Rare-Event Verification for Stochastic Hybrid Systems

Paolo Zuliani
Computer Science Department
Carnegie Mellon University
Pittsburgh, PA, USA
pzuliani@cs.cmu.edu

Christel Baier
Fakultät Informatik
TU Dresden
Dresden, Germany
baier@tcs.inf.tu-dresden.de

Edmund M. Clarke
Computer Science Department
Carnegie Mellon University
Pittsburgh, PA, USA
emc@cs.cmu.edu

ABSTRACT

In this paper we address the problem of verifying in stochastic hybrid systems temporal logic properties whose probability of being true is very small — rare events. It is well known that sampling-based (Monte Carlo) techniques, such as statistical model checking, do not perform well for estimating rare-event probabilities. The problem is that the sample size required for good accuracy grows too large as the event probability tends to zero. However, several techniques have been developed to address this problem. We focus on importance sampling techniques, which bias the original system to compute highly accurate and efficient estimates. The main difficulty in importance sampling is to devise a good biasing density, that is, a density yielding a low-variance estimator. In this paper, we show how to use the cross-entropy method for generating approximately optimal biasing densities for statistical model checking. We apply the method with importance sampling and statistical model checking for estimating rare-event probabilities in stochastic hybrid systems coded as Stateflow/Simulink diagrams.

Categories and Subject Descriptors

C.3 [**Special-purpose and application-base systems**]: Real-time and embedded systems; D.2.4 [**Software Engineering**]: Software/Program Verification—*statistical methods, formal methods*

Keywords

Probabilistic model checking, hybrid systems, stochastic systems, rare events, statistical model checking

1. INTRODUCTION

Stochastic hybrid systems [2] are among the most difficult systems to verify. They combine discrete, continuous, and probabilistic behavior, thereby exacerbating the state explosion problem that afflicts many automated verification techniques (*e.g.*, model checking). In particular, temporal logic verification for stochastic hybrid systems is currently outside the reach of formal verification methods. To deal with this problem, one can instead use *statistical* model checking. This technique blends randomized (*i.e.*, Monte Carlo) simulation, model checking, and statistical analysis, and it enjoys better scalability than other formal verification techniques [17, 16]. With statistical model checking one can compute approximations of the probability that a stochastic hybrid system satisfies a given temporal logic specification. The accuracy of the computed probability can be controlled by the user (the probability is usually given with a *confidence interval*, and the user can control *width* and *coverage* of the interval). Naturally, higher accuracy will require more simulations. Since the vast majority of the computational cost of statistical model checking is due to system simulation, it is important to keep the *sample size* — the number of simulations — as small as possible. In most cases, statistical model checking techniques can give accurate estimates with feasible sample sizes, *i.e.*, smaller than 10^4. However, it is well known that Monte Carlo techniques, such as statistical model checking, suffer from the *rare-event* problem [10]. An event is said rare when it occurs with very small probability. For example, the negation of a safety property can be thought as a rare event: it should be very unlikely that the system is unsafe. Now, the problem is that estimating accurately rare-event probabilities using standard Monte Carlo techniques requires very high sample sizes. Thus, these techniques quickly become unfeasible, as we next explain.

The Monte Carlo approach for estimating probabilities relies on the strong law of large numbers for its correctness, and on relative frequencies for computing estimates. The strong law of large numbers states that if $X_1, X_2, \ldots$ is a sequence of independent and identically distributed (iid) random variables with $\mathrm{E}[|X_1|] < \infty$, then

$$\mathrm{P}\left(\lim_{n\to\infty} \frac{S_n}{n} = \mu\right) = 1$$

where $S_n = \sum_{i=1}^{n} X_i$ and $\mu = \mathrm{E}[X_1]$. Therefore, we can approximate μ by taking the average of a *finite* number of realizations (samples) of X_1, since we know that the average will not converge to μ only for a negligible subset of realizations (a set of measure 0). It can be shown that the condition $\mathrm{E}[|X_1|] < \infty$ is necessary and sufficient for the average $\frac{S_n}{n}$ to converge to a finite limit (with probability 1). Also, the strong law of large numbers holds in the case that $\mu = \mathrm{E}[X_1]$ exists but it is not finite [14, Chapter 4].

Now, suppose we want to estimate $p = \mathrm{P}(X \in B)$, the probability that X belongs to a given Borel set B, where X is

HSCC'12, April 17–19, 2012, Beijing, China.

a random variable defined over a probability space $(\Omega, \mathcal{F}, \mathrm{P})$. First, we obtain a number of independent realizations of $I_B(X)$, the indicator function of B — $I_B(x)$ is 1 if $x \in B$ ("$X \in B$ has occurred"), 0 otherwise. Then, we compute their average and return that as the estimate of p. Note that the random variable $I_B(X)$ is a Bernoulli of success parameter p, that is, $\mathrm{P}(I_B(X) = 1) = p$. Also, note that $p = \mathrm{E}[I_B(X)]$. Therefore, given a finite sequence $X_1, \ldots, X_N$ of random variables iid as X, we define the *crude Monte Carlo estimator* as $\hat{p} = \frac{1}{N}\sum_{i=1}^{N} I_B(X_i)$. By the strong law of large numbers $\hat{p}$ converges to p as $N \to \infty$ (with probability 1). Also, $\hat{p}$ is *unbiased* (*i.e.*, $\mathrm{E}[\hat{p}] = p$).

The speed of convergence of $\hat{p}$ depends on the variance of $I_B(X)$, which is finite (it is of course $p(1-p)$). In particular, from the central limit theorem it follows that for large N the distribution of $\hat{p}$ is approximately a normal distribution of mean p and variance $\mathrm{Var}(I_B(X))/N$. From this we can compute approximate *confidence intervals* for p in the following way. Let z_γ denote the γ-quantile of the standard normal distribution, *i.e.*, the number such that $\mathrm{P}(\mathcal{N} \leqslant z_\gamma) = \gamma$, where $\mathcal{N}$ is a normal random variable with mean 0 and variance 1. Then, for $\alpha < 1$ and large N the following holds:

$$\mathrm{P}\left(|p - \hat{p}| \leqslant z_{1-\frac{\alpha}{2}} \frac{\mathcal{S}}{\sqrt{N}}\right) \approx 1 - \alpha$$

where $\mathcal{S}$ is the square root of *sample variance*

$$\mathcal{S}^2 \stackrel{\text{def}}{=} \frac{1}{N-1}\sum_{i=1}^{N}(I_B(X_i) - \hat{p})^2$$

which, again by the strong law of large numbers, converges to $\mathrm{Var}(I_B(X))$ with probability 1. The term $2z_{1-\frac{\alpha}{2}}\frac{\mathcal{S}}{\sqrt{N}}$ is the *absolute width* of the $(1-\alpha)100\%$ confidence interval. In the rare event case ($p \ll 1$) it is very important to have confidence intervals of *relative* width, *i.e.*, we would like an estimate $\hat{p}$ such that

$$\mathrm{P}\left(|p - \hat{p}| \leqslant \delta p\right) \approx 1 - \alpha$$

for some small $\delta > 0$. Clearly, a confidence interval of absolute width 0.01 would not make much sense if we wanted to estimate, say, $p = 10^{-10}$. For example, it can be shown that a 99% approximate confidence interval of relative width δ needs about $\frac{1-p}{p\delta^2}$ samples. To estimate $p = 10^{-8}$ with a relative width $\delta = 0.01$ we would thus need about $N \approx \frac{1}{p\delta^2} = 10^{12}$ samples — an unfeasible quantity. Furthermore, we see that if $p \to 0$ while δ is fixed, the sample size grows larger and larger.

Finally, an important quantity associated with the estimator $\hat{p}$ is its *relative error*:

$$\mathrm{RE}(\hat{p}) \stackrel{\text{def}}{=} \frac{\sqrt{\mathrm{Var}(\hat{p})}}{\mathrm{E}[\hat{p}]}$$

and intuitively it is a "measure" of the accuracy of the estimator $\hat{p}$ with respect to its standard deviation. Since $\hat{p}$ is unbiased, the sample $X_1, \ldots, X_N$ is iid, and $p \ll 1$, it follows that

$$\mathrm{RE}(\hat{p}) = \frac{\sqrt{\mathrm{Var}(I_B(X))/N}}{p} = \frac{\sqrt{p(1-p)}}{p\sqrt{N}} \approx \sqrt{\frac{1}{Np}}\ .$$

It is easy to see that if N is kept constant and $p \to 0$, then $\mathrm{RE}(\hat{p}) \to \infty$. Therefore, in order to keep the relative error low as $X \in B$ becomes rarer, we need to increase the sample size. This means that the crude MC estimator is useless in the rare-event case.

A possible solution to this problem is to search for another estimator whose variance is smaller than $\mathrm{Var}(\hat{p})$, for a given sample size. Importance sampling is a technique for devising estimators with reduced variance, and thus with low relative error. In particular, in importance sampling the original system is biased to increase the likelihood of the event of interest. The samples are then weighted in order to obtain unbiased estimates. The main difficulty in importance sampling is to devise a good biasing distribution, that is, one yielding a low-variance estimator. The cross-entropy method is a recent technique that can help in devising a biased distribution.

In this work we use statistical model checking with importance sampling and the cross-entropy method for estimating rare-event probabilities in stochastic hybrid systems. The paper is divided as follows. In Section 2 we briefly recapitulate temporal logic and statistical model checking; in Section 3 we define our semantic model for stochastic hybrid systems; in Sections 4 and 5 we introduce importance sampling and the cross-entropy method, respectively. Finally, in Section 6 we apply the techniques to an example of stochastic hybrid system modeled in Stateflow/Simulink.

2. STATISTICAL MODEL CHECKING

We give a short introduction to temporal logic and statistical model checking. In this paper, we use Bounded Linear Temporal Logic (BLTL) [7, 21, 6] as our specification language. BLTL restricts Linear Temporal Logic (LTL) [9] with time bounds on the temporal operators. For example, we can specify that "within 10 time units the system will shut down and the `shutdown` signal will be ON *until* then" as the BLTL formula

$$\texttt{shutdown\_ON}\ \mathbf{U}^{10}\ \texttt{sysdown}$$

where `shutdown_ON` and `sysdown` are predicates over the system's state space and time defined to be true iff the shutdown signal is ON and iff the system is down at that time, respectively. Again, a BLTL formula expressing the specification "it is not the case that in the *future* 25 time units the system is *globally* down for one time unit" is written as

$$\neg(\mathbf{F}^{25}\mathbf{G}^{1}\ \texttt{sysdown})$$

where the $\mathbf{F}^{25}$ operator encodes "future 25 time units", and $\mathbf{G}^1$ expresses "globally for one time unit". Formally, the syntax of BLTL is given by:

$$\phi ::= y \sim v \mid (\phi_1 \vee \phi_2) \mid (\phi_1 \wedge \phi_2) \mid \neg\phi_1 \mid (\phi_1 \mathbf{U}^t \phi_2),$$

where $\sim \in \{\geq, \leq, =\}$, $y \in SV$ (the finite set of state variables), $v \in \mathbb{R}$, $t \in \mathbb{R}_{>0}$, and $\neg, \wedge, \vee$ are the usual Boolean connectives. Formulae of the type $y \sim v$ are also called atomic propositions (AP). The formula $\phi_1 \mathbf{U}^t \phi_2$ holds true if and only if, *within* time t, ϕ_2 will be true and ϕ_1 will hold *until* then. Note that the operators $\mathbf{F}^t$ and $\mathbf{G}^t$ referenced above can be easily defined in terms of the until $\mathbf{U}^t$ operator: $\mathbf{F}^t\phi = \mathit{true}\ \mathbf{U}^t\phi$ requires ϕ to hold true within time t (*true* is the atomic proposition identically true); $\mathbf{G}^t\phi = \neg\mathbf{F}^t\neg\phi$ requires ϕ to hold true up to time t.

The semantics of BLTL formulae [7, 21, 6] is defined with respect to system traces (or executions). A trace is a sequence $\sigma = (s_0, t_0), (s_1, t_1), \ldots$ where the s_i's are states and

the t_i's represent time. The pair (s_i, t_i) expresses the fact that the system moved to state s_{i+1} after having spent t_i time units in state s_i. The trace suffix of σ starting at $k \in \mathbb{N}$ is denoted by σ^k, and σ^0 denotes the full trace σ.

Definition 1. The *semantics* of BLTL for a trace σ^k is:

- $\sigma^k \models AP$ *iff* *AP holds true in state* s_k;
- $\sigma^k \models \phi_1 \vee \phi_2$ *iff* $\sigma^k \models \phi_1$ *or* $\sigma^k \models \phi_2$;
- $\sigma^k \models \phi_1 \wedge \phi_2$ *iff* $\sigma^k \models \phi_1$ *and* $\sigma^k \models \phi_2$;
- $\sigma^k \models \neg\phi_1$ *iff* $\sigma^k \models \phi_1$ *does not hold;*
- $\sigma^k \models \phi_1 \mathbf{U}^t \phi_2$ *iff* $\exists i \geq 0$ *such that*

  a) $\sum_{l=0}^{i-1} t_{k+l} \leq t$, and

  b) $\sigma^{k+i} \models \phi_2$, and

  c) $\forall\ 0 \leq j < i$, $\sigma^{k+j} \models \phi_1$.

If the trace σ satisfies the property ϕ we write $\sigma \models \phi$.

Statistical model checking [19, 18, 5, 13, 4] combines Monte Carlo simulation, model checking, and statistical analysis, for verifying stochastic systems. Its main assumption is the existence of a probability measure P over the set of system traces satisfying a given BLTL formula. In particular, for every BLTL formula ϕ, the probability $\mathrm{P}\{\sigma \mid \sigma \models \phi\}$ must be well-defined. Given a stochastic process and probability measure over it, this requirement does not pose any problem in practice — see [20] for more details. In the next Section we define a model for stochastic hybrid systems and we show that it induces a well-defined stochastic process and a unique probability measure over the process' traces. This is clearly a crucial requirement for statistical model checking to make sense.

Suppose now that $p = \mathrm{P}\{\sigma \mid \sigma \models \phi\}$ for a given formula ϕ. The verification problem is thus to compute (or approximate) p. Statistical model checking treats it as a statistical inference problem, and solves it through randomized sampling of the system traces. The traces are model checked to determine whether ϕ holds, and the number of satisfying traces is used to *estimate* p. Specifically, we seek to approximate probabilistically (*i.e.*, compute with high probability a value close to) p. Note that the system behavior with respect to ϕ can be characterized as a Bernoulli random variable under the measure P. Given a system trace σ we can define the Bernoulli random variable Z to be 1 if $\sigma \models \phi$, and 0 otherwise. Thus, $\mathrm{P}(Z = 1) = p$ (and of course $\mathrm{P}(Z = 0) = 1 - p$). In statistical model checking, one therefore aims at estimating the success parameter of Z. Statistical techniques are applied to independent samples of Z to estimate p. In particular, to obtain n samples of Z we first have to run n iid system simulations that yield the traces $\sigma_1, \ldots, \sigma_n$, and then we check property ϕ on each trace σ_i. To estimate p, one can then use fixed-sample size statistical techniques such as the Chernoff-Hoeffding bound [5], or sequential techniques such as Bayesian credibility intervals [21]. Another statistical model checking approach [19, 18] uses statistical hypothesis testing techniques aimed at deciding whether p is greater than a given threshold. However, such techniques suffer from the rare-event problem, too.

We have seen that in order to generate each sample of Z we need to check property ϕ on a trace. Because BLTL properties are time-bounded, it is possible to decide whether a trace σ satisfies a given property only by checking a finite prefix of σ [21]. That result assumes that the system under verification does not exhibit Zeno behavior. In particular, for any system trace σ it must be $\sum_{i=0}^{\infty} t_i = \infty$, which means that the system cannot make an infinite number of transitions in a finite amount of time. This assumption is widely adopted and it is sufficient for ensuring termination of statistical model checking algorithms. (However, it is not always necessary. For example, for finite-state continuous-time Markov chains it can be shown that the set of traces exhibiting Zeno behavior has measure zero [1].)

3. STOCHASTIC HYBRID SYSTEMS

In this Section we present our semantic model for stochastic hybrid systems, and we prove that it induces a well-defined Markov process. The model is especially suited for capturing the behavior of simulation engines for hybrid systems, such as Stateflow/Simulink.

3.1 Preliminaries

We shall consider stochastic processes over Polish spaces. A Polish space is a separable topological space metrizable by a complete metric. A Borel set in a topological space is a set formed by countable union, intersection, or relative complement of open sets (equivalently, closed sets). Given a Polish space S, we denote its Borel σ-algebra by $\mathcal{B}(S)$.

Definition 2. A *stochastic kernel* on a measurable space $(S, \mathcal{B}(S))$ is a function $K{:}S \times \mathcal{B}(S) \to [0, 1]$ such that:

- for each $x \in S$, $K(x, \cdot)$ is a probability measure on $\mathcal{B}(S)$; and
- for each $B \in \mathcal{B}(S)$, $K(\cdot, B)$ is a (Borel) measurable function on S.

Since we consider discrete time systems, we define the sample space $\Omega = S^\omega$ and the product σ-algebra $\mathcal{F}$ of Ω. Given a stochastic kernel K on $(\Omega, \mathcal{F})$ and an initial state $x \in S$, then Kolmogorov's theorem shows [14, Section II.9] that there exists a unique probability measure P defined on $(\Omega, \mathcal{F})$ and a Markov process $\{X_t : t \in \mathbb{N}\}$ such that for all $B \in \mathcal{B}(S)$ and for all $x_i \in S$:

- $\mathrm{P}(X_1 \in B) = \delta_B(x)$; and
- $\mathrm{P}(X_{t+1} \in B \,|\, (x_1, \ldots, x_t)) = \mathrm{P}(X_{t+1} \in B \,|\, x_t) = K(x_t, B)$

where δ_B is the usual Dirac measure.

Our aim is to introduce a hybrid automaton model and a "probabilistic simulation function" that will induce a stochastic kernel, in order to use Kolmogorov's theorem.

3.2 Hybrid Automata

We first define non-probabilistic hybrid automata.

Definition 3. A *discrete-time hybrid automaton* (DTHA) consists of:

- a continuous state space $\mathbb{R}^n$;
- a finite set Q of *locations*;
- an edge relation $E \subseteq Q \times Q$ (control switches);
- one initial state $(q_0, x_0) \in Q \times \mathbb{R}^n$;
- a *flow* function $\varphi : Q \times \mathbb{R}_{>0} \times \mathbb{R}^n \to \mathbb{R}^n$ representing the time evolution of the (continuous) state, in a specific location. For each $q \in Q$, the flow function

$$\varphi_q : \mathbb{R}_{>0} \times \mathbb{R}^n,\ (t,x) \mapsto \varphi(q,t,x),$$

is (Borel) measurable.

- a *jump* function $jump : E \times \mathbb{R}^n \to \mathbb{R}^n$, representing the (possibly) discontinuous change of state after switching location.

A DTHA may feature nondeterminism because of multiple outgoing edges from a location. We assume that the directed graph (Q, E) of locations does not have self-loops or terminal locations, *i.e.*, $(q,q) \notin E$ for all $q \in Q$ and for each $q \in Q$ there is at least one edge $(q,q') \in E$. Also, note that continuous flow functions are automatically (Borel) measurable.

Notation:. If $e \in E$, we shall write $jump_e(x)$ for $jump(e,x)$. Similarly, if $q \in Q$ then φ_q denotes the function $(t,x) \mapsto \varphi(q,t,x)$.

Definition 4. The *semantics* of a DTHA is a transition system $\mathcal{T}$ with

- state space $S = Q \times \mathbb{R}^n$
- initial state $s_0 = (q_0, x_0) \in S$
- transition relation $\longrightarrow \subseteq S \times (E \cup \mathbb{R}_{>0}) \times S$ given by the following two rules:

$$\frac{x' = \varphi_q(t,x)}{(q,x) \longrightarrow_t (q,x')} \qquad \frac{e = (q,q') \in E, \quad x' = jump_e(x)}{(q,x) \longrightarrow_e (q',x')}$$

continuous transition (time passage) | discrete transition (switching location)

Any nondeterminism in a DTHA is resolved by a *simulation function*. In particular, such function can capture the determinism necessary for simulating a DTHA. An example is the "12 o'clock" graphical rule in Stateflow diagrams. (It states that the first edge, in clockwise orientation from 12, that is enabled shall be selected.)

Definition 5. A *simulation function* for a DTHA is a map

$$\Delta : S \to E \cup \mathbb{R}_{>0}$$

A simulation function induces a subsystem of $\mathcal{T}$ where each state $s \in S$ has a unique successor state, namely the unique state s' such that $s \longrightarrow_\sigma s'$ where $\sigma = \Delta(s)$. In particular, Δ induces an infinite path in $\mathcal{T}$:

$$s_0\, s_1\, s_2\, s_3\, \ldots \quad \text{where} \quad s_i \longrightarrow_{\Delta(s_i)} s_{i+1} \quad \text{for } i = 0,1,2,\ldots$$

An alternative definition of a deterministic simulation function could be to take the absolute time as an additional parameter. That is,

$$\Delta : S \times \mathbb{R}_{\geqslant 0} \to E \cup \mathbb{R}_{>0}$$

which induces a timed path of $\mathcal{T}$:

$$path(\Delta) = (s_0,\theta_0)\,(s_1,\theta_1)\,(s_2,\theta_2)\,\ldots \in (S \times \mathbb{R}^n_{\geqslant 0})^\omega$$

where s_0 is the initial state of $\mathcal{T}$, $\theta_0 = 0$ and for each $i \in \mathbb{N}$:

- if $\Delta(s_i,\theta_i) = e \in E$ then $\theta_{i+1} = \theta_i$ and s_{i+1} is the unique state in S such that $s_i \longrightarrow_e s_{i+1}$
- if $\Delta(s_i,\theta_i) = t \in \mathbb{R}_{>0}$ then $\theta_{i+1} = \theta_i + t$ and s_{i+1} is the unique state in S such that $s_i \longrightarrow_t s_{i+1}$

This is slightly more "powerful" since it allows to make different choices for the same state s when visiting s at different time instances. A corresponding time-dependent (or even history-dependent) definition of probabilistic simulation functions could be defined for the probabilistic case. In this paper we exclusively deal with Markovian systems, so we shall not pursue the more general case.

Now, Discrete Time Stochastic Hybrid Automata (DTSHA) are obtained by replacing the (deterministic) simulation function Δ with a probabilistic version. The *probabilistic* simulation function decides for each state $s = (q,x) \in S$

- either to take some continuous transition; in which case the size of the time step will be "sampled" according to some probability distribution over the non-negative reals;
- or to take some discrete transition; in which case the choice of which edge $(q,q') \in E$ to take is resolved according to a probability distribution over

$$E(q) = \{e \in E : e = (q,q') \text{ for some } q' \in Q\}.$$

Remember that we suppose $E(q) \neq \varnothing$ for all $q \in Q$.

We denote by $\mathbb{P}(\cdot)$ the set of probability measures over $(\cdot)$.

Definition 6. A *probabilistic simulation function* is a map

$$\lambda : S \to \mathbb{P}(\mathcal{B}(\mathbb{R}_{>0})) \cup \mathbb{P}(E)$$

satisfying the conditions (where we assume $s = (q,x) \in S$):

(P1) For each state $s \in S$ such that $\lambda(s) \in \mathbb{P}(E)$ and each edge $e \in E$ we have:

$$\text{if } e = (p,p') \text{ where } p \neq q \text{ then } \lambda(s)(e) = 0$$

This is equivalent to $\sum_{e \in E(q)} \lambda(s)(e) = 1$.

(P2) For each state $s \in S$ such that $\lambda(s) \in \mathbb{P}(\mathcal{B}(\mathbb{R}_{>0}))$, we define the function

$$\Pi_s : \mathcal{B}(S) \to [0,1], \quad \Pi_s(B) \stackrel{\text{def}}{=} \lambda(s)(time(s,B))$$

where $time(s,B)$ is the set of time points for which evolution (from state $s = (q,x) \in S$) of $\varphi_{q,x}(\cdot) = \varphi_q(\cdot,x)$ ends up in B. Formally,

$$time(s,B) \stackrel{\text{def}}{=} \big\{t \in \mathbb{R}_{>0} : \varphi_q(t,x) \in B\big\} = \varphi^{-1}_{q,x}(B_q)$$

where

$$\varphi_{q,x} : \mathbb{R}_{>0} \to \mathbb{R}^n, \quad \varphi_{q,x}(t) = \varphi_q(t,x)$$
$$B_q \stackrel{\text{def}}{=} \big\{y \in \mathbb{R}^n : (q,y) \in B\big\}.$$

(P3) The sets of states for which λ is a probability measure over reals and over E must be measurable. That is, the sets

$$S_c \stackrel{\text{def}}{=} \big\{s \in S : \lambda(s) \in \mathbb{P}(\mathcal{B}(\mathbb{R}_{>0}))\big\}$$
$$S_d \stackrel{\text{def}}{=} \big\{s \in S : \lambda(s) \in \mathbb{P}(E)\big\}$$

are measurable.

(P4) For each edge $e = (q,q') \in E$, the function $\psi_e : S_d \to [0,1]$, $s \mapsto \lambda(s)(e)$ is measurable over S_d.

(P5) For each Borel-set $B \subseteq \mathcal{B}(S)$, the function $\phi_B : S_c \to [0,1]$, $s \mapsto \lambda(s)(time(s,B))$ is measurable over S_c.

Remark 1. In condition (P2) for fixed state $s = (q,x)$, the function $\varphi_{q,x}$ is measurable when we require that φ is measurable. If B is a measurable subset of $S = Q \times \mathbb{R}^n$ then $B_q = \{y \in \mathbb{R}^n : (q,y) \in B\}$ is a measurable subset of $\mathbb{R}^n$. Therefore, the pre-image $\varphi_{q,x}^{-1}(B_q)$ is a measurable subset of $\mathbb{R}_{>0}$.

As Q is finite, the measures over E are discrete. For measures over the non-negative reals if, for example, $\lambda(s)$ corresponds to an exponential distribution of parameter $\kappa(s) \in \mathbb{R}_{>0}$, then for all $T \in \mathcal{B}(\mathbb{R}_{>0})$

$$\lambda(s)(T) \;=\; \int_T \kappa(s) \cdot e^{-\kappa(s)t} dt \;.$$

We can now define the stochastic kernel induced by a probabilistic simulation function.

Definition 7. A probabilistic simulation function λ induces the stochastic kernel $\Pi : S \times \mathcal{B}(S) \to [0,1]$ defined as follows:

$$\Pi(s,B) \stackrel{\text{def}}{=} \begin{cases} \Pi_s(B) & \text{if } s \in S_c \\ \Psi_s(B) & \text{if } s \in S_d \end{cases}$$

where Π_s is as per Definition 6 (condition P2), and Ψ_s is the map

$$\Psi_s(B) \stackrel{\text{def}}{=} \sum_{e \in Edges(s,B)} \psi_e(s) = \sum_{e \in Edges(s,B)} \lambda(s)(e)$$

where $Edges(s,B)$ is the set of edges for which a transition from state s results in a state in B. Formally (where $s = (q,x) \in S$):

$$Edges(s,B) \stackrel{\text{def}}{=} \big\{e = (q,q') \in E : (q', jump_e(x)) \in B\big\}.$$

PROPOSITION 1. *The function Π of Definition 7 is a stochastic kernel.*

PROOF. We start by showing that for each state $s \in S$, $\Pi(s,\cdot)$ is a probability measure over $\mathcal{B}(S)$. If $\lambda(s) \in \mathbb{P}(\mathcal{B}(\mathbb{R}_{>0}))$ then by condition (P2) in Definition 6

$$\Pi(s,B) = \Pi_s(B) = \lambda(s)\big(\varphi_{q,x}^{-1}(B_q)\big)$$

and this is indeed a probability measure. If $\lambda(s) \in \mathbb{P}(E)$, then

$$\Pi(s,B) = \Psi_s(B) = \sum_{(q,q') \in E} \lambda(s)(e) \cdot \delta_B(q', jump_e(x))$$

where δ_B is the Dirac measure over B. Condition (P1) and the assumption that $(q,q') \in E$ for at least one location $q' \in Q$ ensure that $\Pi(s,S) = 1$. Since E is finite, this is a probability measure too.

Next, we need to show that for each $B \in \mathcal{B}(S)$, the function $\Pi_B : S \to [0,1]$, $s \mapsto \Pi(s,B)$ is measurable. We must thus show that for any $I \in \mathcal{B}([0,1])$ the set $\Pi_B^{-1}(I)$ is measurable. Note that:

$$\begin{aligned} \Pi_B^{-1}(I) &= \{s \in S : \Pi(s,B) \in I\} \\ &= \{s \in S_c : \Pi_s(B) \in I\} \;\cup\; \{s \in S_d : \Psi_s(B) \in I\} \\ &= \{s \in S_c : \phi_B(s) \in I\} \;\cup\; \{s \in S_d : \Psi_s(B) \in I\} \\ &= \phi_B^{-1}(I) \;\cup\; \{s \in S_d : \sum_{e \in Edges(s,B)} \psi_s(e) \in I\} \end{aligned}$$

Measurability of $\Pi_B^{-1}(I)$ follows thus directly from conditions (P4)-(P5) and $\cup-$closedness. □

Proposition 1 enables us to show (see Section 3.1) the existence of the discrete-time Markov process and the probability measure over the product σ-algebra $\mathcal{F}$ for our stochastic hybrid systems model.

4. IMPORTANCE SAMPLING

Importance Sampling is a variance-reduction technique for the Monte Carlo method. Here we present a brief overview of the technique — the interested reader can find more details in [15], for example.

4.1 Basics

Consider the general case of estimating $c = \mathrm{E}[g(X)]$ for a random variable X and a measurable function $g{:}\mathbb{R} \to \mathbb{R}^{\geqslant 0}$, assuming $0 < c < \infty$. We also assume that the distribution of X is absolutely continuous with respect to the Lebesgue measure, and denote by f the corresponding density. Recall that in statistical model checking we are interested in determining the probability that a stochastic system satisfies a certain temporal logic formula ϕ. In this setting, the function g is just the model checker that verifies whether a trace satisfies ϕ. Therefore, given a random trace σ, the random variable $g(\sigma)$ is a Bernoulli — 1 if the trace σ satisfies ϕ, and 0 otherwise.

Let $X_1, \ldots, X_N$ be random variables iid as X. The *crude Monte Carlo* (MC) estimator is $\hat{c} \stackrel{\text{def}}{=} \frac{1}{N}\sum_{i=1}^N g(X_i)$. By the strong law of large numbers, $\hat{c}$ converges to c with probability 1. (Clearly, the sequence $g(X_1), \ldots, g(X_N)$ is iid with mean $\mathrm{E}[g(X)]$, so the law of large numbers applies.) Also, $\hat{c}$ is unbiased, and its variance is

$$\mathrm{Var}(\hat{c}) = \frac{1}{N}(\mathrm{E}[g^2(X)] - c^2) \;. \tag{1}$$

We now introduce Importance Sampling. Suppose we had another (absolutely continuous) distribution for X, with corresponding density f_*, such that the ratio f/f_* is well-defined. Importance sampling is based upon the following identity:

$$\begin{aligned} c &= \mathrm{E}[g(X)] \\ &= \int_{\mathbb{R}} g(x) f(x)\, dx \\ &= \int_{\mathbb{R}} g(x) \frac{f(x)}{f_*(x)} f_*(x)\, dx \\ &= \int_{\mathbb{R}} g(x) W(x) f_*(x)\, dx \\ &= \mathrm{E}_*[g(X)W(X)] \end{aligned} \tag{2}$$

where E_* denotes expectation with respect to the density f_*. The term $W(x) = \frac{f(x)}{f_*(x)}$ is the *likelihood ratio*. We require that for all x such that $g(x)f(x) > 0$, it must be $f_*(x) > 0$; the density f_* is known as the *biasing* (or *proposal*) density.

Definition 8. Let $X_1, \ldots, X_N$ be random variables iid with density f_*. The *Importance Sampling* (IS) estimator is

$$\hat{c}_{\mathrm{IS}} = \frac{1}{N}\sum_{i=1}^N g(X_i) W(X_i)$$

where $W(x) = f(x)/f_*(x)$ is the likelihood ratio.

Note that the samples X_i's are drawn from the proposal distribution. The IS estimator is unbiased by (2), and its variance is (see Appendix A):

$$\mathrm{Var}(\hat{c}_{\mathrm{IS}}) = \frac{1}{N}(\mathrm{E}_*[g^2(X)W^2(X)] - c^2) \ . \tag{3}$$

The key problem in importance sampling is to find a proposal density such that the variance (3) of the IS estimator is smaller than the variance (1) of the crude MC estimator.

4.2 Optimal bias

It is not difficult to show that there exists a proposal density which can minimize the variance (3) of the IS estimator. In particular, if the function g is non-negative the following *optimal proposal* density results in a zero-variance estimator:

$$f_*(x) \stackrel{\mathrm{def}}{=} \frac{g(x)f(x)}{c} \ . \tag{4}$$

When g is a real (non necessarily positive) function the variance can be minimized, although the minimum is non-zero — see Appendix A.

The claim that (4) gives a zero-variance estimator can be easily verified:

$$\begin{aligned}\hat{c}_{\mathrm{IS}} &= \frac{1}{N}\sum_{i=1}^{N} g(X_i)W(X_i) = \frac{1}{N}\sum_{i=1}^{N} g(X_i)\frac{f(X_i)}{f_*(X_i)} \\ &= \frac{c}{N}\sum_{i=1}^{N} g(X_i)\frac{f(X_i)}{g(X_i)f(X_i)} = c \ .\end{aligned}$$

Therefore, for any sample size (with at least one sample x for which $g(x) \neq 0$) the IS estimator is constant. But this does not help in practice, since f_* depends on $c = \mathrm{E}[g(X)]$, the (unknown) quantity we are trying to estimate. Therefore, instead of trying to come up with the optimal density, it may be preferable to search in a parametrized family of densities for a biasing density "close" to the optimal one. This is exactly the approach taken by the cross-entropy method, as we show in the next Section.

5. THE CROSS-ENTROPY METHOD

The cross-entropy method was introduced in 1999 by Rubinstein [11]. It assumes that the original (or nominal) density f of X belongs to a parametric family $\{f(\cdot, u) \,|\, u \in \mathcal{U}\}$, and in particular $f(\cdot) = f(\cdot, v)$ for some fixed $v \in \mathcal{U}$. The method seeks the density in the family which minimizes the Kullback-Leibler divergence with the optimal proposal density. Basically, to estimate probabilities using importance sampling and the cross-entropy method, we perform two steps. First, we find a density with minimal Kullback-Leibler divergence with respect to the optimal proposal density. Second, we perform importance sampling with the proposal density computed in the previous step to estimate $\mathrm{E}[g(X)]$. Both steps require sampling, and in practice the number of samples generated for the second step will be much larger than for the first.

Definition 9. The *Kullback-Leibler* divergence of two densities f, h is

$$\mathcal{D}(f,h) = \int_{\mathbb{R}} f(x) \ln \frac{f(x)}{h(x)} \, dx.$$

The Kullback-Leibler divergence is also known as the *cross-entropy* (CE). Formally, it is not a distance, since it is not symmetric, *i.e.*, $\mathcal{D}(f,h) \neq \mathcal{D}(h,f)$ in general. However, it can be shown (see Appendix B) that $\mathcal{D}$ is always non-negative, and that $\mathcal{D}(f,h) = 0$ iff $f = h$. Therefore, the CE can be useful in assessing how close two densities are.

Our task is to estimate $c = \mathrm{E}[g(X)]$, where X is a random variable with density f and g is a non-negative, measurable function. Again, the idea of the CE method is to find a density in the parametric family such that the CE with the optimal proposal density f_* is minimal. Therefore, we need to solve the minimization problem:

$$u^* \stackrel{\mathrm{def}}{=} \operatorname*{argmin}_{u \in \mathcal{U}} \ \mathcal{D}(f_*(\cdot), f(\cdot, u))$$

where $f_*(x) = g(x)f(x,v)/c$ is the optimal proposal density. It is easy to transform the minimization problem into a maximization problem:

$$\begin{aligned}\operatorname*{argmin}_{u \in \mathcal{U}} \mathcal{D}(f_*(\cdot), f(\cdot, u)) &= \operatorname*{argmin}_{u \in \mathcal{U}} \ \mathrm{E}_* \left[\ln \frac{f_*(X)}{f(X,u)} \right] \\ &= \operatorname*{argmin}_{u \in \mathcal{U}} \int_{\mathbb{R}} f_*(x) \ln f_*(x) \, dx - \int_{\mathbb{R}} f_*(x) \ln f(x,u) \, dx \\ &= \operatorname*{argmax}_{u \in \mathcal{U}} \int_{\mathbb{R}} f_*(x) \ln f(x,u) \, dx \\ &= \operatorname*{argmax}_{u \in \mathcal{U}} \int_{\mathbb{R}} g(x) f(x,v) \ln f(x,u) \, dx \\ &= \operatorname*{argmax}_{u \in \mathcal{U}} \mathrm{E}[g(X) \ln f(X,u)]\end{aligned}$$

where in the second step we used the fact is $\mathcal{D}$ is non-negative and that the first integral does not depend on u. It is worth to observe that in the maximization problem the dependency on f_* has disappeared, thus simplifying it. In fact, Rubinstein and Kroese [12] show that for certain families of densities the maximization problem can be solved *analytically*. Assume now that $\mathbf{X}$ is a random vector, *i.e.*, $\mathbf{X}{:}\Omega \to \mathbb{R}^n$ (of course g must be defined over $\mathbb{R}^n$). Note that this does not change what we obtained so far. The following Proposition gives the optimal parameter $\mathbf{u}^* \stackrel{\mathrm{def}}{=} \operatorname{argmax}_{u \in \mathcal{U}} \mathrm{E}[g(\mathbf{X}) \ln f(\mathbf{X}, u)]$ when $\mathbf{X}$ is a vector of independent, one-dimensional exponential family of distributions.

PROPOSITION 2. *[12] Let $\mathbf{X}$ be a random vector of n independent one-dimensional exponential distributions parametrized by the mean; $g{:}\mathbb{R}^n \to \mathbb{R}^{\geqslant 0}$ be a measurable function. Then the optimal parameter $\mathbf{u}^* = (u_1^*, \ldots, u_n^*)$ is*

$$u_j^* = \frac{\mathrm{E}[g(\mathbf{X})X_j]}{\mathrm{E}[g(\mathbf{X})]}$$

where X_j is the j-th component of $\mathbf{X}$.

From the Proposition, we see that the optimal parameter depends on the quantity we are estimating, *i.e.*, $\mathrm{E}[g(\mathbf{X})]$. Therefore, $\mathbf{u}^*$ needs itself to be estimated by Monte Carlo simulation. More specifically, the j-th component of $\mathbf{u}^*$ may be estimated from iid random variables $\mathbf{X}_1, \ldots, \mathbf{X}_N$ (as in the Proposition above) by

$$\hat{u}_j^* = \frac{\sum_{i=1}^N g(\mathbf{X}_i) X_{ij}}{\sum_{i=1}^N g(\mathbf{X}_i)} \tag{5}$$

where X_{ij} is the j-th component of $\mathbf{X}_i$. However, recall that in statistical model checking $g(\mathbf{X}_i)$ is either 1 or 0 — a sample trace either satisfies a given temporal logic property or it

does not. Also, in the rare-event case it will be very unlikely to sample traces that satisfy the temporal logic property. This means that for reasonable sample sizes the estimator in (5) would most likely be $\frac{0}{0}$, thereby of little use.

This problem can be solved by noting that, for an arbitrary *tilting* parameter $w \in \mathcal{U}$, the following holds

$$u_j^* = \frac{\mathrm{E}[g(\mathbf{X})X_j]}{\mathrm{E}[g(\mathbf{X})]} = \frac{\mathrm{E}_w[g(\mathbf{X})W(\mathbf{X},w)X_j]}{\mathrm{E}_w[g(\mathbf{X})W(\mathbf{X},w)]}$$

where $W(x,w) = f(x)/f(x,w)$ and $f(x) = f(x,v)$ is the nominal density of $\mathbf{X}$. It is important to note that the expectation is computed with respect to the proposal density $f(\cdot,w)$. Now, we can use Monte Carlo simulation again to estimate u_j^* by

$$\hat{u}_j^* = \frac{\sum_{i=1}^N g(\mathbf{X}_i)W(\mathbf{X}_i,w)X_{ij}}{\sum_{i=1}^N g(\mathbf{X}_i)W(\mathbf{X}_i,w)} \tag{6}$$

where the $\mathbf{X}_i$'s are sampled from $f(\cdot,w)$. In other terms, we use importance sampling with a proposal density given by the tilting parameter w. Naturally, w must be chosen in such a way to avoid the $\frac{0}{0}$ problem of the estimator (6). Therefore, w should increase the probability of the event $g(\mathbf{X}) = 1$, *i.e.*, we should sample more often traces satisfying the given temporal property. However, it is not required to "guess" a tilting parameter close to the optimal one: it is often sufficient that the chosen w increases the probability of the rare event in the range 0.01-0.1. This enables meaningful estimates of the optimal $\mathbf{u}^*$ using (6).

We point it out that the CE method does not guarantee that the computed proposal density minimizes the variance (3). This is because in general the optimal proposal density may not belong to the parametric family. However, the CE method has been shown to work very well in many applications [12]. As we show in the next Section, the CE method works well with statistical model checking, too.

Finally, Rubinstein [11] has presented a multi-level CE algorithm for estimating the probability of rare events of the form $\{g(X) \geqslant \gamma\}$, where g is a real function and γ a constant. The algorithm first gets N samples $X_1, ...X_N$ of the system under the nominal (unbiased) distribution. Then it computes the sample quantile of the $g(X_i)$'s, and it adaptively tunes γ to make the event of interest more frequent. Besides the more restricted class of events considered, this algorithm does not work in our case. In statistical model checking the function g is in fact the model checker that checks whether a simulation trace satisfies the given temporal logic formula. Function g thus returns either 0 or 1. When computing the sample quantile one has to order the values of the $g(X_i)$'s. But these would most likely all be 0, since the system is sampled with the original distribution, under which the event $\{g(X) \geqslant \gamma\}$ is rare. Therefore, this technique is not directly applicable in our case.

6. EXPERIMENTS

We have applied the cross-entropy method to an example of stochastic hybrid system modeled in Stateflow/Simulink. The model implements a fault-tolerant controller for an aircraft elevator system[1]. The model is part of a larger Simulink modeling of the HL-20 crew rescue vehicle developed by NASA [3]. Typically, the two horizontal tails on the sides of an aircraft fuselage are each governed by one elevator, and there are two independent hydraulic actuators per elevator — four in total. During normal operation, each elevator is positioned by its corresponding outer actuator, and an inner actuator can be used in case of malfunctioning. The two outer actuators are driven by two separate hydraulic circuits, while the two inner actuators are both connected to a third hydraulic circuit. The outer actuators operates during normal use, and in case of failure the inner actuators can be operated. The system should ensure that at any given time only one set of actuators (*i.e.*, either outer or inner) position the elevators. If a fault arises in the outer actuators or in their corresponding hydraulic circuits, the system will activate the inner actuators; the outer actuators will be switched off and eventually isolated if the fault persists. Failures in the hydraulic circuits may be temporary, and a failed circuit can always placed back online if the fault condition terminates. The control logic of the system is implemented as a Stateflow diagram, while the hydraulic actuators and the elevators are modeled using Simulink. More details about the model can be found in [8].

We have modified the Stateflow/Simulink model by adding random failures in the three hydraulic circuits only. A failure is modeled as an out-of-bounds reading of the circuit pressure. We model failure injection as three independent Poisson processes. When a failure in a hydraulic circuit occurs, the circuit will stay in faulty condition for one second, after which the pressure reading returns to its normal value, and the fail condition terminates. The nominal fault rates for the three circuits were all set to 1/3600. In our experiments we estimated the probability of BLTL formula ϕ:

$$\phi = \mathbf{F}^{25}\mathbf{G}^{1}((\mathrm{H1}_{fail} \vee \mathrm{H3}_{fail}) \wedge \mathrm{H2}_{fail})$$

where H1 and H3 denote the hydraulic circuits driving the outer actuators, and H2 denotes the circuit driving the inner actuators. Informally, we want to estimate the probability that, within 25 seconds, the horizontal tails do not respond to the control inputs for a duration of one second. Since the fault rates of the three hydraulic circuits are low (1/3600), we expect ϕ to be a rare event.

We have performed three experiments, depending on the number of samples used to compute the optimal CE rates and in importance sampling (the higher numbers are always used for importance sampling). The proposal density is implemented by changing the fault rates of the three Poisson processes modeling the fault injection. The initial fault rates (tilting rates) for the computation of the optimal bias were 1/10, except for the first experiment (100/1,000 samples) where we used 1/8. This because during the computation of the CE rates the proportion of satisfying traces was too low. All our experiments have been performed on a 3.2GHz Intel Xeon computer running Matlab R2010a.

In Table 1 we report the estimate for the probability that ϕ holds, the (approximate) relative error, the total computation time (*i.e.*, simulation, model checking, and cross-entropy calculations), and the (approximately) optimal rates computed by the CE method. These were the actual rates used in importance sampling. The relative error is computed as the ratio between the estimated standard deviation of the estimate and the estimate itself. (We recall that the standard deviation can be estimated by the square root

[1]More information about the model is available at `http://mathworks.com/products/stateflow/demos.html?file=/products/demos/shipping/stateflow/sf_aircraft.html`

Samples	Estimate	RE	Time	Rates
100 1,000	1.58×10^{-14}	0.58	0.23	1/1.00 1/2.00 1/1.00
1,000 10,000	8.54×10^{-14}	0.24	2.45	1/0.98 1/2.01 1/1.02
10,000 100,000	8.11×10^{-14}	0.17	23.9	1/0.52 1/2.01 1/1.48

Table 1: Cross-Entropy and Importance Sampling. Samples used for CE rates computation and importance sampling; probability estimate; relative error; total computation time (hours); computed cross-entropy rates.

of the sample variance $\frac{1}{N-1}\sum_{i=1}^{N}(g(X_i)W(X_i)-\hat{c})^2$, where $X_1,\ldots,X_N$ are iid as the proposal distribution, and $\hat{c}$ is the probability estimated by importance sampling on the same sample $X_1,\ldots,X_N$.)

The table shows that statistical model checking with importance sampling and cross-entropy can efficiently estimate rare-event probabilities. In particular, with a feasible sample size of 10^4 it is possible to estimate probabilities in the order of 10^{-14} with reasonable accuracy (RE = 0.24). Clearly, standard statistical model checking and crude Monte Carlo would need an unfeasible number of samples to provide similar levels of accuracy. Also, we see that by increasing the sample size the relative error decreases — as one would expect — thereby yielding more accurate estimates.

Finally, since each sample can be generated independently from the others, Monte Carlo methods can readily take advantage of parallel/multi-core systems. That further contributes in making statistical model checking an effective technique.

7. CONCLUSIONS

In this paper we have addressed the verification of rare events for stochastic hybrid systems via statistical model checking. We have proposed a semantic model for stochastic hybrid systems that is tailored for simulation environments. We have shown that the model induces a Markov process and a well-defined probability measure, and it is thus usable with statistical model checking. Previous works have shown that statistical model checking can efficiently verify large, "difficult" systems. However, this verification technique suffers from the rare-event problem: if the property to verify is true with an extremely small probability, then statistical model checking becomes inefficient. In particular, a large number of simulations is required to obtain an accurate estimate of the probability. The problem can be tackled by combining importance sampling and the cross-entropy method with statistical model checking, as we have proposed. Our initial findings indicate that this combination can efficiently address the verification of rare events for stochastic hybrid systems.

8. ACKNOWLEDGMENTS

This research was sponsored by the GSRC under contract no. 1041377 (Princeton University), the National Science Foundation under contracts no. CNS0926181 and no. CNS0931985, the Semiconductor Research Corporation under contract no. 2005TJ1366, General Motors under contract no. GMCMUCRLNV301, the Office of Naval Research under award no. N000141010188, the DFG-project QuaOS, and the Collaborative Research Center HAEC (SFB 912) funded by the DFG.

9. REFERENCES

[1] C. Baier, B. R. Haverkort, H. Hermanns, and J.-P. Katoen. Model-checking algorithms for continuous-time Markov chains. *IEEE Trans. Software Eng.*, 29(6):524–541, 2003.

[2] H. A. P. Blom and J. Lygeros, editors. *Stochastic Hybrid Systems*, volume 337 of *Lecture Notes in Control and Information Sciences*. Springer, 2006.

[3] S. Gage. NASA HL-20 lifting body airframe modeled with Simulink and the aerospace blockset. *MATLAB Digest*, 10(4), 2002.

[4] R. Grosu and S. A. Smolka. Monte Carlo Model Checking. In *TACAS*, volume 3440 of *LNCS*, pages 271–286, 2005.

[5] T. Hérault, R. Lassaigne, F. Magniette, and S. Peyronnet. Approximate probabilistic model checking. In *VMCAI*, volume 2937 of *LNCS*, pages 73–84, 2004.

[6] S. K. Jha, E. M. Clarke, C. J. Langmead, A. Legay, A. Platzer, and P. Zuliani. A Bayesian approach to Model Checking biological systems. In *CMSB*, volume 5688 of *LNCS*, pages 218–234, 2009.

[7] O. Maler and D. Nickovic. Monitoring temporal properties of continuous signals. In *FORMATS*, volume 3253 of *LNCS*, pages 152–166, 2004.

[8] P. J. Mosterman and J. Ghidella. Model reuse for the training of fault scenarios in aerospace. In *Proceedings of the AIAA Modeling and Simulation Technologies Conference*, 2004.

[9] A. Pnueli. The temporal logic of programs. In *FOCS*, pages 46–57. IEEE, 1977.

[10] G. Rubino and B. Tuffin, editors. *Rare Event Simulation using Monte Carlo Methods*. Wiley, 2009.

[11] R. Y. Rubinstein. The cross-entropy method for combinatorial and continuous optimization. *Methodology and Computing in Applied Probability*, 1:127–190, 1999.

[12] R. Y. Rubinstein and D. P. Kroese. *The Cross-Entropy Method*. Springer, 2004.

[13] K. Sen, M. Viswanathan, and G. Agha. Statistical model checking of black-box probabilistic systems. In *CAV*, volume 3114 of *LNCS*, pages 202–215, 2004.

[14] A. N. Shiryaev. *Probability*. Springer, 1995.

[15] R. Srinivasan. *Importance Sampling*. Springer, 2002.

[16] H. L. S. Younes, E. M. Clarke, and P. Zuliani. Statistical verification of probabilistic properties with unbounded until. In *SBMF*, volume 6527 of *LNCS*, pages 144–160, 2010.

[17] H. L. S. Younes, M. Z. Kwiatkowska, G. Norman, and D. Parker. Numerical vs. statistical probabilistic model checking. *STTT*, 8(3):216–228, 2006.

[18] H. L. S. Younes and D. J. Musliner. Probabilistic plan verification through acceptance sampling. In *AIPS*

Workshop on Planning via Model Checking, pages 81–88, 2002.

[19] H. L. S. Younes and R. G. Simmons. Probabilistic verification of discrete event systems using acceptance sampling. In *CAV*, volume 2404 of *LNCS*, pages 223–235, 2002.

[20] H. L. S. Younes and R. G. Simmons. Statistical probabilistic model checking with a focus on time-bounded properties. *Inf. Comput.*, 204(9):1368–1409, 2006.

[21] P. Zuliani, A. Platzer, and E. M. Clarke. Bayesian statistical model checking with application to Stateflow/Simulink verification. In *HSCC*, pages 243–252, 2010.

APPENDIX

For completeness, we report some standard results about importance sampling [15] and the cross-entropy [12].

A. IMPORTANCE SAMPLING

We calculate the variance of the IS estimator of Definition 8. In the following, Var_* denotes variance taken with respect to the biasing density f_*. The variance of the IS estimator is

$$\begin{aligned}\mathrm{Var}(\hat{c}_{\mathrm{IS}}) &= \mathrm{Var}_*\left(\frac{1}{N}\sum_{i=1}^{N} g(X_i)W(X_i)\right)\\ &= \frac{1}{N^2}\sum_{i=1}^{N}\mathrm{Var}_*(g(X_i)W(X_i))\\ &= \frac{1}{N}\mathrm{Var}_*(g(X)W(X))\\ &= \frac{1}{N}(\mathrm{E}_*[g^2(X)W^2(X)] - \mathrm{E}_*^2[g(X)W(X)])\\ &= \frac{1}{N}(\mathrm{E}_*[g^2(X)W^2(X)] - c^2) \qquad (7)\end{aligned}$$

where by (2) it is $c = \mathrm{E}_*[g(X)W(X)] = \mathrm{E}[g(X)]$. Also, the variance can be expressed in a slightly different form. Continuing from (7) we have

$$\begin{aligned}\mathrm{Var}(\hat{c}_{\mathrm{IS}}) &= \frac{1}{N}\left(\int_{\mathbb{R}} g^2(x)\frac{f^2(x)}{f_*^2(x)}f_*(x)\,dx - c^2\right)\\ &= \frac{1}{N}\left(\int_{\mathbb{R}} g^2(x)\frac{f(x)}{f_*(x)}f(x)\,dx - c^2\right)\\ &= \frac{1}{N}(\mathrm{E}[g^2(X)W(X)] - c^2)\end{aligned}$$

We now calculate the optimal biasing density. Since the variance is always non-negative, we need to minimize the expectation term in (7). By Jensen's inequality we get

$$\begin{aligned}E_*[g^2(X)W^2(X)] &\geqslant E_*^2[|g(X)|W(X)] \qquad (8)\\ &= E_*^2[|g(X)|\frac{f(X)}{f_*(X)}]\\ &= \left(\int |g(x)|\frac{f(x)}{f_*(x)}f_*(x)\,dx\right)^2\\ &= E^2[|g(X)|] \qquad (9)\end{aligned}$$

and, since the square function is strictly convex, equality holds in (8) iff the random variable $|g(X)|W(X)$ is constant, *i.e.*,

$$|g(X)|W(X) = k$$

for some constant k and $X \sim f_*$ (because in (8) the expectation is computed with respect to f_*). But $W(x) = \frac{f(x)}{f_*(x)}$, so we deduce that

$$f_*(x) = \frac{1}{k}|g(x)|f(x) \qquad (10)$$

is the *optimal* biasing density, *i.e.*, the density which minimizes the variance (7) by attaining the lower bound in (9). It remains to calculate k: when $|g(X)|W(X) = k$, we have immediately from (9) that $k = \mathrm{E}[|g(X)|]$. Therefore, from (7) the variance of the IS estimator is

$$\mathrm{Var}_*(\hat{c}_{\mathrm{IS}}) = \frac{1}{N}(k^2 - c^2)$$

which in general may be non-zero, but will of course tend to zero as $N \to \infty$. Note that when g is non-negative, then $k = \mathrm{E}[|g(X)|] = \mathrm{E}[g(X)] = c$ and therefore $\mathrm{Var}_*(\hat{c}_{\mathrm{IS}}) = 0$.

B. CROSS-ENTROPY

We show that the cross-entropy (or Kullback-Leibler divergence) of two densities f ang g is always non-negative. Recall its definition

$$\mathcal{D}(f,g) = \int_{\mathbb{R}} f(x)\ln\frac{f(x)}{g(x)}\,dx = \mathrm{E}\left[\ln\frac{f(X)}{g(X)}\right]$$

where X is a random variable with density f. The proof is a simple application of Jensen's inequality (note that $-\ln$ is a convex function):

$$\begin{aligned}\mathrm{E}\left[\ln\frac{f(X)}{g(X)}\right] = \mathrm{E}\left[-\ln\frac{g(X)}{f(X)}\right] \geqslant\\ -\ln\mathrm{E}\left[\frac{g(X)}{f(X)}\right] = -\ln\int_{\mathbb{R}} g(x)\,dx = 0\end{aligned}$$

where the last equality holds because g is a probability density. Also, it follows that $\mathcal{D}(f,g) = 0$ iff $f = g$.