Adaptive Gridding for Abstraction and Verification of Stochastic Hybrid Systems

Sadegh Esmaeil Zadeh Soudjani
Delft Center for Systems & Control
TU Delft - Delft University of Technology
Delft, The Netherlands
S.EsmaeilZadehSoudjani@tudelft.nl

Alessandro Abate
Delft Center for Systems & Control
TU Delft - Delft University of Technology
Delft, The Netherlands
A.Abate@tudelft.nl

***Abstract*—This work is concerned with the generation of finite abstractions of general Stochastic Hybrid Systems, to be employed in the formal verification of probabilistic properties by means of model checkers. The contribution employs an abstraction procedure based on a partitioning of the state space, and puts forward a novel adaptive gridding algorithm that is expected to conform to the underlying dynamics of the model and thus at least to mitigate the curse of dimensionality related to the partitioning procedure. With focus on the study of probabilistic safety over a finite horizon, the proposed adaptive algorithm is first benchmarked against a uniform gridding approach from the literature, and finally tested on a known applicative case study.**

***Keywords*-Stochastic Hybrid Systems, Markov Chains, Abstractions, Approximations, Reachability and Safety**

I. Introduction

Stochastic Hybrid Systems (SHS) are dynamical models with interleaved discrete, continuous, and probabilistic dynamics. Motivated by their application in a number of diverse domains, the study of SHS has recently flourished and has witnessed interesting advances at the intersection of the fields of Systems and Control and of Formal Verification.

In particular, [1] has characterized the concept of probabilistic reachability for discrete time SHS, proposing an algorithm to compute this quantity. Theoretically, the connection between the solution of problems related to probabilistic reachability and the verification of PCTL properties has been investigated in [2] and extended to general automata properties in [3]. On the other hand, from a computational perspective, [4] has looked at the numerical evaluation of the specifications discussed in [1]. This evaluation can be achieved by a formal abstraction approach that is based on the partitioning of the state space, which originates a (discrete time) Markov chain (MC) from the original SHS. The approach is formal in that it allows for the computation of explicit bounds on the error associated with the abstraction. In the end, this technique allows to express classes of probabilistic specifications [3], [4] over SHS and to compute them over MC abstractions via available probabilistic model checkers [5], [6], with explicit error bounds.

From a different perspective and over classes of continuous time probabilistic hybrid models, [7] has developed an approach based on satisfiability modulo theory to attain the verification of similar probabilistic properties, but without necessitating a state-space partitioning procedure. Similarly, [8] has looked at the concept of probabilistic reachability for continuous time models and put forward approximation techniques for its computation.

This work looks at extending the applicability of the technique developed in [4] by addressing its known bottleneck: the issue of state-space scalability of the abstraction, which is limited by the "curse of dimensionality" related to the partitioning procedure. In contrast to [4], which has leveraged a uniform partitioning algorithm based on the quantification of a global error, this work puts forward an adaptive procedure that exploits the knowledge of local quantities. This procedure is expected to adapt to the underlying dynamics of the SHS, which is characterized by a (set of) stochastic kernels. Furthermore, this work looks at the implementation of the adaptive procedure: the choice of the shape of the partitioning sets, the execution of the refinement step in the adaptive generation of the grid, as well as the generation of the transition probabilities over the partitioning sets (which involves a marginalization procedure), are discussed.

The article is structured as follows. Initially, Section II-A introduces the SHS model, whereas Section II-B presents the problem of probabilistic invariance. Section III discusses the abstraction of a SHS as a MC. Furthermore, with focus on the probabilistic invariance problem, the quantification of the error in the abstraction procedure is presented in Section IV, under three different assumptions on the underlying dynamics. Section V deals with the generation of the abstraction and elaborates on a number of choices. Finally, Section VI develops two numerical studies: a benchmark compares the adaptive approach versus the uniform procedure; and a case study from the SHS literature tests the scalability of the adaptive approach.

This work is supported by the European Commission MoVeS project FP7-ICT-2009-257005, by the European Commission Marie Curie grant MANTRAS 249295, and by the NWO VENI grant 016.103.020.

II. Preliminaries

A. Model: Stochastic Hybrid System (SHS)

We consider discrete time Markov processes defined over a general state space, characterized by a pair $(\mathcal{S}, T_s)$, where

- $\mathcal{S}$ is the 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;
- 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 $A \in \mathcal{B}(\mathcal{S}), P_s(A) = \int_A T_s(ds|s)$, where P_s denotes the conditional probability $P(\cdot|s)$.

In this work we focus on a particular state space that is "hybrid" in nature [1], 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) $\mathcal{Q} = \{q_1, q_2, \ldots, q_m\}$. The continuous domains have a dimension $n(q)$ that is mode-dependent and characterized by a function $n : \mathcal{Q} \to \mathbb{N}$.

Given a 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 [1]:

$$\begin{aligned} &T_s(\{q'\} \times A_{q'}|(q,x)) \\ &\quad = T_q(q'|(q,x)) \times \begin{cases} T_x(A_{q'}|(q,x)), & \text{if } q' = q, \\ T_r(A_{q'}|(q,x), q'), & \text{if } q' \neq q. \end{cases} \end{aligned} \tag{1}$$

Here $T_q : \mathcal{Q} \times \mathcal{S} \to [0,1]$ assigns to each $s \in \mathcal{S}$ a discrete probability distribution $T_q(\cdot|s)$ over $\mathcal{Q}$. If the selected location 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 $q' \neq q$, 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 $q' \in \mathcal{Q}$ a probability measure $T_r(\cdot|s, q')$ over the continuous domain associated with $q' \in \mathcal{Q}$.

Finally, the initial condition for the model is sampled from $Init : \mathcal{B}(\mathcal{S}) \to [0,1]$, a probability measure on $\mathcal{S}$. We shall denote such a discrete-time stochastic hybrid model with $\mathfrak{S} = (\mathcal{Q}, n, Init, T_q, T_x, T_r)$, and refer the reader to [1] for technical details on its topological and measurability properties and for an algorithmic definition of its execution.

B. Problem: Probabilistic Invariance

The problem of finite horizon probabilistic invariance can be formalized as follows: consider a compact Borel set $A \in \mathcal{B}(\mathcal{S})$, representing a set of safe states (we shall thus alternatively refer to the problem of "probabilistic safety"). Characterize and compute the probability that an execution of $\mathfrak{S}$, associated with the initial condition $s_0 \in \mathcal{S}$, remains within set A during the finite time horizon $[0, N]$:

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

The following theorem provides a theoretical framework to study the above problem.

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

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

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 finite horizon probabilistic invariance as a dynamic programming problem. However, since its explicit solution is rarely 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, which leads to two major questions: whether the numerical output can be precisely related to the actual solution; and whether the approach is scalable. In the next section we answer the first question by introducing a numerical approximation of the original model, and by explicitly computing the error related to the computation of finite-horizon probabilistic reachability with the abstraction. Furthermore, by focusing on the algorithmic implementation of the abstraction, we investigate the scalability properties of the proposed approach (computational complexity, memory usage), thus addressing the second question.

III. Abstraction by a Finite State Markov Chain

We recall a procedure presented in [4] to approximate a SHS $\mathfrak{S} = (\mathcal{S}, T_s)$, by a finite state Markov chain (MC) $\mathfrak{P} = (\mathcal{P}, T_p)$. Here $\mathcal{P} = \{n_1, n_2, \ldots, n_p\}$ is a finite set of states and $T_p : \mathcal{P} \times \mathcal{P} \to [0,1]$ is a probability matrix, such that $T_p(n_2|n_1) = P_{n_1}(n_2)$ characterizes the probability of transitioning from state n_1 to state n_2 and thus induces a conditional probability distribution over the finite space $\mathcal{P}$.

Consider the safe set $A \in \mathcal{B}(\mathcal{S}), A = \cup_{q \in \mathcal{Q}} \{q\} \times A_q$, with $A_q \in \mathcal{B}(\mathbb{R}^{n(q)})$. Algorithm 1 provides a procedure to abstract a SHS $\mathfrak{S}$ 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 representative point $(q, v_{q,i}) \in A_{q,i} \in \mathcal{P}$ the corresponding partition set $A_{q,i} \subset \mathcal{S}$. Furthermore, the map $\xi : A \to A_p$ associates to any point $s \in A$ of the SHS the corresponding discrete state in $\mathcal{P}$. Additionally, notice that the absorbing set ϕ is added to the definition of the MC $\mathfrak{P}$ in order to render the probability matrix T_p stochastic.

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

Require: input SHS $\mathfrak{S}$

1: For all $q \in \mathcal{Q}$, select a finite (m_q-dimensional) partition of set A_q as $A_q = \cup_{i=1}^{m_q} A_{q,i}$ ($A_{q,i}$ are non-overlapping)
2: For each $A_{q,i}$, select a single representative point $(q, v_{q,i}) \in A_{q,i}$
3: Define $A_p = \{(q, v_{q,i}), i = 1, ..., m_q, q \in \mathcal{Q}\}$ and take $\mathcal{P} = A_p \cup \{\phi\}$ as the finite state space of the MC $\mathfrak{P}$
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. *Algorithm 1 can be applied to abstract a general SHS by a finite state MC, regardless of the specifics of the probabilistic invariance problem studied in this work, (that is regardless of the particular safe set A) by assuming $A = \mathcal{S}$. The quantification of the abstraction error, to be carried out in Section IV, will however require that the hybrid set A is bounded (namely, a disjoint union of bounded sets).*

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

$$p_{p_0}(A_p) := P\{p(k) \in A_p \text{ for all } k \in [0, N], p(0) = p_0\}.$$

We now formulate the discrete version of Theorem 1.

Theorem 2 ([4]). *Consider value functions $V_k^p : \mathcal{P} \to [0, 1], k = 0, 1, ..., N$, computed through the backward recursion:*

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

initialized with:

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

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

It is of interest to provide a quantitative comparison between the finite outcome obtained by Theorem 2 and the continuous solution in Theorem 1. The following section accomplishes this goal.

IV. Quantification of the Abstraction Error

We first introduce a technical lemma discussing a bound on the distance between evaluations of the function $V_k, k = 0, 1, ..., N$. The result is inspired by [4, Theorem 1].

Lemma 1. *Consider the safe set $A = \cup_{q \in Q} \{q\} \times A_q$. Then the following inequality holds for all $s = (q, x), s' = (q, x') \in A$:*

$$\begin{aligned} &|V_k(s) - V_k(s')| \\ &\leq \int_{A_q} |T_x(d\bar{x}|(q, x)) - T_x(d\bar{x}|(q, x'))| \\ &+ \sum_{\bar{q} \in Q} |T_q(\bar{q}|(q, x)) - T_q(\bar{q}|(q, x'))| \\ &+ \sum_{\bar{q} \neq q} \int_{A_{\bar{q}}} |T_r(d\bar{x}|(q, x), \bar{q}) - T_r(d\bar{x}|(q, x'), \bar{q})| . \end{aligned}$$

The following continuity conditions restrict the generality of the stochastic kernels characterizing T_s in (1).

Assumption 1. *Assume that the kernels T_x, T_r admit densities t_x, t_r respectively, and that the following continuity assumptions are valid:*

$$\begin{aligned} &|T_q(\bar{q}|(q, x)) - T_q(\bar{q}|(q, x'))| \leq h_1(q, \bar{q}, i) \|x - x'\|, \\ &\quad \forall x, x' \in A_{q,i}, \\ &|t_x(\bar{x}|(q, x)) - t_x(\bar{x}|(q, x'))| \leq h_2(q, i, j) \|x - x'\|, \\ &\quad \forall \bar{x} \in A_{q,j}, \forall x, x' \in A_{q,i}, \\ &|t_r(\bar{x}|(q, x), \bar{q}) - t_r(\bar{x}|(q, x'), \bar{q})| \leq h_3(q, \bar{q}, i, k) \|x - x'\|, \\ &\quad \bar{q} \neq q, \forall \bar{x} \in A_{\bar{q},k}, \forall x, x' \in A_{q,i}, \end{aligned}$$

where $q, \bar{q} \in \mathcal{Q}; i, j = 1, \ldots, m_q; k = 1, \ldots, m_{\bar{q}}$; and $h_1(\cdot), h_2(\cdot), h_3(\cdot)$ are finite positive constants.

Given this assumption, a continuity property can be established on the value functions V_k for the SHS $\mathfrak{S}$.

Theorem 3. *Suppose that the stochastic kernels of the SHS $\mathfrak{S}$ satisfy Assumption 1. 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, $k \in [0, N]$:*

$$|V_k((q, x)) - V_k((q, x'))| \leq \mathcal{K}_{q,i} \|x - x'\|,$$

$\forall x, x' \in A_{q,i} = \Xi(\xi(x)), i \in \{1, \ldots, m_q\}$, where the constant $\mathcal{K}_{q,i}$ is given by:

$$\begin{aligned} \mathcal{K}_{q,i} = &\sum_{j=1}^{m_q} h_2(q, i, j) \mathscr{L}(A_{q,j}) + \sum_{\bar{q} \in Q} h_1(q, \bar{q}, i) \\ &+ \sum_{\bar{q} \neq q} \sum_{k=1}^{m_{\bar{q}}} h_3(q, \bar{q}, i, k) \mathscr{L}(A_{\bar{q},k}), \end{aligned}$$

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

The result in Theorem 3 can be employed to quantify the error between the value $p_{p_0}(A_p)$ for the MC from $p_{s_0}(A)$ for the SHS.

Theorem 4. *Assume that Assumption 1 holds. Then the invariance probability $p_{s_0}(A)$ for the SHS $\mathfrak{S}$, initialized at $s_0 \in A$, satisfies:*

$$|p_{s_0}(A) - p_{p_0}(A_p)| \leq \max\{\gamma_{q,i}\delta_{q,i} | i = 1, ..., m_q, q \in \mathcal{Q}\}, \tag{2}$$

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 $A_{q,i} \subseteq A$, namely

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

and the constants $\gamma_{q,i}$ are specified as $\gamma_{q,i} = N\mathcal{K}_{q,i}$, as per Theorem 3.

A. Relaxation of Assumption 1

We propose a simplification of Assumption 1 and tailor Theorems 3, 4 accordingly. The new assumption is simpler, less tight, however the error bounds are more conservative.

Assumption 2. *Assume that the kernels T_x, T_r admit densities t_x, t_r respectively, and that the following holds for a choice of finite positive $h_1(\cdot), h_2(\cdot), h_3(\cdot)$:*

$$\begin{aligned}
&|T_q(\bar{q}|(q,x)) - T_q(\bar{q}|(q,x'))| \leq h_1(q,i)\,\|x - x'\|\,,\\
&\quad \forall \bar{q} \in Q, \forall x, x' \in A_{q,i},\\
&|t_x(\bar{x}|(q,x)) - t_x(\bar{x}|(q,x'))| \leq h_2(q,i)\,\|x - x'\|\,,\\
&\quad \forall \bar{x} \in A_q, \forall x, x' \in A_{q,i},\\
&|t_r(\bar{x}|(q,x),\bar{q}) - t_r(\bar{x}|(q,x'),\bar{q})| \leq h_3(q,i)\,\|x - x'\|\,,\\
&\quad \forall \bar{q} \neq q \in Q, \forall x, x' \in A_{q,i}, \forall \bar{x} \in A_{\bar{q}},
\end{aligned}$$

where $q \in \mathcal{Q}; i = 1, \ldots, m_q$.

Theorem 5. *Suppose the stochastic kernels of the SHS $\mathfrak{S}$ satisfy Assumption 2. Then the value functions $V_k : \mathcal{S} \to [0,1]$, characterizing the probabilistic invariance problem for the $\mathfrak{S}$ over $A \in \mathcal{B}(\mathcal{S})$, satisfy the following Lipschitz continuity, $k \in [0, N]$:*

$$|V_k((q,x)) - V_k((q,x'))| \leq \mathcal{K}_{q,i}\,\|x - x'\|\,,$$

$\forall x, x' \in A_{q,i} = \Xi(\xi(x)), i \in \{1, \ldots, m_q\}$, where the constant $\mathcal{K}_{q,i}$ is given by:

$$\mathcal{K}_{q,i} = mh_1(q,i) + h_2(q,i)\mathscr{L}(A_q) + h_3(q,i)\sum_{\bar{q}\neq q}\mathscr{L}(A_{\bar{q}}),$$

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

Theorem 6. *Under Assumption 2 the invariance probability $p_{s_0}(A)$ for the SHS $\mathfrak{S}$, initialized at $s_0 \in A$, satisfies:*

$$|p_{s_0}(A) - p_{p_0}(A_p)| \leq \max\{\gamma_{q,i}\delta_{q,i} | i = 1, ..., m_q, q \in \mathcal{Q}\}, \tag{3}$$

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 $\gamma_{q,i} = N\mathcal{K}_{q,i}$, as per Theorem 5, and where $\delta_{q,i}$ is the diameter of the set $A_{q,i} \subseteq A$: $\delta_{q,i} = \max\{\|x - x'\| \,|x, x' \in A_{q,i}\}$.

B. Special case: uniform gridding

We now further relax Assumption 2 and connect with the results presented in [4], which have inspired most of this work. The new assumptions entail continuity results and bounds that can be again obtained from those above.

Assumption 3. *Assume that the kernels T_x, T_r admit densities t_x, t_r respectively, and that the following holds for a choice of finite positive h_1, h_2, h_3:*

$$\begin{aligned}
&|T_q(\bar{q}|(q,x)) - T_q(\bar{q}|(q,x'))| \leq h_1\,\|x - x'\|\,,\\
&\quad \forall \bar{q} \in Q, \forall (q,x), (q,x') \in A,\\
&|t_x(\bar{x}|(q,x)) - t_x(\bar{x}|(q,x'))| \leq h_2\,\|x - x'\|\,,\\
&\quad \forall (q,\bar{x}), (q,x), (q,x') \in A,\\
&|t_r(\bar{x}|(q,x),\bar{q}) - t_r(\bar{x}|(q,x'),\bar{q})| \leq h_3\,\|x - x'\|\,,\\
&\quad \forall \bar{q} \neq q \in Q, \forall (q,x), (q,x'), (\bar{q},\bar{x}) \in A.
\end{aligned}$$

Theorem 7 ([4], Theorem 2)**.** *Under Assumption 3 the invariance probability $p_{s_0}(A)$ for the SHS $\mathfrak{S}$ initialized at $s_0 \in A$ satisfies:*

$$|p_{s_0}(A) - p_{p_0}(A_p)| \leq \gamma\delta, \tag{4}$$

where $p_{p_0}(A_p)$ is the invariance probability for the MC $\mathfrak{P}$ initialized at the discrete state $p_0 = \xi(s_0) \in A_\delta$. The constant $\gamma = N\mathcal{K}$, where

$$\mathcal{K} = mh_1 + \max_{q\in\mathcal{Q}}\mathscr{L}(A_q)\,(h_2 + (m-1)h_3)\,,$$

and where δ is the largest diameter of the partition sets $A_{q,i} \subset A$: $\delta = \max_{q\in\mathcal{Q}, i=1,\ldots,m_q}\{\|x - x'\| \,|x, x' \in A_{q,i}\}$.

V. Algorithms for Abstraction

In the previous sections we considered arbitrary partitions of the state space and, with focus on finite time probabilistic invariance over a 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 SHS $\mathfrak{S}$ and on its MC abstraction $\mathfrak{P}$. In this section we focus on a few alternatives for the generation of the abstraction $\mathfrak{P} = (\mathcal{P}, T_p)$ from $\mathfrak{S} = (\mathcal{S}, T_s)$. The abstraction procedure consists of two main steps:

1) partitioning of $\mathcal{S}$, which leads to $\mathcal{P}$; and
2) marginalization of T_s, which leads to T_p.

A. Algorithms for Grid Generation

Let us first focus on the state space partitioning, which involves a grid generation. The grid can be either uniform and generated instantaneously [4], or be 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 7 and is thus instantaneous. On the other hand, the adaptive partitioning requires the knowledge of errors that are local to the existing partition sets (see Theorems 4 and 6) and proceeds via a progressive refinement of the grid. We will thus perform adaptive

gridding either under Assumption 1 or under Assumption 2 (less tight) over the existing partition, whereas Assumption 3 will be associated with the generation of uniform gridding [4]. Comparing Assumption 1 against Assumption 2, 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 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 the adaptive grid generation. Consider for the sake of discussion an n-dimensional model. Given a grid, there are two main options over the shape of its cells [9], [10]: n-dimensional simplices, or Cartesian hyper-rectangles. The first option leads to the known Kuhn triangulation [10] and is widely used in numerical solution of partial differential equations. The second approach generates hyper-rectangular cells aligned with the main axes which, for our problem at hand, appears to be advantageous. Cartesian cells in fact better accommodate the subsequent step that involves the marginalization of probability laws, which generates T_s. (Marginalization over general convex polygons is known to be a computationally expensive problem [11].)

With focus on the refinement step, consider a single Cartesian cell. We are again presented with two options for partitioning it: replace the cell with 2^n smaller cells by splitting it along its centroid; or replace the cell with 2 smaller cells by partitioning along one axis. The second approach is also known as variable resolution approach [9]. While the first approach decreases the error (which depends on the cell diameter, see Theorems 4 and 6) faster than the second, it is also associated with the generation of partitions with larger cardinality. Since we aim at economizing over 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 1. *For an n-dimensional model, the error convergence rate for a partitioning procedure based on a Cartesian grid, which proceeds by splitting the longest axis, is lower bounded by the factor $\sqrt{1-\frac{3}{4n}}$.*

The grid generation procedures are formally presented in Algorithms 2, 3, 4.

B. Marginalization

The generation of a grid and the choice of representative points $(q, v_{q,i})$ for each of the resulting partition sets $A_{q,i}, i = 1, \ldots, m_q, q \in \mathcal{Q}$ (let us recall that the choice of representative points is arbitrary), fully characterizes 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 is highly dependent on the shape of the kernels T_s, we have attempted to alleviate it 1) by working with hyper-rectangular partitions, 2) by exploiting vectorial representations of the quantities of interest, and 3) by leveraging as much as possible the sparsity of the manipulated matrices. The sparsity of the generated transition probability matrix (number of its non-zero entries) depends on the kernels underlying T_s, particularly on their variance terms. 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), as well as to higher error bounds via the Lipschitz constants. It is possible to use and tune of 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 case the marginalization procedure is greatly simplified, given the regular arrangement of the partition cells.

Algorithm 2 Generation of the adaptive grid

Require: SHS $\mathfrak{S} = (\mathcal{S}, T_s)$ under Assumption 1 over current partition; error threshold ϵ

1: set initial partition over the hybrid state space $\mathcal{S}$
2: compute the error e according to (2) in Theorem 4
3: **if** $e > \epsilon$ **then**
4: refine the partition by splitting along its main axis the single cell with maximum local error
5: go to step 2
6: **end if**

Ensure: $\mathcal{P}$, error e

Algorithm 3 Generation of the adaptive grid

Require: SHS $\mathfrak{S} = (\mathcal{S}, T_s)$ under Assumption 2 over current partition; error threshold ϵ

1: set initial partition over the hybrid state space $\mathcal{S}$
2: compute the error e according to (3) in Theorem 6
3: **if** $e > \epsilon$ **then**
4: refine the partition by splitting along its main axis all the cells with error greater than threshold ϵ
5: go to step 2
6: **end if**

Ensure: $\mathcal{P}$, error e

VI. Case Studies

A. Numerical Benchmark

Let us consider an n-dimensional linear, controlled stochastic difference equation

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

Algorithm 4 Generation of the uniform grid

Require: SHS $\mathfrak{S} = (\mathcal{S}, T_s)$ under Assumption 3; error threshold ϵ

1: pick partition diameter δ based on bound (4) in Theorem 7 and on threshold ϵ

2: perform partitioning of $\mathcal{S}$ into uniformly-packed hypercubes

Ensure: $\mathcal{P}$, error $e = \epsilon$

where $w(k), k \geq 0$, is the process noise, taken to be Normal 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 Normal with zero mean and covariance X : $x(0) \sim \mathcal{N}(0, X)$. 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 weighting matrices $Q \succeq 0$ and $R \succ 0$. The optimal control law for this stochastic control problem (also known as stochastic linear quadratic regulator, or LQR) is given as a stationary linear state feedback $u(k) = K_s x(k)$, where K_s represents the steady-state LQR feedback gain matrix $K_s = -\left(R + B^T P_s B\right)^{-1} B^T P_s A$, and P_s is the solution of the following matrix equation:

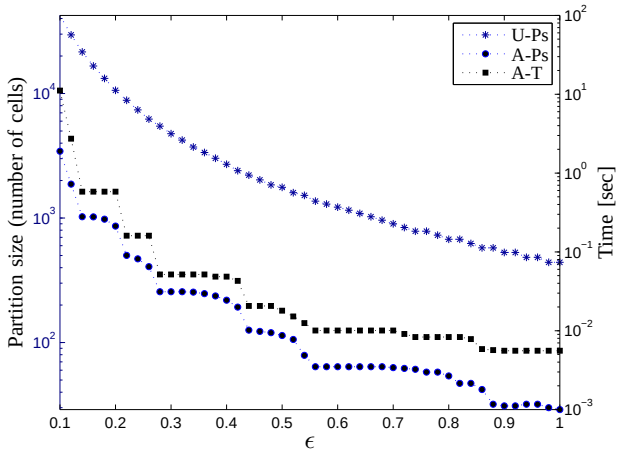
$$P_s = Q + A^T P_s A - A^T P_s B \left(R + B^T P_s B\right)^{-1} B^T P_s A.$$

The closed loop system can be represented as

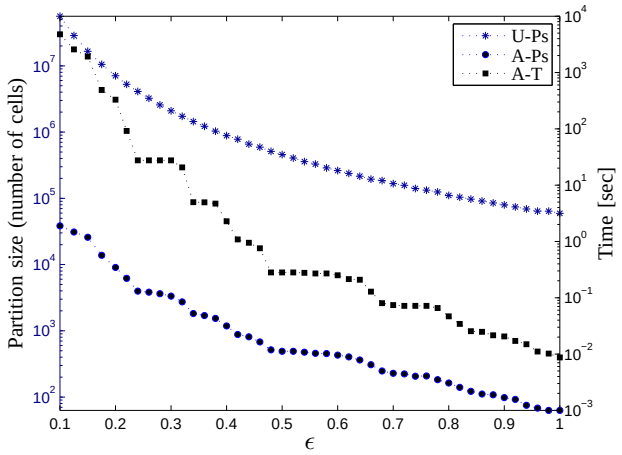
$$x(k+1) = (A + BK_s)x(k) + w(k), \quad k \in \mathbb{N},$$

which is a stochastic difference equation 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 $T_s(\cdot|x) \sim \mathcal{N}(\cdot; (A + BK_s)x, W)$. The computation of the Lipschitz constant of this kernel can be adapted from [4].

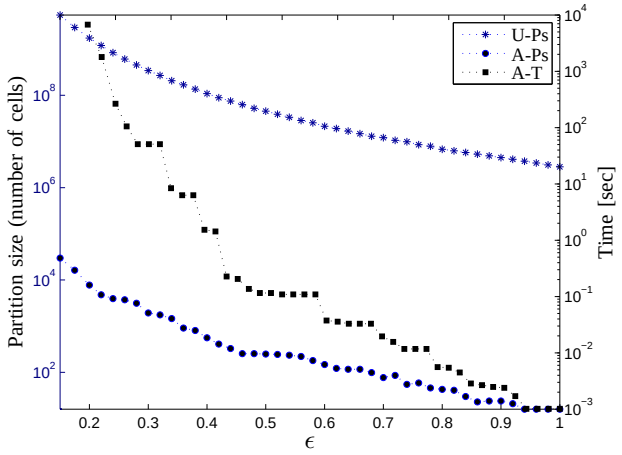
With focus on the closed loop model, let us consider the probabilistic invariance problem over a safe set $A = [-1, 1]^n$, namely a hypercube pointed at the origin, and a time horizon $[0, N]$. For the LQR cost function, we have selected the weighting matrices $Q = \mathbb{I}_{n\times n}, R = \mathbb{I}_{m\times m}$ (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, and A has been further scaled so that $\max_{i=1,\ldots,n} |\lambda_i(A)| = 1$, where $\lambda_i(A)$ denotes the i-th eigenvalue of matrix A. The variance of the initial condition has been selected to be $X = 10\,\mathbb{I}_{n\times n}$.



(a) $n = 2$

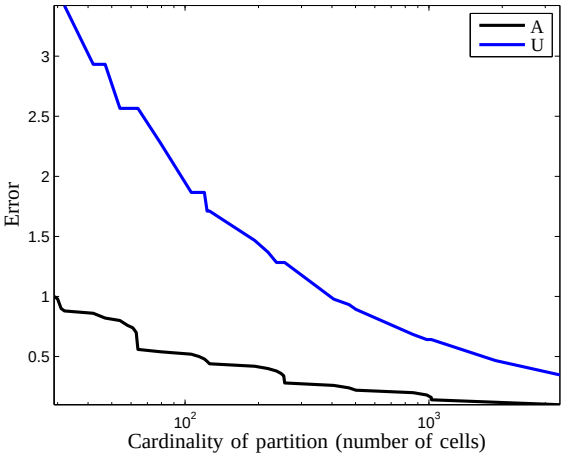


(b) $n = 3$

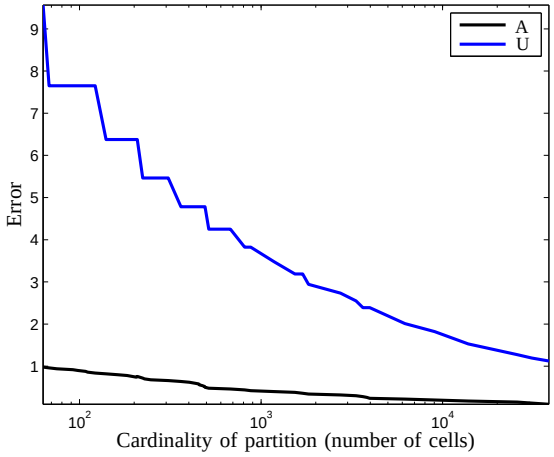


(c) $n = 4$

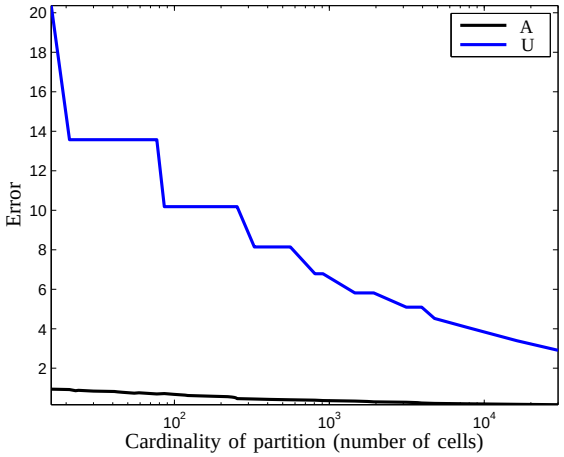
Figure 1. Numerical benchmark. For dimensions $n = 2$ (a), $n = 3$ (b), and $n = 4$ (c) and for different levels of the error threshold ϵ, partition size (number of cells), generated by adaptive (Algorithm 2, labeled A-Ps) vs. uniform gridding (Algorithm 4, labeled U-Ps), and simulation time, required to generate the adaptive partitioning (Algorithm 2, labeled A-T) (average over 30 runs).



(a) $n = 2$

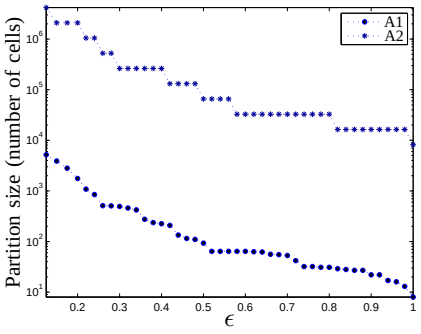


(b) $n = 3$

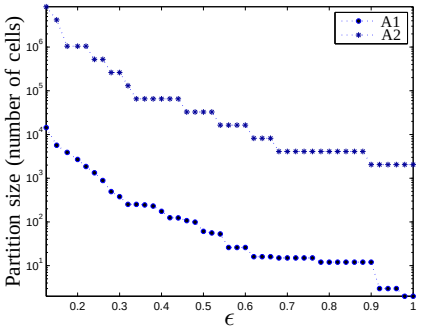


(c) $n = 4$

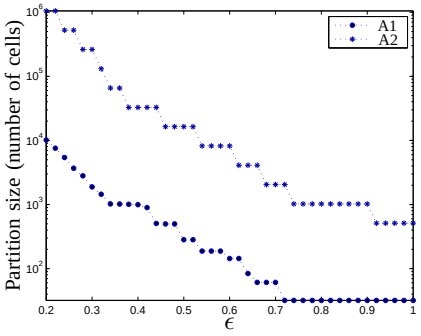
Figure 2. Numerical benchmark. Errors obtained selecting the same number of cells (same partition size), for dimensions $n = 2$ (a), $n = 3$ (b), $n = 4$ (c), for the adaptive gridding of Algorithm 2 (labeled A) vs. the uniform gridding of Algorithm 4 (labeled U).



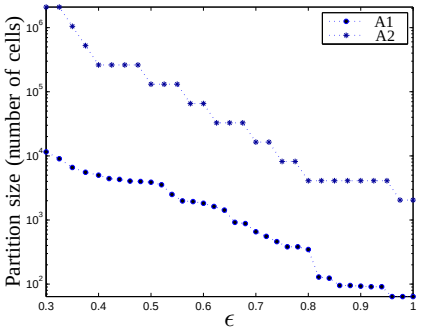
(a) $n = 3$



(b) $n = 4$

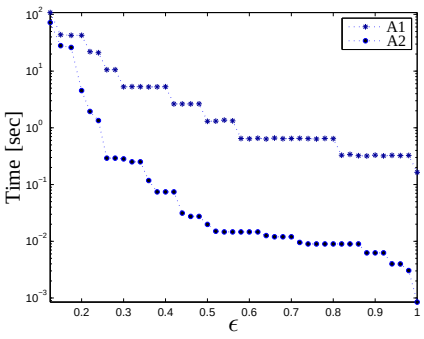


(c) $n = 5$

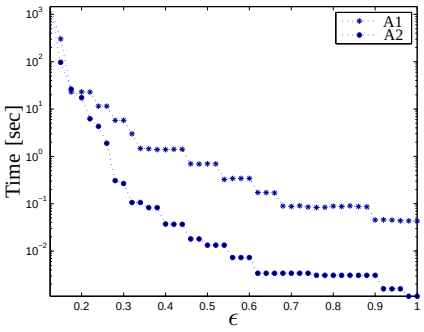


(d) $n = 6$

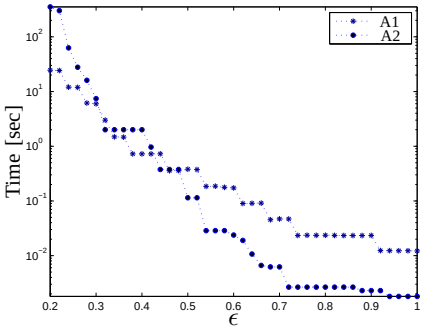
Figure 3. Numerical benchmark. Partition size (number of cells), for dimensions $n = 3$ (a), $n = 4$ (b), $n = 5$ (c), and $n = 6$ (d), generated by the adaptive gridding of Algorithm 2 (labeled A1) vs. the adaptive gridding of Algorithm 3 (labeled A2), for different levels of error threshold ϵ.



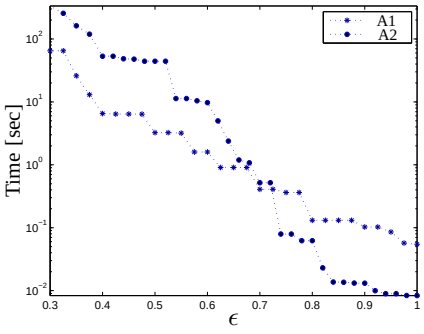
(a) $n = 3$



(b) $n = 4$



(c) $n = 5$



(d) $n = 6$

Figure 4. Numerical benchmark. Simulation time, for dimensions $n = 3$ (a), $n = 4$ (b), $n = 5$ (c), and $n = 6$ (d), required to generate the adaptive partitioning of Algorithm 2 (labeled A1) and the adaptive gridding of Algorithm 3 (labeled A2), for different levels of error threshold ϵ (average over 30 runs).

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 2 for the adaptive gridding, and by Algorithm 4 for the uniform one. The horizontal axis represents the threshold ϵ, namely the upper bound for the abstraction error. The error is based on, respectively, equation (2) in Theorem 4 and (4) in Theorem 7. This batch of simulations 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 thresholds ϵ.

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

Figure 2 compares the error obtained by generating the adaptive gridding with Algorithm 2 (see Theorem 4) against that obtained by generating the uniform gridding of Algorithm 4 (see Theorem 7), given a fixed number of cells for both methods (these values are represented on the horizontal axis). The simulations are again performed for dimensions $n = 2, 3, 4$. 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 is not practically useful.)

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 2 vs. the adaptive gridding of Algorithm 3. This batch of simulations is performed for dimensions $n = 3, 4, 5, 6$. Similarly, Figure 4 compares the simulation time required for generating the adaptive partitioning of Algorithm 2 and the adaptive gridding of Algorithm 3 (average over 30 runs). Figure 3 confirms that, since the continuity bounds related to Assumption 2 are less tight, Algorithm 3 ends up requiring a larger number of cells, given any threshold ϵ. However, Algorithm 3 works faster than Algorithm 2 in the partition refinement step, since it requires a local error update for the partitions with error greater than the given threshold, whereas Algorithm 2 requires in the worst case a global update of the error of each cell. Thus, for smaller accuracy threshold ϵ and larger dimensions (and large number of generated cells) the method based on Algorithm 3 ends being the faster (Figure 4). Algorithm 2 can alternatively be made faster by substituting its refinement step (4:) with that of Algorithm 3.

error threshold ϵ	1	0.95	0.9	0.85	0.8	0.75	0.7	0.65	0.6	0.55	0.5	0.45	0.4	0.35	0.3	0.25	0.2	0.15	0.1
size uniform (per dimension)	21	22	23	25	26	28	30	32	35	38	42	46	52	59	69	83	103	137	206
total size adaptive	28	30	32	45	56	60	63	64	64	82	113	124	221	250	256	447	864	1024	3447
total time adaptive [sec]	0.008	0.008	0.009	0.01	0.01	0.01	0.01	0.01	0.01	0.02	0.03	0.04	0.09	0.09	0.09	0.74	5.0	7.4	266
grid generation, % time adaptive	46	46	47	42	47	42	46	43	45	45	49	49	50	43	36	19	9.6	4.1	5.1
total time uniform [sec]	0.06	0.06	0.07	0.08	0.09	0.11	0.13	0.15	0.19	0.20	0.26	0.30	0.42	0.56	0.85	1.34	21	OOM	OOM

(a) $n = 2$

error threshold ϵ	1	0.95	0.9	0.85	0.8	0.75	0.7	0.65	0.6	0.55	0.5	0.45	0.4	0.35	0.3	0.25
size uniform (per dimension)	39	41	43	45	48	51	55	59	64	70	77	85	96	110	128	153
total size adaptive	63	75	101	120	146	206	227	347	433	473	492	752	1209	1771	3327	3915
total time adaptive [sec]	0.02	0.02	0.04	0.05	0.09	0.13	0.14	0.66	1.3	1.7	1.8	5.8	19	31	190	275
grid generation, % time adaptive	44	41	44	44	54	51	43	25	17	15	12	15	13	16	11	6.5

(b) $n = 3$

error threshold ϵ	1	0.95	0.9	0.85	0.8	0.75	0.7	0.65	0.6	0.55	0.5	0.45	0.4	0.35	0.3	0.25
size uniform (per dimension)	41	43	46	48	51	55	59	63	68	75	82	91	102	117	136	163
total size adaptive	15	16	21	25	28	60	83	112	123	228	248	254	573	1127	1923	3868
total time adaptive [sec]	0.005	0.006	0.008	0.01	0.01	0.02	0.03	0.05	0.06	0.17	0.18	0.21	3.1	18	40	255
grid generation, % time adaptive	6.0	21	34	27	42	40	42	47	47	49	42	34	18	12	14	13

(c) $n = 4$

error threshold ϵ	1	0.95	0.9	0.85	0.8	0.75	0.7	0.65	0.6	0.55	0.5	0.45	0.4	0.35	0.3	0.25
size uniform (per dimension)	126	133	140	148	158	168	180	194	210	229	252	280	315	359	419	503
total size adaptive	8	9	12	18	32	38	44	66	114	221	422	523	987	1119	3711	12411
total time adaptive [sec]	0.002	0.002	0.002	0.003	0.01	0.01	0.01	0.02	0.05	0.18	0.83	1.6	4.9	5.9	139	2330
grid generation, % time adaptive	55	48	50	46	27	32	38	49	34	34	24	17	27	15	29	29

(d) $n = 5$

error threshold ϵ	1	0.95	0.9	0.85	0.8	0.75	0.7	0.65	0.6	0.55	0.5	0.45	0.4
size uniform (per dimension)	1289	1357	1432	1516	1611	1718	1841	1983	2148	2343	2577	2863	3221
total size adaptive	64	121	195	254	304	699	933	1609	1965	2985	4038	5121	14342
total time adaptive [sec]	0.03	0.05	0.09	0.15	0.20	1.0	3.2	11	14	29	87	111	1785
grid generation, % time adaptive	26	21	18	21	29	14	20	23	48	38	50	42	60

(e) $n = 6$

Figure 5. Numerical benchmark. Given a threshold ϵ for the error, the table reports for various dimensions ($n = 2, \ldots, 6$ from (a) to (e)) the number of bins per dimension in the uniform abstraction (total partition size is its n-th power), the total size of the adaptive abstraction (according to Algorithm 2), the total abstraction time for the adaptive approach with its percentage for the gridding procedure, and the total abstraction time for the uniform approach (when not reported it is to be considered out of memory, or OOM). The reported figures are obtained by running averages over 100 simulations.

2) Marginalization: The outcomes of the marginalization procedure are recapitulated by the data displayed on Figure 5. These figures are obtained by running averages over 100 simulations. We focus on the adaptive gridding according to Algorithm 2, and compare the time spent generating the grid to that needed in performing the marginalization. It can be observed that that the gridding procedure increases in relative importance as the dimension n grows. A few outcomes of the uniform approach (for $n = 2$), which soon runs into memory issues, are also displayed. Notice that the outcomes hinge on the actual dynamics (matrices A, B), which are newly generated for each of the 100 simulations.

B. The Multi-Room Heating Problem

We test the adaptive abstraction procedure on a case study with hybrid dynamics elaborated in [4]. Consider h rooms controlled by a centralized heater, which can be switched ON and OFF. The 2 discrete modes of the hybrid model are $Q = \{\mathsf{ON}, \mathsf{OFF}\}$. The continuous state space lies in $\mathbb{R}^h$ and evolves according to the equation $x(k+1) = x(k) + \Sigma x(k) + \Gamma(q(k)) + W(k)$, where the matrix $\Sigma \in \mathbb{R}^{h \times h}$, is made up of the entries $[\Sigma]_{ij} = a_{ij}$ if $i \neq j$, and $[\Sigma]_{ii} = -(b_i + \sum_{k \neq i} a_{ik})$. The vector-valued function $\Gamma(q(k)) \in \mathbb{R}^h$ depends on the configuration of the heater. In particular $[\Gamma(q)]_i = b_i x_a + c_i \delta(q)$. The constants a_{ij}, b_i, c_i, x_a represent the average heat transfer rate from room i to room j (a_{ij}) and to the ambient (b_i), the heat rate supplied by the heater to room i (c_i), and the ambient temperature (x_a). In this dynamical equation $W(k)$ is the input noise, which is Gaussian $\mathcal{N}(0, \mu^2)$. The transition kernel is $T_x(\cdot|(q,x)) = \mathcal{N}(\cdot; x + \Sigma x + \Gamma(q), \mu^2 I)$. The reset kernel is $T_r(\cdot|(q,x), q') = T_x(\cdot|(q,x))$. The heater switches status based on the average temperature of the rooms: this is modeled by a discrete kernel $T_q(q'|(q,x)) = \sigma\left(\sum_{i=1}^h x_i/h\right)$, if $q' = \mathsf{OFF}$, and equal to the complement $(1-\sigma)$ if $q' = \mathsf{ON}$, and where $\sigma(y) = y^d/(\alpha^d + y^d), y \in \mathbb{R}$. The Lipschitz constants of these kernels are derived in [4]. For $h = 2$, Figure 8 displays the adaptive grid generated with Algorithm 2 with $\epsilon = 0.5$. The continuous domain is $[16, 23] \times [16, 23] \subset \mathbb{R}^2$ for both modes. The selected parameters are $a_{12} = a_{21} = 0.0625, b_1 = 0.0375, b_2 = 0.025, c_1 = 0.65, c_2 = 0.6, x_a = 6, \alpha = 19.5, d = 10$. Figure 8 also displays the invariance probability computed over the abstraction, for a time horizon $N = 10$ and noise variance $\mu^2 = 1.3$. We perform a comparison between the adaptive gridding technique (Algorithm 2) and the uniform one (Algorithm 4, as used in [4]) in Figure 6, where we match the errors obtained by selecting a comparable partition size, for different numbers of rooms h. The total time for

number of rooms h	1	2	3	4	5
total size uniform	2500	2500	2744	4096	1024
total size adaptive	2493	2501	2690	3913	1063
adaptive/uniform error	0.0806	0.0178	0.0039	0.0007	0.0001
total time adaptive [sec]	1.4	7.1	243	926	1287

Figure 6. The Multi-Room Heating Problem. Adaptive gridding technique (Algorithm 2) vs. uniform one (Algorithm 4). For comparable partition sizes, over different numbers of rooms h, the ratio of the errors obtained by the two approaches is displayed. The total time for the abstraction procedure based on the adaptive technique is also reported (average over 30 simulations).

number of rooms h	2	3	4	5
error threshold ϵ	0.5	0.5	0.5	0.5
total size adaptive	213	1540	6033	15742
total time adaptive	29.57 [s]	446 [s]	53 [m]	126.4 [m]
grid generation, % time	0.6	1.4	5.9	17.5

Figure 7. The Multi-Room Heating Problem. The adaptive gridding technique (Algorithm 2) is tested, given a fixed error threshold ϵ, for an increasing number of rooms h. The table displays the size of the generated abstraction, the time required to obtain it, and the percentage of time spent on the gridding procedure.

the abstraction procedure based on the adaptive technique is also reported (average over 30 simulations). In Figure 7 the adaptive gridding technique (Algorithm 2) is tested, given a fixed error threshold ϵ, for an increasing number of rooms h. The table displays the size of the generated abstraction, the time required to obtain it, and the percentage of time spent on the gridding procedure.

REFERENCES

[1] A. Abate, M. Prandini, J. Lygeros, and S. Sastry, "Probabilistic reachability and safety for controlled discrete time stochastic hybrid systems," *Automatica*, vol. 44, no. 11, pp. 2724–2734, November 2008.

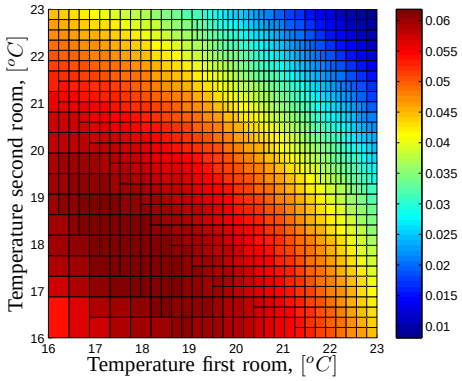
[2] F. Ramponi, D. Chatterjee, S. Summers, and J. Lygeros, "On the connections between PCTL and dynamic programming," in *ACM Proceedings of the 13th International Conference on Hybrid Systems: Computation and Control*, April 2010, pp. 253–262.

[3] A. Abate, J.-P. Katoen, and A. Mereacre, "Quantitative automata model checking of autonomous stochastic hybrid systems," in *ACM Proceedings of the 14th International Conference on Hybrid Systems: Computation and Control*, Chicago, IL, April 2011, pp. 83–92.

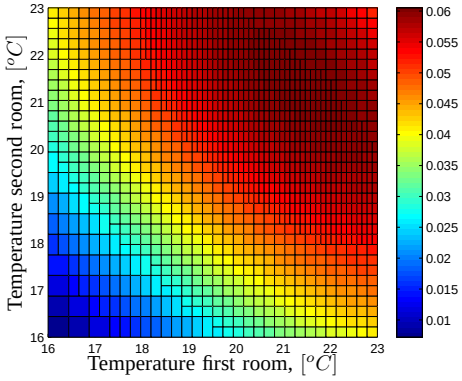
[4] A. Abate, J.-P. Katoen, J. Lygeros, and M. Prandini, "Approximate model checking of stochastic hybrid systems," *European Journal of Control*, no. 6, pp. 624–641, 2010.

[5] 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*, ser. Lecture Notes in Computer Science, H. Hermanns and J. Palsberg, Eds. Springer Verlag, Berlin Heidelberg, 2006, vol. 3920, pp. 441–444.

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



(a) Discrete mode $q =$ `ON`



(b) Discrete mode $q =$ `OFF`

Figure 8. The Multi-Room Heating Problem, case $h = 2$ rooms. Generated adaptive partitions (Algorithm 2) for the two discrete modes. The continuous domains denote temperature intervals for each of the two rooms, whereas color intensity represents the invariance probability as a function of the initial condition.

[7] 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*, ser. Lecture Notes in Computer Sciences, M. Egerstedt and B. Misra, Eds. Springer Verlag, Berlin Heidelberg, 2008, no. 4981, pp. 172–186.

[8] M. Prandini and J. Hu, "Stochastic reachability: Theory and numerical approximation," in *Stochastic hybrid systems*, ser. Automation and Control Engineering Series 24, C. Cassandras and J. Lygeros, Eds. Taylor & Francis Group/CRC Press, 2006, pp. 107–138.

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

[10] C. Traxler, "An algorithm for adaptive mesh refinement in n dimensions," *Computing*, vol. 59, pp. 115–137, 1997.

[11] P. Somerville, "Numerical computation of multivariate normal and multivariate-t over convex regions," *Journal of Computational and Graphical Statistics*, vol. 7, no. 4, pp. 529–544, 1998.