When to Stop Verification?
Statistical Trade-off between Expected Loss and Simulation Cost

Sumit Kumar Jha
Computer Science Department
University of Central Florida
Orlando, Florida
Email: jha@eecs.ucf.edu

Christopher James Langmead
Computer Science Department and
Lane Center for Computational Biology
Carnegie Mellon University
Email: cjl@cs.cmu.edu

Swarup Mohalik Sethu Ramesh
India Science Lab
General Motors Research and Development
Bangalore, India
Email: swarup.mohalik, ramesh.s@gm.com

Abstract—Exhaustive state space exploration based verification of embedded system designs remains a challenge despite three decades of active research into Model Checking. On the other hand, simulation based verification of even critical embedded system designs is often subject to financial budget considerations in practice. In this paper, we suggest an algorithm that minimizes the overall cost of producing an embedded system including the cost of testing the embedded system and expected losses from an incompletely tested design. We seek to quantify the trade-off between the budget for testing and the potential financial loss from an incorrect design. We demonstrate that our algorithm needs only a logarithmic number of test samples in the cost of the potential loss from an incorrect validation result. We also show that our approach remains sound when only upper bounds on the potential loss and lower bounds on the cost of simulation are available. We present experimental evidence to corroborate our theoretical results.

I. INTRODUCTION

The validation of stochastic embedded system models is an important challenge for the modern transportation and biomedical devices industry. Embedded system designs often interact with models of an incompletely predictable world and such cyber-physical interactions are often modeled using stochastic models. State space exploration based verification of such stochastic embedded system models has not scaled enough so as to be suitable for verifying large embedded system models despite three decades of active research into Model Checking.

On the other hand, simulation based verification suffers from the fundamental problem that 100% model coverage is difficult to achieve. Very often, a system is tested to the extent permitted by the testing budget and the time available to test the model before production starts. The testing budget (and time) are often derived heuristically and not from first principles. We suggest that the amount of simulation based testing that a system should be exposed to should depend on the expected losses from a possible malfunctioning of the system [7]. In this paper, we seek to present a strategy that minimizes the overall cost of verifying a system including (i) the cost of running verification tests, and (ii) the expected loss from a potential bug in the system that is not discovered using simulation. A logical stopping point for testing these models is the scenario where the expected loss from a possible failure of the embedded system is lower than the cost of generating and running another test on the embedded system model. In such a setting, it makes economic sense to stop testing a stochastic embedded system model when the cost of running the next test exceeds the expected loss from an uncaught bug in the model.

II. BACKGROUND

We survey the required background in stochastic model validation, Bayesian statistics and probabilistic temporal logics that will be needed to understand our algorithms and proofs in Section III.

A. Validation of Stochastic Systems

Model validation refers to the task of algorithmically deciding whether a given model $M$ with the initial state $s$ meets a given behavior $\phi$, denoted $M, s \models \phi$. For non-stochastic systems, exhaustive model checking algorithms are often able to solve the validation problem. For example, a combination of model checking and static analysis techniques [2] have been used to verify Windows device drivers. Unfortunately, these same algorithms cannot be applied to stochastic embedded system models without substantial degradation in performance.

Because of the stochastic nature of the model, it becomes necessary to compute (or bound) the probability that a given behavior exists. The key problem addressed in this paper is validating properties of stochastic embedded system models.

Definition 1 (Probabilistic model validation problem). Given a stochastic model $M$, an initial state $s$, a specification $\phi$, and a probability bound $p$, algorithmically decide whether the stochastic model $M$ satisfies the specification $\phi$ with probability at least $p$, i.e. $M, s \models P_{\geq p}(\phi)$.

There are two basic strategies that have been employed to solve the probabilistic model validation problem. The first approach treats the problem as a probabilistic state space exploration problem [13]. Such methods compute a numerical estimate of the probability that the system satisfies $\phi$, and then compare the numerically computed value to $p$. The second approach, which we use in this paper, uses a simulation-based approach to compute a statistical estimate of the probability that the system satisfies $\phi$. Here, a finite number of sample trajectories are drawn from the model. Each sample trajectory is evaluated to determine whether it satisfies $\phi$, and the number of

978-3-9810801-7-9/DATE11©2011 EDAA
satisfying and non-satisfying traces is used to determine whether $M, s \models P_{\geq \rho}(\phi)$.

Performing multiple stochastic simulations of the model can be computationally expensive. Thus, it is important to minimize the number of samples or test cases needed to validate the model. The validation algorithm presented in this paper minimizes the overall cost of simulation based model validation given the loss from a possible failure of the embedded system. Our methods build on existing work [16], [15], [9], and combine techniques from the fields of model checking and Bayesian statistics [4].

### B. Probabilistic Adapted Finitely Monitorable Specifications

We formally define the nature of high-level behavioral specifications that we use to express the expected behavior of stochastic embedded system models. A specification is said to be adapted to a stochastic process if the truth of the specification can be determined on a trajectory sampled from the stochastic process. Naturally, we are interested in specifications whose truth value can be decided by observing only a finite prefix of the simulation of the stochastic process. We call the logical formulae that represent such properties adapted finitely monitorable (AFM) specifications.

Due to the stochastic nature of our models, we generally seek to verify that a desired behavior holds with a threshold probability. Given an AFM specification $\phi$, the property $Pr_{\geq \rho}(\phi)$ is true for a stochastic system if and only if the property $\phi$ is true for a random simulation of the stochastic system with probability at least $\rho$. We call such probabilistic properties probabilistic adapted finitely monitorable (PAFM) specifications. We survey an important kind of PAFM specifications called Probabilistic Bounded Metric Temporal Logic (PBMTL).

**Probabilistic Bounded Metric Temporal Logic:** A widely used logic for defining behavioral expectations from simulations of continuous (and possibly stochastic) systems is the Metric Temporal Logic [12]. Monitoring algorithms [6] can automatically generate “run-time monitors” from suitable logical specifications on simulations such that the monitor accepts a simulation of the system if and only if the simulation satisfies the logical specification. There exist efficient monitoring algorithms for Metric Temporal Logic. The logic extended with a probability operator naturally defines a subset of PAFM specifications.

Metric Temporal Logic (MTL) can specify both lower and upper bounds on the time bounds associated with the temporal operators. The syntax of the MTL property is given by the following grammar:

$$\phi := x \sim v \mid (\phi_1 \lor \phi_2) \mid (\phi_1 \land \phi_2) \mid \neg \phi_1 \mid (\phi_1 U[t\sim t']\phi_2),$$

where $\sim \in \{\geq, \leq, =\}$, $x$ is a model variable, $v \in \mathbb{R}$, and $t \in \mathbb{Q}_{\geq 0}$. The semantics of Bounded MTL for a trace $\sigma^k$ starting at the $k^{th}$ state ($k \in \mathbb{N}$) is defined as follows:

- $\sigma^k \models x \sim v$ if and only if $V(\sigma, k, x) \sim v$;
- $\sigma^k \models \phi_1 \lor \phi_2$ if and only if $\sigma^k \models \phi_1$ or $\sigma^k \models \phi_2$;
- $\sigma^k \models \phi_1 \land \phi_2$ if and only if $\sigma^k \models \phi_1$ and $\sigma^k \models \phi_2$;
- $\sigma^k \models \neg \phi_1$ if and only if $\sigma^k \models \phi_1$ does not hold (written $\sigma^k \not\models \phi_1$);
- $\sigma^k \models \phi_1 U[t\sim t']\phi_2$ if and only if there exists $i \in \mathbb{N}$ such that (a) $t \leq \sum_{0 \leq i < 1} \Delta_k \leq t'$; (b) $\sigma^{k+i} \models \phi_2$; and (c) for each $0 \leq j < i$, $\sigma^{k+j} \models \phi_1$.

**Definition 2.** A Probabilistic Bounded Metric Temporal Logic formula is a formula of the form $P_{\geq \rho}(\phi)$, where $\phi$ is a Bounded Metric Temporal Logic formula and $\rho \in [0, 1]$.

We say that $M$ satisfies Probabilistic Bounded Metric Temporal Logic property $P_{\geq \rho}(\phi)$, denoted by $M \models P_{\geq \rho}(\phi)$, if and only if the probability that a randomly sampled execution of $M$ satisfies Bounded Metric Temporal Logic property $\phi$ is greater than or equal to $\rho$.

### C. Bayesian Statistics

We recall the notions of KL divergence, affinity, $\delta$-separation and then an important result on the concentration of Bayesian posteriors [4], [3]. We will use these results to theoretically characterize the performance of our proposed algorithm in Section III.

**Definition 3 (Kullback-Leibler (KL) Divergence).** Given a parameterized family of probability distributions $\{f_\rho\}$, the Kullback-Leibler (KL) divergence $K(\rho_0, \rho)$ between the parameter distributions corresponding to the two parameters $\rho$ and $\rho_0$ is $E_{\rho_0}\left[\log\left(\frac{f_\rho}{f_{\rho_0}}\right)\right]$. Note that $E_{\rho_0}$ is the expectation computed under the probability measure $f_{\rho_0}$.

The KL divergence between two distributions is intuitively a measure of the difference between two probability distributions.

**Definition 4 (Kullback-Leibler (KL) Neighborhood).** Given a parameterized family of probability distributions $\{f_\rho\}$, the KL neighborhood $K_i(\rho_0)$ of a distribution corresponding to the parameter value $\rho_0$ will be given by the parameter values $\{\rho : K(\rho_0, \rho) < \epsilon\}$.

Given a parameterized family of distributions, the KL neighborhood of a distribution is the set of parameter values corresponding to distributions that are similar to this distribution under the KL divergence notion of similarity.

**Definition 5 (Kullback-Leibler (KL) Support).** A parameter value $\rho_0$ is said to be in the KL support of a prior $\Pi$ if and only if for all $\epsilon > 0$, $\Pi(K_i(\rho_0)) > 0$.

Given an arbitrary prior probability distribution $\Pi$, a parameter value $\rho_0$ is in the Kullback-Leibler (KL) support of the prior $\Pi$ if and only if every positive KL neighborhood of $\rho_0$ has a non-zero prior probability.

**Definition 6 (Affinity).** Let $f$ and $g$ be two probability distributions on a probability space $\mathbb{R}$ with probability measure $\mu$. The affinity $\text{Aff}(f, g)$ between the two densities $f$ and $g$ is defined as the Lebesgue integral $\int_{\mathbb{R}} \sqrt{f g} \, d\mu$.

The affinity of two probability distributions is another measure of the similarity between those distributions. In particular, the
affinity between two distributions is 0 if and only if they are identical. If both the distributions never together assign a non-zero probability to the same sample event, then the affinity between these two distributions is zero.

**Definition 7** (Marginal Density). For a probability measure \( v \) on \( \rho \) where \( \rho \in \varrho \), \( q^{(\rho)}_v \) is the marginal density of \( X_1, \ldots, X_n \), where \( X_i \) (1 \( \leq i \leq n \)) are sampled i.i.d. from the distribution \( f_\rho \).

\[
q^{(\rho)}_v(x_1, x_2, \ldots, x_n) = \int f_\rho(x_1) \ldots f_\rho(x_n) v(d\rho)
\]

**Definition 8** (Strong \( \delta \)-Separation). Let \( A \subset [0, 1] \) and \( \delta > 0 \). The set \( A \) and the point \( \rho_0 \) are said to be strongly \( \delta \)-separated if and only if for any probability measure \( v \) on \( A \),

\[
\text{Aff} (f_{\rho_0}, q^{(\delta)}_v) < \delta
\]

Note that \( q^{(\delta)}_v \) is the marginal density of one sample under the distribution \( f_\rho \).

**Theorem II.1.** If \( \rho_0 \) and \( A = [a_0, a_1] \) are strongly \( \delta \)-separated, and \( b_0 = -\log \delta \), then, for all probability \( v \) on \( A \), for all \( n \),

\[
\text{Aff} \left( \int f(x_i|\rho_0) \ldots f(x_n|\rho_0), \int f(x_i|u) \ldots f(x_n|u) g(u) du \right) < e^{-nb_0}
\]

**Proof:** Proof in [11].

**Theorem II.2** (Bayesian Consistency Theorem [3]). If \( x_i \) are i.i.d. samples of the Bernoulli random variable \( X_i \) (1 \( \leq i \leq n \)) with probability of success \( \rho_0 \) such that \( \rho_0 \) lies in the KL support of the prior \( g \), \( A = [a_0, a_1] \) is strongly \( \delta \)-separated from \( \rho_0 \) (for some \( \delta > 0 \)), and the prior probability measure on \( A \) is finite, then the posterior probability of \( A \) decreases exponentially to 0 almost everywhere.

\[
P \left( \frac{\int f(x_1|u) \ldots f(x_n|u) \cdot g(u) du}{\int f(x_1|u) \ldots f(x_n|u) \cdot g(u) du} > e^{-nb} \text{ i.o.} \right) = 0
\]

Here, \( b \) is a constant and the abbreviation i.o. stands for infinitely often.

**Proof:** For the details of the proof, please see [3], [4] and also [11].

**III. Algorithm: Cost Based Bayesian Statistical Verification**

Most of the existing work in Statistical Verification is equipped only with Frequentist guarantees of Type I and II errors. Optimal values of the Type I and Type II errors are not easily apparent to an end-user, who would actually really want these errors to be zero. In this paper, we take a Bayesian approach to statistical verification. One of the key advantages of a Bayesian approach is that it can easily incorporate readily available decision criteria, such as the expected monetary costs for generating samples and for returning an incorrect answer. In this paper, we develop a cost-based algorithm for validation of stochastic embedded system models. That is, one where we know (lower bound on) the cost of drawing samples from the model and (upper bound on) the cost of making an incorrect decision [7]. Our algorithm will stop when the cost of making an additional observation is larger than the expected loss from making a wrong decision.

Consider the validation query that a newly designed electronic braking system is correct with 99.9999% probability. Each detailed simulation of such a system may take a few minutes, and is associated with a non-trivial computational cost. Each day of simulation time of a single CPU node costs about $20 on a commercial cluster. The cost of making an incorrect decision is substantial. If an erratic model is released into the market, the cost of a recall is prohibitive. In 2010, the cost of an automotive recall in the US was estimated to be over $2 billion. The goal of the cost based statistical verification approach is to use the information about the cost of sampling and the cost of making an incorrect decision to decide the number of test samples that ought to be observed.

We now present our Bayesian approach for deciding the model validation problem when the cost of generating each simulation and the cost of making a wrong decision is known. Recall that for any finite trace \( \sigma_1 \) from a stochastic model and an adapted finitely monitorable formula \( \phi \), we can deterministically decide whether \( \sigma_1 \) satisfies \( \phi \). Therefore, we can define a random variable \( X_i \) denoting the outcome of \( \sigma_i \models \phi \). Then, \( X_i \) will be a Bernoulli random variable with probability mass function

\[
f(x_i|u) = u^{x_i}(1 - u)^{1-x_i}
\]

where \( x_i = 1 \) if and only if \( \sigma_i \models \phi \), otherwise \( x_i = 0 \). Note that the random variables \( X_i \) (1 \( \leq i \leq n \)) are i.i.d. Since the true probability with which the system \( M \) satisfies \( \phi \) is unknown, we can model it as a random variable \( U \), with a density \( g(u) \) called the prior density. The prior probability distribution is usually based on our previous experiences and beliefs about the system. Even a complete lack of information about the probability of the system satisfying the AFM formula can be summarized formally by a non-informative or objective prior probability distribution [4].

Suppose we have a sequence of random variables \( X_1, \ldots, X_n \) defined as above, and let \( d = (x_1, \ldots, x_n) \) denote a sample of those variables. When \( n \) independent simulation traces of the system have been analyzed against an AFM formula \( \phi \) and exactly \( x \) of these traces indeed satisfy the specification, then the posterior distribution of the probability with which \( M \) satisfies the AFM formula \( \phi \) is given by the product of the prior probability and the probability of the observed samples. After having made \( n \) observations, the probability that the system \( M \) satisfies the AFM formula \( \phi \) with probability at least \( \rho \) is

\[
\frac{\int f(x_1|u) \ldots f(x_n|u) \cdot g(u) du}{\int f(x_1|u) \ldots f(x_n|u) \cdot g(u) du} = C_0
\]

If \( C_0 \) is the possible cost of the hazard associated with rejecting the property \( Pr_{\geq \rho}(\phi) \) when the property is actually true, the expected cost \( \text{Cost} (N = n) \) associated with making a wrong decision after \( N \) samples is then given by
Require: Model $\mathcal{M}$, PAFM Formula $Pr_{\geq \rho}(\phi)$, Cost of each simulation $s$, the costs $C_0$ and $C_1$ of the possible hazard associated with the failure modes of the property, Indifference Region $[\rho - \epsilon_1, \rho + \epsilon_2]$, Prior density $g$ for unknown parameter $\theta$

$n \leftarrow 0$; /* Total Number of Samples observed so far */
i \leftarrow 0; /* Samples satisfying the AFM formula */

$ExpectedLoss = \infty$

while $ExpectedLoss > s$ do
    $n \leftarrow n + 1$;
    Observe an i.i.d. sample simulation $\sigma_n$;
    if $\sigma_n \models \phi$ then
        $x_n = 1$, $i \leftarrow i + 1$; /* simulation satisfies the AFM formula */
    else
        $x_n = 0$; /* simulation does not satisfy the AFM formula */
    end if
    $ExpectedLoss = \min \left( \int_0^{\rho - \epsilon_1} f(x_1|u) \cdot f(x_n|u) \cdot g(u) \, du + C_0, \int_{\rho + \epsilon_2}^1 f(x_1|u) \cdot f(x_n|u) \cdot g(u) \, du + C_1 \right)$
end while

if $\left( \int_0^{\rho - \epsilon_1} f(x_1|u) \cdot f(x_n|u) \cdot g(u) \, du < \int_{\rho + \epsilon_2}^1 f(x_1|u) \cdot f(x_n|u) \cdot g(u) \, du \right)$ then
    print The formula $Pr_{\geq \rho}(\phi)$ holds on $\mathcal{M}$.
else
    print The formula $Pr_{\geq \rho}(\phi)$ does not hold on $\mathcal{M}$.
end if

Fig. 1. Cost based Bayesian Model Checking algorithm

Cost($N = n$) = $\int_0^\rho f(x_1|u) \cdot f(x_n|u) \cdot g(u) \, du$ - $C_0$ $\quad$ (1)

If $C_1$ is the possible cost of the hazard associated with accepting the property $Pr_{\geq \rho}(\phi)$ when the property is actually false, the expected cost Cost($N = n$) associated with making a wrong decision after $N$ samples is then given by

Cost($N = n$) = $\int_0^\rho f(x_1|u) \cdot f(x_n|u) \cdot g(u) \, du$ - $C_1$ $\quad$ (2)

The Cost based Bayesian Model Checking algorithm is illustrated in Figure 1. The algorithm takes six inputs:

(i) the stochastic embedded system model $\mathcal{M}$ under investigation,
(ii) the PAFM Formula $Pr_{\geq \rho}(\phi)$,
(iii) the cost of generating each simulation sample $s$,
(iv) the costs $C_0$ and $C_1$ of the possible loss [7] associated with the failure modes of the property,
(v) the indifference region $[\rho - \epsilon_1, \rho + \epsilon_2]$ such that $0 \leq \rho - \epsilon_1 < \rho < \rho + \epsilon_2 \leq 1$, such that we are indifferent to the answer of the Statistical Verification algorithm if the true probability of satisfying the formula $\phi$ lies within the tolerance region, and
(vi) the prior probability distribution $g$ for the probability of the model $\mathcal{M}$ satisfying the formula $\phi$.

The algorithm then performs i.i.d simulations of the model under investigation and records the total number of simulation so far performed ($n$) and the number of simulations that actually satisfy the AFM formula ($x$). After observing each sample, the algorithm verifies whether the cost of observing another additional sample exceeds the reduction in the expected loss from an incorrect decision by the Model Checking algorithm (irrespective of the outcome of this additional sample). If an additional sample is less expensive to generate than the associated reduction in the expected loss from an incorrect decision by the Model Checking algorithm, the algorithm loops back and generates a new sample. Otherwise, the algorithm stops and generates the answer to the model checking query of the system $\mathcal{M}$ satisfying the PAFM formula $Pr_{\geq \rho}(\phi)$.

A. Theoretical Results

We now show that our Cost based Bayesian Statistical Verification algorithm is guaranteed to terminate almost surely. Then, we investigate the number of samples needed by our algorithm and the influence of the variation of the cost of producing test samples and the possible loss from an incorrect decision.

Theorem III.1. The Cost based Bayesian Statistical Model Verification algorithm terminates almost surely if the true prob-
ability with which the model \( M \) satisfies the formula \( \phi \) lies in the KL support of the proper prior probability distribution \( g \).

Proof: There are two cases:

(i) Suppose the PAFM specification is true i.e. \( \rho_0 = \rho + \epsilon_2 \), then the interval \([0, \rho + \epsilon_2]\) is strongly \( \delta \)-separated from \( \rho_0 \) for some non-zero constant \( \delta \) (See [11]). Then, we know that the following holds infinitely often almost surely (from Theorem II.2).

\[
\int_{0}^{\rho+\epsilon_2} \frac{f(x_1|u) \cdots f(x_n|u) \cdot g(u) \, du}{\int_{0}^{\rho} f(x_1|u) \cdots f(x_n|u) \cdot g(u) \, du} C_0 \leq e^{-nb} C_0
\]

(ii) Suppose the PAFM specification is false i.e. \( \rho_0 < \rho - \epsilon_1 \), then the following holds infinitely often almost surely.

\[
\int_{\rho-\epsilon_1}^{\rho+\epsilon_2} \frac{f(x_1|u) \cdots f(x_n|u) \cdot g(u) \, du}{\int_{0}^{\rho-\epsilon_1} f(x_1|u) \cdots f(x_n|u) \cdot g(u) \, du} C_1 \leq e^{-nb} C_1
\]

In both cases, the computed expected loss vanishes exponentially to zero as we increase the number of samples obtained. Thus, it will become strictly smaller that the cost of generating each sample \( s \) and the algorithm stops.

Theorem III.2. The number of samples needed by Algorithm 1 is logarithmic in the cost of making an incorrect decision.


Hence, the number of samples needed by the algorithm is only logarithmic in the loss from making an incorrect decision. The characterization of the algorithm for two special cases is worth further study:

1) Cost of Simulation much smaller than the Loss associated with an Erratic decision: An interesting scenario happens when the cost of generating each sample observation through simulation for the model i.e. \( s \) is much smaller than the loss associated with an error \( C \) i.e. \( \frac{C}{S} \to 0 \). The number of samples needed before the algorithm terminates is bounded by the logarithm of the ratio \( \frac{S}{C} \), i.e. \( \frac{C}{S} \to 0 \). Hence, even though the ratio \( \frac{C}{S} \) itself may be large, the number of samples needed for the algorithm to terminate only grows logarithmically in this ratio.

2) Cost of Simulation close to the Loss associated with an Erratic decision: When the cost of generating each sample observation through simulation for the model i.e. \( s \) is close to the loss associated with an error \( C \) i.e. \( \frac{S}{C} \to 1 \), then the number of samples needed by our algorithm is approximately a linear function of the ratio \( \frac{S}{C} \) itself. This can be demonstrated by using the observation that the Taylor series expansion for the term \( \log x \) is approximately \( x - 1 \), when \( x \) is close to 1.

IV. EMPIRICAL PERFORMANCE OF ALGORITHM

A. Cost Model

As we know the time needed to simulate the model under test and the cost of using CPU cycles for the same time, we can estimate (lower bounds on) the cost of simulating our stochastic embedded system model. For example, in the next section, we assume that it takes at least 4 seconds to run a simulation of our stochastic embedded system model on a given processor. We also know that the cost of running the simulation on an EC2 cluster for an hour is 8.5 cents. Thus, the cost of generating a single sample for this model \( s \) is at least \( 9 \times 10^{-5} \) cents.

The loss from the algorithm producing an incorrect answer depends on the context of the application in which this verification problem arises [7]. For example, if the incorrect answer could lead to an automobile recall that costs $1 billion, that would be the cost of producing an incorrect answer. We note that we only need upper bounds on the cost of making a wrong decision and lower bounds on the cost of simulating the system.

B. Logarithmic Dependence of Number of Samples on Costs

We study the number of samples required by the algorithm as a function of the loss from the incorrect answer. We analyzed a property that should hold on the model with at least probability 0.95. We used uniform priors and no indifference region. We also know that this property is true on the model with probability 0.88. We varied the cost of making an incorrect decision but kept the losses due to the algorithm producing an incorrect answer symmetric i.e. \( C_0 = C_1 \) (say).

For the same system and the same property, we sought an answer to the question that the property is true with probability at least 0.5. Again, we used uniform priors and no indifference region for our algorithm. We varied the cost of making an incorrect decision but kept the costs symmetric.

In Figure 2(a) and Figure 2(b), we plotted the number of samples observed before our algorithm terminates against the logarithm of the symmetric cost of making an incorrect decision. An important point to note is that the number of samples only increases logarithmically in the cost of an incorrect decision. This is explained by the fact that Bayesian posteriors concentrate exponentially as discussed in Theorem III.1. We also plot the expected loss from making an incorrect decision at the termination of the algorithm. We see that the expected loss has fallen below the cost of a simulation when the algorithm terminates.

The exponential decay in expected loss from an incorrect decision merits further discussion from the perspective of practical applications of this algorithm. If our upper estimate of making a wrong decision is off by a constant factor, the algorithm would only need to draw constant many additional samples. Similarly, if our lower estimate on the cost of drawing a single observation using simulation is off by a constant factor, the cost based Bayesian validation algorithm only needs to draw constant many additional samples. While it is probably impossible to precisely compute the cost of each simulation or the cost of making a wrong decision in advance, it is reasonable to assume that good lower bounds on the cost of simulating the model and upper bounds on the cost of making a wrong decision are readily available [7]. The performance of our algorithm is not severely affected by the use of these approximate costs for simulation and expected losses because of the exponential concentration of the Bayesian posterior.

We analyzed a Simulink model of the Automatic Transmission Control subsystem using 33,814,710 samples. The
subsystem has two inputs: Brake and Throttle, and two outputs: Vehicle Speed and Engine RPM. We wanted to verify the property that the Engine RPM does not exceed 5000 for more than 5 seconds during the simulation. We assume that the cost of running a simulation is at least $0.000001, while the upper bound on the cost of a wrong decision is assumed to be $1 million. We wanted to verify that the property holds on our model with at least 0.999999 probability.

V. CONCLUSION AND FUTURE WORK

Statistical Verification based on Bayesian analysis reduces the overall cost of testing a system and the expected loss from system failure [7]. Our algorithm provides a reasonable basis to allocate testing resources among multiple systems. Systems that are likely to lead to large losses (due to failure) need to be tested much more than those whose failure is likely to lead to smaller losses. Our algorithm quantifies the amount of testing that one ought to undertake given the cost of performing each test and the overall cost of a potential failure of the system under verification.

Several interesting directions for future work remain open. The use of i.i.d. sampling algorithms during the proposed verification technique is not satisfactory. It is well known that i.i.d. sampling algorithms are a particularly poor way of exploring rare errors in stochastic models of cyber-physical systems. In ongoing work, we are investigating the use of non-i.i.d. sampling algorithms for statistical verification. The goal or our research is to merge the theory of statistical verification with the practice of industrial testing.

While the design and verification of stochastic cyber-physical models is a challenge [14], the automated correct-by-construction synthesis of such cyber-physical systems has been a subject of increasing attention [1], [8]. We believe that statistical verification techniques are particularly well suited [5], [10] for the automated synthesis of stochastic cyber-physical systems from high level behavioral specifications. Statistical verification approaches can be used to automatically explore design spaces of stochastic models against both correctness and performance specifications.

REFERENCES