Statistically Sound Verification and Optimization for Complex Systems [⋆]

Yan Zhang, Sriram Sankaranarayanan and Fabio Somenzi

University of Colorado, Boulder, CO 80309, USA
{yan.zhang, srirams, fabio}@colorado.edu

Abstract. This paper discusses verification and optimization of complex systems with respect to a set of specifications under stochastic parameter variations. We introduce a simulation-based statistically sound model inference approach that considers systems whose responses depend on a few design parameters and many stochastic parameters. The technique iteratively searches over the space of design parameters by alternating between verification and optimization phases. The verification phase uses statistical model checking to check if the model using the current design parameters satisfies the specifications. Failing this, we seek new values of the design parameters for which statistical verification could potentially succeed. This is achieved through repeated simulations for various values of the design and stochastic parameters, and quantile regression to construct a model that predicts the spread of the responses as a function of the design parameters. The resulting model is used to select a new set of values for the design parameters. We evaluate this approach over several benchmark examples. In each case, the performance is improved significantly compared to the nominal design.

1 Introduction

We address the problem of selecting design parameter values for complex systems that are "robust" with respect to varying stochastic parameters. For instance, a control designer often faces the problem of selecting gain values of a controller so that the design is robust under stochastic disturbances and variations in the plant model parameters. Elsewhere, the problem of designing "robust" analog circuits that can function correctly under stochastic process variations is also well known. Thus, the problem of finding appropriate design parameter values for a complex system whose output responses depend on a few controllable (tunable) design parameters, and numerous uncontrollable stochastic parameters with known probability distributions, is quite common. In this work, we present an automatic search method that seeks to adjust the design parameters so that the resulting system satisfies the specifications with a given probability bound.

We introduce an approach that combines simulation, quantile regression [12] and a generalization procedure. The approach iterates between two phases: *verification* and *optimization*. The verification phase determines whether the system is safe given the

[⋆] This work was supported by the US National Science Foundation (NSF) through grants CNS 1016994 and CCF 1320069. All opinions expressed are those of the authors, and not necessarily of the NSF.

currently chosen design parameters. If not, we search for a new set of values for the design parameters (design point) that can potentially yield a safe system. The new design point is chosen by constructing a relational model that captures the spread of the responses as a function of the design parameters using simulations and *quantile regression*. This relational model is then constructed to search for new design points that potentially satisfy the specifications with the given probability bound. Repeated iterations of this process checks correctness over a sequence of design points, while iteratively refining the relational model, converging to optimal values for the design parameters.

The relational model effectively marginalizes the effects of the stochastic parameters. It is constructed using *quantile regression* to fit through the upper and lower quantiles of the responses as a function of the design parameters, followed by a *generalization* procedure that relaxes the model into a statistical over-approximation of the response. The procedure iterates until it successfully finds a design point that satisfies the specifications, or stops when a new design point cannot be found. In the latter case, we report that we cannot find a safe design point and suggest that the specifications may be too stringent.

The main contribution of this paper is the introduction of a simulation-based statistically sound model inference approach that combines verification and optimization. This problem is hard for formal verification techniques that reason symbolically about the distribution of an output response. In recent years, statistical verification techniques have received increasing attention [24, 19, 10, 22, 27, 17, 13]. They are simulation-based, requiring just the ability to simulate the model efficiently for various values of design and stochastic parameters. Such a technique can be used to place "high confidence" bounds on the probability that a response satisfies a given specification. Statistical model checking (SMC) [24, 10] is a family of statistical verification techniques that relies on sequential hypothesis testing [21, 11]. An SMC technique checks whether a time-bounded LTL property is satisfied with a certain probability bound by deciding between two mutually exclusive hypotheses through simulation.

SMC provides a "likely yes/no" answer for a system and its specifications. In contrast, we wish to find design points for which the system is likely to satisfy the specifications. A straightforward, but impractical, approach iterates through individual design points, and runs SMC for each of them. Hence, it is desirable to build a model that characterizes the relationship between design parameters and responses. For this purpose, regression-based performance modeling techniques are natural candidates and have been studied extensively [20, 15, 14, 3, 26, 6]. They use simulation data to fit functions that approximate the true response. However, since the outcome of a regression-based approach is an approximation, rather than a sound model of the response function, few guarantees can be provided. Our previous work attempts to combine regression and hypothesis testing techniques to provide a *statistically sound model inference* approach [25]. A statistically sound model provides an envelope of a response that is guaranteed to contain the corresponding response with a high probability. Such a model is useful when dealing with complex systems, in which case a formally sound model cannot be obtained.

In the control community, similar problems have been considered, such as robust convex optimization [2] and chance-constrained optimization [16]. A classic technique to solve for these problems is known as the scenario approach [4], which provides solutions that guaranteed to be optimal with a desired probability. The similarity between the scenario approach and our approach lies in that both of them deal with uncertainties in a system and provide statistical guarantees on the solutions. However, the scenario approach assumes that the system dynamics are available in a closed form, while our approach only relies on the ability to simulate the system.

To our knowledge, the idea of this paper, which combines quantile regression with SMC is unique. Nevertheless, the use of SMC for tuning model parameters has received some attention in the past. Jha *et al.* present the use of SMC to tune parameters for closed loop controller models in order to satisfy a given set of temporal logic specifications [9]. Their approach uses Monte-Carlo sampling over the design parameter values, wherein the number of simulation runs required to resolve the hypothesis testing problem is used as the fitness function for each design parameter. A similar idea is used by Palaniappan *et al.* to fit parameter values for biological models based on experimental observations as well as model specifications [17]. In their work, SMC is used to derive a fitness function that seeks to measure the fraction of the specifications satisfied by a particular choice of model parameters. Our approach builds a more sophisticated "global" model of how the properties depend on the design parameters using quantile regression, and is expected to use fewer number of simulations.

While our approach considers controllable design parameters, a significant body of work treats problems involving uncontrollable, non-deterministic parameters along with stochastic parameters using SMC. We refer the reader to recent papers by Zuliani *et al.* [8] and Ellen *et al.* [7] that use reinforcement learning techniques to verify the correctness properties under the worst case values of non-deterministic parameters.

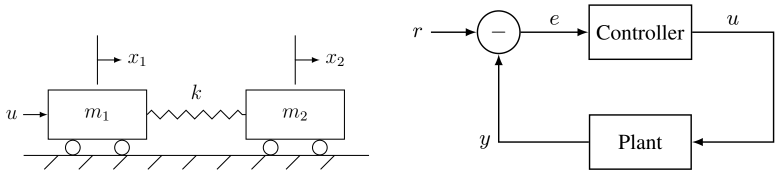
The paper is organized as follows. Section 2 presents an overview of the proposed approach. Section 3 formulates the use of quantile regression. Section 4 discusses how to manipulate the model from quantile regression to achieve statistical soundness. Section 5 introduces a method to find new design points that are potentially safe. Section 6 shows applications of the proposed approach.

2 Overview

Consider a system with design parameters $\mathbf{u} \in \mathbb{U}$ and stochastic parameters $\mathbf{x} \in \mathbb{X}$, where $\mathbb{U}$ and $\mathbb{X}$ are the domains of the parameters. Assume that the design parameters are controllable, i.e., we can choose values for them, and the stochastic parameters, following a joint distribution $F(\mathbf{x})$, are uncontrollable. We also assume give nominal design parameters $\mathbf{u}_{nom}$. A response ϕ is defined by a function $r(\mathbf{u}, \mathbf{x})$ where r is computable as a *black-box*, but has an complex analytic form. A specification of such a system has the form $\phi \in [a, b]$, with $a, b \in \mathbb{R}$. We wish to find a design parameter $\mathbf{u}$ that satisfies the specification with probability at least θ_0 (a given probability threshold):

$$\Pr_{\mathbf{x} \sim F(\mathbf{x})} \left(r(\mathbf{u}, \mathbf{x}) \in [a, b] \right) \geq \theta_0 \,, \tag{1}$$

First, we statistically verify whether the system with the nominal parameters $\mathbf{u}_{nom}$ satisfies (1). If the verification fails, we search for new design point $\mathbf{u}_{new} \in \mathbb{U}$.



(a) A two-mass-spring system.

(b) Closed-loop control.

Fig. 1: A two-mass-spring system and the closed-loop system with a controller.

Example 1 (A Two-Mass-Spring System). A two-mass-spring system [23] is shown in Figure 1a. It consists of two rigid bodies and a spring. The model is uncertain in which $m_1 = 1.0 \pm 20\%$, $m_2 = 1.0 \pm 20\%$ and $k = 1.0 \pm 20\%$ with appropriate units. We apply force u to m_1 and measure $y = x_2$, the position of m_2. In Figure 1b, a controller is used to track y with r, the reference position.

A lead compensator controls the plant. It has two tunable parameters, the pole $p \in [-1200, -800]$ and the zero $z \in [-1.2, -0.8]$. Nominally, $p = -1000$ and $z = -1$. The goal is to design a controller so that the step response of the system satisfies: (1) the settling time $t \leq 2.5$ and (2) the overshoot $r \leq 15\%$ of the steady state value.

The key idea of the proposed approach is to fit a *relational model* for the response $r(\mathbf{u}, \mathbf{x})$. Let $\mathbb{I}$ be the set of real-valued intervals. A relational model g maps design parameters $\mathbf{u} \in \mathbb{U}$ to intervals $g(\mathbf{u}) \in \mathbb{I}$. In effect, $g(\mathbf{u})$ marginalizes the effects of the stochastic parameters. Such a model attempts to over-approximate the spread of $r(\mathbf{u}, \mathbf{x})$ over $\mathbf{x} \sim F(\mathbf{x})$. The key notion that we seek to satisfy is called *statistical soundness*.

Definition 1 (Statistical Soundness). *Given a probability $\theta_0 \in (0, 1)$, a relational model $g : \mathbb{U} \to \mathbb{I}$ is θ_0-statistically sound if for all $\mathbf{u} \in \mathbb{U}$*

$$\Pr_{\mathbf{x} \sim F(\mathbf{x})} \left(r(\mathbf{u}, \mathbf{x}) \in g(\mathbf{u}) \right) \geq \theta_0 \,. \tag{2}$$

While constructing an accurate but fully sound relational model is often expensive, if not impossible, a statistically sound model can be used instead with guarantees that are probabilistic rather than absolute.

In Definition 1 there is a universal quantifier over $\mathbf{u}$. Since the response function r is assumed to be a black-box, finding a model that satisfies (2) is not possible. In the proposed approach, we will restrict ourselves to show that (2) is true for some finite subset of design points. Furthermore, checking if a model $g(\mathbf{u})$ is statistically sound at a given design point $\mathbf{u}$ requires detailed knowledge of the function $r(\mathbf{u}, \mathbf{x})$, which is not available. To address this, we will use hypothesis testing techniques such as the sequential Bayesian test to conclude statistical soundness *with high confidence* at a given design point $\mathbf{u}$.

Figure 2 shows the basic flow of the proposed approach. First, using quantile regression, we compute a relational model $\hat{g}(\mathbf{u}) = [\hat{g}_\ell(\mathbf{u}), \hat{g}_u(\mathbf{u})]$ with affine functions $\hat{g}_\ell$ and $\hat{g}_u$, to approximate the response function $r(\mathbf{u}, \mathbf{x})$ with $\mathbf{u} \in \mathbb{U}$ and $\mathbf{x} \in \mathbb{X}$. Quantile regression is carried out using randomly sampled design and stochastic parameters, and the corresponding values of the response. However, $\hat{g}$ is not guaranteed

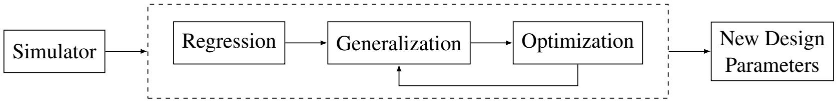
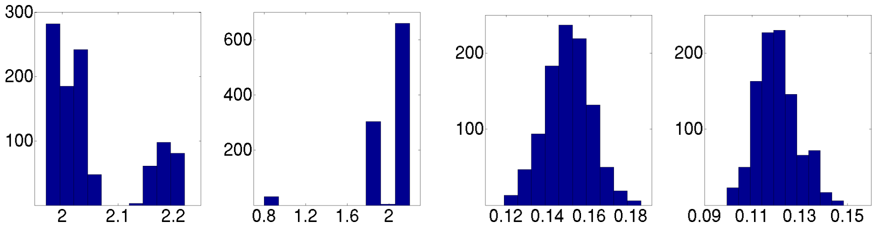


Fig. 2: Basic working flow of the proposed approach.



(a) t at $\mathbf{u}_{nom}$ (left) and $\mathbf{u}_{new}$ (right).

(b) r at $\mathbf{u}_{nom}$ (left) and $\mathbf{u}_{new}$ (right).

Fig. 3: Histogram of t (in seconds) and r (percentage) in the two-mass-spring example.

to be statistically sound. Next, we check whether the nominal design point $\mathbf{u}_{nom}$ satisfy the specifications under stochastic parameter variations. This is achieved by a generalization technique [25], which derives a relational model $g(\mathbf{u})$ that is θ_0-statistically sound for $\mathbf{u} = \mathbf{u}_{nom}$ with high confidence. Intuitively, the procedure fixes the design parameters to $\mathbf{u}_{nom}$ and samples the stochastic parameters sequentially. A tolerance interval $I : [\ell, u]$ is computed so that a long enough sequence of the observed responses fall in the interval $[\hat{g}_\ell(\mathbf{u}_{nom}) + \ell\,,\, \hat{g}_u(\mathbf{u}_{nom}) + u]$. This procedure is guaranteed to yield $g(\mathbf{u}) \equiv [\hat{g}_\ell(\mathbf{u}) + \ell, \hat{g}_u(\mathbf{u}) + u]$ that is statistically sound at $\mathbf{u} = \mathbf{u}_{nom}$, with high confidence. For a specification $\phi \in [a, b]$, if $g(\mathbf{u}_{nom})$ is contained in $[a, b]$, we conclude that with a high probability (which depends on θ_0) the system is safe at $\mathbf{u}_{nom}$. Otherwise, we search for new design point that yields a safe system.

To carry out the search, the response $r(\mathbf{u}, \mathbf{x})$ is modeled by $g(\mathbf{u})$. We then look for a point $\mathbf{u}_1 \in \mathbb{U}$ that has the largest margin from violating the specifications. Since g is statistically sound only at $\mathbf{u}_{nom}$, generalization is applied again so that g becomes statistically sound at $\{\mathbf{u}_{nom}, \mathbf{u}_1\}$. Then we check whether the specifications hold. The procedure continues until either the system is safe at some $\mathbf{u}_i$ at the i^{th} iteration, or a limit on the number of iterations is exceeded, in which case, a failure is returned.

Example 2. Let us continue with Example 1. First, we simulate the system with randomly sampled design and stochastic parameters

$$p \in [-1200, -800],\ z \in [-1.2, -0.8],\ m_1 \in [0.8, 1.2],\ m_2 \in [0.8, 1.2],\ k \in [0.8, 1.2].$$

We use quantile regression to fit a lower and an upper bound function for the responses t and r. For instance, $1.157 + 0.03966p + 0.7071z$ is the lower bound of t, with p and z normalized to $[-1, 1]$. Figure 3 shows the histograms of t and r at $\mathbf{u}_{nom}$ and $\mathbf{u}_{new}$. Apparently, the system violates the specification $r \leq 15\%$ at $\mathbf{u}_{nom}$ ($p = -1000$ and $z = -1$). After optimization, we have $p = -1200$ and $z = -0.928$. The histograms show that both specifications are satisfied.

3 Quantile Regression

In this section, we present the basic notion of quantile regression. For a real-valued random variable X with a distribution $F_X(x) = \mathsf{Pr}(X \leq x)$, the τ^{th} quantile of X is defined as $Q_X(\tau) = \inf\{x : F_X(x) \geq \tau\}$. Informally, it is the smallest x such that $\mathsf{Pr}(X \geq x)$ is at most $1 - \tau$.

Consider a complex system with design parameters $\mathbf{u}$, stochastic parameters $\mathbf{x}$ and a response $\phi = r(\mathbf{u}, \mathbf{x})$. For a fixed $\mathbf{u}$, $r(\mathbf{u}, \mathbf{x})$ can be regarded as a random variable, denoted as $\tilde{r}_{\mathbf{u}}$. The random variable $\tilde{r}_{\mathbf{u}}$ follows the distribution of $r(\mathbf{u}, \mathbf{x})$, which depends on r and the distribution of $\mathbf{x}$. A τth quantile function $g_\tau(\mathbf{u}) = Q_{\tilde{r}_{\mathbf{u}}}(\tau)$ maps the design parameters onto the τth quantile of the random variable $\tilde{r}_{\mathbf{u}}$. In the proposed approach, the goal of quantile regression is to approximate the quantile function $g_\tau(\mathbf{u})$ with an affine function of the form $\hat{g}_\tau(\mathbf{u}; \mathbf{c}) = c_0 + \sum_{i=1}^{k} c_i u_i$, where $\mathbf{c} = (c_0, c_1, \ldots, c_k)$ are unknown coefficients and u_i is the i^{th} design parameter. The coefficients $\mathbf{c}$ are computed by minimizing the residual between $g_\tau(\mathbf{u})$ and $\hat{g}_\tau(\mathbf{u})$,

$$\min_{\mathbf{c}=(c_0,c_1,\ldots,c_k)} \left\| g_\tau(\mathbf{u}) - \hat{g}_\tau(\mathbf{u}; \mathbf{c}) \right\| . \tag{3}$$

Since $g_\tau(\mathbf{u})$ is often not available, (3) is merely conceptually useful. We show a general approach to solve for $\hat{g}_\tau(\mathbf{u}; \mathbf{c})$. For a given set of simulation data with m data points, quantile regression relies on the following penalty function,

$$\rho_\tau(\mathbf{e}) = \sum_{\substack{i=1 \\ e_i \geq 0}}^{m} \tau e_i + \sum_{\substack{i=1 \\ e_i \leq 0}}^{m} (\tau - 1) e_i \, , \tag{4}$$

where $e_i = r(\mathbf{u}^{(i)}, \mathbf{x}^{(i)}) - \hat{g}_\tau(\mathbf{u}^{(i)})$ are the residuals between the response function and the approximation, evaluated at $(\mathbf{u}^{(i)}, \mathbf{x}^{(i)})$. Here $\mathbf{u}^{(i)}$ and $\mathbf{x}^{(i)}$ refers to the i^{th} observations of the design and the stochastic parameters, respectively. For a fixed τ (except for 0.5), (4) incurs an asymmetric penalty on the positive and the negative side of the residual $\mathbf{e}$. For $\tau > 0.5$ ($\tau < 0.5$), a positive (negative) residual incurs more penalty and thus is minimized. The penalty function (4) leads to the following optimization problem.

$$\min_{\mathbf{c}=(c_0,c_1,\ldots,c_k)} \rho_\tau \left(r(\mathbf{u}, \mathbf{x}) - \hat{g}_\tau(\mathbf{u}; \mathbf{c}) \right) . \tag{5}$$

Since (4) is piecewise linear, it has a unique minimum.

The problem in (5) is solved as a linear program [12]. The penalty function in (4) is encoded by adding auxiliary variables $\mathbf{s} = (s_1, \ldots, s_m)$ and $\mathbf{t} = (t_1, \ldots, t_m)$. The auxiliary variables $\mathbf{s}$ and $\mathbf{t}$ correspond to the cases that the response ϕ is greater and less than the approximation $\hat{g}_\tau$, respectively. With them, we write (5) as

$$\begin{aligned} & \min_{\mathbf{c}=(c_0,c_1,\ldots,c_k)} \sum_{i=1}^{m} \tau s_i + \sum_{i=1}^{m} (1 - \tau) t_i \\ & \text{subject to} \\ & r\left(\mathbf{u}^{(i)}, \mathbf{x}^{(i)}\right) - \hat{g}_\tau\left(\mathbf{u}^{(i)}; \mathbf{c}\right) = s_i - t_i, \;\; i = 1, 2, \ldots, m \, , \\ & \mathbf{s} \geq 0 \, , \; \mathbf{t} \geq 0 \, . \end{aligned} \tag{6}$$

To minimize the objective function, at most one of s_i and t_i should be non-zero. The first constraint forces that either $\mathbf{s}$ or $\mathbf{t}$ equals to the residuals. The last two constraints ensures $\mathbf{s}$ and $\mathbf{t}$ to be non-negative (notice the sign change in the second sum of the objective function in (4) and (6)).

It is important to understand that the formulation in (6) only solves for $\tau \in (0, 1)$. For $\tau = 0$ and $\tau = 1$, (6) fails to find the maximum lower bound and the minimum upper bound. This is because in the two cases, (4) penalizes only one side of the residuals and thus allows the approximation to behave arbitrarily on the opposite side. Such a solution is meaningless in practice. For instance, for $\tau = 0$, the lower bound function of t in Example 2 can be either $0 + 0p + 0z$ or $-100 + 0p + 0z$, with the same objective value of 0. To obtain a meaningful lower (upper) bound approximation from quantile regression, we set τ close to 0 (1). Note that $\hat{g}_\tau$ is not necessarily close to the true lower (upper) bound. In the case that there are outliers in the simulation data, $\hat{g}_\tau$ can be distant from the true bound. In contrast, $\hat{g}_\tau$ tends to leave out the outliers and only concerns with the normal data. Such a property is often desirable when dealing with data from practical settings. In the following, we write $\hat{g}_\ell$ and $\hat{g}_u$ to indicate the estimated lower and the upper bound, respectively. By default, we assume that $\hat{g}_\ell$ is computed with $\tau = 0.01$ and $\hat{g}_u$ with $\tau = 0.99$.

4 Generalization and Verification

As mentioned in Section 2, $\hat{g}_\ell$ and $\hat{g}_u$ form a relational model $\hat{g}(\mathbf{u}) \equiv [\hat{g}_\ell(\mathbf{u}), \hat{g}_u(\mathbf{u})]$. Clearly, $\hat{g}$ is not necessarily statistically sound (see Definition 1) and thus does not provide guarantees on the behavior of the underlying system. We now present a generalization technique that converts $\hat{g}$ into a statistically sound model with high likelihood, and statistically verifies whether specifications of the form $\phi \in [a, b]$ are satisfied.

Generalization Recall that Definition 1 defines statistical soundness for all $\mathbf{u} \in \mathbb{U}$. Such a condition is too strong since our goal is to (1) learn whether the specifications hold at $\mathbf{u}_{nom}$ and if not, (2) find a new point $\mathbf{u}_{new}$ that satisfies them. Hence we are only concerned with statistical soundness at these two design points.

Once the design parameters are fixed, $\hat{g}$ becomes an interval. We derive a tolerance interval $[\ell, u]$ so that the interval $[\hat{g}_\ell(\mathbf{u}) + \ell, \hat{g}_u(\mathbf{u}) + u]$ is a statistically sound bound for the response ϕ under stochastic parameter variations. The procedure is based on sequential Bayesian test which is briefly reviewed here.[1] Sequential Bayesian test investigates statistical hypotheses through a sequence of observations and determine which one should be accepted. It computes Bayes factor

$$B = \frac{\mathsf{Pr}(z_1, \ldots, z_N \mid \mathcal{H}_1)}{\mathsf{Pr}(z_1, \ldots, z_N \mid \mathcal{H}_2)}, \quad z_i = \begin{cases} 1, & \mathcal{H}_1 \vdash s_i \\ 0, & \mathcal{H}_2 \vdash s_i \end{cases},$$

where $\mathcal{H}_1$ and $\mathcal{H}_2$ are mutually exclusive hypotheses, each z_i is a random variate of a Bernoulli random variable Z, and $\mathcal{H} \vdash s$ is interpreted as s is in favor of $\mathcal{H}$. A large

[1] The interested readers are referred to Kass and Raftery [11] and Zhang *et al.* [25].

```
Data: Model $\hat{g}(\mathbf{u}) = [\hat{g}_\ell, \hat{g}_u]$, Design Parameters $\mathbf{u}$, Probability $\theta_0$, Threshold $T$
Result: Tolerance Interval $[\ell, u]$, Model $g(\mathbf{u})$
$K = -\log(T+1)/\log\theta_0 - 1$ ;
$\ell$, $u$, count = 0 ;
while count $< K$ do
    $\mathbf{x}$ = Sample the stochastic parameter space ;
    $\phi$ = Simulate the system at design parameters $\mathbf{u}$ and measure response ;
    if $\hat{g}_\ell(\mathbf{u}) + \ell \leq \phi \leq \hat{g}_u(\mathbf{u}) + u$ then
        count = count + 1 ;
        continue ;
    else
        count = 0 ;
        $\ell, u = \min(\phi - \hat{g}_\ell(\mathbf{u}), \ell)\,,\ \max(\phi - \hat{g}_u(\mathbf{u}), u)$ ;
    end
end
Return $[\ell, u]$, $[\hat{g}_\ell(\mathbf{u}) + \ell, \hat{g}_u(\mathbf{u}) + u]$ ;
```

Algorithm 1: Generalization that achieves statistical soundness at fixed $\mathbf{u}$.

Bayes factor indicates that the observed data support $\mathcal{H}_1$ over $\mathcal{H}_2$. Thus we specify a threshold T such that we accept $\mathcal{H}_1$ when B grows beyond T, and accept $\mathcal{H}_2$ when it falls below $1/T$. Usually $\mathcal{H}_1$ and $\mathcal{H}_2$ have the form $\mathsf{Pr}(\Psi) \geq \theta_0$ and $\mathsf{Pr}(\Psi) < \theta_0$, where θ_0 is a specified probability and Ψ denotes the assertion

$$r(\mathbf{u}, \mathbf{x}) \in [\hat{g}_\ell(\mathbf{u}) + \ell, \hat{g}_u(\mathbf{u}) + u]\,,\ \text{for fixed } \mathbf{u} \text{ and } \mathbf{x} \sim F(\mathbf{x})\,, \tag{7}$$

The goal is to derive proper ℓ and u for given θ_0 and T such that $\mathcal{H}_1$ is accepted.

Algorithm 1 shows the generalization procedure to derive a tolerance interval to achieve statistical soundness at fixed design parameters. The inputs are the model $\hat{g}(\mathbf{u}) = [\hat{g}_\ell(\mathbf{u}), \hat{g}_u(\mathbf{u})]$, fixed design parameters $\mathbf{u}$, a probability θ_0 which indicates the desired probability that (7) should happen, and a Bayes factor threshold T. The algorithm first computes a sequence length K with the specified θ_0 and T. Intuitively, it is the minimum number of consecutive supportive observations required to accept $\mathcal{H}_1$ for the given θ_0 and T. Then the tolerance interval $[\ell, u]$, as well as a *count* variable, is initialized to 0. The *count* variable records the number of consecutive supportive observations. Next, we sample the stochastic parameters $\mathbf{x}$ according to the distribution $F(\mathbf{x})$ and simulate the system to obtain the response ϕ. The observation supports $\mathcal{H}_1$ if (7) holds. In this case, the variable *count* is incremented, terminating when it reaches K. Otherwise, *count* is reset to 0, and ℓ and u are updated to satisfy (7).

One may have noticed that Algorithm 1 did not employ the comparison between the Bayes factor B and its threshold T. Instead, it derives a sequence length K and lets a *count* variable grows towards K. In fact, there is a natural correspondence between *count* and B, as well as K and T (see Zhang *et al.* [25] for details). An important observation is that *count* is only incremented when we find a supportive observation. Therefore, for fixed θ_0 and T, that *count* reaches K is equivalent to that the Bayes factor B grows to at least T.

Theorem 1. *Algorithm 1 terminates and when it terminates, we have $B \geq T$.*

Verification Algorithm 1 yields a tolerance interval $[\ell, u]$ and a model

$$g(\mathbf{u}) = [\hat{g}_\ell(\mathbf{u}) + \ell, \hat{g}_u(\mathbf{u}) + u] \tag{8}$$

that is θ_0 statistically sound at the fixed design parameters. It means that for a fixed $\mathbf{u}$, we have a high level of confidence to claim that the response ϕ has a probability of at least θ_0 to lie in the interval (8). It has been shown that the level of confidence is linked to the Bayes factor threshold T such that the type I/II error is bounded by $\frac{1}{T+1}$ [10, 27]. Hence with large θ_0 and T, the interval (8) is almost an over-approximation of the response ϕ under stochastic parameter variations. To verify whether specifications $\phi \in [a, b]$ hold at some $\mathbf{u}$, we simply check whether (8) is contained in $[a, b]$. If yes, we conclude that with a confidence level of at least $1 - \frac{1}{T+1}$, the system is safe with a probability of at least θ_0 at $\mathbf{u}$. Otherwise, we continue to search for new design point.

5 Optimization

To find a new design point, we introduce an iterative procedure. At the i^{th} iteration, we try to find a candidate $\mathbf{u}_{new}^{(i)}$ that is safe *with respect to the model in* (8). We may fail if either the specifications are too stringent or our approximation is too excessive. In these cases, we stop and report that for $\mathbf{u} \in \mathbb{U}$ and $\mathbf{x} \in \mathbb{X}$, we cannot find a design point which satisfies all the specifications. Suppose $\mathbf{u}_{new}^{(i)}$ is found. Since (8) is not guaranteed to be statistically sound at $\mathbf{u}_{new}^{(i)}$, we apply generalization so that (8) becomes statistically sound at $\mathbf{u}_{new}^{(i)}$, and check whether the system is safe there. If yes, $\mathbf{u}_{new}^{(i)}$ is the final design point. Otherwise, we try again with the updated model in (8). After the i^{th} iteration, $\mathbf{u}_{new}^{(i)}$ is included in the set of points at which (8) is statistically sound.

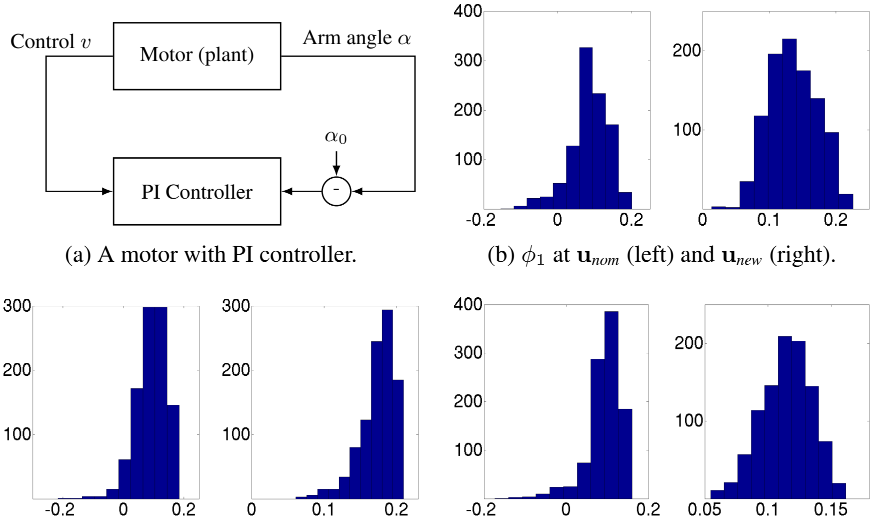
It is easy to pick up a candidate point from (8) that satisfies the specifications. However, an arbitrary choice can easily lead to a failed attempt in verification. As a consequence, more iterations and thus more simulations would be required. Therefore, the candidate should be the one that is most likely to satisfy the specifications. Our solution is to search for the point that has the largest margin from violating the specifications using the following linear program:

$$\begin{aligned} &\max_{\mathbf{u}_{new} \in \mathbb{U}} (b - \hat{g}_u(\mathbf{u}_{new}) - u) + (\hat{g}_\ell(\mathbf{u}_{new}) + \ell - a) \\ &\text{subject to} \\ &a \leq \hat{g}_\ell(\mathbf{u}_{new}) + \ell \leq \hat{g}_u(\mathbf{u}_{new}) + u \leq b\,. \end{aligned} \tag{9}$$

In the case of multiple specifications, (9) consists of multiple constraints, each corresponding to a specification. Also, the objective function of becomes the sum of the margin for each specification. Clearly, (9) is infeasible if and only if we cannot find any candidate.

6 Experimental Evaluation

We present four applications: (1) a motor with a rigid arm controlled by a PI controller, (2) a ring oscillator circuit modeled at the transistor-level, (3) an insulin pump that controls the blood glucose level of diabetic patients, and (4) an aircraft flight control model.



(a) A motor with PI controller.

(b) ϕ_1 at $\mathbf{u}_{nom}$ (left) and $\mathbf{u}_{new}$ (right).

(c) ϕ_2 at $\mathbf{u}_{nom}$ (left) and $\mathbf{u}_{new}$ (right).

(d) ϕ_3 at $\mathbf{u}_{nom}$ (left) and $\mathbf{u}_{new}$ (right).

Fig. 4: A motor with a rigid arm controlled by a PI controller is shown in (a). Figure (b), (c) and (d) shows the histograms of ϕ_1, ϕ_2 and ϕ_3.

All models have stochastic parameter variations. We use our approach to search for a design point that maximizes the empirical probability of satisfying the given specifications. The experiments are performed on a AMD Athlon II quad-core 2.8 GHz CPU with 4 G RAM. The proposed approach is implemented in Python-2.7.

6.1 Motor with PI Controller

Figure 4a shows a DC motor with an attached rigid arm controller by a PI controller. We control the input voltage v of the motor which determines the angle α of the rigid arm. The goal is to set α to a reference α_0, thus holding the arm at a constant angle. The design parameters are the proportional gain K_p and the integral gain K_i. There are 5 stochastic parameters, such as the resistance and the inductance in the motor model.

The step response $\alpha(t)$ should satisfy the following specifications. Over $t \in [0, 2]$, $\alpha(t) \le 1.5$. The specification is $\phi_1 \ge 0$ where

$$(1)\ \phi_1 = \min(1.5 - \alpha(t))\,,\ t \in [0, 2]\,;$$

Over $t \in [2, T]$ where T is the total simulation time, $\alpha(t) \in [0.8, 1.2]$. The specifications are $\phi_2 \ge 0$ and $\phi_3 \ge 0$ where

$$(2)\ \phi_2 = \min(\alpha(t) - 0.8)\,,\ (3)\ \phi_3 = \min(1.2 - \alpha(t))\,,\ t \in [2, T]\,.$$

The nominal design point $\mathbf{u}_{nom}$ is $K_p = -2.5$ and $K_i = -1$. Our goal is to verify whether the specifications hold at $\mathbf{u}_{nom}$ and if not, find a new design point $\mathbf{u}_{new}$ from $K_p \in [-3, -2]$ and $K_i \in [-1.2, -0.8]$ to satisfy the specifications.

Table 1: Results for the motor example ($\theta_0 = 0.95$ and $T = 100$).

Spec	MC-1000		Proposed Approach							
	$\mathbf{u}_{nom}$	$\mathbf{u}_{new}$	I_{nom}	Sim_R	T_R	Iters	Sim_W	T_W	T_O	I_{new}
1	93.1%	100%	$[-0.08, 0.19]$	500	189 s	1	307	148 s	1 s	$[0.06, 0.22]$
2	95.8%	100%	$[-0.13, 0.17]$				247			$[0.06, 0.22]$
3	95.5%	100%	$[-0.13, 0.16]$				398			$[0.06, 0.17]$
all	92.1%	100%	-				-			-

The system is designed in Matlab$^{®}$with Simulink$^{®}$. Table 1 shows the results of this example. The column "MC-1000" shows the yields of each specification at $\mathbf{u}_{nom}$ and $\mathbf{u}_{new}$ estimated through 1000 Monte-Carlo simulations. Sim_R and T_R are the number of simulations and time spent, respectively, for quantile regression; Sim_W, T_W represent the same for generalization and Sim_O, T_O for optimization. "Iters" is the number of iterations of our search. Finally, I_{nom} and I_{new} are the statistically sound performance bounds at $\mathbf{u}_{nom}$ and $\mathbf{u}_{new}$.

First, notice that the the system fails to satisfy all the three specifications at $\mathbf{u}_{nom}$ as shown by the Monte-Carlo simulations. The proposed approach makes the same conclusion by showing that the performance bounds I_{nom} are not contained in the specifications. The bounds are derived from a relational model g that is statistically sound at $\mathbf{u}_{nom}$. Next, we pick up a new design point $\mathbf{u}_{new}$ from the model g according to the linear program (9), and check whether it satisfies the specifications. In fact it does, as shown by the performance bounds I_{new}. Having yields of 100%, the conclusion is also confirmed by the Monte-Carlo simulations at $\mathbf{u}_{new}$. The new design parameters for this application is $K_p = -2$ and $K_i = -0.8$. To obtain this result, 500 simulations are spent in quantile regression and 398 simulations in generalization.[2]

Figure 4b, 4c and 4d present the histograms of the responses ϕ_1, ϕ_2 and ϕ_3 at $\mathbf{u}_{nom}$ and $\mathbf{u}_{new}$. We choose $\theta_0 = 0.95$ and $T = 100$ in generalization. This means that the probability that the intervals under I_{nom} and I_{new} are the true performance bounds is at least 95%. Given that, we have at least $100\% - \frac{1}{T+1} \times 100\% \approx 99\%$ confidence that $\mathbf{u}_{new}$ satisfies the specifications. Yield estimation from the Monte-Carlo simulations is a strong support to our conclusion.

6.2 Ring Oscillator

Figure 5 shows a ring oscillator. It is is designed to oscillate at a frequency f of 2.1 GHz with a power consumption w of 5 mW. However, a real circuit suffers from process variations, such as the doping concentration and oxide layer thickness, resulting in deviation from the ideal performance. The performance specifications are

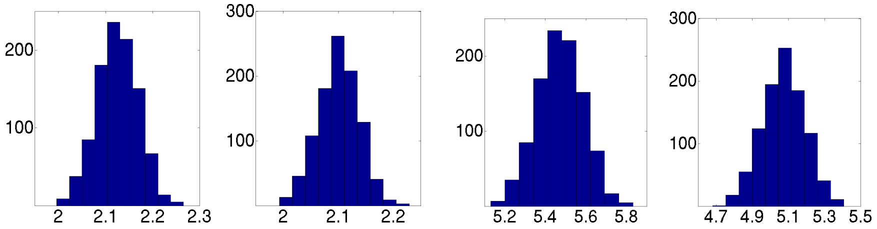
$$(1)\ f \in [2.0, 2.2]\mathrm{GHz}\,,\ (2)\ w \leq 5.5\,\mathrm{mW}\,.$$

We choose 12 design parameters. They are the channel widths and lengths of each transistor. Also, 54 stochastic parameters are considered, arising from process variations in the transistor parameters. The goal is to verify whether the two specifications can be

[2] Simulation data are reusable with respect to different specifications.

Table 2: Results for the 3-stage ring oscillator ($\theta_0 = 0.95$ and $T = 100$).

Spec	MC-1000		Proposed Approach							
	$\mathbf{u}_{nom}$	$\mathbf{u}_{new}$	I_{nom}	Sim_R	T_R	Iters	Sim_W	T_W	T_O	I_{new}
1	95.8%	98.9%	$[2.05, 2.23]$GHz	500	307 s	1	309	233 s	1 s	$[2.04, 2.19]$GHz
2	60.1%	100%	$[5.18, 5.85]$mW				332			$[4.75, 5.41]$mW
all	60.0%	98.9%	-				-			-



(a) f at $\mathbf{u}_{nom}$ (left) and $\mathbf{u}_{new}$ (right).

(b) w at $\mathbf{u}_{nom}$ (left) and $\mathbf{u}_{new}$ (right).

Fig. 6: Histograms of f (left, GHz) and w (right, mW) at in the ring oscillator.

satisfied under the nominal design point and if not, choose new values for the width and length of each transistor.

We use LTSpice® [1], a freely available SPICE simulator, to simulate the circuit. The results are shown in Table 2. The columns have the same meanings as in Table 1. The circuit at the nominal widths and lengths has a poor performance in the power consumption w, which has a yield of only 60.1%. The upper bound of I_{nom} violates the specification (2) excessively. Our approach finds a new design point that has performance bounds that satisfies both specifications, which is confirmed by the Monte-Carlo simulations. The yield is boosted from 60% to almost 100%. Figure 6 shows the histograms of the two responses, f and w, at $\mathbf{u}_{nom}$ and $\mathbf{u}_{new}$. Obviously, we have a significant performance improvement.

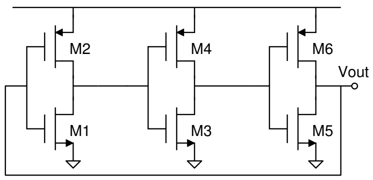
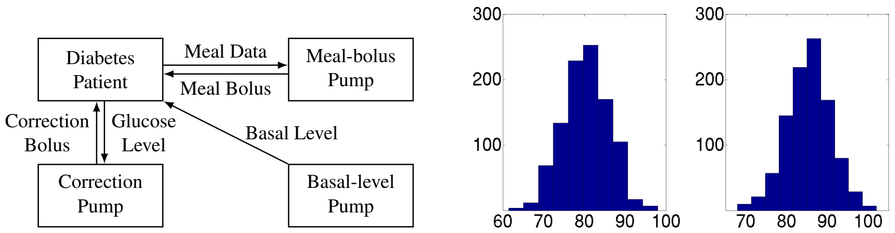


Fig. 5: A 3-stage ring oscillator.

6.3 Insulin Pump

We study a previously published model of an insulin pump used by type-1 diabetic patients [18, 5]. Our model incorporates a physiological model of the human insulin-glucose response from Dalla-Man *et al.* [5], models of sensor errors and a typical pump usage by type-1 diabetic patients [18]. A type-1 diabetic patient uses their insulin pump with at least three "design parameters" that include (a) the *basal rate* (basal) that represents the rate at which background insulin is delivered, (b) the insulin-to-carbohydrates ratio (icRatio) that controls how much bolus insulin is to be administered to the patient for each gram of carbohydrate to be consumed, and (c) a *correction factor* (cor) to correct blood glucose levels that are higher than normal. Clinically, these values are tuned manually by a physician upon close observation of the patient's blood glucose levels,



(a) A model of an insulin pump.

(b) g_{min} at $\mathbf{u}_{nom}$ (left) and $\mathbf{u}_{new}$ (right).

Fig. 7: A model of an insulin pump (left) and the histograms of $\min(g(t))$, the minimum glucose level during simulation (right).

meal and sleep patterns over time. Our study attempts to automate this choice assuming that personalized models are available for patients.

The stochastic parameters include the time of the meal, the amount of carbohydrates in each meal, sensor noise and the discrepancies between the planned and actual meals [18]. Overall, the model has 3 design parameters and 10 stochastic parameters. We used virtual patient parameters published for 30 patients by Dalla Man et al. [5]. Our study here focuses on a single model patient. The total simulation time is $1400\,\text{min}$.

There are many important correctness properties. Ideally, the human blood glucose level should be between $70\,\text{mg/dl}$ and $180\,\text{mg/dl}$. A level lower than $70\,\text{mg/dl}$ is called *hypoglycemia*, and a level higher than $180\,\text{mg/dl}$ is called *hyperglycemia*. In practice, hypoglycemia is usually much more critical than hyperglycemia since it can cause seizures, unconsciousness and even death. Therefore, our goal is to control the blood glucose level higher than $70\,\text{mg/dl}$ at all time time and reduce the time that the patient stays in hyperglycemia as much as possible.

The above description yields the following specifications. The blood glucose level $g(t)$ should be between $70\,\text{mg/dl}$ and $240\,\text{mg/dl}$ over $t \in [0, T]$ where T is the total simulation time.

$$(1)\ \min(g(t)) \geq 70\,\text{mg/dl}\,,\ (2)\ \max(g(t)) \leq 240\,\text{mg/dl}\,;$$

The maximum period p_h for hyperglycemia is at most $240\,\text{min}$, and the total time in hyperglycemia is at most 20% of the total simulation time.

$$(3)\ p_h \leq 240\,\text{min}\,,\ (4)\ r_h \leq 20\%\,.$$

Table 3 shows the results of applying our approach to the data for model that pertains to a single patient, whose insulin pump is tuned to a nominal design point basal $= 0.3$, icRatio $= 0.06$ and cor $= 0.06$. Observe that the pump works well except that it has a 3.8% chance of dangerous hypoglycemia. *Our approach lowers this chance to 0.4%*, a significant lowering of a risk. Another observation comes from the number of iterations. Unlike the other examples, our approach takes 3 iterations to find a new design point. It indicates that the system has a relatively small margin from violating the specifications, as shown by I_{new}. The new design point basal $= 0.225$, icRatio $= 0.080$ and cor $= 0.049$. Histograms of $\min(g(t))$ at $\mathbf{u}_{nom}$ and $\mathbf{u}_{new}$ are shown in Figure 7b.

Table 3: Results for the insulin pump example ($\theta_0 = 0.95$ and $T = 100$). The units of I_{nom} and I_{new} for specification (1) and (2) are mg/dl.

Spec	MC-1000		Proposed Approach							
	$\mathbf{u}_{nom}$	$\mathbf{u}_{new}$	I_{nom}	Sim_R	T_R	Iters	Sim_W	T_W	T_O	I_{new}
1	96.2%	99.6%	$[68.12, 95.28]$	500	624s	3	567	701s	4s	$[70.0, 102.1]$
2	100%	100%	$[186.6, 219.3]$				549			$[189.2, 227.0]$
3	100%	100%	$[41.44, 209.8]$min				423			$[48.6, 213.3]$min
4	100%	100%	$[6.0\%, 18.8\%]$				420			$[6.2\%, 20.0\%]$
all	96.2%	99.6%					-			-

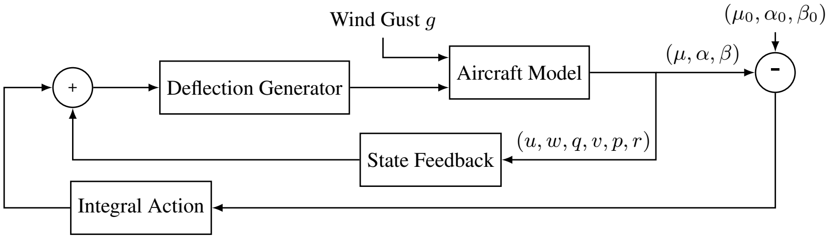


Fig. 8: An aircraft flight control model.

6.4 Aircraft Flight Control System

Figure 8 shows a model of the flight control system in an aircraft. This model is available in Matlab® R2014a Robust Control Toolbox™. The aircraft is modeled as a 6th-order state-space system. The state variables include the velocity on x, y and z-body axis (u, v, w), the pitch rate q, the roll rate p and the yaw rate r. These variables together with three responses, the flight-path bank angle μ, the angle of attack α and the sideslip angle β, are available to the controller. The controller, which consists of a state feedback control and an integral control, is designed to generate the deflections of the elevators, the ailerons and the rudder so that a good tracking performance is maintained on the responses with respect to the reference μ_0, α_0 and β_0.

The controller has two gain matrices, K_x and K_i, that maps the controller inputs to deflections. K_x is a 3×6 state-feedback matrix, and K_i is a 3×3 matrix for integrating the three tracking errors. In all, we have 27 design parameters. The stochastic parameters arise from uncertainties in the state matrix and the input matrices along with the stochastic wind disturbance. In all, we have 73 stochastic parameters. The following specifications concern the step response of $\mu(t)$, $\alpha(t)$ and $\beta(t)$. First, the settling time of each trajectory should be smaller than 7.5 s.

$$(1)\ t_\mu \leq 7.5\,\text{s}\,,\ (2)\ t_\alpha \leq 7.5\,\text{s}\,,\ (3)\ t_\beta \leq 7.5\,\text{s}\,;$$

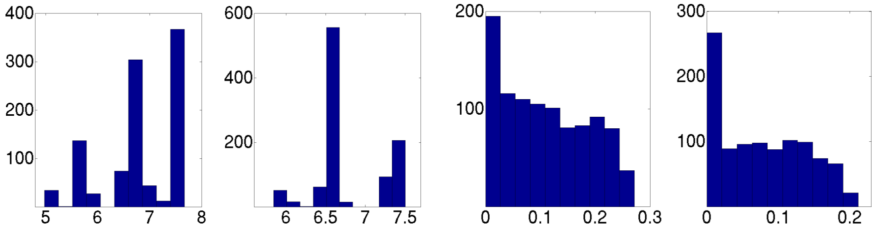
Also, the overshoot should be less than 20% of the steady state value.

$$(4)\ r_\mu \leq 20\%\,,\ (5)\ r_\alpha \leq 20\%\,,\ (6)\ r_\beta \leq 20\%\,.$$

Table 4 presents the results of applying our approach. Observe that the specification (2) and (5) are not satisfies at $\mathbf{u}_{nom}$, confirmed by both the Monte-Carlo simulations and

Table 4: Results for the aircraft flight control example ($\theta_0 = 0.95$ and $T = 100$).

Spec	MC-1000		Proposed Approach							
	$\mathbf{u}_{nom}$	$\mathbf{u}_{new}$	I_{nom}	Sim_R	T_R	Iters	Sim_W	T_W	T_O	I_{new}
1	100%	100%	$[1.40, 6.47]$s	500	307s	1	326	341s	2s	$[1.98, 6.42]$s
2	76.7%	99.9%	$[5.00, 7.79]$s				332			$[5.86, 7.48]$s
3	100%	100%	$[3.82, 6.23]$s				479			$[3.80, 6.34]$s
4	100%	100%	$[3.8\%, 9.5\%]$				399			$[0, 11.7\%]$
5	82.5%	99.5%	$[0, 26\%]$				402			$[0, 19.5\%]$
6	100%	100%	$[5.3\%, 9.4\%]$				507			$[7.7\%, 12.7\%]$
all	74.1%	99.5%	-				-			-



(a) t_α at $\mathbf{u}_{nom}$ (left) and $\mathbf{u}_{new}$ (right).

(b) r_α at $\mathbf{u}_{nom}$ (left) and $\mathbf{u}_{new}$ (right).

Fig. 9: Histograms of t_α (left, in seconds) and r_α (right, as percentage) in the aircraft flight control model.

the performance bounds I_{nom}. We use 500 simulations in quantile regression and 507 in generalization, and find a new design point in one iteration. The new point leads to better performance on t_α and r_α and thus a boost of the overall yield from 74.1% to 99.5%. Figure 9 shows the histograms of t_α and r_α at $\mathbf{u}_{nom}$ and $\mathbf{u}_{new}$, which clearly shows the performance improvement.

Now let us compare I_{nom} with I_{new}. Note that except for t_α and r_α in specification (2) and (5), all the other responses have larger performance bounds at $\mathbf{u}_{new}$ but still satisfy the specifications. It indicates that the proposed approach trades off the performance of the other responses so that (2) and (5) can be satisfied.

7 Conclusion

In this paper, we have introduced a statistically sound model inference approach for the verification and optimization of complex systems. First, using quantile regression, a relational model is computed to approximate the marginalized response function. Then a generalization procedure is employed to relax the model so that it becomes statistically sound at the nominal design point. The resulting model is used to verify the specifications. If fail, the model is then used to search for a new design point. We show several interesting examples that through the application of our approach, the yield of these systems are improved significantly.

References

1. LTSpice: A high performance SPICE simulator, schematic capture and waveform viewer, http://www.linear.com/designtools/software/
2. Ben-Tal, A., Nemirovski, A.: Robust convex optimization. Mathematics of Operations Research 23(4), 769–805 (1998)
3. Bernardinis, F.D., Jordan, M.I., Sangiovanni-Vincentelli, A.: Support vector machines for analog circuit performance representation. In: DAC. pp. 964–969 (2003)
4. Campi, M.C., Garatti, S., Prandini, M.: The scenario approach for systems and control design. Annual Reviews in Control 33(2), 149–157 (2009)
5. Dalla-Man, C., Rizza, R., Cobelli, C.: Meal simulation model of the glucose-insulin system. IEEE Transactions on Biomedical Engineering 54(10), 1740–1749 (2007)
6. Doostan, A., Iaccarino, G.: A least-squares approximation of partial differential equations with high-dimensional random inputs. Journal of Computational Physics 228(12), 4332–4345 (2009)
7. Ellen, C., Gerwinn, S., Fränzle, M.: Statistical model checking for stochastic hybrid systems involving nondeterminism over continuous domains (2014), to appear in special issue on Statistical Model Checking
8. Henriques, D., Martins, J., Zuliani, P., Platzer, A., Clarke, E.: Statistical model checking for markov decision processes. In: QEST'12 (2012)
9. Jha, S.K., Datta, R., Langmead, C., Jha, S., Sassano, E.: Synthesis of insulin pump controllers from safety specifications using bayesian model validation. In: Proceedings of 10th Asia Pacific Bioinformatics Conference, (APBC) (2012)
10. Jha, S.K., Clarke, E.M., Langmead, C.J., Legay, A., Platzer, A., Zuliani, P.: A Bayesian approach to model checking biological systems. In: CMSB. pp. 218–234 (2009)
11. Kass, R.E., Raftery, A.E.: Bayes factors. Journal of the American Statistical Association 90(430), 774–795 (1995)
12. Koenker, R.: Quantile regression. No. 38, Cambridge university press (2005)
13. Lagoa, C.M., Dabbene, F., Tempo, R.: Hard bounds on the probability of performance with application to circuit analysis. IEEE Transactions on Circuits and Systems 55(10), 3178–3187 (2008)
14. Li, X.: Finding deterministic solution from underdetermined equation: large-scale performance variability modeling of analog/RF circuits. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 29(11), 1661–1668 (2010)
15. Mitev, A., Marefat, M., Ma, D., Wang, J.M.: Principle Hessian direction-based parameter reduction for interconnect networks with process variation. IEEE Transactions on VLSI Systems 18(9), 1337–1347 (2010)
16. Nemirovski, A., Shapiro, A.: Convex approximations of chance constrained programs. SIAM Journal on Optimization 17(4), 969–996 (2006)
17. Palaniappan, S., Gyori, B., Liu, B., Hsu, D., Thiagarajan, P.: Statistical model checking based calibration and analysis of bio-pathway models. In: CMSB. pp. 120–134 (2013)
18. Sankaranarayanan, S., Miller, C., Raghunathan, R., Ravanbakhsh, H., Fainekos, G.: A model-based approach to synthesizing insulin infusion pump usage parameters for diabetic patients. In: Communication, Control, and Computing (Allerton), 2012 50th Annual Allerton Conference on. pp. 1610–1617 (2012)
19. Sen, K., Viswanathan, M., Agha, G.: Statistical model checking of black-box probabilistic systems. In: CAV. pp. 202–215 (2004)
20. Singhee, A., Rutenbar, R.A.: Beyond low-order statistical response surfaces: latent variable regression for efficient, highly nonlinear fitting. In: DAC. pp. 256–261 (2007)

21. Wald, A.: Sequential tests of statistical hypotheses. The Annals of Mathematical Statistics 16(2), 117–186 (1945)
22. Wang, Y.C., Komuravelli, A., Zuliani, P., Clarke, E.M.: Analog circuit verification by statistical model checking. In: ASP-DAC. pp. 1–6 (2011)
23. Wie, B., Bernstein, D.S.: A benchmark problem for robust control design. In: American Control Conference. pp. 961–962 (May 1990)
24. Younes, H.L.S., Simmons, R.G.: Probabilistic verification of discrete event systems using acceptance sampling. In: CAV. pp. 223–235 (2002)
25. Zhang, Y., Sankaranarayanan, S., Somenzi, F., Chen, X., Ábraham, E.: From statistical model checking to statistical model inference: Characterizing the effect of process variations in analog circuits. In: ICCAD (2013)
26. Zhang, Y., Sankaranarayanan, S., Somenzi, F., Chen, X., Ábraham, E.: Sparse statistical model inference for analog circuits under process variations. In: ASP-DAC. pp. 449–454 (2014)
27. Zuliani, P., Platzer, A., Clarke, E.M.: Bayesian statistical model checking with application to stateflow/simulink verification. Formal Methods in System Design 43(2), 338–367 (2013)