2121, 13)$;
- length of edges of the partition cells: $\boldsymbol{\delta} = (0.0424, 2.60)$;
- resulting number of bins per dimension: $(5, 5)$;
- resulting total number of cells: $2 \times 5^2 = 50$;
- resulting partition size: $\delta = 2.6003$.

Notice that the safe set and the partition sets are mode invariant. Let us denote the safe set by A and its partition by $\cup_{i=1}^m A_i$. Since the probability distribution T_q does not depend on the state x, we have that $k_q(q, \bar{q}, i) = 0$. Furthermore, since $t_r(\bar{x}|(q, x), \bar{q}) = t_x(\bar{x}|(q, x))$, then $k_r(q, \bar{q}, i, j) = k_x(q, i, j) \geq |t_x(\bar{x}|(q, x)) - t_x(\bar{x}|(q, x'))|$ for any $x, x' \in A_i, \bar{x} \in A_j$. These observations simplify local error computations to

$$E_{q,i} = 2N \sum_{j=1}^m k_x(q, i, j)\mathcal{L}(A_j), \quad q \in \{q_1, q_2\}, \quad i = 1, 2, \ldots, n.$$

The local form of the abstraction error is $\max\{E_{q,i}|q \in \{q_1, q_2\}, i = 1, 2, \ldots, n\}$, while its global form is $E = 2N\mathcal{L}(A)k_x$. The upper bound $k_x(q, i, j)$ is computable as in the nonhybrid case. The computation of E_5 is simplified to $E_5 = \max_{q,i} 2NH(q, i)$, where

$$\int_A |t_x(\bar{x}|(q, x)) - t_x(\bar{x}|(q, x'))|d\bar{x} \leq H(q, i) \quad \forall x, x' \in A_i, q \in \{q_1, q_2\}.$$

Table 4 reports the abstraction error for the chosen set of parameters. For the bounds in the global form, two maximization problems (one per mode) need to be solved. The optimization time for the local error computations is also reported in the table.

Again, with the uniform discretization approach, let us increase number of bins per dimension (from 5) to 30 and compute a more accurate approximation for the safety problem. Abstraction error and optimization times are presented in Table 5. The resulting total number of cells is thus $2 \times 30^2 = 1800$, which is dimensionally higher than the previous instance, as well as than the experiments in the global case presented in Tables 2 and 3). This of course
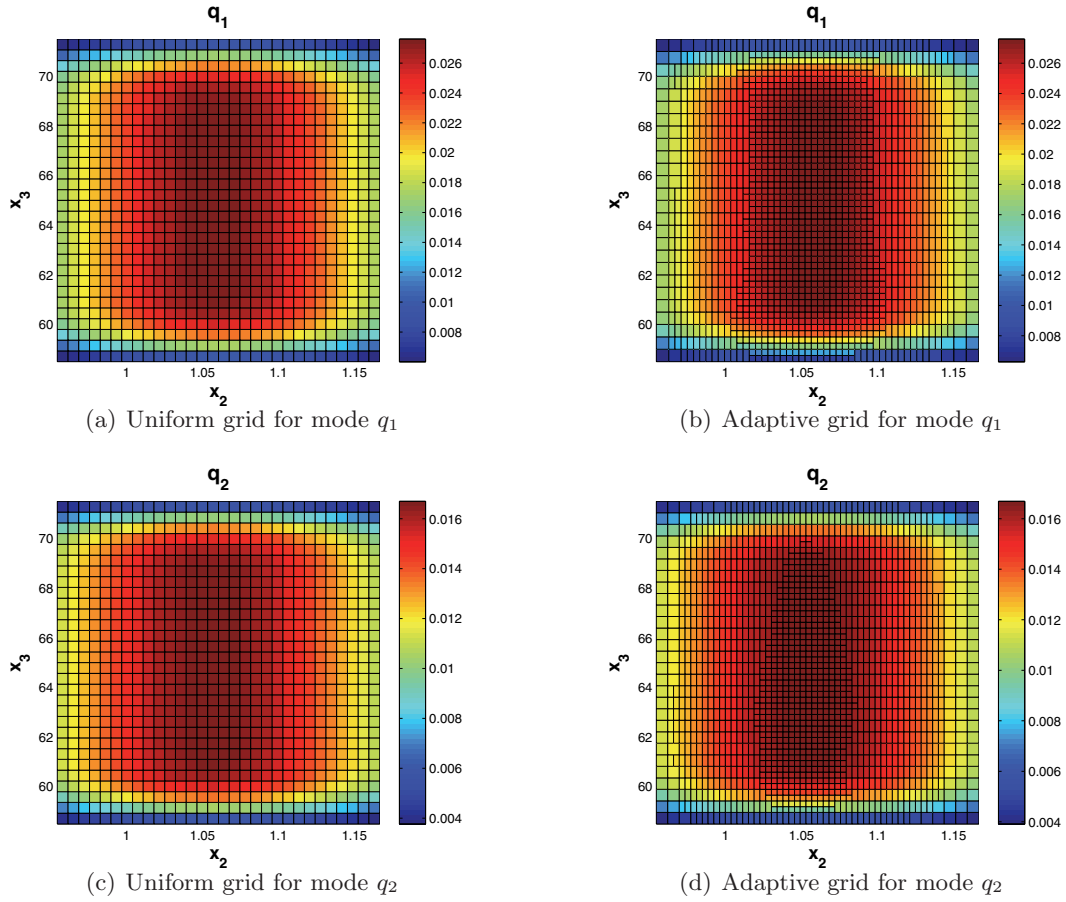
(a) Uniform grid for mode q_1 (b) Adaptive grid for mode q_1

(c) Uniform grid for mode q_2 (d) Adaptive grid for mode q_2

Figure 7. *Visualization of the quantity $p_{x_0}(A)$ (probabilistic invariance) over the grid points of the two discrete modes of the stochastic hybrid model. On the left are outcomes obtained using uniform grids, and on the right are outcomes obtained using adaptive grids.*

comes at a computational cost (compare with optimization time in Table 4). The output of the safety invariance problem is presented in Figures 7(a) and 7(c).

As a second step, we have implemented Algorithm 3 to generate an adaptive grid. The associated errors have been computed based on local Lipschitz constants of the distribution, using the error quantification of Theorem 4.1. Notice that the reset kernel does not depend on the next mode and coincides with t_x, which does not imply $h_r(q, \bar{q}, i, k) = h_x(q, i, k)$ in general. Consequently, the adaptive grid is in general mode dependent. Figures 7(b) and 7(d) present the adaptive grid together with the invariance probability for points over the state-space of the stochastic hybrid system. The grid has been generated for an error E_1 (based on (3.7)) equal to 14.47 and has resulted in a total of 3504 cells. The runtime has amounted to 87 seconds. An a posteriori analysis of the adaptive grid, based on (3.14), ensures an improved abstraction error equal to $E_5 = 5.21$.

7. Conclusions. This contribution has presented an abstraction procedure based on a partitioning of the state-space and has discussed an adaptive gridding generation technique exploiting local formula-based error computation and state-space rescaling. By conforming to the underlying dynamics of the model, the method alleviates the "curse of dimensionality" that is in general related to partitioning procedures. While the focus of the work has been on the study of probabilistic safety over a finite horizon, the technique can be employed in the formal abstraction and verification of stochastic models over more general probabilistic properties, by means of model checkers.

The authors are interested in extending the results toward more general dynamics, and in the development of a software tool based on the presented theoretical results.

REFERENCES

[1] A. ABATE, J.-P. KATOEN, J. LYGEROS, AND M. PRANDINI, *Approximate model checking of stochastic hybrid systems*, Eur. J. Control, 6 (2010), pp. 624–641.

[2] A. ABATE, J.-P. KATOEN, AND A. MEREACRE, *Quantitative automata model checking of autonomous stochastic hybrid systems*, in Proceedings of the 14th ACM International Conference on Hybrid Systems: Computation and Control, Chicago, IL, 2011, pp. 83–92.

[3] A. ABATE, M. PRANDINI, J. LYGEROS, AND S. SASTRY, *Probabilistic reachability and safety for controlled discrete time stochastic hybrid systems*, Automatica J. IFAC, 44 (2008), pp. 2724–2734.

[4] C. BAIER AND J.-P. KATOEN, *Principles of Model Checking*, MIT Press, Cambridge, MA, 2008.

[5] D. BARNES AND D. CHU, *Introduction to Modelling for Biosciences*, Springer-Verlag, New York, 2010.

[6] D. P. BERTSEKAS, *Convergence of discretization procedures in dynamic programming*, IEEE Trans. Automat. Control, 20 (1975), pp. 415–419.

[7] H. A. P. BLOM AND J. LYGEROS, EDS., *Stochastic Hybrid Systems: Theory and Safety Critical Applications*, Lecture Notes in Control and Inform. Sci. 337, Springer-Verlag, Berlin, Heidelberg, 2006.

[8] M. L. BUJORIANU AND J. LYGEROS, *Reachability questions in piecewise deterministic Markov processes*, in Hybrid Systems: Computation and Control, Lecture Notes in Comput. Sci. 2623, O. Maler and A. Pnueli, eds., Springer-Verlag, Berlin, 2003, pp. 126–140.

[9] R. BUNDSCHUH, F. HAYOT, AND C. JAYAPRAKASH, *The role of dimerization in noise reduction of simple genetic networks*, J. Theoret. Biol., 220 (2003), pp. 261–269.

[10] C. G. CASSANDRAS AND J. LYGEROS, EDS., *Stochastic Hybrid Systems*, Autom. Control Eng. 24, CRC Press, Boca Raton, FL, 2007.

[11] M. H. A. DAVIS, *Markov Models and Optimization*, Chapman & Hall/CRC Press, London, 1993.

[12] S. ESMAEIL ZADEH SOUDJANI AND A. ABATE, *Adaptive gridding for abstraction and verification of stochastic hybrid systems*, in Proceedings of the 8th International Conference on Quantitative Evaluation of Systems, Aachen, Germany, 2011, pp. 59–69.

[13] M. FRÄNZLE, H. HERMANNS, AND T. TEIGE, *Stochastic satisfiability modulo theory: A novel technique for the analysis of probabilistic hybrid systems*, in Hybrid Systems: Computation and Control, Lecture Notes in Comput. Sci. 4981, M. Egerstedt and B. Misra, eds., Springer-Verlag, Berlin, Heidelberg, 2008, pp. 172–186.

[14] D. T. GILLESPIE, *A general method for numerically simulating the stochastic time evolution of coupled chemical reactions*, J. Comput. Phys., 22 (1976), pp. 403–434.

[15] D. T. GILLESPIE, *Exact stochastic simulation of coupled chemical reactions*, J. Phys. Chem., 81 (1977), pp. 2340–2361.

[16] D. T. GILLESPIE, *The chemical Langevin equation*, J. Chem. Phys., 113 (2000), pp. 297–306.

[17] A. HINTON, M. KWIATKOWSKA, G. NORMAN, AND D. PARKER, *PRISM: A tool for automatic verification of probabilistic systems*, in Tools and Algorithms for the Construction and Analysis of Systems, Lecture Notes in Comput. Sci. 3920, H. Hermanns and J. Palsberg, eds., Springer-Verlag, Berlin, Heidelberg, 2006, pp. 441–444.

[18] J.-P. Katoen, M. Khattri, and I. S. Zapreev, *A Markov reward model checker*, in Proceedings of the IEEE International Conference on Quantitative Evaluation of Systems, Torino, Italy, 2005, pp. 243–244.

[19] R. Khanin and D. Higham, *Chemical master equation and Langevin regimes for a gene transcription model*, in Computational Methods in Systems Biology, Lecture Notes in Comput. Sci. 4695, M. Calder and S. Gilmore, eds., Springer, Berlin, Heidelberg, 2007, pp. 1–14.

[20] P. E. Kloeden and E. Platen, *Numerical Solution of Stochastic Differential Equations*, Springer-Verlag, New York, 1992.

[21] K. Koutsoukos and D. Riley, *Computational methods for reachability analysis of stochastic hybrid systems*, in Hybrid Systems: Computation and Control, Lecture Notes in Comput. Sci. 3927, J. Hespanha and A. Tiwari, eds., Springer-Verlag, Berlin, 2006, pp. 377–391.

[22] H. J. Kushner and P. G. Dupuis, *Numerical Methods for Stochastic Control Problems in Continuous Time*, Springer-Verlag, New York, 2001.

[23] R. Munos and A. Moore, *Variable resolution discretization in optimal control*, Machine Learning, 49 (2002), pp. 291–323.

[24] S. Prajna, A. Jadbabaie, and G. J. Pappas, *A framework for worst-case and stochastic safety verification using barrier certificates*, IEEE Trans. Automat. Control, 52 (2007), pp. 1415–1428.

[25] M. Prandini and J. Hu, *Stochastic reachability: Theory and numerical approximation*, in Stochastic Hybrid Systems, Autom. Control Eng. 24, C. G. Cassandras and J. Lygeros, eds., CRC Press, Boca Raton, FL, 2007, pp. 107–138.

[26] F. Ramponi, D. Chatterjee, S. Summers, and J. Lygeros, *On the connections between PCTL and dynamic programming*, in Proceedings of the 13th ACM International Conference on Hybrid Systems: Computation and Control, ACM, New York, 2010, pp. 253–262.

[27] M. Rathinam, L. R. Petzold, Y. Cao, and D. T. Gillespie, *Stiffness in stochastic chemically reacting systems: The implicit tau-leaping method*, J. Chem. Phys., 119 (2003), pp. 12784–12794.

[28] P. Somerville, *Numerical computation of multivariate normal and multivariate-t over convex regions*, J. Comput. Graph. Statist., 7 (1998), pp. 529–544.

[29] C. Traxler, *An algorithm for adaptive mesh refinement in n dimensions*, Computing, 59 (1997), pp. 115–137.