A simulation-based approach to the approximation of stochastic hybrid systems ⋆

S. Garatti * **M. Prandini** *

* *Politecnico di Milano, Milano, Italy (e-mail: {sgaratti,prandini}@elet.polimi.it)*

Abstract: We study the problem of approximating a stochastic, possibly hybrid, system by means of some abstracted model to the purpose of simplifying the analysis of properties such as probabilistic safety and reachability. We suppose that the property to be analyzed depends on the behavior of some output signal of the system and that the model is designed in order to reproduce that signal as close as possible, for the different possible realizations of the stochastic input affecting the system. The idea developed in this paper is to assess the quality of a model as an approximation of a stochastic system by testing how close are their output signals over a finite number of input realizations. Under suitable assumptions, we show that, with high confidence, the quality assessed on a few input realizations is guaranteed to hold also for all the unseen ones except for a set of pre-defined probability ϵ. The proposed approach can be applied to an arbitrary system, the only requirement being to be able to run multiple simulations of its behavior for different input realizations.

Keywords: Model approximation; stochastic hybrid systems; randomized algorithms; scenario optimization; reachability analysis.

1. INTRODUCTION

Stochastic hybrid systems (SHS) represent a powerful modeling framework for describing complex, large scale systems that involve the interaction between continuous dynamics, discrete dynamics and probabilistic uncertainty. Because of their versatility, SHS have been effectively adopted in diverse application domains such as control of telecommunication networks, air traffic management, manufacturing, biology and finance (see, for example, Blom and J. Lygeros [2006], Cassandras and J. Lygeros [2006] for an overview).

The verification of properties related to the system evolution, like, e.g., safety and reach/avoid properties, is typically addressed through numerical methods involving state-space gridding (Abate et al. [2010]), and, as such, is affected by an exponential growth of the computational effort as a function of the state-space dimension. To the purpose of scaling-up numerical methods for system verification, it is then important to find a way of introducing simpler descriptions of a given SHS that mimic the behavior of the original system and can be used in place of it to assess the property of interest.

This motivates our work in this paper, where we study the problem of approximating a stochastic, possibly hybrid, system by means of some (simpler) model, see Julius et al. [2006], Petreczky and Vidal [2007], Julius and Pappas [2009]. The interested reader is referred to Lygeros and Prandini [2010] for a discussion on further challenges in the advancement of analysis and control methods for SHS.

We adopt a setting that is similar to that in Julius and Pappas [2009], Julius et al. [2006], where a notion of approximate stochastic (bi)simulation is introduced for SHS. More specifically, we consider a system S that is characterized in terms of some stochastic output signal y^S and suppose that the candidate abstracted model M generates a signal y^M that takes values in the same set as y^S. The system and the model are fed by the same stochastic input which may influence both the continuous evolution within each mode and the spontaneous transitions between modes. The initial state is supposed to be stochastic.

According to the notion of approximate (bi)simulation in Julius and Pappas [2009], the quality of M as an approximate abstraction of S can be assessed by evaluating the maximal distance between the system and the model output realizations over all possible input realizations and initial conditions except for a set of probability ε. Let γ be this distance. If we are interested in evaluating the probability that y^S enters some unsafe set A, then, we can suitably enlarge A so as to obtain a set A_γ such that if y^S enters A, then, y^M enters A_γ. The probability that y^S enters A can then be upper bounded by the probability that y^M enters A_γ increased by ε.

Differently from Julius and Pappas [2009], we suppose that the property of interest depends on the system behavior over a *finite horizon* and we provide a computational method that is of general applicability and not restricted to some specific class of systems.

⋆ Research supported by the European Commission under the MoVeS project, FP7-ICT-2009-257005.

The key idea is to assess the quality of a model M as an approximation of a stochastic system S by testing its behavior over a finite number of realizations of the initial state and of the stochastic input affecting the system. Under suitable assumptions, the quality assessed on a few scenarios is guaranteed to hold also for all unseen scenarios except for a set of probability ϵ, with high confidence. This idea was first mentioned in Campi et al. [2009] as one of the possible applications of the scenario approach to systems and control design, and is further elaborated here, leading to a significant improvement in terms of problem formulation and extent of the results with respect to Abate and Prandini [2011].

The proposed approach should be combined with computational verification techniques to allow for the analysis of probabilistic safety and reachability properties of large scale stochastic systems.

Interestingly, the computational complexity of the approach in terms of number of scenarios to consider depends on the complexity of the model to be designed and not on that of the system S to be approximated. Indeed, the only assumption on S is that one should be able to run multiple executions of S and to determine the corresponding output realizations. If feasible, one could even run experiments on the real system without the need of determining a mathematical description and building a simulator for it.

The rest of the paper is organized as follows. We start by formulating the problem of approximating a stochastic system S in Section 2, where we precisely define the issues of performance assessment and optimization of an abstracted model M of S. In Section 3, we develop a simulation-based approach to address both these issues. Final conclusions are drawn in Section 4.

Notation: Throughout the paper, we use small letters like s to denote a signal defined over the look-ahead time horizon $[0,T]$, and s_t to denote the value taken by s at time $t \in [0,T]$. For each $t \in [0,T]$, s_t takes value in the space $\mathcal{S}$. $\mathcal{S}$ may be e.g. $\mathbb{R}^n$ or, when we are dealing with hybrid systems and s_t has both a continuous and a discrete component, $\mathbb{R}^n \times \{1,2,\ldots,q\}$. $\mathcal{S}^{[0,T]}$ denotes the set of all signals defined over the time interval $[0,T]$ and taking values in $\mathcal{S}$ at each time instant $t \in [0,T]$.

2. PROBLEM FORMULATION

System S is described as an operator that maps the initial state x_0 and the input signal w on the time horizon $[0,T]$ into the signal y^S of interest:

$$y^S = f^S(x_0, w).$$

x_0 and w are defined over $\mathcal{X} \times \mathcal{W}^{[0,T]}$ and are assumed to be stochastic with known probability measure P. Signal y^S takes values in $\mathcal{Y}^{[0,T]}$.

Model M is defined as follows

$$y^M = f^M(x_0, w),$$

where $y^M \in \mathcal{Y}^{[0,T]}$.

Note that function f^M defining M depends on the initial condition x_0 of S. This does not mean that the state space of M has the same size as that of S but that f^M incorporates the mapping from the initialization of the state of S to that of the (possibly lower-dimensional) state of M.

Example 1. (JLSS). Consider a stochastic system S with state $x_t \in \mathbb{R}^n$ that evolves within $[0,T]$ according to the following Stochastic Differential Equation (SDE):

$$dx_t = Ax_t dt + Fx_t dB_t, \tag{1}$$

in-between the jump times $0 = \tau_0 < \tau_1 < \cdots < \tau_i < \cdots \leq T$ of a Poisson process P with rate $\lambda > 0$. The Brownian motion B in equation (1) is assumed to be independent of the Poisson process P. At each jump time $\tau_i > 0$, the state is reset according to

$$x_{\tau_i} = R \lim_{s \to \tau_i^-} x_s, \tag{2}$$

whereas $x_{\tau_0} = x_0$. The initial state $x_0 \in \mathcal{X} := \mathbb{R}^n$ is independent of the stochastic input $w = (B, P)$ that includes both the Brownian motion B and the Poisson process P and takes values in $\mathcal{W} = \mathbb{R} \times \mathbb{Z}_+$ at each time $t \in [0,T]$.

This system is known as Jump Linear Stochastic System (JLSS) since its evolution between jump times is characterized by a SDE with drift and diffusion terms that are linear in x_t, and the state resets at the jump times are linear in x_t as well.
A JLSS can be seen as a SHS with a single operating mode characterized by a SDE. When a (auto)transition occurs, the continuous state is subject to some deterministic reset and the continuous dynamics keeps unchanged after the transition.
The output signal of interest y_t^S takes values in $\mathcal{Y} = \mathbb{R}^p$ and is given by

$$y_t^S = Cx_t.$$

A reduced model of the system with output signal

$$y_t^M = \tilde{C} z_t$$

can be obtained by taking only a subset z_t of the state variables x_t and appropriately re-defining the matrices entering the original JLSS definition. The resulting jump diffusion process $z_t \in \mathbb{R}^{\tilde{n}}$ (with $\tilde{n} < n$) satisfies the SDE

$$dz_t = \tilde{A} z_t dt + \tilde{F} z_t dB_t, \tag{3}$$

in-between the jump times $0 = \tau_0 < \tau_1 < \cdots < \tau_i < \cdots \leq T$, and is reset according to

$$z_{\tau_i} = \tilde{R} \lim_{s \to \tau_i^-} z_s \tag{4}$$

at the jump times $\tau_i > 0$. The initial condition $z_{\tau_0} = z_0$ is a function of x_0: $z_0 = l(x_0)$. $\square$

The quality of M as an approximation of S is evaluated by looking at the similarity of the signals y^M and y^S of the two systems. To this purpose, we introduce a quasi-metric

$$D : \mathcal{Y}^{[0,T]} \times \mathcal{Y}^{[0,T]} \to \mathbb{R}_+$$

to assess how close signal y^M is to y^S. For example, letting d be any metric defined over $\mathcal{Y}$, the metric

$$D(y^S, y^M) = \sup_{t \in [0,T]} d(y_t^S, y_t^M),$$

can be used if we are interested in y^S and y^M being close each other at each time instant. If, otherwise, we are interested in the distance between trajectories only, the directional Hausdorff metric can be used

$$D(y^S, y^M) = \sup_{t \in [0,T]} \inf_{\tau \in [0,T]} d(y_t^S, y_\tau^M).$$

As for the metric d, it highly depends on the space $\mathcal{Y}$ and on the problem itself. For example, if $\mathcal{Y} = \mathbb{R}^p$ it is customary to use the Euclidean metric $d(y_t^S, y_t^M) = \|y_t^S - y_t^M\|$. If instead $\mathcal{Y} = \mathbb{R}^p \times \{1, 2, \dots, q\}$ so that $y_t \in \mathcal{Y}$ has both a continuous and a discrete component, say $y_t = (y_t^c, y_t^d)$, the metric d can be

$$d(y_t^S, y_t^M) = \begin{cases} +\infty & \text{if } y_t^{S,d} \neq y_t^{M,d} \\ \|y_t^{S,c} - y_t^{M,c}\| & \text{otherwise.} \end{cases}$$

The meaning of this metric is that we want first to check whether S and M are in the same operation mode, and then, if so, how close the continuous components of the y variables are.

When evaluating the quality of M as an approximation of S, we can require either that y^M is close to y^S for every and each realization of x_0 and w or, alternatively, that y^M is close to y^S for all realizations of x_0 and w except a set of pre-specified probability $\varepsilon \in (0, 1)$. This latter approach is adopted in Julius and Pappas [2009] and presents the advantage that if there exist some "bad" but quite unlikely realizations that would over-penalize the performance of M as an approximation of S, then, they can be discarded.

Definition 1. System M is said to be an ε-abstraction of S with accuracy function $h : \mathcal{X} \to \mathbb{R}_+$ if

$$\mathsf{P}\{D(y^S, y^M) \leq h(x_0)\} \geq 1 - \varepsilon. \tag{5}$$

Note that, according to Definition 1, $D(y^S, y^M)$ is upper bounded by some positive function $h(x_0)$ of the initial condition x_0. This means that different initializations are allowed to correspond to a different similarity level of y^M and y^S, which can avoid the conservativeness of using a uniform bound.

Remark 1. Evidently, the performance assessed over a set of realizations of measure $1 - \varepsilon$ improves as ε grows, but becomes meaningless if ε is too close to 1. The probability ε has then to be chosen so as not to penalize performance, while leading to sensible statements on the properties of S through the analysis of M.

2.1 Abstraction performance assessment

In this case we suppose that both the operators f^S and f^M defining S and M are given and the objective is to assess the performance of M as an ε-abstraction of S. This entails to determining an accuracy function $h(x_0)$ so that the condition (5) is satisfied. Clearly, the solution of this problem is not unique, and we are interested in determining the "smallest possible" $h(x_0)$. Since x_0 is stochastic, a sensible measure of the size of $h(x_0)$ is its expectation. This leads naturally to the following chance-constrained optimization problem:

$$\begin{aligned} &\min_{h(\cdot) \in \mathbb{H}} \mathsf{E}[h(x_0)] \\ &\text{subject to: } \mathsf{P}\{D(y^S, y^M) \leq h(x_0)\} \geq 1 - \varepsilon, \end{aligned} \tag{6}$$

where $\mathbb{H}$ is a set of functions from $\mathcal{X}$ to $\mathbb{R}_+$.

Remark 2. In Julius and Pappas [2009] a method is proposed for finding a $h(x_0)$ which is feasible for (6) (i.e. a $h(x_0)$ that satisfies the probabilistic constraint). This method is based on the introduction of the so-called stochastic (bi-)simulation functions and provides no guarantees about the optimality of the obtained $h(x_0)$. This may lead to a severe underestimation of the abstraction capabilities of M and to conservative results.

Remark 3. Note that if the accuracy function $h(x_0)$ is assumed to be constant, then the problem reduces to

$$\begin{aligned} &\min_{h \in \mathbb{R}} h \\ &\text{subject to: } \mathsf{P}\{D(y^S, y^M) \leq h\} \geq 1 - \varepsilon, \end{aligned}$$

which was previously considered in Abate and Prandini [2011] that can be seen as a particular case of our setting.

2.2 Abstraction performance optimization

In this second case, f^M is no more given and our goal is to design M so that M is an ε-abstraction of S with the smallest possible accuracy $h(x_0)$. The problem becomes:

$$\begin{aligned} &\min_{f^M \in \mathcal{F}, h(\cdot) \in \mathbb{H}} \mathsf{E}[h(x_0)] \\ &\text{subject to: } \mathsf{P}\{D(y^S, y^M) \leq h(x_0)\} \geq 1 - \varepsilon, \end{aligned} \tag{7}$$

where $\mathcal{F}$ is some given family of operators and $\mathbb{H}$ is a set of functions from $\mathcal{X}$ to $\mathbb{R}_+$

3. SCENARIO IMPLEMENTATION

The optimization problems introduced in Sections 2.1 and 2.2 are called *chance-constrained* problems since we have to minimize a cost function subject to a constraint which holds in probability. Unfortunately, the constraint $\mathsf{P}\{D(y^S, y^M) \leq h(x_0)\} \geq 1 - \varepsilon$ is in general non-convex even in the case when the constraint $D(y^S, y^M) \leq h(x_0)$ is convex with respect to the optimization variables for every realization of x_0 and w. For this reason, chance-constrained problems are usually hard to solve and, indeed, they are NP-hard with few exceptions.

The scenario approach, Calafiore and Campi [2005, 2006], Campi and Garatti [2008], Campi et al. [2009], Campi and Garatti [2011], is a recent paradigm for computing approximate solutions to chance-constrained problems at relatively low computational effort when the constraint $D(y^S, y^M) \leq h(x_0)$ and the cost function $\mathsf{E}[h(x_0)]$ are convex in the optimization variables. Algorithmically speaking, the scenario approach builds on a very intuitive and basic idea: a number, say N, of realizations of x_0 and w, say $x_0^{(i)}$ and $w^{(i)}$ for $i = 1, 2, \dots, N$, are extracted according to the underlying probability measure P and optimization is performed over this finite number of instances of x_0 and w only. More precisely, letting α be a user chosen parameter such that $0 \leq \alpha < \varepsilon$, and letting $y^{S,(i)} = f^S(x_0^{(i)}, w^{(i)})$ and $y^{M,(i)} = f^M(x_0^{(i)}, w^{(i)})$, $i = 1, 2, \dots, N$, the Scenario Algorithm described in Algorithm 1 aims at finding a solution that violates the condition

$$D\left(y^{S,(i)}, y^{M,(i)}\right) \leq h(x_0^{(i)})$$

$\lfloor \alpha N \rfloor$ times out of N, i.e. with an *empirical* probability equal to α ($\lfloor \cdot \rfloor$ = integer part).

In Algorithm 1, the constraints to be violated are selected by progressively discarding one constraint at a time, that one giving the largest immediate cost improvement (*greedy* algorithm). This way, the obtained solution is not the best possible one violating $\lfloor \alpha N \rfloor$ constraints out of N, yet a fair sub-optimality is achieved while keeping the computational effort at a reasonable level. Each optimization

Algorithm 1 The Scenario Algorithm (SA)

1: Solve problem

$$\min \mathsf{E}[h(x_0)]$$
$$\text{subject to: } D\left(y^{S,(i)}, y^{M,(i)}\right) \leq h(x_0^{(i)}),$$
$$i \in \{1, 2, \ldots, N\}.$$

Store the solution.

2: Find the constraints violated by the stored solution, i.e. find the indexes i such that

$$D\left(y^{S,(i)}, y^{M,(i)}\right) > h(x_0^{(i)}).$$

Let these indexes be $j_1, j_2, \ldots, j_p$ (if $\{j_1, j_2, \ldots, j_p\} = \emptyset$, then take $p = 0$). If $p = \lfloor \alpha N \rfloor$, then halt the algorithm and return the stored solution.

3: Find the active constraints for the stored solution, i.e. find the indexes i such that

$$D\left(y^{S,(i)}, y^{M,(i)}\right) = h(x_0^{(i)}).$$

Let these indexes be $i_1, i_2, \ldots, i_m$.

4: *for* $k = 1, 2, \ldots, m$

Solve problem

$$\min \mathsf{E}[h(x_0)]$$
$$\text{subject to: } D\left(y^{S,(i)}, y^{M,(i)}\right) \leq h(x_0^{(i)}),$$
$$i \in \{1, 2, \ldots, N\}/\{i_k, j_1, j_2, \ldots, j_p\}.$$

If the obtained cost is better than the cost of the stored solution, then delete this latter and store the last computed solution.

end

5: Goto 2.

problem which is required to be solved in the algorithm is of standard type, i.e. with a finite number of convex constraints, and is amenable to a resolution via standard solvers like CVX, Grant and Boyd [2011], and YALMIP, Löfberg [2004]. Algorithm 1 comes to termination as long as each time Step 3 is called one active constraint whose removal improves the cost can be found. This condition is satisfied in normal situation and is assumed here for granted.

Although obtained based on a finite number of samples of x_0 and w only, the scenario solution comes with precise guarantees about its feasibility for the original chance-constrained problem, i.e. the problem with the probabilistic constraint over the whole infinite domain $\mathcal{X} \times \mathcal{W}^{[0,T]}$. This is the main feature of the scenario approach that hence can be reliably (as opposed to empirically) used to tackle chance-constrained problems otherwise deemed intractable. The following theorem precisely states this feasibility property and can be derived quite directly from Theorem 2.1 in Campi and Garatti [2011].

Theorem 1. (Feasibility of the scenario solution). If N is big enough so that (r is the dimensionality of the optimization variable)

$$\binom{\lfloor \alpha N \rfloor + r}{\lfloor \alpha N \rfloor} \sum_{i=0}^{\lfloor \alpha N \rfloor + r} \binom{N}{i} \varepsilon^i (1-\varepsilon)^{N-i} \leq \beta, \qquad (8)$$

then the scenario solution is such that

$$\mathsf{P}\left\{D\left(y^S, y^M\right) \leq h(x_0)\right\} \geq 1 - \varepsilon$$

with confidence $1 - \beta$.

Unfortunately, we cannot guarantee that the scenario solution is always feasible, because it depends on the N extracted samples $x_0^{(i)}$, $w^{(i)}$ and it may well happen that these samples are not enough representative. Yet, this latter case is very unlikely for large N and, indeed, Theorem 1 says that if N is chosen as indicated, then, the probability of such a bad event is no greater than β. It can be shown that the smallest N satisfying (8) increases logarithmically with β. Hence, we can enforce a very small value for β – like $\beta = 10^{-7}$ or even $\beta = 10^{-10}$ which guarantee the achievement of $\mathsf{P}\left\{D\left(y^S, y^M\right) \leq h(x_0)\right\} \geq 1 - \varepsilon$ beyond any reasonable doubt – without affecting N too much.

Remark 4. (Choice of α). As for the choice of the empirical probability of violation α, one should note that the closer α to the desired violation probability ε the better the approximation of the chance-constrained solution; yet, at the same time, it holds that $N \to \infty$ as $\alpha \to \varepsilon$. Intuitively, if α equals ε, then, $\mathsf{P}\left\{D\left(y^S, y^M\right) \leq h(x_0)\right\}$ will fluctuate around $1 - \varepsilon$ depending of the extracted samples of x_0 and w, and it is not possible to guarantee that $\mathsf{P}\left\{D\left(y^S, y^M\right) \leq h(x_0)\right\}$ is bigger than $1 - \varepsilon$ with high confidence for a finite N. The ultimate choice for α remains to the user, who selects its own best comprise between the accuracy required by the application and computational tractability.

Summarizing, the scenario approach provides a reliable and computationally low demanding tool by means of which the chance-constrained problems in Sections 2.1 and 2.2 can be tackled. Clearly, its application requires that the convexity assumption is verified and this, in turn, poses some conditions on the choices of $\mathbb{H}$ and $\mathcal{F}$, the class of accuracy functions and of candidate abstracted models over which optimization is performed. These choices are now discussed in the next two sections.

3.1 Abstraction performance assessment

In performance assessment, the sole optimization variable is $h(\cdot) \in \mathbb{H}$, $\mathbb{H}$ being a parametric class of positive functions, while f^M defining the model is given and fixed.

No restrictions have to be posed on f^S and f^M, which hence can be arbitrary.

Let $h_\vartheta(\cdot)$ denote the parameterized version of $h(\cdot)$. In order to apply the scenario approach, we need to ensure both the convexity of $\mathsf{E}[h_\vartheta(x_0)]$ and of the constraint $D\left(y^S, y^M\right) \leq h_\vartheta(x_0)$ with respect to ϑ, for every value of x_0 and w. Since the convexity $\mathsf{E}[h_\vartheta(x_0)]$ is achieved when $h_\vartheta(x_0)$ is a convex in ϑ, while the convexity of $D\left(y^S, y^M\right) \leq h_\vartheta(x_0)$ requires that $h_\vartheta(x_0)$ is concave in ϑ, the sole possibility is that $h_\vartheta(x_0)$ is linearly parameterized in ϑ.

One possibility is that of considering as $\mathbb{H}$ the class of positive quadratic hybrid functions of the continuous part of x_0. More precisely, letting $x_0 = (x_0^c, x_0^d)$ be the decomposition of x_0 into its continuous part x_0^c, taking value in $\mathbb{R}^n$, and its discrete part x_0^d, taking value in the finite alphabet $\{1, 2, \ldots, q\}$, $h_\vartheta(\cdot)$ can be parameterized as follows

$$h_\vartheta(x_0) = \sum_{k=1}^{q} \left[x_0^{c\prime} \Theta_k^A x_0^c + 2\Theta_k^b x_0^c + \Theta_k^c\right] \mathbf{1}_{[x_0^d = k]},$$

where $\mathbf{1}_{[\cdot]}$ is the indicator function.

Let us define the $(n+1)\times(n+1)$ symmetric matrix:

$$\Theta_k = \begin{bmatrix} \Theta_k^A & \Theta_k^{b\prime} \\ \Theta_k^b & \Theta_k^c \end{bmatrix}.$$

Then, we have that

$$x_0^{c\prime}\Theta_k^A x_0^c + 2\Theta_k^b x_0^c + \Theta_k^c = \begin{bmatrix} x_0^{c\prime} & 1 \end{bmatrix} \Theta_k \begin{bmatrix} x_0^c \\ 1 \end{bmatrix},$$

and the condition of positiveness of $h_\vartheta(x_0)$ simply translates into a positive semi-definite condition on the matrices Θ_k, i.e. $\Theta_k \succeq 0$, $k = 1, 2, \ldots, q$, which is a linear constraint on Θ_k. Moreover, $\mathsf{E}\left[h_\vartheta(x_0)\right]$ can be expanded as follows:

$$\begin{aligned}
\mathsf{E}\left[h_\vartheta(x_0)\right] &= \mathsf{E}\left[\sum_{k=1}^{q} \mathrm{tr}\left(\begin{bmatrix} x_0^{c\prime} & 1 \end{bmatrix} \Theta_k \begin{bmatrix} x_0^c \\ 1 \end{bmatrix} \mathbf{1}_{[d_0=k]}\right)\right] \\
&= \mathsf{E}\left[\sum_{k=1}^{q} \mathrm{tr}\left(\Theta_k \begin{bmatrix} x_0^c \\ 1 \end{bmatrix} \begin{bmatrix} x_0^{c\prime} & 1 \end{bmatrix} \mathbf{1}_{[x_0^d=k]}\right)\right] \\
&= \sum_{k=1}^{q} \mathrm{tr}\left(\Theta_k \mathsf{E}\left[\begin{bmatrix} x_0^c x_0^{c\prime} & x_0^c \\ x_0^{c\prime} & 1 \end{bmatrix} \mathbf{1}_{[x_0^d=k]}\right]\right) \\
&= \sum_{k=1}^{q} \mathrm{tr}\left(\Theta_k \mathsf{E}\left[\begin{bmatrix} x_0^c x_0^{c\prime} & x_0^c \\ x_0^{c\prime} & 1 \end{bmatrix} \middle| x_0^d = k\right]\right) \mathsf{P}(x_0^d = k),
\end{aligned}$$

where the conditional expectation in the last equality can be computed from the knowledge of P. In the case when $x_0 \in \mathbb{R}^n$, i.e. the state has no discrete component, then the parametrization simplifies to $h_\vartheta(x_0) = \begin{bmatrix} x_0{}' & 1 \end{bmatrix} \Theta \begin{bmatrix} x_0 \\ 1 \end{bmatrix}$, $\Theta \succeq 0$, while $\mathsf{E}[h_\vartheta(x_0)] = \mathrm{tr}\left(\Theta \mathsf{E}\begin{bmatrix} x_0 x_0{}' & x_0 \\ x_0{}' & 1 \end{bmatrix}\right)$.

The scenario implementation for the abstraction performance assessment is given in Algorithm 2.

Algorithm 2 SA for abstraction performance assessment

1: Choose $\varepsilon \in (0,1)$, $\beta \in (0,1)$, and $\alpha \in [0, \varepsilon)$. Let N the smallest integer satisfying (8).
2: Extract N realizations of the stochastic input $w^{(i)}$, $i = 1, 2, \ldots, N$, and of the initial condition $x_0^{(i)}$, $i = 1, 2, \ldots, N$;
3: Run the corresponding N executions of S and M to compute via simulation N realizations of the output signals

$$\begin{aligned} y^{S,(i)} &= f^S(x_0^{(i)}, w^{(i)}), \qquad i = 1, 2, \ldots, N \\ y^{M,(i)} &= f^M(x_0^{(i)}, w^{(i)}), \qquad i = 1, 2, \ldots, N. \end{aligned}$$

Compute $D(y^{S,(i)}, y^{M,(i)})$, $i = 1, 2, \ldots, N$.
4: Run the Scenario Algorithm with the following cost function

$$\min_{\Theta_1 \succeq 0, \ldots, \Theta_q \succeq 0} \sum_{k=1}^{q} \mathrm{tr}\left(\Theta_k \mathsf{E}\left[\begin{bmatrix} x_0^c x_0^{c\prime} & x_0^c \\ x_0^{c\prime} & 1 \end{bmatrix} \mathbf{1}_{[x_0^d=k]}\right]\right)$$

and constraints

$$D(y^{S,(i)}, y^{M,(i)}) \leq \sum_{k=1}^{q} \begin{bmatrix} x_0^{c\prime} & 1 \end{bmatrix} \Theta_k \begin{bmatrix} x_0^c \\ 1 \end{bmatrix} \mathbf{1}_{[x_0^d=k]}.$$

The class of positive quadratic function of the continuous part of x_0 seems to be rich enough for many situations of interest, where, for each mode x_0^d, the approximation capability of model M is better for a certain initial condition $x_0^c = \bar{x}_0^c$ and decreases as x_0^c moves away from $\bar{x}_0^c$. Yet, another possible parametrization for non-quadratic functions is

$$h_\vartheta(x_0) = \sum_{k=1}^{r} \vartheta_k h_k(x_0),$$

where $h_k(x_0)$, $k = 1, 2, \ldots, r$, are given positive basis functions [1], subject to the linear condition $\vartheta_k \geq 0$, $\forall k$. A scenario implementation, similar to Algorithm 2, can be given in this case too.

3.2 Abstraction performance optimization

In order to apply the scenario approach to the chance-constrained optimization problem (7), we need to parameterize both the accuracy function $h(\cdot)$ and the model M (as specified by f^M) such that both the average performance $\mathsf{E}[h(x_0)]$ and the sample constraint $D\left(y^S, y^M\right) \leq h(x_0)$ are convex as a function of the optimization variables.

If, on one hand, for $h(\cdot)$ the considerations made in the previous section continue to hold, the convexity of the constraint $D\left(y^S, y^M\right) \leq h(x_0)$ poses some restrictions on the problems that can be addressed. In particular, if we denote with $f_\lambda^M(x_0, w)$ a parametrization of M, we have to ensure that $D\left(y^S, y^M\right) = D\left(y^S, f_\lambda^M(x_0, w)\right)$ is convex as a function of λ, for all the values taken by x_0 and w. Note that a sufficient condition is that $f_\lambda^M(x_0, w)$ is linear in λ and $D\left(y^S, y^M\right)$ is convex in y^M.

This poses quite restrictive conditions on the admissible parametrization of the candidate abstracted models M. No restrictions apply instead to the system S that can be considered.

Here, we shall focus on the setting of Example 1 and address the JLSS model optimization.

Example 2. (JLSS optimization). Consider the system S and the model M described in Example 1. Let function D be defined through the standard Euclidean metric on $\mathbb{R}^p$. Clearly, D is convex in y^M.

Suppose that the matrices $\tilde{A}$, $\tilde{F}$, $\tilde{C}$, and $\tilde{R}$, defining the dynamics of M, are given, while we want to optimize the function $l : \mathbb{R}^n \to \mathbb{R}^{\tilde{n}}$ that maps the initial state x_0 of S into the initial state z_0 of M.

Given that the JLSS is characterized by linear drift and diffusion terms and by a linear reset map, it is easily seen that if we linearly parameterize function l through a $\tilde{n} \times n$ matrix L by setting $l(x_0) = Lx_0$ then, the resulting function $f_\lambda^M(x_0, w)$ is linear in $\lambda = L$ and the scenario approach can be applied to optimize the performance of M as an abstraction of S, with respect to the parameters λ (and ϑ).

Before introducing the scenario implementation of the chance-constrained optimization, we need to specify how to determine $y^M = f_\lambda^M(x_0, w)$ as an explicit function of $\lambda = L$ for each pair of initial condition x_0 and input realization w. To this purpose one can simulate $\tilde{n}$ executions of equations (3) and (4), each with the same input w and

[1] E.g., if $x_0 \in \mathbb{R}^n$, $h_k(x_0) = \exp(-(x_0 - m_k)' V_k (x_0 - m_k))$ with m_k and V_k given.

for the $\tilde{n}$ initial conditions $z_0 = e_1, \ldots, z_0 = e_{\tilde{n}}$, where e_i is the vector with all elements equal to 0 except for the i-th element equal to 1. Then, y^M can be obtained as a linear combination of these executions according to Lx_0. More precisely, letting $\xi_{i,t}$ be the execution of (3) and (4) associated with the initial condition e_i at time t, and letting

$$\Xi_t = [\xi_{1,t}\ \xi_{2,t}\ \cdots \xi_{\tilde{n},t}]$$

be the matrix with $\xi_{i,t}$ as columns, then we have $y_t^M = \tilde{C}\Xi_t L x_0$, $\forall t \in [0, T]$.

This eventually leads to Algorithm 3. □

Algorithm 3 SA for abstraction performance optimization

1: Choose $\varepsilon \in (0,1)$, $\beta \in (0,1)$, and $\alpha \in [0, \varepsilon)$. Let N the smallest integer satisfying (8).
2: Extract N realizations of the stochastic input $w^{(i)}$, $i = 1, 2, \ldots, N$, and of the initial condition $x_0^{(i)}$, $i = 1, 2, \ldots, N$;
3: Run the corresponding N executions of S to compute via simulation N realizations of the system output

$$y^{S,(i)} = f^S(x_0^{(i)}, w^{(i)}), i = 1, 2, \ldots, N$$

4: *for* $i = 1, \ldots, N$
Run $\tilde{n}$ executions of (3) and (4) with varied initial conditions $z_0 = e_1, \ldots, z_0 = e_{\tilde{n}}$ and same input equal to $w^{(i)}$ so as to generate $\Xi_t^{(i)}$.
end
5: Run the Scenario Algorithm with the following cost function

$$\min_{\Theta \succeq 0, L} \operatorname{tr}\left(\Theta \mathsf{E}\begin{bmatrix} x_0 x_0{}' & x_0 \\ x_0{}' & 1 \end{bmatrix}\right)$$

and constraints

$$D(y^{S,(i)}, y^{M,(i)}) \leq \begin{bmatrix} x_0^{(i)\prime} & 1 \end{bmatrix} \Theta \begin{bmatrix} x_0^{(i)} \\ 1 \end{bmatrix}$$

where $y_t^{M,(i)} = \tilde{C}\Xi_t^{(i)} L x_0^{(i)}, \quad t \in [0, T]$.

4. CONCLUSIONS

In this paper, we propose a simulation-based approach to the analysis and design of an approximate abstraction of a SHS. This approach rests on recent results on the randomized solution to chance-constrained programs via the scenario method.

A main advantage of the proposed approach is that it does not require specific assumptions on the system S to be approximated. In the design part, however, some restrictions are posed on the admissible parameterizations of the candidate abstracted model class. This is due to the assumptions needed for the scenario method to provide (probabilistic) guarantees on the obtained randomized solution.

It is worth noticing that some of the approaches in the literature to the design of simpler abstracted models of a hybrid system (non necessarily stochastic) do not provide a quantification of the accuracy of the obtained abstracted model, see e.g. Mazzi et al. [2008]. A possibility is then to adopt a two-step approach, where, in the first step, some method – possibly based on some heuristic – is adopted to obtain an abstracted model of the system, and, in the second step, the proposed simulation-based method is used for assessing the performance of the designed abstraction.

REFERENCES

A. Abate and M. Prandini. Approximate abstractions of stochastic systems: a randomized method. In *Proceedings of the 50th IEEE Conference on Decision and Control and European Control Conference*, Orlando, FL, 2011.

A. Abate, J.P. Katoen, J. Lygeros, and M. Prandini. Approximate model checking of stochastic hybrid systems. *European Journal of Control*, 16(6):624–641, 2010.

H.A.P. Blom and J. Lygeros, editors. *Stochastic Hybrid Systems: Theory and Safety Critical Applications*. Number 337 in Lecture Notes in Control and Information Sciences. Springer-Verlag, Berlin, 2006.

G. Calafiore and M.C. Campi. Uncertain convex programs: randomized solutions and confidence levels. *Mathematical Programming*, 102(1):25–46, 2005.

G. Calafiore and M.C. Campi. The scenario approach to robust control design. *IEEE Transactions on Automatic Control*, 51(5):742–753, 2006.

M.C. Campi and S. Garatti. The exact feasibility of randomized solutions of uncertain convex programs. *SIAM Journal on Optimization*, 19(3):1211–1230, 2008.

M.C. Campi and S. Garatti. A sampling-and-discarding approach to chance-constrained optimization: Feasibility and optimality. *Journal of Optimization Theory and Applications*, 148(2):257–280, 2011.

M.C. Campi, S. Garatti, and M. Prandini. The scenario approach for systems and control design. *Annual Reviews in Control*, 33(2):149–157, 2009.

C.G. Cassandras and J. Lygeros, editors. *Stochastic Hybrid Systems*. Number 24 in Control Engineering. CRC Press, Boca Raton, 2006.

M. Grant and S. Boyd. CVX: Matlab software for disciplined convex programming, version 1.21. `http://cvxr.com/cvx`, Feb 2011.

A.A Julius and G.J. Pappas. Approximate abstraction of stochastic hybrid systems. *IEEE Trans. Automatic Control*, 54(6):1193–1203, 2009.

A.A. Julius, A. Girard, and G.J. Pappas. Approximate bisimulation for a class of stochastic hybrid systems. In *Proceedings of the American Control Conference*, Minneapolis, MA, 2006.

J. Löfberg. Yalmip : A toolbox for modeling and optimization in MATLAB. In *Proceedings of the CACSD Conference*, Taipei, Taiwan, 2004. URL `http://control.ee.ethz.ch/ joloef/yalmip.php`.

J. Lygeros and M. Prandini. Stochastic hybrid systems: a powerful framework for complex, large scale applications. *European Journal of Control*, 16(6):583–594, 2010.

E. Mazzi, A. Sangiovanni Vincentelli, A. Balluchi, and A. Bicchi. Hybrid system model reduction. In *Proc. 47th IEEE Conf. on Decision and Control*, pages 227–232, Cancun, Mexico, 2008.

M. Petreczky and R. Vidal. Metrics and topology for nonlinear and hybrid systems. In *Proceedings of the 10th International Conference on HYBRID SYSTEMS: COMPUTATION AND CONTROL*, volume 4416 of *LCNS*, pages 459–472, 2007.