Poster Abstract: StocHy - automated verification and synthesis of stochastic processes

Nathalie Cauchi
Department of Computer Science
Oxford, United Kingdom
nathalie.cauchi@cs.ox.ac.uk

Alessandro Abate
Department of Computer Science
Oxford, United Kingdom
aabate@cs.ox.ac.uk

ABSTRACT

Stochastic hybrid systems (SHS) are a rich mathematical modelling framework capable of describing complex systems, where uncertainty and hybrid (that is, both continuous and discrete) components are relevant. We introduce a new software tool - StocHy- aimed at simplifying both the modelling of SHS and their analysis. StocHy can (i) perform verification tasks, e.g., compute the probability of staying within a certain region of the state space from a given set of initial conditions; (ii) automatically synthesise strategies maximising this probability, and (iii) simulate the SHS evolution over time. We highlight the performance of StocHy, via a set of experiment that are run on a standard laptop, with an Intel Core i7-8550U CPU at 1.80GHz × 8 and with 8 GB of RAM. StocHy is available at `gitlab.com/natchi92/StocHy`.

CCS CONCEPTS

• **Mathematics of computing** → *Markov processes.*

KEYWORDS

formal methods, verification, synthesis, stochastic hybrid systems

ACM Reference Format:
Nathalie Cauchi and Alessandro Abate. 2019. Poster Abstract: StocHy - automated verification and synthesis of stochastic processes. In *22nd ACM International Conference on Hybrid Systems: Computation and Control (HSCC '19), April 16–18, 2019, Montreal, QC, Canada.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3302504.3311805

1 IMPLEMENTATION

StocHy is implemented in C++ and employs manipulations based on vector calculus, the symbolic construction of probabilistic kernels, and multi-threading. SHS are described by parsing well-known and -used state-space models from which StocHy generates a standard SHS model automatically and formats it to be analysed. StocHy is modular, and has separate simulation, verification and synthesis engines, which are implemented as independent libraries.

HSCC '19, April 16–18, 2019, Montreal, QC, Canada

ACM ISBN 978-1-4503-6282-5/19/04.
https://doi.org/10.1145/3302504.3313349

2 DISCRETE-TIME STOCHASTIC HYBRID SYSTEMS

A SHS [1] is a discrete-time model defined as the tuple

$$\mathcal{H} = (\mathcal{Q}, n, \mathcal{U}, T_x, T_q), \quad where \tag{1}$$

- $\mathcal{Q} = \{q_1, q_2, \ldots, q_m\}$, $m \in \mathbb{N}$, represents a finite set of modes (locations);
- $n \in \mathbb{N}$ is the dimension of the continuous space $\mathbb{R}^n$ of each mode; the hybrid state space is then $\mathcal{D} = \cup_{q\in\mathcal{Q}}\{q\} \times \mathbb{R}^n$;
- $\mathcal{U}$ is a continuous set of actions, e.g. $\mathbb{R}^v$;
- $T_q : \mathcal{Q} \times \mathcal{D} \times \mathcal{U} \to [0, 1]$ is a discrete stochastic kernel on $\mathcal{Q}$ given $\mathcal{D} \times \mathcal{U}$, which assigns to each $s = (q, x) \in \mathcal{D}$ and $u \in \mathcal{U}$, a probability distribution over $\mathcal{Q} : T_q(\cdot|s, u)$;
- $T_x : \mathcal{B}(\mathbb{R}^n) \times \mathcal{D} \times \mathcal{U} \to [0, 1]$ is a Borel-measurable stochastic kernel on $\mathbb{R}^n$ given $\mathcal{D} \times \mathcal{U}$, which assigns to each $s \in \mathcal{D}$ and $u \in \mathcal{U}$ a probability measure on the Borel space $(\mathbb{R}^n, \mathcal{B}(\mathbb{R}^n)) : T_x(\cdot|s, u)$.

In this model the discrete component takes values in a finite set $\mathcal{Q}$ of modes (a.k.a. locations), each endowed with a continuous domain (the Euclidean space $\mathbb{R}^n$). The semantics of transitions at any point over a discrete time domain, are as follows: given a point $s \in \mathcal{D}$, the discrete state is chosen from T_q, and depending on the selected mode $q \in \mathcal{Q}$ the continuous state is updated according to the probabilistic law T_x. Non-determinism in the form of actions can affect both discrete and continuous transitions.

3 FORMAL VERIFICATION

StocHy performs *formal verification* of SHS via either of two abstraction techniques: (i) for discrete-time, continuous-space models with additive disturbances, and possibly with multiple discrete modes,we employ formal abstractions as general Markov chains (MC) or Markov decision processes (MDP); StocHy improves techniques the state-of-the-art FAUST^2 tool [4] by simplifying the input model description and by reducing the computational time needed to generate the abstractions; and (ii) for models with a finite number of actions, we employ interval Markov decision processes (IMDP) and the model checking framework in [3]; StocHy incorporates a novel abstraction algorithm allowing for efficient computation [2].

3.1 Comparison of verification methods

We consider a simple SHS, consisting of one discrete mode $\mathcal{Q} = \{q_0\}$ with two continuous variables $x \in \mathbb{R}^2$ which evolve according to

$$T_x = \mathcal{N}(\cdot; A_q x, G_q). \tag{2}$$

Here, $\mathcal{N}(\cdot; \eta, \nu)$ denotes a Gaussian density function with mean η and covariance matrix ν^2; $A_q = \left[\begin{smallmatrix} 0.935 & 0 \\ 0 & 0.916 \end{smallmatrix}\right]$ and $G_q = \left[\begin{smallmatrix} 1.782 & 0 \\ 0 & 0.511 \end{smallmatrix}\right]$. We are interested in computing the probability of remaining within

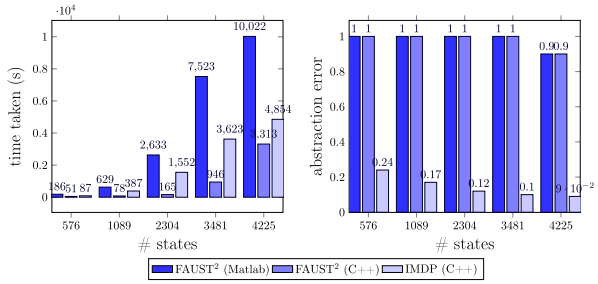

Figure 1: Comparison of verification algorithms within StocHy and the tool FAUST^2.

a safe-set $X_{safe} = \left[\begin{smallmatrix} 18 & 24 \\ 18 & 24 \end{smallmatrix}\right]$ over $K = 3$ time steps. This can be encoded into a bounded LTL property, $\varphi_1 := \Box^{\leq K=3} X_{safe}$ where $\Box$ is the "*always*" temporal operator considered over a finite horizon K. We instantiate this SHS and feed this into StocHy. We verify the model against φ_1 using both verification algorithm's within StocHy and perform a comparison with the tool FAUST^2 [4] with respect to the total time taken and the corresponding abstraction error for a fixed number of states. We depict the results of this comparison within Fig. 1. It can be seen that StocHy provides a significant improvement over the state of the art. □

4 FORMAL SYNTHESIS

StocHy carries out *control* (strategy, policy) *synthesis* via formal abstractions, employing : (i) stochastic dynamic programming; StocHy exploits the use of symbolic kernels; and (ii) robust synthesis using IMDP; StocHy automates the synthesis algorithm with the abstraction procedure, and exploits the use of sparse matrices.

4.1 Strategy synthesis example

We consider a stochastic process with two modes $Q = \{q_0, q_1\}$ and with two continuous variables $x \in \mathbb{R}^2$ evolve using (2), with,

$A_{q_0} = \left[\begin{smallmatrix} 0.43 & 0.52 \\ 0.65 & 0.12 \end{smallmatrix}\right]$, $G_{q_0} = \left[\begin{smallmatrix} 1 & 0.1 \\ 0 & 0.1 \end{smallmatrix}\right]$, $A_{q_1} = \left[\begin{smallmatrix} 0.65 & 0.12 \\ 0.52 & 0.43 \end{smallmatrix}\right]$, $G_{q_1} = \left[\begin{smallmatrix} 0.2 & 0 \\ 0 & 0.2 \end{smallmatrix}\right]$.

For this example, the actions are associated to a deterministic selection of locations.The continuous variables x are bounded within the domain shown Fig. 2a. We would like to synthesise a strategy such that given any initial condition we avoid the "*purple*" region until we reach the "*green*" region. This requirement can be expressed as an LTL formula $\varphi_2 := (\neg purple)\ \mathsf{U}\ green$ where U is the "*until*" operator and the atomic propositions $\{purple, green\}$ denote regions within the set $X = [-1.5, 1.5]^2$. We synthesise a strategy, $\pi^\star$, using IMDP algorithm within StocHy. This generates an abstraction with a total of 2410 states, a maximum probability of 1, a maximum abstraction error of 0.21 and takes 1639.3 [s]. The lower probabilities of satisfying φ_2 for each mode are shown in Fig.2b and Fig.2c. Fig.2a shows the simulation of a trajectory under $\pi^\star$ with a starting point of $(-0.5, -1)$ in q_0. □

5 SIMULATION

StocHy allows *simulation* of complex stochastic processes by means of Monte Carlo techniques; StocHy automatically generates statistics from the simulations in the form of histograms, visualising the evolution of both the continuous random variables and the discrete modes. We consider a SHS consisting of $Q = \{q_0, q_1, q_2, q_3\}$ with the

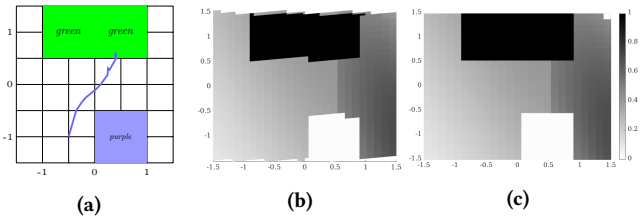

Figure 2: (a) Gridded domain and a superimposed simulation of trajectory initialised at $(-0.5, -1)$ within q_0, under π^*. Lower probabilities of satisfying φ_2 for q_0 (b) and q_1 (c).

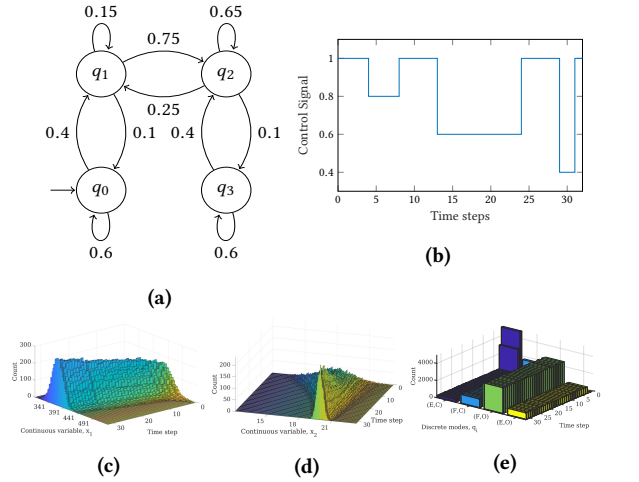

Figure 3: Histogram plots with respect to time step for (d) x_1, (e) x_2 and discrete modes (f) q.

continuous dynamics are characterised using the continuous kernel $Tx = \mathcal{N}(\cdot; A_q x + B_q u + x \sum_{i=1}^{v} N_{q,i} u_i + F_q, G_q)$. where A_q, B_q, G_q are appropriately sized matrices, $N_{q,i}$ represents the bilinear influence of the $i-$th input component u_i. The actual values of the matrices A_q, B_q, G_q, N_q are provided within the tool distribution. We depict the MC for the discrete modes and the input control signal u within Fig.3a. We simulate the evolution of this dynamical model over a fixed time horizon $K = 32$ steps, with an initial $x_1 \sim \mathcal{N}(450, 25)$ and $x_2 \sim \mathcal{N}(17, 2)$. The generated histograms depicting the range of values the continuous variables can be in during each time step and the associated count are shown in Fig. 3c for x_1 and Fig. 3d for x_2; and a histogram showing the likelihood of being in a discrete mode within each time step in Fig. 3e. The total time taken ito generate the simulations is 48.6 [s].

REFERENCES

[1] Alessandro Abate, Joost-Pieter Katoen, John Lygeros, and Maria Prandini. 2010. Approximate Model Checking of Stochastic Hybrid Systems. *European Journal of Control* 16, 6 (2010), 624–641.

[2] Nathalie Cauchi, Luca Laurenti, Morteza Lahijanian, Alessandro Abate, Marta Kwiatkowska, and Luca Cardelli. 2019. Efficiency through Uncertainty: Scalable Formal Synthesis for Stochastic Hybrid Systems. In *22nd ACM International Conference on Hybrid Systems: Computation and Control (HSCC)*. arXiv: 1901.01576.

[3] Morteza Lahijanian, Sean B Andersson, and Calin Belta. 2015. Formal verification and synthesis for discrete-time stochastic systems. *IEEE Trans. Automat. Control* 60, 8 (2015), 2031–2045.

[4] Sadegh Esmaeil Zadeh Soudjani, Caspar Gevaerts, and Alessandro Abate. 2015. FAUST^2: Formal Abstractions of Uncountable-STate STochastic Processes.. In *TACAS*, Vol. 15. 272–286.