

**Statistically Sound Verification and Optimization of
Black-Box Systems**

by

Yan Zhang

B.S., Shanghai Jiao Tong University, Shanghai, China, 2009

M.S., University of Colorado, Boulder, CO, U.S., 2012

A thesis submitted to the
Faculty of the Graduate School of the
University of Colorado in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
Department of Electrical, Computer, and Energy Engineering

2014

This thesis entitled:
Statistically Sound Verification and Optimization of Black-Box Systems
written by Yan Zhang
has been approved for the Department of Electrical, Computer, and Energy Engineering

Sriram Sankaranarayanan

Fabio Somenzi

Date _____

The final copy of this thesis has been examined by the signatories, and we find that both the content and the form meet acceptable presentation standards of scholarly work in the above mentioned discipline.

Zhang, Yan (Ph.D., Electrical Engineering)

Statistically Sound Verification and Optimization of Black-Box Systems

Thesis directed by Prof. Sriram Sankaranarayanan

This thesis discusses two important problems for the design of practical systems under stochastic parameter variations: verification and optimization. Verification is concerned with the safety of a system, i.e., whether a system satisfies its specifications. If not, optimization is applied to tune the design parameters in the system so that the new design is safe. This thesis treats systems as black-boxes, assuming that the systems can be simulated efficiently but without detailed knowledge of the internal workings. It presents a series of simulation-based techniques to solve the problems of design verification and optimization. A notion called statistical soundness is introduced in this thesis, which guarantees that the outcome of the proposed techniques are “statistically certified” in the sense that the probability of drawing a wrong conclusion is bounded. For the problem of verification, this thesis develops a **statistically sound model inference (SSMI)** approach. SSMI constructs statistically sound models to explain the relationship between the stochastic parameters and the responses of a system. To improve the scalability of SSMI, a sparse approximation algorithm is also introduced. For the problem of optimization, this thesis presents a statistically sound optimization technique, SSMI-opt. SSMI-opt aims to find values of the design parameters for which the system satisfies the specifications. The proposed techniques can be applied to many interesting areas, including analog/mixed-signal circuits, embedded systems, biological systems, and medical devices. This thesis demonstrates the utility of this methodology on several interesting benchmark examples.

Dedication

To my wife and my parents.

Acknowledgements

I would like to thank Prof. Sriram Sankaranarayanan and Prof. Fabio Somenzi. As a Ph.D. student, the experience in the past five years is life-changing. Both of you have taught me what it takes to become an researcher and provided valuable advices on my research. Without you, it would be impossible to complete this thesis. Also, I would like to say sorry to Sriram for my consistently poor pronunciation of his last name.

I would like to thank my committee members, Prof. Alireza Doostan, Prof. Dragan Maksimovic, and Prof. Chris Myers. Your comments and suggestions have broaden my knowledge and made the thesis a better work.

I would like to thank my wife and my parents. You have provided enormous supports in life and in spirit. I would like to thank all my colleagues, especially Zyad Hassan, Erwin Dunbar, Arlen Cox, Saqib Sohail and Michael Dooley. You have made the lab a wonderful place to work, to exchange ideas, and to chat for fun.

Finally, I would like to thank the US National Science Foundation (NSF) under the award numbers SHF-1016994 and SHF-1320069. All opinions expressed are those of the author, and not necessarily of the NSF.

Contents

Chapter

1	Introduction	1
1.1	Background and Motivation	1
1.2	Contributions of this Thesis	3
1.3	Limitations of Symbolic Techniques	4
1.4	Related Work	7
1.4.1	Monte-Carlo Simulation	7
1.4.2	Statistical Model Checking	8
1.4.3	Uncertainty Quantification	9
1.4.4	Other Simulation-Based Methods	11
1.5	Organization of this Thesis	11
2	Background	12
2.1	Sequential Hypothesis Testing	12
2.1.1	Sequential Probability Ratio Test	14
2.1.2	Sequential Bayesian Test	18
2.2	Statistical Model Checking	23
2.2.1	Bounded Linear Temporal Logic	25
2.2.2	Bayesian Statistical Model Checking	27
2.3	Regression	29

2.3.1	Ordinary Least Squares	29
2.3.2	Regularization	30
3	Statistical Soundness	33
3.1	Black-Box Systems and Specifications	34
3.1.1	Black-Box Systems	34
3.1.2	Design and Stochastic Parameters	35
3.1.3	Response Specifications	36
3.2	Statistical Soundness	38
3.3	Statistically Sound Yield Computation	42
3.4	Comparison with Statistical Model Checking	48
4	Statistically Sound Model Inference	53
4.1	Overview	54
4.1.1	Regression	57
4.1.2	Generalization	58
4.2	Ordinary Least Squares Regression	59
4.2.1	A Resampling Heuristic	60
4.2.2	Complexity	62
4.3	Generalization	63
4.3.1	Tolerance Interval	63
4.3.2	Algorithms for the Derivation of Tolerance Intervals	64
4.3.3	Complexity	70
4.4	Applications	70
4.4.1	Motor Controller	71
4.4.2	Low-Pass Filter	73
4.4.3	Buck Converter	76
4.5	Summary	78

5	A Sparse Approximation Method	80
5.1	Generalized Polynomial Chaos	81
5.1.1	Orthogonal Polynomials	82
5.1.2	Orthogonal Projection	83
5.1.3	Generalized Polynomial Chaos	85
5.2	A Low-Degree Approximation Algorithm	86
5.2.1	Overview	86
5.2.2	Choosing a Subset of Basis Functions via gPC	89
5.2.3	Computing Unknown Coefficients	90
5.3	Discussion of the Algorithm	91
5.4	Applications	92
5.4.1	Randomly Generated Sparse Polynomials	93
5.4.2	Ring Oscillator	93
5.4.3	Digital-to-Analog Converter	97
5.4.4	Low-Pass Filter	100
5.5	Summary	101
6	Statistically Sound Optimization	102
6.1	Overview	103
6.2	Quantile Regression	108
6.3	An Iterative Optimization Algorithm	111
6.3.1	Generalization of Relational Models	111
6.3.2	Optimization	114
6.4	Applications	115
6.4.1	Ring Oscillator	116
6.4.2	Insulin Pump	118
6.4.3	Aircraft Flight Control System	119

6.5	Summary	121
7	Conclusion	123
7.1	Summary of this Thesis	123
7.2	Future Work	124
7.2.1	Statistically Sound Model Inference	124
7.2.2	Sparse Approximation	125
7.2.3	Statistically Sound Optimization	125
7.3	Combining Statistical and Symbolic Techniques	125
	Bibliography	127

Tables

Table

2.1	Interpretation of Bayes factor.	20
4.1	Run length K for common values of θ_0 and T	68
4.2	Stochastic parameters in the motor plant.	72
4.3	Verification results of the motor controller.	73
4.4	Verification results of the low-pass filter.	75
4.5	Stochastic parameters of the transistor S_n in the buck converter.	77
4.6	Verification results of the buck converter.	78
5.1	Random variables and orthogonal polynomials.	85
5.2	Stochastic parameters of an NMOS transistor in the ring oscillator.	96
5.3	Verification results of the ring oscillator.	97
5.4	Stochastic parameters of an NMOS transistor in the DAC.	98
5.5	Verification results of the DAC.	99
5.6	Verification results of the low-pass filter.	100
6.1	Optimization results for the three-stage ring oscillator.	117
6.2	Optimization results for the insulin pump model.	120
6.3	Optimization results for the aircraft flight control model.	121

Figures

Figure

1.1	A three-stage ring oscillator verified by a symbolic technique.	6
2.1	Strengths of SPRT without and with indifference region.	17
2.2	Comparison between ridge regression and LASSO.	32
3.1	A two-mass-spring system and the closed-loop system with a controller.	41
3.2	Sound models versus statistically sound models.	44
3.3	Statistically sound model, response surface and specification.	45
4.1	A basic buck converter with two stochastic parameters.	55
4.2	A high-level flow of SSML.	56
4.3	A comparison between the safe regions of the basic buck converter.	59
4.4	Snapshots of the interval I during generalization of the basic buck converter.	70
4.5	A motor with a PI controller and its response specifications.	71
4.6	Safe regions of the motor controller.	74
4.7	An analog low-pass filter.	74
4.8	A buck converter with realistic switches and control logic.	76
4.9	Safe region of the buck converter.	78
5.1	An interpretation of orthogonal projection.	84
5.2	Comparisons between the proposed sparse approximation approach and LASSO	94

5.3	A three-stage ring oscillator.	95
5.4	An eight-bit digital-to-analog converter.	97
6.1	A two-mass-spring system and the closed-loop system with a controller.	104
6.2	A high-level flow of SSMI-opt.	106
6.3	Histograms of the responses in the two-mass-spring system.	107
6.4	Comparison of penalty functions.	109
6.5	A three-stage ring oscillator.	116
6.6	Histograms of the responses in the ring oscillator.	117
6.7	A model of an insulin pump and histograms of its response.	118
6.8	An aircraft flight control model.	120
6.9	Histograms of the responses in the aircraft flight control model.	122

Chapter 1

Introduction

1.1 Background and Motivation

Model-based design (MBD) has become an increasingly popular approach for the design of large systems with complicated dynamics. In the domain of analog/mixed-signal (AMS) circuits, modeling languages such as Verilog-AMS are used to support the simulation-based design and verification process. For embedded systems, modeling a system with Simulink/Stateflow has been a standard step in the design flow. A system in MBD often involves two types of parameters: **design parameters** and **stochastic parameters**. Design parameters are controllable, i.e., can be assigned to certain values by designers. Examples of design parameters include the channel width and length of a CMOS transistor in an analog circuit, the gains in a PID controller, and so on. On the other hand, stochastic parameters are uncontrollable and arise from the randomness in the process of manufacturing, the environment and many other aspects. The exact values of stochastic parameters usually vary in different instances of a system. Hence, it is common to assume that they follow certain statistical distributions.

The design of practical systems often involves the following theme. Based on previous knowledge, the stochastic parameters are assumed to have fixed values, which are called the nominal values. The design parameters are then tuned with respect to the nominal values of the stochastic parameters so that the system can meet the desired performance requirements. Such a design is known as the **nominal design**. However, a nominal design may suffer from stochastic parameter variations, such as process variations or variations in the environment. Although the nominal de-

sign satisfies the performance requirements when the stochastic parameters are fixed to the nominal values, there is no guarantee that an actual system with stochastic parameters that are different than the nominal values also does so.

To design systems that are robust under stochastic parameter variations, verification and optimization play important roles. Verification focuses on checking whether a system meets all the performance requirements. For example, consider a ring oscillator (see Section 5.4 for an example) which is an analog circuit that produces a voltage oscillating at a fixed frequency. The circuit contains design parameters such as the channel width and length of CMOS transistors, and stochastic parameters such as the gate oxide thickness, doping concentrations, and so on. It is desirable to show that the oscillation frequency is within a certain range for some given values of design parameters under stochastic parameter variations. If the system fails to do so, a designer must tune the design parameters so that the performance requirements are satisfied as much as possible. This process is called design optimization. The two techniques presented in this thesis aim to guarantee that a system satisfies the performance requirements not only for the nominal values of the stochastic parameters, but also for a large proportion of the possible variations.

As systems become larger, it is often difficult to reason about their behaviors in a symbolic way. Such a system is often regarded as a black-box, for which a designer can simulate it to obtain values of the output with given input values, but do not need to know information about its internal workings. In this thesis, the input of a system refers to the design and the stochastic parameters and the output refers to the responses of the system, such as the oscillation frequency in a ring oscillator. The relationship between the input and output is called the response surface. For a black-box system, the response surface can rarely be written in a closed-form and is only computable through simulation of the system. Hence, in most cases, simulation is the only effective method to learn how the responses are affected by the design and the stochastic parameters. As a consequence, it is important to develop simulation-based techniques for the analysis of black-box systems.

A key problem of simulation-based techniques lies in their lack of coverage. Since these

techniques do not fully explore the stochastic parameter space, any conclusion drawn about the behavior of the system is not guaranteed to be true. This thesis provides statistical guarantees by introducing a notion called statistical soundness and developing a series of statistically sound techniques. Informally, a statistically sound model is a model that with a large probability, it over-approximates the behavior of a system. This thesis will show how to construct such models and how to use them to aid the verification and optimization of black-box systems. The proposed techniques can be applied to many interesting areas, including AMS circuits, embedded systems, biological systems, and medical devices.

1.2 Contributions of this Thesis

The main contributions of this thesis are as follows:

- Chapter 4 introduces **statistically sound model inference (SSMI)**, a technique for modeling and verification of black-box systems. SSMI combines regression techniques and **statistical model checking (SMC)** [43] to provide models that explain how the stochastic parameters affects the responses of a system in a statistically sound manner. Such a model statistically over-approximates the response surface of interest and is used to verify the specification in regards to the response. The outcome of verification is a yield computed with respect to the model, which is shown to form a lower bound of the true yield. The yield shows not only whether the specification is satisfied, but also the the probability that it is satisfied. In addition, the model can generate plots of safe parameter regions. Section 4.4 shows a couple of such plots in terms of interesting stochastic parameters. This work is originally published in ICCAD 2013 [97].
- Chapter 5 presents a sparse approximation algorithm that aims to extend the ability of SSMI to handle systems with many stochastic parameters. Compared to classic sparse approximation algorithms, such as least absolute shrinkage and selection operator (LASSO) [80] and basis pursuit [18], the algorithm has two salient features. First, it combines **general-**

ized polynomial chaos (gPC) [90], an uncertain quantification technique, and LASSO, a sparse approximation algorithm that is widely used in the area of compressed sensing. The resulting algorithm is more efficient than LASSO alone. Second, the algorithm produces polynomial approximations of degrees as low as possible. This method is useful in practice since lower-degree polynomials are generally preferred over higher-degree ones. This work is originally published in ASPDAC 2014 [99].

- Chapter 6 proposes a statistically sound design optimization technique, SSMI-opt. For a black-box system, SSMI-opt aims to find values of the design parameters such that the system robustly satisfies the specifications under the stochastic parameter variations. As the name suggests, it borrows the idea of SSML. Instead of constructing models in terms of the stochastic parameters, SSMI-opt “marginalizes” the effects of the stochastic parameters and approximates the lower and the upper bound of a response as a function of the design parameters. Such a model is obtained using **quantile regression** [50], a regression technique that estimates a certain quantile of response variables, and a generalization procedure introduced in SSML. This technique is used to find values of the design parameters that lead to a statistically certified safe system. This work is published in ATVA 2014 [98].

1.3 Limitations of Symbolic Techniques

Those who are familiar with symbolic verification techniques may wonder why one should consider statistical instead of symbolic techniques, given that the latter can provide formal guarantees on the verification results. This section provides an overview the state-of-the-art symbolic techniques that are applied to continuous/hybrid systems. Also, it contains a brief introduction to our previous work on symbolic verification, which demonstrates the poor scalability of many symbolic techniques.

Given a system and a property of interest, the verification problem decides whether or not the system satisfies the property. Model checking techniques are a general class of algorithms to solve

the verification problem. Various model checking algorithms use approaches such as exhaustive search and symbolic exploration. Symbolic model checking techniques solve the model checking problem via symbolically reasoning about the behavior of a system. Such a technique encodes the states and the transition relation of a system into symbolic representations, such as BDDs [61] or SAT formulas [12], and considers a large number of states at a single step. These techniques have been widely used in the verification of digital circuits and achieved great successes. The applications to continuous/hybrid systems dates back to the work of Kurshan and McMillan [52] and Hedrich and Barke [37] on the verification of analog circuits. The main theme of symbolic model checking involves performing reachability analysis [32, 57, 84, 82, 3, 92, 91], which explores the reachable state-space of a system. In order to apply symbolic techniques to the verification of a continuous/hybrid system, it requires detailed models which models the internal workings of the system. However, obtaining such a model can be challenging in many practical cases. Little et al. [56] introduce a technique which generates labeled hybrid Petri Net using simulation data to model the behavior of a system. This work is extended by Batchu [6] and Kulkarni [51]. Tiwary et al. [82] propose a piecewise interval approach to model the dynamics of non-linear analog devices, such as CMOS transistors and diodes. Zhang et al. [96] further extend the idea of Tiwary et al. [82] and consider the use of piecewise linear functions.

A key shortcoming of existing symbolic approaches lies in their scalability. Usually they can only handle small systems with relatively simple dynamics. For example, for the verification of transistor-level analog circuits, the capability of most existing symbolic techniques is restricted to the order of ten transistors with simple device models, such as the Schichman-Hodges models. Our previous work [96] illustrates an experimental study on the performance of a representative symbolic technique, which provides evidence for the poor scalability of many symbolic techniques. The goal of the experimental study is to analyze the performance of satisfiability-modulo-theory (SMT) solvers to verify transistor-level analog circuits with conservative piecewise linear approximations of non-linear circuit elements such as diodes and transistors. To obtain conservative approximations, the characteristics of non-linear devices are piecewise linearized. The piecewise linear function is then

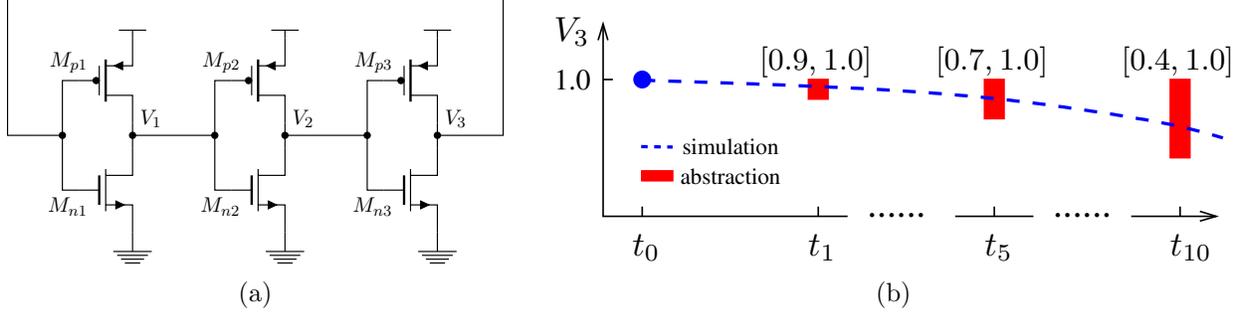


Figure 1.1: A three-stage ring oscillator (a) and the predicted reachable region of V_3 by the approach in [96] (b). The time step $\Delta t = t_i - t_{i-1}$ is small enough to ignore integration error.

“generalized” into a relational model which encloses the original device model.

A bounded model checking approach is used to perform the verification [13]. Since an analog circuit is a continuous system, time is discretized into fixed-step points at which the states of the system is expressed using the piecewise linearized dynamics. A transition relation, which shows how the state variables change from one state to another, is defined as the integration rules that relate the current state variables to the variables at the next time point. To verify whether a property is satisfied, the piecewise linear dynamics, together with the transition relation, are “unrolled” at a sequence of time points. The unrolling is encoded as an SMT formula and solved by SMT solvers.

The above approach is used to verify the reachability of a ring oscillator, which is shown in Figure 1.1a. The time step is chosen so that it is sufficiently small to ignore the integration error. The result is shown in Figure 1.1b. It illustrates that the reachable region quickly becomes too conservative to provide any meaningful reachability information. Our previous work [96] also shows that using a more fine-grained piecewise linearized model provides little improvement on the result but requires more solving time. This work demonstrates that although symbolic techniques can provide formal guarantees, it is a challenging task to apply them to even small continuous systems.

1.4 Related Work

1.4.1 Monte-Carlo Simulation

Conventionally, the safety of a black-box system is verified by **Monte-Carlo simulation**. A common theme involves random sampling of the stochastic parameters and simulating the system accordingly. The yield is estimated by the proportion of data points that satisfy the specifications. In the limiting case where the sample size is infinite, Monte-Carlo estimation is always accurate. In practice, however, it is well-known that they suffer from a slow convergence rate of $O\left(\frac{1}{\sqrt{N}}\right)$, where N is the sample size [71]. In other words, to improve the accuracy of an estimation by an order of 10, it requires 100 times more simulations. For large systems, running so many simulations can be prohibitively expensive.

To have a better convergence rate, **quasi Monte-Carlo (QMC)** methods are worth mentioning (see, e.g., Singhee and Rutenbar [76]). Essentially, these methods construct deterministic sequences for points in the stochastic parameter space rather than doing random sampling. Such a sequence is called a **low discrepancy sequence (LDS)** [78] and consists of points distributed “uniformly” in the parameter space. It guarantees that the parameter space is evenly explored. QMC methods have an empirical convergence rate of $O\left(\frac{1}{N}\right)$, which is better than the Monte-Carlo methods. However, to achieve this rate, the number of stochastic parameters should not be too large.

For many practical systems, an important problem is to detect rare events, i.e., events that happen in a very low probability. A classic approach for this problem is **importance sampling** [31], which is a modification of the standard Monte-Carlo methods. Importance sampling changes the distribution with respect to which sampling is performed, and allows Monte-Carlo methods to avoid those unimportant data points (i.e., points that are unlikely to result in rare events). Another interesting approach is developed by Singhee and Rutenbar [77], which exploits ideas from extreme value theory [62] and **support-vector machine (SVM)** classifiers. Unlike importance sampling, this approach does not modify the underlying distribution, and thus can be used to gather statistics

of the rare events.

Compared to Monte-Carlo techniques, the proposed approaches in this thesis aim to derive conservative yield estimations using less simulations. Chapter 3 and Chapter 4 show that by introducing the notion of statistical soundness, one can effectively provide lower bound on the yield of a system with less computational cost than conventional Monte-Carlo techniques.

1.4.2 Statistical Model Checking

As systems grow larger, symbolic techniques become out of reach. In recent years, researchers have been seeking statistical solutions to verify complex systems. Such techniques often rely on repeated simulations, enriched with statistical inference techniques, to provide statistical guarantees on the behavior of the systems. The seminal work by Younes and Simmons [95] initiates the research area called **statistical model checking (SMC)**. In their work, the model checking of stochastic systems is regarded as a hypothesis testing problem and solved using **sequential probability ratio test (SPRT)** [83]. Later, Sen et al. [74] propose a p -value significance test for the verification of black-box systems. Hérault et al. [39] introduce an approach that approximates the satisfaction probability of some probabilistic properties. They use a single sampling plan, which fixed the sample size upfront, and estimate the satisfaction probability by the proportion of the data points that satisfies the property. Their technique is known as **approximate model checking**. Jha et al. [43] introduce a Bayesian SMC framework based on the **sequential Bayesian test** [40, 47]. Compared with SPRT, the sequential Bayesian test is both practically and theoretically more convenient since it does not require indifference regions as SPRT does. Instead, it computes Bayes factors by integrating over a given prior density. Bayesian SMC has also been applied to the verification of analog circuits [85], medical devices [44], and embedded systems [100]. An introduction to SMC can be found in Chapter 2.

A main drawback of SMC techniques is that they are designed to answer “likely yes/no” questions. In many cases, it may be desirable to understand how the stochastic parameters affect the responses in a system. This thesis provides techniques to solve this problem. The proposed

techniques (see Chapter 4 and Chapter 5) construct statistical models that explain the relationship between stochastic parameters and responses. Their applications are not restricted to showing whether a system satisfies the specifications. It is also possible to derive regions of the stochastic parameter space that are safe with respect to the specifications.

SMC techniques have also been applied to the optimization of black-box systems. Jha et al. [44] present the use of SMC to tune parameters for closed loop controller models in order to satisfy a given set of temporal logic specifications. 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 introduced by Palaniappan et al. [68] to fit parameter values for biological models based on experimental observations, as well as, model specifications. 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.

In contrast to the work by Jha et al. [44] and Palaniappan et al. [68], our approach (see Chapter 6) on design optimization, SSMI-opt, is more straightforward. Our approach employs only linear programs and is therefore computationally inexpensive. Moreover, SSMI-opt builds models that characterize the behavior of a system, which can be reused in different design phases.

While this thesis considers the design parameters as controllable, a significant body of work treats problems involving uncontrollable non-deterministic parameters along with stochastic parameters in the context of SMC. Recent papers by Henriques et al. [38] and Ellen et al. [28] use reinforcement learning techniques to verify the correctness properties under the worst-case values of the non-deterministic parameters.

1.4.3 Uncertainty Quantification

Uncertainty quantification (UQ) is an emerging area that studies how to characterize uncertainties in a system and their effects on the responses of the system. Conventionally, Monte-Carlo methods have been the main approach for UQ. In recent years, alternative approaches,

such as stochastic Galerkin schemes based on polynomial chaos expansion [23, 4, 60, 90, 79] and stochastic collocation schemes [5, 67, 89, 72, 59], have been proposed. Stochastic Galerkin methods transform a stochastic system into a deterministic system in which the stochastic parameters are substituted by a finite polynomial chaos expansion. They are often used in an intrusive manner, i.e., modify simulators so that the transformed deterministic system can be simulated directly from the description of the original system. On the other hand, stochastic collocation methods are non-intrusive. They rely on the legacy code of simulators and perform computations using simulation data from sparse grids or other quadrature rules. Compared to Monte-Carlo methods, these approaches are more effective in modeling and propagating uncertainties.

Challenges arise when the system has a high-dimensional stochastic parameter space. In this case, both stochastic Galerkin and stochastic collocation methods become inefficient. Many efforts have been spent to solve this problem. Li et al. [55] introduce an approach based on reduced rank regression. They use quadratic polynomials to model nonlinear response surfaces. A similar idea is demonstrated by Feng and Li [30] to handle the problem of interconnect modeling of integrated circuits. Singhee and Rutenbar [75] develop a nonlinear regression approach based on latent variable regression and neural networks. Doostan and Iaccarino [25] propose to decompose a high-dimensional response surface into a summation over products of univariate functions. Doostan and Owhadi [26] introduce a non-intrusive sparse approximation method based on Legendre polynomials and \mathcal{L}_1 minimization. Li [54] use matching pursuit to find the “best” projection of the response surface onto an orthogonal polynomial basis.

Our work, especially the sparse approximation algorithm presented in Chapter 5, have brought a lot of ideas from UQ. However, models from UQ techniques do not provide guarantees that they lead to correct conclusions, e.g., on the safety of a system. This thesis combines the strength of UQ techniques in model building with that of SMC techniques in providing statistical guarantees, and develop the proposed approaches in this thesis.

1.4.4 Other Simulation-Based Methods

A few other interesting techniques originates from the hardware testing community. Yoon et al. [93] propose a hierarchical model inference approach to derive statistical distributions of circuit properties. Dang and Nahhal [22] use motion planning techniques for **rapidly-exploring random trees (RRTs)** to verify specifications of analog circuits. Ahmadyan et al. [2] also use RRTs to generate property-oriented test cases for analog circuits.

1.5 Organization of this Thesis

This thesis is organized as follows. The next chapter presents background knowledge that is used extensively. It includes an introduction to sequential hypothesis testing, statistical model checking and regression techniques. Chapter 3 develops the notion of statistical soundness. It serves as the foundation of the work in this thesis. Chapter 4 introduces statistically sound model inference and shows how it is applied to the verification of black-box systems. Chapter 5 discusses a sparse approximation algorithm that combines generalized polynomial chaos and \mathcal{L}_1 minimization. The algorithm is demonstrated in the context of SSML. Chapter 6 presents a design optimization technique for systems suffering from stochastic parameter variations. The final chapter summarizes this thesis and points out some directions for future research.

Chapter 2

Background

This chapter introduces background knowledge that is used extensively throughout the thesis. It is divided into three parts. First, it reviews statistical hypothesis testing techniques. Next, this chapter discusses statistical model checking, which is based on sequential hypothesis testing. Finally, it presents a short discussion on regression algorithms. The notations appearing in this chapter are used consistently in the rest parts of the thesis. The presentation is not meant to be exhaustive. Readers who are interested to learn more about the topics should refer to the references mentioned in the text.

2.1 Sequential Hypothesis Testing

A hypothesis is a statement about an unknown population parameter. For instance, the mean lifetime of rabbits are greater than 6 years, or the yield of a production line is no less than 90%. Usually, we have a pair of hypotheses, \mathcal{H}_1 versus \mathcal{H}_2 . To learn whether a hypothesis is true, one takes a set of observations from the population and uses a technique called hypothesis testing. Hypothesis testing is a statistical decision procedure that decides [17]:

- For which observations the hypothesis \mathcal{H}_1 should be accepted to be true;
- For which observations the hypothesis \mathcal{H}_2 should be accepted to be true.

Examples of classical hypothesis testing techniques include the likelihood ratio test (LRT) and the p -value significance test. Since a test is conducted with finite observations from the population,

it is unavoidable that the conclusion can sometimes be incorrect. The probability of accepting \mathcal{H}_2 when \mathcal{H}_1 is true is known as the Type I error α . Similarly, the probability of accepting \mathcal{H}_1 when \mathcal{H}_2 is true is known as the Type II error β . The pair (α, β) indicates the **strength** of the test. For any hypothesis testing technique, it is important to be able to bound the Type I/II error. Conventionally, this is achieved by selecting a proper test statistic and then fixing a sample size N based on the test (see Casella and Berger [17, Ch. 8] for details).

Sequential hypothesis testing, a.k.a. sequential analysis, is a form of hypothesis testing where the sample size is not fixed in advance. Instead, it evaluates the observed data sequentially until a decision can be made by satisfying some predefined stopping criterion. Compared with the classical approaches, sequential hypothesis testing often reaches a conclusion at a much earlier stage of the inference process, thus saving the effort of performing more experiments.

Like other hypothesis testing approaches, sequential hypothesis testing requires a pair of hypotheses \mathcal{H}_1 and \mathcal{H}_2 concerned with some unknown population parameter θ . The form of a hypothesis depends on specific applications. This thesis focuses on the following form: $\mathcal{H}_1 : \theta \geq \theta_0$ versus $\mathcal{H}_2 : \theta < \theta_0$, where θ is the unknown probability that a system satisfies a specification, and $\theta_0 \in [0, 1]$ is a probability that θ is desired to exceed. Sequential hypothesis testing draws observations in sequence and for each observation, it conducts one of the following actions:

- Accept \mathcal{H}_1 to be true;
- Accept \mathcal{H}_2 to be true;
- Draw another observation and continue testing without making any conclusion.

The process terminates as soon as either the first or the second action is taken, regardless how much data have been collected. In practice, the number of observations is much less than that in a hypothesis testing with predefined sample size.

Note that the names of the hypotheses are \mathcal{H}_1 and \mathcal{H}_2 instead of the conventional null hypothesis \mathcal{H}_0 and alternative hypothesis \mathcal{H}_2 . This may initially be confusing to those who are familiar with hypothesis testing. We make this choice because for the problem that we are interested

in, it is more nature to think of accepting a “good” hypothesis rather than rejecting a “bad” one. Therefore, we try to avoid the name “null hypothesis” and the associated symbol \mathcal{H}_0 .

The following parts introduce two sequential hypothesis testing approaches: sequential probability ratio test (SPRT) developed by Wald [83] in Section 2.1.1, and Bayesian sequential hypothesis testing formulated by Jeffreys [40, 41] in Section 2.1.2.

2.1.1 Sequential Probability Ratio Test

Consider a pair of hypotheses, $\mathcal{H}_1 : \theta \geq \theta_0$ versus $\mathcal{H}_2 : \theta < \theta_0$. Let Z be a Bernoulli random variable with a probability mass function (pmf)

$$f_Z(z | \theta) = \theta^z(1 - \theta)^{1-z}, \quad z \in \{0, 1\}. \quad (2.1)$$

Assume that the sequence of observations $D = (z_1, \dots, z_m)$ are independent and identically distributed (i.i.d.). SPRT computes the probability ratio $\frac{p_{1m}}{p_{2m}}$ where $p_{im} = \Pr(D | \mathcal{H}_i)$ is the probability of D when \mathcal{H}_i is assumed to be true. It defines the following rule:

- Accept \mathcal{H}_1 to be true if $\frac{p_{1m}}{p_{2m}} \geq A$;
- Accept \mathcal{H}_2 to be true if $\frac{p_{1m}}{p_{2m}} \leq B$;
- Draw another observation and continue testing if $B < \frac{p_{1m}}{p_{2m}} < A$.

Example 2.1.1 (Tossing A Biased Coin). Suppose that we have a biased coin which prefers one side over the other. It has an unknown probability θ of getting heads. To see whether $\theta \geq 0.7$, we need to construct a pair of hypothesis $\mathcal{H}_1 : \theta \geq 0.7$ versus $\mathcal{H}_2 : \theta < 0.7$. We define the outcome of a toss experiment as a Bernoulli random variable Z ,

$$Z = 1, \quad \text{if the outcome is a head,}$$

$$Z = 0, \quad \text{otherwise.}$$

To decide which hypothesis should be accepted, we make a sequence of observations $D = (z_1, \dots, z_m)$ and compute the probability ratio as discussed in the following. ||

2.1.1.1 Computation of the Probability Ratio

A simple hypothesis, which involves only a single point in the parameter space, has the form $\mathcal{H} : \theta = \theta_0$. For a pair of simple hypotheses $\mathcal{H}_1 : \theta = \theta_1$ versus $\mathcal{H}_2 : \theta = \theta_2$, SPRT computes the probability ratio as follows.

$$\frac{p_{1m}}{p_{2m}} = \frac{\Pr(D | \theta = \theta_1)}{\Pr(D | \theta = \theta_2)} = \prod_{i=1}^m \frac{\Pr(z_i | \theta = \theta_1)}{\Pr(z_i | \theta = \theta_2)}, \quad (2.2)$$

Now consider the pair of hypotheses, $\mathcal{H}_1 : \theta \geq \theta_0$ versus $\mathcal{H}_2 : \theta < \theta_0$. Clearly, the formulation in (2.2) does not work if $\theta_1 = \theta_2 = \theta_0$. Wald [83] proposed to relax the original hypotheses such that they became

$$\mathcal{H}_1 : \theta \geq \theta_1 \text{ versus } \mathcal{H}_2 : \theta \leq \theta_2, \quad (2.3)$$

where $\theta_1 = \theta_0 + \delta$, $\theta_2 = \theta_0 - \delta$ and δ is a positive number. The interval (θ_2, θ_1) is called the **indifference region** for that if $\theta \in (\theta_2, \theta_1)$, it makes no difference in which hypothesis is accepted. We say that δ is the **half-width** of the indifference region. The probability ratio (2.3) of the hypotheses is then computed by (2.2).

It may not be immediately clear that the ratio (2.2) formulates a test on the relaxed hypotheses (2.3), given that (2.3) concerns with a range of parameter values rather than single points. However, notice that for a given probability p

$$\Pr(z_i | \theta = p) = p^{z_i} (1 - p)^{1 - z_i},$$

since the observation $z_i \in \{0, 1\}$ is taken from the pmf (2.1). Therefore, the ratio

$$\frac{\Pr(z_i | \theta = p_1)}{\Pr(z_i | \theta = p_2)} = \frac{p_1^{z_i} (1 - p_1)^{1 - z_i}}{p_2^{z_i} (1 - p_2)^{1 - z_i}}, \quad p_1 \in [\theta_1, 1], \quad p_2 \in [0, \theta_2]$$

has a minimum value at $p_1 = \theta_1$ and $p_2 = \theta_2$ if $z_i = 1$, and a maximum value if $z_i = 0$. When SPRT terminates, if \mathcal{H}_1 is true, the ratio (2.2) has the smallest value among $p_1 \in [\theta_1, 1]$ and $p_2 \in [0, \theta_2]$. Similarly, if \mathcal{H}_2 is true, (2.2) has the largest value. Choosing any other values for p_1 and p_2 would have lead to the same conclusion. Therefore, (2.2) is used to compute the probability ratio of the relaxed hypotheses (2.3).

Example 2.1.2 (Tossing A Biased Coin - Computation). Continue with the example 2.1.1. Let $A = 100$, $B = 0.01$ and $\delta = 0.05$. Hence, $\theta_1 = 0.75$ and $\theta_2 = 0.65$. Suppose that we draw observations in sequence and at some point, we have collected total of $m = 40$ observations without leading to a conclusion. The probability ratio (2.2) becomes

$$\frac{p_{1m}}{p_{2m}} = \left(\frac{0.75}{0.65}\right)^{m_1} \left(\frac{0.25}{0.35}\right)^{m_2}, \quad m_1 + m_2 = m,$$

where m_1 and m_2 are the number of times that we see heads and tails, respectively. The following cases illustrate the rule of SPRT.

- If $m_1 = 38$ and $m_2 = 2$, the ratio $\frac{p_{1m}}{p_{2m}} \approx 117$ is greater than A . Thus we accept the relaxed hypothesis $\mathcal{H}_1 : \theta \geq 0.75$;
- If $m_1 = 18$ and $m_2 = 22$, the ratio $\frac{p_{1m}}{p_{2m}} \approx 0.008$ is less than B . Thus we accept the relaxed hypothesis $\mathcal{H}_2 : \theta < 0.65$.
- If $m_1 = 30$ and $m_2 = 10$, the ratio $\frac{p_{1m}}{p_{2m}} \approx 2.5$. We need more observations to conclude.

Note that the conclusion is with respect to the relax hypotheses. Interested readers can verify that the smaller δ is, the larger m is required to draw conclusions. ||

2.1.1.2 Strength of SPRT

Figure 2.1a plots the probability of accepting \mathcal{H}_1 as a function of θ for an SPRT with an ideal strength, i.e., the Type I and the Type II error are exactly α and β . To find a test with a desired strength (α, β) , one needs to establish the relationship between (α, β) and (A, B) . Wald [83] proved that to for fixed A and B , the strength (α, β) satisfies

$$\frac{\beta}{1 - \alpha} \leq \frac{1}{A} \quad \text{and} \quad \frac{\alpha}{1 - \beta} \leq B. \quad (2.4)$$

The inequalities (2.4) provides upper bounds on α and β once A and B are chosen. It is non-trivial to determine the exact strength of a test. It can be shown that setting $A = \frac{1 - \alpha}{\beta}$ and $B = \frac{\alpha}{1 - \beta}$ yields a test that has a strength very close to (α, β) . In practice, this actual strength is often better than (α, β) . As a consequence, the rule becomes

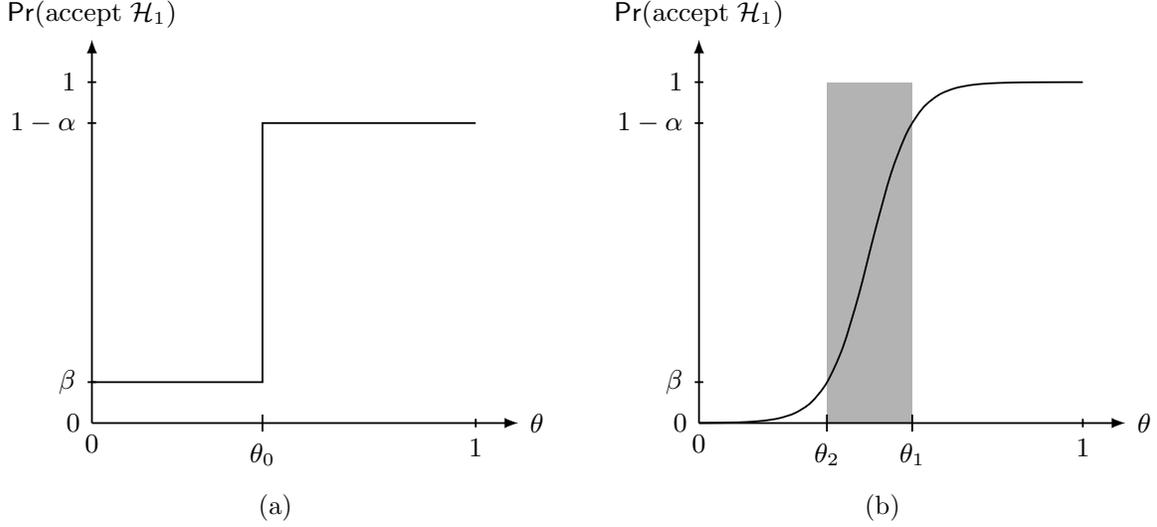


Figure 2.1: Probability of accepting $\mathcal{H}_1 : \theta \geq \theta_0$ over $\mathcal{H}_2 : \theta < \theta_0$ as a function of θ (left) and probability of accepting $\mathcal{H}_1 : \theta \geq \theta_1$ over $\mathcal{H}_2 : \theta \geq \theta_2$ (right). (adapted from Younes [94].)

- Accept \mathcal{H}_1 to be true if $\frac{p_{1m}}{p_{2m}} \geq \frac{1 - \alpha}{\beta}$;
- Accept \mathcal{H}_2 to be true if $\frac{p_{1m}}{p_{2m}} \leq \frac{\alpha}{1 - \beta}$;
- Draw another observation and continue testing if $\frac{\alpha}{1 - \beta} < \frac{p_{1m}}{p_{2m}} < \frac{1 - \alpha}{\beta}$.

We provide an intuitive explanation on the inequalities (2.4). Let Q_1 and Q_2 be the events that \mathcal{H}_1 and \mathcal{H}_2 are accepted, respectively. Clearly,

$$\Pr_i(Q_1) + \Pr_i(Q_2) = 1, \quad i = 1, 2, \quad (2.5)$$

where \Pr_i is the probability under the assumption that \mathcal{H}_i is true. Notice that $\Pr_1(Q_2) = \alpha$ and $\Pr_2(Q_1) = \beta$. For an arbitrary sequence of observations S_∞ with infinite length, the probability that one of the hypotheses is accepted at a finite length is 1. Let m denote the length. According to the rule,

$$\begin{aligned} p_{1m} &\geq A p_{2m}, & \text{if } \mathcal{H}_1 \text{ shall be accepted,} \\ p_{1m} &\leq B p_{2m}, & \text{if } \mathcal{H}_2 \text{ shall be accepted.} \end{aligned} \quad (2.6)$$

Since S_∞ is arbitrary,

$$\begin{aligned} p_{1m} &= \Pr_1(Q_1) = 1 - \alpha, \quad p_{2m} = \Pr_2(Q_1) = \beta, \quad \text{if } \mathcal{H}_1 \text{ shall be accepted,} \\ p_{1m} &= \Pr_1(Q_2) = \alpha, \quad p_{2m} = \Pr_2(Q_2) = 1 - \beta, \quad \text{if } \mathcal{H}_2 \text{ shall be accepted.} \end{aligned} \tag{2.7}$$

Combining (2.5), (2.6) and (2.7), we have shown the inequalities (2.4).

Remark 1. In general, the performance of a sequential analysis (including SPRT as well as other sequential techniques) is characterized by the number of required observations, which degrades when θ approaches θ_0 . Consider the two cases, $\theta = \theta_0 + \epsilon$ and $\theta = \theta_0 - \epsilon$ where ϵ is an arbitrarily small positive number. For a sequential test with a desired strength (α, β) to distinguish them, it has to accept \mathcal{H}_1 with a probability at least $1 - \alpha$ in the first case, and accept \mathcal{H}_1 with a probability at most β in the second case. This is impractical unless $\beta = 1 - \alpha$, which means that either the Type I or the Type II error would be meaninglessly large. Consequently, a sequential test is usually used in those cases that θ is assumed not too close to θ_0 . Figure 2.1b shows a realistic curve for the probability of accepting \mathcal{H}_1 as a function of θ . As θ grows towards 1 and 0, the realistic Type I and Type II error approach 0. \diamond

Example 2.1.3 (Tossing A Biased Coin - Strength). Let us analyze the strength of the test in Example 2.1.2. With $A = 100$ and $B = 0.01$, we have

$$\frac{\beta}{1 - \alpha} \leq 0.01 \quad \text{and} \quad \frac{\alpha}{1 - \beta} \leq 0.01.$$

Given that $\alpha, \beta \in [0, 1]$, the inequalities are relaxed: $\alpha \leq 0.01$ and $\beta \leq 0.01$. Thus we have a Type I error less than 1% and a Type II error less than 1%. \parallel

2.1.2 Sequential Bayesian Test

As in Section 2.1.1, this section considers the hypotheses $\mathcal{H}_1 : \theta \geq \theta_0$ versus $\mathcal{H}_2 : \theta < \theta_0$. A Bernoulli random variable Z is defined the same as (2.1) and the sequence of observations $D = (z_1, \dots, z_m)$ are assumed i.i.d. A Bayesian test, which computes a Bayes factor B (not confused with the parameter B in SPRT) rather than the probability ratio in SPRT, has a similar rule as follows.

- Accept \mathcal{H}_1 to be true if $B \geq T$;
- Accept \mathcal{H}_2 to be true if $B \leq \frac{1}{T}$;
- Draw another observation and continue testing if $\frac{1}{T} < B < T$;

The parameter T is called the threshold of the Bayes factor.

2.1.2.1 Bayes' Theorem and Bayes Factor

The theory of Bayesian test relies on Bayes' theorem. It states that for two events P and Q ,

$$\Pr(P | Q) = \frac{\Pr(Q | P)\Pr(P)}{\Pr(Q)}. \quad (2.8)$$

In the Bayesian interpretation, P represents a hypothesis whose probability we are interested in and Q represents the outcome of an experiment. $\Pr(P)$, the **prior**, is interpreted as the initial belief in P . $\Pr(P | Q)$, the **posterior**, the belief in P after observing the outcome of Q . The quotient $\frac{\Pr(Q | P)}{\Pr(Q)}$, which transforms the prior to the posterior, shows the supports (positive or negative) Q provided to P . If Q is in favor of P , the posterior shows that the belief in P is strengthened taking Q into account. Otherwise, the belief is weakened.

In the simplest case, P is a simple hypothesis, i.e., the parameters are completely specified. The prior and the posterior are the probability of P being true before and after considering Q , respectively. In more complicated cases, however, P is a composite hypothesis and the parameters may not be fixed values. To define the probability with respect to such a hypothesis, we are required to provide a **prior distribution** on the free parameters, which reflects our initial belief in how these parameters are distributed.

Let $\mathcal{H}_1 : \theta \geq \theta_0$ versus $\mathcal{H}_2 : \theta < \theta_0$ be two competing hypotheses with a prior distribution $\pi(\theta)$. It is natural to consider using Bayes' theorem to check which hypothesis is more probable given some observations D . Plugging \mathcal{H}_1 and \mathcal{H}_2 into (2.8), the posteriors are

$$\Pr(\mathcal{H}_1 | D) = \frac{\Pr(D | \mathcal{H}_1)\Pr(\mathcal{H}_1)}{\Pr(D)}, \quad \Pr(\mathcal{H}_2 | D) = \frac{\Pr(D | \mathcal{H}_2)\Pr(\mathcal{H}_2)}{\Pr(D)}. \quad (2.9)$$

Table 2.1: A scale of Bayes factor in supporting the hypothesis \mathcal{H}_1 (adapted from Jeffreys [41]).

B	1 to 3	3 to 10	10 to 30	30 to 100	> 100
Strength in supporting \mathcal{H}_1	Barely worth mentioning	Substantial	Strong	Very strong	Decisive

In general, it is difficult to compute $\Pr(D)$. Thus we convert (2.9) into the odds ratio,

$$\frac{\Pr(\mathcal{H}_1 | D)}{\Pr(\mathcal{H}_2 | D)} = \frac{\Pr(D | \mathcal{H}_1) \Pr(\mathcal{H}_1)}{\Pr(D | \mathcal{H}_2) \Pr(\mathcal{H}_2)}. \quad (2.10)$$

The ratio $\frac{\Pr(\mathcal{H}_1)}{\Pr(\mathcal{H}_2)}$ and $\frac{\Pr(\mathcal{H}_1 | D)}{\Pr(\mathcal{H}_2 | D)}$ are known as the prior odds and the posterior odds. $\frac{\Pr(D | \mathcal{H}_1)}{\Pr(D | \mathcal{H}_2)}$, which is a ratio between two likelihoods, is called **Bayes factor**.

Equation (2.10) describes a fundamental relationship in Bayesian test. It shows how the observations change our initial belief on the two hypothesis. For example, initially, we believe that both \mathcal{H}_1 and \mathcal{H}_2 have a 50% chance of being true. After making several observations in favor of \mathcal{H}_1 , such a belief is altered such that we may think \mathcal{H}_1 is much more likely to happen. In this process, Bayes factor serves as a “transformation power” that convinces us the truth of \mathcal{H}_1 . Jeffreys [41] introduced a scale of Bayes factor in supporting the hypothesis \mathcal{H}_1 , as shown in Table 2.1. A similar table can be derived for \mathcal{H}_2 by taking the inverse of B . In practice, we often choose $T = 100$ as the threshold in the Bayesian test. As shown later, it yields a good strength of the test.

2.1.2.2 Computation of Bayes Factor

Let $\pi(\theta)$ be the prior distribution of θ in the hypotheses $\mathcal{H}_1 : \theta \geq \theta_0$ over $\mathcal{H}_2 : \theta < \theta_0$, and $D = (z_1, \dots, z_m)$ be the i.i.d. observations of the Bernoulli random variable Z defined in (2.1). Recall that $f_Z(z | \theta) = \theta^z(1 - \theta)^{1-z}$ is the pmf of Z . The Bayes factor is computed as follows.

$$\begin{aligned} B = \frac{\Pr(D | \mathcal{H}_1)}{\Pr(D | \mathcal{H}_2)} &= \frac{\int_{\theta_0}^1 f_Z(z_1 | \theta) \cdots f_Z(z_m | \theta) \cdot \pi(\theta) d\theta}{\int_0^{\theta_0} f_Z(z_1 | \theta) \cdots f_Z(z_m | \theta) \cdot \pi(\theta) d\theta} \\ &= \frac{\int_{\theta_0}^1 \theta^{m_1} (1 - \theta)^{m_2} \cdot \pi(\theta) d\theta}{\int_0^{\theta_0} \theta^{m_1} (1 - \theta)^{m_2} \cdot \pi(\theta) d\theta}, \end{aligned} \quad (2.11)$$

where $m_1 = \sum_{i=1}^m z_i$ and $m_2 = m - m_1$ are the number of observations in favor of \mathcal{H}_1 and \mathcal{H}_2 , respectively. Unlike SPRT, the computation for Bayes factor is not straightforward since it involves integration over the prior distribution of θ .

It is a deep question in Bayesian statistics how the prior distribution $\pi(\theta)$ should be chosen, which has a great impact on the computation of Bayes factor (see Berger [7, 8, 9] for discussions). In the case that little information is available regarding to the distribution of θ , a **non-informative prior** $\pi(\theta) = 1$ is often assumed. Since such a prior does not provide any implication on the prior probability of the hypotheses, we additionally assume that $\Pr(\mathcal{H}_1) = \Pr(\mathcal{H}_2) = 0.5$. This is known as the **objective Bayesian analysis** [9]. With a non-informative prior, (2.11) is simplified into

$$B = \frac{\int_{\theta_0}^1 \theta^{m_1} (1 - \theta)^{m_2} d\theta}{\int_0^{\theta_0} \theta^{m_1} (1 - \theta)^{m_2} d\theta}. \quad (2.12)$$

Remark 2. Sequential Bayesian test does not require an indifference region as SPRT does, and thus the computation is possible for hypotheses $\mathcal{H}_1 : \theta \geq \theta_0$ versus $\mathcal{H}_2 : \theta < \theta_0$. Consequently, it is often more convenient to use in practice. As Remark 1 suggested, sequential Bayesian test, same as SPRT, experiences a poor performance in the case that the true θ is close to θ_0 . On the other hand, in those cases that θ is distant from θ_0 , sequential Bayesian test usually has a better performance than SPRT. Jha [42] contains a detailed discussion on the performance of sequential Bayesian test. ◇

Example 2.1.4 (Tossing A Biased Coin - A Bayesian Approach). We consider the pair of hypotheses $\mathcal{H}_1 : \theta \geq 0.7$ versus $\mathcal{H}_2 : \theta < 0.7$ in Example 2.1.1 using sequential Bayesian test. As in Example 2.1.2, at some point of the computation, we have collected 40 observations without leading to a conclusion. Let $T = 100$ be the threshold of Bayes factor. Consider the following three cases.

- If $m_1 = 38$ and $m_2 = 2$, the Bayes factor $B \approx 13261$ is greater than T . Thus we accept the hypothesis \mathcal{H}_1 . In fact, $B \approx 146 \geq T$ for $m_1 = 35$ and $m_2 = 5$, which means that we could have accepted \mathcal{H}_1 using less observations;

- If $m_1 = 18$ and $m_2 = 22$, the Bayes factor $B \approx 0.0004$ is less than $\frac{1}{T}$. Thus we accept the hypothesis \mathcal{H}_2 . In fact, $B \approx 0.009 \leq \frac{1}{T}$ for $m_1 = 21$ and $m_2 = 19$, which means that we could have accepted \mathcal{H}_2 using less observations;
- If $m_1 = 30$ and $m_2 = 10$, the Bayes factor $B \approx 2.6$. As in SPRT, we need more observations to conclude.

These cases illustrate that sequential Bayesian test has a superior performance over SPRT. ||

Now let us compare (2.2) and (2.12). For the ease of reading, the two equations are repeated.

$$\frac{p_{1m}}{p_{2m}} = \prod_{i=1}^m \frac{\Pr(z_i | \theta = \theta_1)}{\Pr(z_i | \theta = \theta_2)} = \frac{\theta_1^{m_1} (1 - \theta_1)^{m_2}}{\theta_2^{m_1} (1 - \theta_2)^{m_2}}, \quad B = \frac{\int_{\theta_0}^1 \theta^{m_1} (1 - \theta)^{m_2} d\theta}{\int_0^{\theta_0} \theta^{m_1} (1 - \theta)^{m_2} d\theta},$$

where the Bayes factor B is computed with respect to a non-informative prior. It is not hard to see the similarities between the two equations. $\frac{p_{1m}}{p_{2m}}$ is a ratio between fixed values of $\theta = \theta_1$ and $\theta = \theta_2$, while B is between an integral from θ_0 to 1 and that from 0 to θ_0 . In fact, SPRT can be regarded as a sequential Bayesian test with a prior

$$\pi(\theta) = \begin{cases} \frac{1}{2} & \text{if } \theta = \theta_1, \\ \frac{1}{2} & \text{if } \theta = \theta_2, \\ 0 & \text{otherwise.} \end{cases}$$

Apparently, B takes into account more information than $\frac{p_{1m}}{p_{2m}}$. This leads to an intuitive explanation on why sequential Bayesian test has a superior performance over SPRT. We refer the interested readers to Jeffreys [41] for a theoretical treatment.

2.1.2.3 Strength of Sequential Bayesian Test

Now we consider the Type I/II error of sequential Bayesian test. Conventionally, the notion of Type I/II error probabilities, which is rooted in Neyman's treatment of hypothesis testing [66], does not apply to the theory of Bayesian test advocated by Jeffreys [40]. The connection is established

by Berger et al. [10]. They showed that in the objective Bayesian analysis, the Type I error α and the Type II error β are bounded for fixed T ,

$$\alpha \leq \frac{1}{T+1} \quad \text{and} \quad \beta \leq \frac{1}{T+1}. \quad (2.13)$$

An intuitive proof is presented here (see [42] for details). The proof for Type II error is shown. The proof for Type I error can be derived in a similar way. Consider (2.10) which is repeated below.

$$\frac{\Pr(\mathcal{H}_1 | D)}{\Pr(\mathcal{H}_2 | D)} = B \cdot \frac{\Pr(\mathcal{H}_1)}{\Pr(\mathcal{H}_2)}.$$

Since we assume an objective Bayesian analysis, $\frac{\Pr(\mathcal{H}_1)}{\Pr(\mathcal{H}_2)} = 1$. Suppose that \mathcal{H}_1 is accepted. Thus we have

$$\frac{\Pr(\mathcal{H}_1 | D)}{\Pr(\mathcal{H}_2 | D)} \geq T.$$

Adding 1 on each side and rearranging, we have

$$\frac{\Pr(\mathcal{H}_1 | D) + \Pr(\mathcal{H}_2 | D)}{\Pr(\mathcal{H}_2 | D)} = \frac{1}{\Pr(\mathcal{H}_2 | D)} \geq T + 1.$$

Since $\beta = \Pr(\mathcal{H}_2 | D)$, we have proved that $\beta \leq \frac{1}{T+1}$.

Example 2.1.5 (Tossing A Biased Coin - Strength of Bayesian Test). Now let us consider the strength of the test in Example 2.1.4. Since $T = 100$, according to (2.13) the Type I and the Type II error are bounded such that

$$\alpha \leq \frac{1}{100+1} \leq 0.01 \quad \text{and} \quad \beta \leq \frac{1}{100+1} \leq 0.01.$$

This shows that sequential Bayesian test can achieve a similar strength as SPRT does but with a better performance as shown in Example 2.1.4. ||

2.2 Statistical Model Checking

Model checking concerns with the problem that for a model of a system, check whether a given property is satisfied. It can be solved in different ways, such as exhaustive search, symbolic exploration and automatic theorem proving. Statistical model checking (SMC) is a family of model

checking techniques based on statistical inference. Unlike symbolic model checking techniques, SMC does not require explicit knowledge of systems under verification, such as the transition relation in a discrete-state model and the differential equation in a continuous-state model. Instead, it relies heavily on simulation to learn the behavior of the underlying systems. As a consequence, SMC provides statistical rather than formal guarantees. Compared to symbolic techniques, SMC has the following advantages:

- It is equation-free;
- It scales well with the dimension of problems;
- It is easy to implement.

Therefore, it can handle many problems that are far beyond the capability of symbolic techniques. In recent years, SMC has been applied to the verification of AMS circuits [53, 19, 85, 97], embedded systems [95, 74, 100], biological systems [43, 68, 65], medical devices [44] and many other areas.

The idea of SMC was first proposed by Younes and Simmons [95]. They formulate the model checking of probabilistic systems as a hypothesis testing problem, and introduce a solution based on sequential probability ratio test (SPRT) [83] (see Section 2.1.1). Later, Sen et al. [74] proposed to use a standard p -value significance test for the verification of black-box systems. Jha et al. [43] introduced a new SMC framework based on sequential Bayesian test [40, 47]. Compared to SPRT, Bayesian test is more convenient in practice since it does not require one to define indifference regions (see Section 2.1.2). Also, it is shown that Bayesian test usually has a better performance than SPRT. Zuliani et al. [100] proposed a Bayesian estimation approach that computed an interval estimate for the probability of satisfying **bounded linear temporal logic (BLTL)** properties.

This section presents an SMC technique based on sequential Bayesian test proposed by Jha et al. [43]. This technique can be used to verify BLTL properties. In the next, the formalism of BLTL is introduced first. Then we show how BLTL properties are model checked via Bayesian test.

2.2.1 Bounded Linear Temporal Logic

Linear temporal logic (LTL) is a formalism in which the statements are referring to paths in a state transition system (possibly infinite) over time. It was first proposed by Kamp [46] and later introduced to formal verification by Pnueli [69]. Examples of LTL properties include: some assertion is eventually true, and assertion ϕ will be true until another ψ becomes true. A LTL formula can reason about properties over paths with infinite length. In general,¹ such paths represent behaviors of a reactive system over infinite time. A bounded LTL (BLTL) formula is an LTL formula that is restricted to a finite time horizon. For instance, some condition becomes true in 10 seconds. The bounded time makes a BLTL property easier to verify in many cases.

2.2.1.1 Syntax of BLTL

This section skips the syntax definition of LTL since it is not relevant, and proceeds directly to BLTL. A BLTL formula is built upon a set of propositions AP, the logical operator \neg and \wedge , and the temporal operator \mathbf{X} , $\mathbf{U}^{(T)}$, where T represents discrete time steps. It is defined inductively as follows.

- If $p \in \text{AP}$, then p is a BLTL formula;
- If ϕ and ψ are BLTL formulas, then $\neg\phi$, $\phi \wedge \psi$, $\mathbf{X}\phi$ and $\phi\mathbf{U}^{(T)}\psi$ are all BLTL formulas.

The operator \mathbf{X} is the **next-state operator** and $\mathbf{U}^{(T)}$ is the **bounded until operator**. For convenience, two more temporal operators, $\mathbf{G}^{(T)}$ and $\mathbf{F}^{(T)}$, are defined such that

- $\mathbf{G}^{(T)}\phi$, ϕ being true globally up to time T , is equivalent to $\phi\mathbf{U}^{(T)}\mathbf{false}$;
- $\mathbf{F}^{(T)}\phi$, ϕ being true eventually up to time T , is equivalent to $\mathbf{trueU}^{(T)}\phi$;

Note that the above definition is just one way of defining BLTL syntax. It is chosen because it offers the most succinct syntax. One could use a different set of logical and temporal operators to construct virtually the same definition.

¹ Namely, we assume that the state transition system is non-zeno.

2.2.1.2 Semantics of BLTL

BLTL concerns only the bounded behavior of a system. A time-bounded path $\pi^{(T)}$ in a state transition system is a sequence of states (s_0, s_1, \dots, s_n) such that

- s_0 is an initial state in the system;
- It is possible for the system to evolve from s_i to s_{i+1} , which takes time t_{i+1} ;
- The path is over a time horizon at most T , i.e., $\sum_{i=1}^n t_i \leq T$.

For simplicity, this thesis assumes that the time steps t_i are equal. Hence, the next state of a state is naturally defined. A labeling function $L : \mathcal{S} \rightarrow 2^{\text{AP}}$, where \mathcal{S} is the state space of the system and 2^{AP} is the power set of the propositions, maps a state to a set of propositions that are true at the state. We use $\pi^{(T)} \models \phi$ to denote that a time-bounded path $\pi^{(T)} = (s_0, s_1, \dots, s_n)$ satisfies a BLTL formula ϕ . Formally, the relation \models is defined as follows.

- For $p \in \text{AP}$, $\pi^{(T)} \models p$ if $s_0 \in L(p)$;
- $\pi^{(T)} \models \phi$ if and only if $\pi^{(T)} \not\models \neg\phi$;
- $\pi^{(T)} \models \phi$ and $\pi^{(T)} \models \psi$ if and only if $\pi^{(T)} \models \phi \wedge \psi$;
- $\pi^{(T)} \models \mathbf{X}\phi$ if and only if $\pi_1^{(T)} \models \phi$ where $\pi_i^{(T)} = (s_i, s_2, \dots, s_n)$;
- $\pi^{(T)} \models \phi \mathbf{U}^{(T)} \psi$ if and only if there exists $0 \leq i \leq n$ such that $\pi_i^{(T)} \models \psi$ and for all $0 \leq j < i$, the path $\pi^{j(T)} \models \phi$ with $\pi^{j(T)} = (s_0, \dots, s_j)$.

The semantics of the syntactic sugar $\mathbf{G}^{(T)}$ and $\mathbf{F}^{(T)}$ can be derived from $\mathbf{U}^{(T)}$.

- $\pi^{(T)} \models \mathbf{G}^{(T)}\phi$ if for all $0 \leq i \leq n$, $\pi_i^{(T)} \models \phi$;
- $\pi^{(T)} \models \mathbf{F}^{(T)}\phi$ if there exists $0 \leq i \leq n$, $\pi_i^{(T)} \models \phi$.

Example 2.2.1 (BLTL Properties). Consider a continuous-state system \mathcal{M} with a stochastic parameter x and an output y such that $y = f(x, t)$. The behavior of the system is affected by x . Let

$y(t)$ be the step response of the system. The trajectory is desired to have the following properties under the variations of x :

- For $t \in [0, 2]$, $y(t) \leq 1.5$;
- For $t \in [2, 5]$, $y(t) \in [0.8, 1.2]$;
- For $t \in [0, 5]$, $y(t)$ eventually stays within $[0.95, 1.05]$.

Assume that time is discretized with a step size of 1. To write the properties in BLTL, we first define three propositions:

$$\phi_1 : y(t) \leq 1.5, \phi_2 : y(t) \in [0.8, 1.2], \phi_3 : y(t) \in [0.95, 1.05].$$

The BLTL formulas for the three properties are then written as

- $\mathbf{G}^{(2)}\phi_1$. ϕ_1 is true in two time steps;
- $\mathbf{G}^{(3)}\mathbf{XX}\phi_2$. $\mathbf{XX}\phi_2$ is true in five time steps and thus ϕ_2 is true for $t \in [2, 5]$;
- $\mathbf{F}^{(5)}\phi_3$. In five time steps, ϕ_3 is eventually true.

It is easy to check that the BLTL formulas are equivalent to the original properties. ||

For a system \mathcal{M} , we say that $\mathcal{M} \models \phi$ if the BLTL formula ϕ is satisfied by every path of \mathcal{M} starting from an initial state.

2.2.2 Bayesian Statistical Model Checking

BLTL properties can be statistically verified by Bayesian SMC. Instead of showing whether a BLTL property holds, Bayesian SMC concerns whether the probability that the property is satisfied is greater than a certain probability. Formally, it aims to show that

$$\Pr(\mathcal{M} \models \phi) \geq \theta_0, \tag{2.14}$$

where \mathcal{M} is an interested system, ϕ is a BLTL property and θ_0 is a desired probability.

Notice that a trajectory of the system either satisfies or violates the BLTL property ϕ . We define a Bernoulli random variable Z for the event that a trajectory satisfies ϕ , in which case $Z = 1$. Hence, it has the following pmf:

$$f_Z(z | \theta) = \theta^z(1 - \theta)^{1-z},$$

where θ is the probability that an arbitrary trajectory satisfies ϕ . Obviously, $\theta = \Pr(\mathcal{M} \models \phi)$. Therefore, the problem of verifying (2.14) is reduced to verify that

$$\theta \geq \theta_0.$$

Section 2.1.2 shows that this problem can be solved by sequential Bayesian test. To do that, one needs to introduce the pair of hypotheses $\mathcal{H}_1 : \theta \geq \theta_0$ versus $\mathcal{H}_2 : \theta < \theta_0$, and compute the Bayes factor B according to (2.12) with the observations $D = (z_1, \dots, z_m)$ drawn in sequence. Each observation z_i corresponds to a trajectory in the system obtained from simulation. Once B grows beyond a predefined threshold T , we accept \mathcal{H}_1 and conclude that (2.14) is true. Similarly, if B is below $\frac{1}{T}$, we accept \mathcal{H}_2 and conclude that (2.14) is false. The chance that the conclusion is wrong is bounded by $\frac{1}{T+1}$ as shown by the inequalities (2.13).

Example 2.2.2 (Statistical Verification using Bayesian SMC). Let us continue with the system \mathcal{M} in Example 2.2.1 and its BLTL properties

$$\mathbf{G}^{(2)}\phi_1, \mathbf{G}^{(3)}\mathbf{XX}\phi_2, \mathbf{F}^{(5)}\phi_3.$$

The goal is to show whether

$$\Pr\left(\mathcal{M} \models \mathbf{G}^{(2)}\phi_1 \wedge \mathbf{G}^{(3)}\mathbf{XX}\phi_2 \wedge \mathbf{F}^{(5)}\phi_3\right) \geq 0.95. \quad (2.15)$$

To verify using Bayesian SMC, we need to sequentially sample the stochastic parameter x , which is regarded as a random variable X . Suppose that the distribution of X is characterized by a pdf $f_X(x)$. The sampling is then carried out with respect to $f_X(x)$. A Bernoulli random variable Z is defined such that $Z = 1$ if a trajectory satisfies $\mathbf{G}^{(2)}\phi_1 \wedge \mathbf{G}^{(3)}\mathbf{XX}\phi_2 \wedge \mathbf{F}^{(5)}\phi_3$ and $Z = 0$ otherwise. The problem is reduced to verify whether $\theta \geq 0.95$ where θ is the probability of $Z = 1$.

Let the threshold of Bayes factor be 100. Suppose that we have collected 88 observations, all of which satisfy the BLTL properties. The current Bayes factor is $B \approx 95$. If the next observation still satisfies the properties, then B grows beyond 100 and we can conclude that (2.15) is true. Otherwise, we need to sample more to make a conclusion. ||

2.3 Regression

In statistics, regression is a process for estimating the relationships among different variables. Usually, the variables are categorized into two non-overlapping sets: independent variables and dependent variables. Regression takes a set of data, which consists of observations on the independent and the dependent variables, and aims to derive a function that explains how the dependent variables change when the independent variables vary. This section presents short reviews of two commonly used regression techniques, ordinary least squares (OLS) and regularization.

2.3.1 Ordinary Least Squares

OLS is one of the most common regression techniques. It has a long history and was first published by Legendre in 1805 and by Gauss in 1809. OLS estimates the unknown parameters in a linear regression model by minimizing the sum of squared error between the observed dependent variables in the dataset and the values predicted by the linear approximation.

Consider a set of n independent variables $\mathbf{x} = (x_1, \dots, x_n)$ and a single dependent variable y . Suppose that we do not know the relationship between \mathbf{x} and y and would like to estimate y using a linear combination of the variables \mathbf{x} . To do this, we establish a linear function (or more precisely, an affine function)

$$\hat{y} = \hat{f}(\mathbf{x}) = \beta_0 + \sum_{i=1}^n \beta_i x_i,$$

where $\boldsymbol{\beta} = (\beta_0, \beta_1, \dots, \beta_n)$ are unknown coefficients. We need to solve for these coefficients such that the estimate \hat{y} is as close to the true y as possible.

Suppose that we have a set of N independent observations $\{\mathbf{x}^{(i)}, y^{(i)}\}$ on the independent

and the dependent variables. Let \mathbf{X} be an $N \times (n + 1)$ matrix and \mathbf{y} be an N vector,

$$\mathbf{X} = \begin{pmatrix} 1 & x_1^{(1)} & \cdots & x_n^{(1)} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_1^{(N)} & \cdots & x_n^{(N)} \end{pmatrix}, \quad \mathbf{y} = \begin{pmatrix} y^{(1)} \\ \vdots \\ y^{(N)} \end{pmatrix}.$$

The goal of OLS is to find the coefficients $\boldsymbol{\beta}$ such that the sum of the squared errors between $\mathbf{X}\boldsymbol{\beta}$ and \mathbf{y} is minimized, i.e.,

$$\min_{\boldsymbol{\beta}} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2. \quad (2.16)$$

The vector $\mathbf{y} - \mathbf{X}\boldsymbol{\beta}$ is known as the **in-sample error**. In most cases, $\boldsymbol{\beta}$ is used for prediction. Hence, besides the in-sample error, it is also important that the computed $\boldsymbol{\beta}$ leads to small errors for new observations, which is called the **out-of-sample error**.

It has been shown in many textbooks that the problem (2.16) has a closed form solution (see, e.g., Kleinbaum et al. [48])

$$\boldsymbol{\beta} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}. \quad (2.17)$$

The linear system $\mathbf{X}\boldsymbol{\beta} = \mathbf{y}$ has at most one solution if $N \geq n$. Such a system is said to be over-determined since there are more observations than the number of unknowns. In this case, the solution (2.17) usually leads to small out-of-sample errors and forms a good estimation of the true relationship between \mathbf{x} and y . On the other hand, if $N < n$, the system $\mathbf{X}\boldsymbol{\beta} = \mathbf{y}$ is said to be under-determined and has infinitely many solutions. In this case, (2.17) can lead to excellent fit with very small in-sample errors for the observations in the dataset. But for new observations that are not used for fitting, it may result in large out-of-sample errors. Such a phenomenon is known as over-fitting and is the key problem that restricts the scalability of OLS. Hawkins [36] provides a good introduction to the problem of over-fitting.

2.3.2 Regularization

For a normalized dataset, an over-fitted model is likely to have coefficients that range over many magnitudes. For instance, we may have $\beta_1 = 1$ and $\beta_2 = 1000$ when x_1 and x_2 have similar

contribution to y . Such a model over-emphasizes the importance of certain independent variables and thus leads to large out-of-sample errors. An easy solution is to collect more observations. However, this is not always achievable due to many practical reasons, such as high cost of simulation, limited access to the dataset, and so on.

Regularization is a common approach to avoid over-fitting when the number of observations is smaller than that of the independent variables. It introduces additional constraints on the vector $\boldsymbol{\beta}$, preventing the coefficients from taking extreme values. A regularized least squares problem has the following form:

$$\min_{\boldsymbol{\beta}} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \lambda\|\boldsymbol{\beta}\|, \quad (2.18)$$

where λ is a free parameter that needs to be tuned empirically (typically by cross validation, see Golub et al. [35]). Intuitively, a large coefficient in $\boldsymbol{\beta}$ leads to a large $\|\boldsymbol{\beta}\|$ and thus is penalized. Thus, the regularized term $\|\boldsymbol{\beta}\|$ forces the coefficients to behave “normally”.

Depending on which type of norm the regularization term $\|\boldsymbol{\beta}\|$ takes, (2.18) has different names. The problem

$$\min_{\boldsymbol{\beta}} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \lambda\|\boldsymbol{\beta}\|_2^2$$

is termed ridge regression [81], where the regularized term takes the \mathcal{L}_2 norm. Ridge regression admits a closed form solution

$$\boldsymbol{\beta} = (\mathbf{X}^T \mathbf{X} + \lambda^2 \mathbf{I})^{-1} \mathbf{X}^T \mathbf{y},$$

where \mathbf{I} is the identity matrix. On the other hand, if $\|\boldsymbol{\beta}\|$ takes the \mathcal{L}_1 norm, i.e.,

$$\min_{\boldsymbol{\beta}} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \lambda\|\boldsymbol{\beta}\|_1,$$

it is called the LASSO problem, where LASSO stands for “least absolute shrinkage and selection operator” [80]. Unlike ridge regression, LASSO does not have closed form solutions. Discussion of solving LASSO problems is out of the scope of this thesis. Interested readers are referred to Tibshirani [80] for the original LASSO paper, Efron et al. [27] for an algorithm based on least angle regression (LAR).

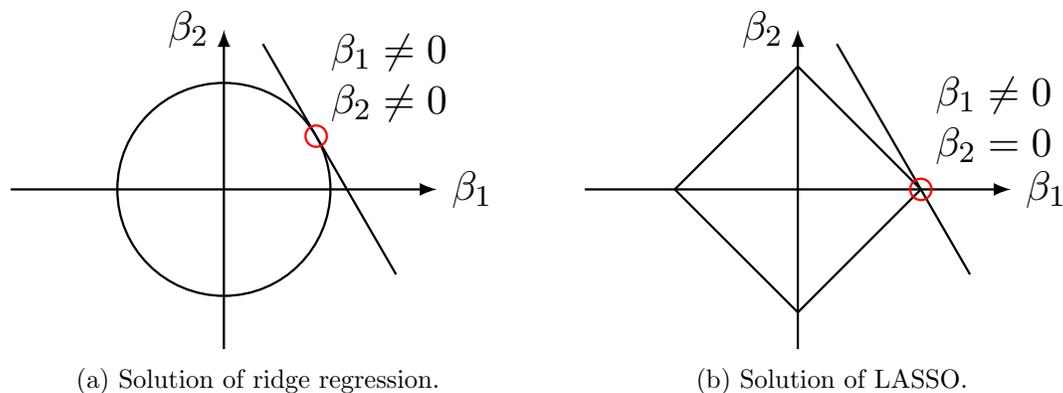


Figure 2.2: Comparison between ridge regression (a) and LASSO (b).

Both ridge regression and LASSO can be used to solve under-determined regression problems. But their outcomes are different. The solution produced by ridge regression is usually dense, i.e., all the coefficients take non-zero values. On the contrast, LASSO tends to produce sparse solutions. To understand this difference, let us take a look at Figure 2.2. In the two figures, the straight line shows a infinite set of solutions for the coefficient vector β obtained from the under-determined linear system $\mathbf{X}\beta = \mathbf{y}$. The circle in Figure 2.2a represents the regularization terms $\|\beta\|_2^2$. Similarly, the diamond in Figure 2.2b represents $\|\beta\|_1$ (i.e., circle in \mathcal{L}_1 norm). The regularized problem (2.18) has a solution when the circle/diamond becomes tangent to the line, as shown by the small circle in both figures. The solution for ridge regression contains zero coefficients only if the line is orthogonal to some axis. By contrast, LASSO tends to set some of the coefficients to 0. Because of this, LASSO is often preferred in practice.

Chapter 3

Statistical Soundness

In symbolic (formal) verification, soundness is a widely used notion.¹ The precise meaning of soundness depends on the context in which it is used. This thesis refers to the soundness of a model that abstracts some aspect of a system’s behavior. Informally, a model of a system is sound, if any behavior of the system is also that of the model. On the other hand, if there are behaviors of the system that the model misses, it is said to be unsound.

A notion called statistical soundness is introduced in this chapter, which is the basis of the rest of this thesis. It is developed by Zhang et al. [97]. Statistical soundness is a probabilistic argument that relaxes the soundness notion in symbolic verification. Intuitively, it says that a model is sound for at least some fraction of its input values. Such a relaxation transforms soundness for all possible inputs to the same notion for a fraction of them with a desired level of confidence, and enables us to discuss many practical cases where it is impossible to prove soundness using limited computational resources. In particular, statistical soundness is shown to be an important concept when one deals with black-box systems, for which sound models may be hard to obtain.

This chapter is organized as follows. First, two important concepts, black-box systems and response specifications, are defined. Next, statistical soundness is introduced. The following section shows what kind of guarantee a statistically sound model can provide in terms of the yield of a black-box system. Finally, the connections between statistical soundness and statistical model checking (SMC) is discussed.

¹ It originates from mathematical logic and refers to the fact that any formula that can be proved by the inference rules in a logical system is valid with respect to the semantics of the rules.

3.1 Black-Box Systems and Specifications

3.1.1 Black-Box Systems

The notion of statistical soundness is developed to handle the verification and optimization problem of black-box systems with respect to a set of specifications. First, we formally define black-box systems.

Definition 3.1.1 (Black-Box System). A black-box system \mathcal{M} is a tuple $(\mathbf{u}, \mathbf{x}, \phi, F_{\mathbf{X}}, \mathbf{r})$ that

- $\mathbf{u} = (u_1, \dots, u_m) \in \mathbb{R}^m$ is a set of real-valued design parameters;
- $\mathbf{x} = (x_1, \dots, x_n) \in \mathbb{R}^n$ is a set of real-valued stochastic parameters;
- $\phi = (\phi_1, \dots, \phi_k) \in \mathbb{R}^k$ is a set of real-valued responses;
- $F_{\mathbf{X}} : \mathbb{R}^n \rightarrow [0, 1]$ is a cumulative distribution function (cdf) of stochastic parameters \mathbf{x} ;
- $\mathbf{r} : \mathbb{R}^{m+n} \rightarrow \mathbb{R}^k$ is a response surface that maps design parameters \mathbf{u} and stochastic parameters \mathbf{x} to the responses ϕ , $\phi = \mathbf{r}(\mathbf{u}, \mathbf{x})$.

Let \mathbb{U} denote the domain of design parameters \mathbf{u} , and \mathbb{X} denote the domain of stochastic parameters \mathbf{x} (and thus the domain of the distribution $F_{\mathbf{X}}$). In a black-box system \mathcal{M} , we assume that design parameters \mathbf{u} are **controllable**, i.e., we can directly change the values of \mathbf{u} in \mathcal{M} . Usually, \mathcal{M} has a nominal design point \mathbf{u}_0 , which is the nominal values of the design parameters. Stochastic parameters \mathbf{x} , on the other hand, are considered as random variables $\mathbf{X} = (X_1, \dots, X_n)$ following the distribution $F_{\mathbf{X}}(\mathbf{x})$. They are **uncontrollable** and there is no way to predict the exact values of them. We further assume that the random variables are independent, each with a cdf $F_{X_i}(x_i)$. Note that this assumption is for the convenience of discussion. In practice, dependent random variables can be transformed using Rosenblatt transformation [70]. We denote $\mathbf{x}_0 = \bar{\mathbf{X}}$, the mean of \mathbf{X} , as the nominal point of the stochastic parameters. Note that unlike \mathbf{u}_0 which can be implemented physically, \mathbf{x}_0 merely represents an “ideal” situation and does not correspond to any real implementation.

A response ϕ stands for a certain behavior of \mathcal{M} . A behavior of a black-box system can be characterized using a response as long as it can be measured as a real-valued quantity. For instance, the settling time of an amplifier reflects how quick the system can react with the change of its input. It is defined as the time from the application of a step input to the time that the output stays within a certain band around the steady-state value. For any given trajectory, it is possible to measure/compute the settling time. Hence, it is a valid response of the system. The response surface \mathbf{r} reflects the dependence of responses ϕ on design parameters \mathbf{u} and stochastic parameters \mathbf{x} . \mathbf{r} is not required to be in a closed-form. For instance, it can be a flow function that is implicitly defined by a set of ODEs. However, \mathbf{r} must be **computable**. Given fixed values of \mathbf{u} and \mathbf{x} we should be able to evaluate $\mathbf{r}(\mathbf{u}, \mathbf{x})$ through either numerical simulation or physical measurement.

3.1.2 Design and Stochastic Parameters

In a black-box system, the design and stochastic parameters can be defined in different ways. For example, in a CMOS transistor, the channel width can be regarded as a stochastic parameter following some distribution. The mean of the distribution is usually the nominal value of the channel width. Alternatively, it can be interpreted as two components, a design parameter that indicates the nominal value and a stochastic parameter that represents the variations upon the nominal value. The two types of definition provide distinct interpretations of the stochastic parameter variations, and are used in different contexts. In this thesis, Chapter 4 and Chapter 5 follow the first type of definition, and Chapter 6 uses the second type of definition.

The distribution of stochastic parameters can either be independent or dependent on the values of design parameters. This thesis assumes that the variations are independent on the values of design parameters. But it should be mentioned that the proposed techniques (in particular, the statistically sound optimization technique in Chapter 6) can be extended in a straightforward way to handle the other case. Also, without loss of generality, this thesis only concerns with time-invariant stochastic parameters. For a stochastic parameter whose value varies with time, it can

be modeled as a discrete probabilistic model, or more sophisticatedly, a stochastic process that is parametrized by a few time-invariant parameters.

3.1.3 Response Specifications

In a black-box system \mathcal{M} , the responses ϕ are desired to meet certain performance requirements. These requirements are known as **response specifications**.

Definition 3.1.2 (Response Specification). For a black-box system \mathcal{M} with a response ϕ , a response specification S_ϕ with respect to ϕ is an inequality

$$\phi \in [a, b], \quad a, b \in \mathbb{R} \cup \{+\infty, -\infty\}.$$

A response specification constrains the allowed values of a response. We write $\mathcal{M}_{(\mathbf{u}, \mathbf{x})} \models S_\phi$ to indicate that the system \mathcal{M} satisfies the response specification S_ϕ when the design and the stochastic parameters take the particular values \mathbf{u} and \mathbf{x} , respectively. We write $\mathcal{M}_{\mathbf{u}} \models S_\phi$ if

$$\mathcal{M} \models S_\phi \text{ for fixed } \mathbf{u} \text{ and } \mathbf{x} \in \mathbb{X}.$$

Usually, a black-box system is accompanied with a set of response specifications, each of which restricts a particular behavior of the system. In the following, we occasionally refer response specifications as specifications for short.

Both Definition 3.1.2 and the BLTL formalism introduced in Section 2.2.1 can be used to assert the behaviors of a black-box system. One may be interested in the difference between them in terms of the expressive power. Fainekos and Pappas [29] showed that any BLTL formula can be expressed as response specifications and vice versa.² A response can be instrumented as an output of the system and thus a corresponding response specification can be written as a BLTL formula. On the other hand, a BLTL formula can be converted to a robustness metric, which is a real-valued measurement on how robust the formula is, and can be expressed as response specifications. Although BLTL and response specifications have the same expressive power, there

² In fact, their introduction is based on metric temporal logic (MTL), a stronger logic than BLTL.

are properties that are easier to express in BLTL, and similarly, properties that more natural in response specifications. The following example shows properties in both categories.

Example 3.1.1 (Response Specification versus BLTL). A D flip-flop is a digital circuit used to store state information. It has two inputs D and clk and two outputs Q and \bar{Q} . When clk is in a rising (or falling) edge, the value of D is recorded and the outputs becomes $Q = D$ and $\bar{Q} = \neg D$. Otherwise, the outputs stay unchanged regardless the input D. Let us assume that the flip-flop is only sensitive to rising edges of clk. Consider the following properties:

- (1) If $D = 1$, then $Q = 1$ after the next rising edge of clk;
- (2) If $Q = 1$, it stays unchanged until the next rising edge after $D = 0$;
- (3) If $Q = 1$, the voltage $V(Q)$ should be at least $0.8V_{DD}$ where V_{DD} is the supply voltage;
- (4) The propagation delay t_d is less than 10 ns;
- (5) The power consumption w of the circuit should be less than $5 \mu\text{W}$.

Property (1) can be expressed in BLTL as $D \rightarrow \mathbf{X}Q$ if we assume that the time step is the clock cycle. Similarly, property (2) can be written as $Q \rightarrow Q\mathbf{U}^{(T)}\mathbf{Y}\neg D$, where \mathbf{Y} , the dual of \mathbf{X} , is the previous-state operator. These two properties cannot be easily cast as response specifications in Definition 3.1.2. Property (3) can be expressed in both formalisms. In BLTL, it is written as $Q \rightarrow (V(Q) \geq 0.8V_{DD})$. On the contrast, a specification $\phi \geq 0.8V_{DD}$ is constructed where $\phi = \min_{V(Q) \geq 0.5V_{DD}} V(Q)$. The last two properties are handled naturally by response specifications. But it is a bit involved if we express them in BLTL. ||

Although BLTL and response specifications have similar expressive power, in this thesis we consider only response specifications. This is because many important properties in real systems can be expressed in a more straightforward way as response specifications.

Example 3.1.2 (Response Specifications in a Black-Box System). Consider the properties of the system in Example 2.2.1. For convenience, they are stated again. The step response $y(t)$ of the system is desired to satisfy the following properties:

- For $t \in [0, 2]$, $y(t) \leq 1.5$;
- For $t \in [2, 5]$, $y(t) \in [0.8, 1.2]$;
- For $t \in [0, 5]$, $y(t)$ eventually stays within $[0.95, 1.05]$.

To write down specifications that fulfill these properties, we define the following responses:

$$\phi_1 = \min_{t \in [0, 2]} (1.5 - y(t)) , \phi_2 = \max_{t \in [2, 5]} (y(t) - 0.8) , \phi_3 = \min_{t \in [2, 5]} (1.2 - y(t)) , \phi_4 = y(5) .$$

Then the properties can be expressed as

$$\phi_1 \geq 0 , \phi_2 \geq 0 , \phi_3 \geq 0 , \phi_4 \in [0.95, 1.05] . \quad \parallel$$

3.2 Statistical Soundness

Consider the case that we want to learn how the design parameters \mathbf{u} and the stochastic parameters \mathbf{x} in a black-box system \mathcal{M} affect a response ϕ of the system.³ Since the response surface r does not necessarily have a closed-form, it naturally leads to the solution that approximates r using simpler functions. Such a technique is known as performance modeling and is discussed in detail in Chapter 4. At this point, suppose that we have a function $\hat{g}(\mathbf{u}, \mathbf{x})$ that approximates $r(\mathbf{u}, \mathbf{x})$. Let us think about this question: **what kind of guarantee can \hat{g} provide in order to reason about the real behavior of \mathcal{M} ?** In other words, if \hat{g} satisfies a response specification, what can we conclude about \mathcal{M} with the response surface r ?

It turns out that we cannot guarantee anything beyond a statement like “ \mathcal{M} perhaps also satisfies the specification”. The problem with performance modeling is that essentially, an approximation \hat{g} is merely a function that is close to the real response surface r in some metric. It does not know whether for an individual set of parameter values, \hat{g} is below or above r . To overcome this drawback, we introduce a series of notions as follows.

³ For simplicity, here we consider only one response of the system.

Definition 3.2.1 (Relational Model). A relational model is a Cartesian product $\mathbb{D} \times \mathbb{I}$ where \mathbb{D} is a set and \mathbb{I} is the set of real-valued intervals,

$$\mathbb{I} \equiv \{[a, b] \mid a, b \in \mathbb{R}\}.$$

In our work, relational models are used to map parameter values to intervals. To emphasize that a relational model $f = \mathbb{D} \times \mathbb{I}$ is treated as a model rather than a relation, we write f as

$$f : \mathbb{D} \rightarrow \mathbb{I}.$$

Definition 3.2.2 (Soundness). Consider a black-box system $\mathcal{M} = (\mathbf{u}, \mathbf{x}, \phi, F_{\mathbf{X}}, r)$ and a relational model $g : \mathbb{U} \times \mathbb{X} \rightarrow \mathbb{I}$ that maps design parameters \mathbf{u} and stochastic parameters \mathbf{x} to real-valued intervals. We say that g is sound if

$$\forall \mathbf{u} \in \mathbb{U}, \mathbf{x} \in \mathbb{X}. r(\mathbf{u}, \mathbf{x}) \in g(\mathbf{u}, \mathbf{x}). \quad (3.1)$$

Note that without loss of generality, we focus on systems with a single response. Intuitively, a sound model g over-approximates r such that for each individual set of parameter values, the value of the response ϕ is enclosed by an interval. It is not hard to see that for a black-box system, we cannot guarantee the soundness of a relational model since (1) it is impossible to enumerate the parameter values, and (2) the response surface does not have a closed-form. Hence instead of soundness, we introduce a relaxed notion called **statistical soundness**.

Definition 3.2.3 (Statistical Soundness). Consider a black-box system $\mathcal{M} = (\mathbf{u}, \mathbf{x}, \phi, F_{\mathbf{X}}, r)$ and a relational model $g : \mathbb{U} \times \mathbb{X} \rightarrow \mathbb{I}$ that maps design parameters \mathbf{u} and stochastic parameters \mathbf{x} to real-valued intervals. For a probability $\theta_0 \in (0, 1)$, we say that g is θ_0 statistically sound with respect to a finite set of values $\{\mathbf{u}_1, \dots, \mathbf{u}_n\}$ of the design parameters if

$$\Pr_{F_{\mathbf{X}}(\mathbf{x})} (r(\mathbf{u}, \mathbf{x}) \in g(\mathbf{u}, \mathbf{x})) \geq \theta_0, \mathbf{u} \in \{\mathbf{u}_1, \dots, \mathbf{u}_n\}. \quad (3.2)$$

We write $r(\mathbf{u}, \mathbf{x}) \prec_{\theta_0}^{\{\mathbf{u}_1, \dots, \mathbf{u}_n\}} g(\mathbf{u}, \mathbf{x})$ to indicate that $g(\mathbf{u}, \mathbf{x})$ is a θ_0 statistically sound model of $r(\mathbf{u}, \mathbf{x})$ at the design points $\{\mathbf{u}_1, \dots, \mathbf{u}_n\}$. When θ_0 and $\{\mathbf{u}_1, \dots, \mathbf{u}_n\}$ are clear from the context, we simply write $r(\mathbf{u}, \mathbf{x}) \prec g(\mathbf{u}, \mathbf{x})$ and say that g is a statistically sound model of r .

Compared to the soundness notion, we make two relaxations in order to define statistical soundness. First, the universal quantifications over design parameter space \mathbb{U} and stochastic parameter space \mathbb{X} in (3.1) are substituted into an enumeration over a finite set and a probability distribution, respectively. Second, the inclusion relation is no longer required to be definite, but only needs to be true for a desired probability. With Definition 3.2.3, we are able to reason about how the parameters of a black-box system affect its behavior in a manner that provide statistical guarantees. Later, we discuss the guarantees obtained from statistical soundness.

A statistically sound model g is usually constructed from some approximation \hat{g} (either functional or relational). In Chapter 4 we introduce a generalization technique that transforms \hat{g} , the approximation into g , a statistically sound model. We regard g as a statistical over-approximation of the response surface r under stochastic parameter variations such that the probability that r is bounded by g at the given set of design parameter values is at least θ_0 . Obviously, the larger θ_0 is, the closer g is to a true over-approximation as in Definition 3.2.2⁴. When $\theta_0 \rightarrow 1$, g is guaranteed to over-approximate r almost everywhere. Formal reasoning with a statistically sound model needs to account for the small probability of leading to a wrong conclusion, which depends on the inference procedure used to achieve statistical soundness. Moreover, in the case that the precise dynamics of the underlying system are not available, statistical soundness seems to be the best guarantee that we can achieve.

Remark 3. The probability θ_0 can sometimes be regarded as a proportion of the parameter space. For example, consider a system with a single uniformly distributed parameter. Then the inequality (3.2) is equivalent to that r is bounded by g in at least θ_0 proportion of the parameter space. But the statement is valid if the parameter space is “uniformly weighted”. Consider another system with a normally distributed parameter in which the center part of the parameter space is clearly more important. In this case, the meaning of θ_0 is no longer equivalent to a proportion of the parameter space. ◇

⁴ Of course, we are not concerned with those over-approximations that are meaninglessly excessive.

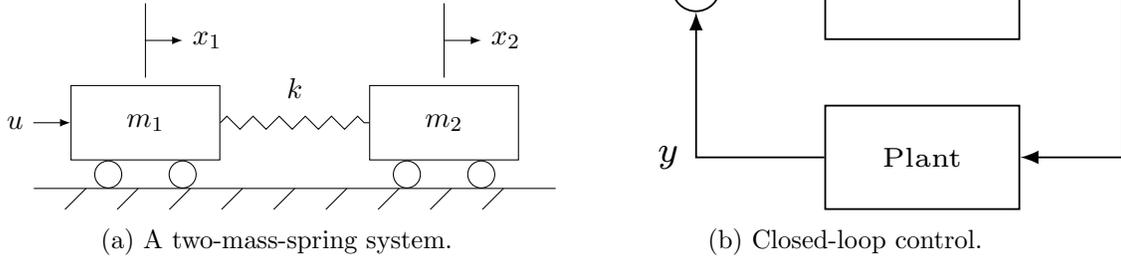


Figure 3.1: A two-mass-spring system and the closed-loop system with a controller.

Example 3.2.1 (Statistically Soundness in A Two-Mass-Spring System). A two-mass-spring system [86] is shown in Figure 3.1a. 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 3.1b a controller is used to track y with r , the reference position. A lead compensator, which has two tunable parameters, the pole location $p \in [-1200, -800]$ and the zero location $z \in [-1.2, -0.8]$, controls the plant. The nominal values are $p_0 = -1000$ and $z_0 = -1$.

Suppose that we are interested in the overshoot r_o of the step response $y(t)$ as the percentage of the steady-state value. The closed-loop system has 7 state variables. It is not trivial to find a solution for r_o . We show two statistically sound models⁵ that are introduced in Chapter 4 and Chapter 6. Both of them are 95% statistically sound models at the nominal design point (p_0, z_0) with respect to r_o . The first model g_1 defines the following relation at the nominal design point:

$$\begin{aligned} g_{1\ell}(k, m_1, m_2) &= 0.146 - 0.026k + 0.021m_1 + 0.021m_2, \\ g_{1u}(k, m_1, m_2) &= 0.155 - 0.026k + 0.021m_1 + 0.021m_2, \\ g_1(k, m_1, m_2) &= [g_{1\ell}(k, m_1, m_2), g_{1u}(k, m_1, m_2)]. \end{aligned}$$

It maps the stochastic parameters (k, m_1, m_2) into an interval so that there is a high probability that the true response r_o lies in the interval. Note that the lower and the upper bound functions are parallel. The other model g_2 is defined in terms of the design parameters (p, z) but is only

⁵ The variables in these models are normalized with a domain $[-1, 1]$.

statistically sound at the nominal design point (p_0, z_0) :

$$g_{2\ell}(p, z) = 0.121 + 0.006p + 0.078z,$$

$$g_{2u}(p, z) = 0.198 + 0.017p - 0.086z,$$

$$g_2(p, z) = [g_{2\ell}(p, z), g_{2u}(p, z)].$$

Notice that g_2 does not include any stochastic parameter. It “marginalizes” the effects of the stochastic parameters in the system and maps each design parameter values to an interval. The interval $g_2(p_0, z_0)$ is a statistical over-approximation of the response r_o . We show different applications of the two types of models in Chapter 4 and Chapter 6. ||

3.3 Statistically Sound Yield Computation

Now suppose that $g(\mathbf{u}, \mathbf{x})$ is a θ_0 statistically sound model of the response surface $r(\mathbf{u}, \mathbf{x})$ in a black-box system \mathcal{M} with respect to a set of design point $(\mathbf{u}_1, \dots, \mathbf{u}_n)$, i.e.,

$$r(\mathbf{u}, \mathbf{x}) \prec_{\theta_0}^{(\mathbf{u}_1, \dots, \mathbf{u}_n)} g(\mathbf{u}, \mathbf{x}).$$

Definition 3.2.3 defines statistical soundness in a mathematics point of view. However, for system designers, it is often useful to interpret the concept from an engineering perspective. An important notion for designers is the yield of a system. Simply speaking, yield is the probability that a specification is satisfied. To be precise, we define the notion as follows.

Definition 3.3.1 (Yield). Consider a black-box system $\mathcal{M} = (\mathbf{u}, \mathbf{x}, \phi, F_{\mathbf{X}}, r)$ with fixed design parameters. We denote such a system as $\mathcal{M}_{\mathbf{u}}$ and the response surface as $r_{\mathbf{u}}(\mathbf{x})$. The yield $Y_{\mathcal{M}_{\mathbf{u}}, S_{\phi}}$ of $\mathcal{M}_{\mathbf{u}}$ with respect to a response specification $S_{\phi} : \phi \in [a, b]$ is defined as:

$$Y_{\mathcal{M}_{\mathbf{u}}, S_{\phi}} = \Pr_{F_{\mathbf{X}}(\mathbf{x})} (\mathcal{M}_{\mathbf{u}} \models S_{\phi}) = \Pr_{F_{\mathbf{X}}(\mathbf{x})} (r_{\mathbf{u}}(\mathbf{x}) \in [a, b]). \quad (3.3)$$

Recall that $\mathcal{M}_{\mathbf{u}} \models S_{\phi}$ if \mathcal{M} satisfies S_{ϕ} for fixed \mathbf{u} and every $\mathbf{x} \in \mathbb{X}$. The extension of Definition 3.3.1 to multiple response specifications is natural. All we need is to substitute S_{ϕ} into $S_{\phi_1} \wedge \dots \wedge S_{\phi_n}$, where ϕ_1, \dots, ϕ_n are the interested specifications and \wedge stands for the logical AND. Obviously, the yield of multiple specifications cannot be greater than that of each individual.

Remark 4. In this thesis, yield is defined at fixed design parameters. Such a definition is well justified in practice since for the designing of real systems, designers try to make their designs as deterministic as possible. This means that they always choose a set values for the design parameters rather than leaving them incompletely specified. On the other hand, Definition 3.3.1 can be easily extended to handle a finite set of design points. But since those cases are unusual in real system designs, we do not provide a formal definition for them. \diamond

We write $g_{\mathbf{u}}(\mathbf{x})$ to denote the statistically sound model $g(\mathbf{u}, \mathbf{x})$ of the response surface $r(\mathbf{u}, \mathbf{x})$ in which the design parameters \mathbf{u} are fixed. It is not hard to see that $g_{\mathbf{u}}(\mathbf{x})$ implicitly defines a region that satisfies a response specification $S_\phi : \phi \in [a, b]$ in the stochastic parameter space. Let such a region be R . Formally,

$$R \equiv \{\mathbf{x} \mid g_{\mathbf{u}}(\mathbf{x}) \subseteq [a, b], \mathbf{x} \in \mathbb{X}\} . \quad (3.4)$$

If $g_{\mathbf{u}}(\mathbf{x})$ were a sound model of $r_{\mathbf{u}}(\mathbf{x})$, i.e., satisfied Definition 3.2.2, the region R would be a true under-approximation of the set of stochastic parameter values that satisfy S_ϕ ,

$$R \subseteq \{\mathbf{x} \mid r_{\mathbf{u}}(\mathbf{x}) \in [a, b]\} .$$

In this case, the yield with respect to $g_{\mathbf{u}}(\mathbf{x})$ would be lower than the true yield of $\mathcal{M}_{\mathbf{u}}$,

$$\Pr_{F_{\mathbf{X}}(\mathbf{x})}(g_{\mathbf{u}}(\mathbf{x}) \subseteq [a, b]) \leq \Pr_{F_{\mathbf{X}}(\mathbf{x})}(r_{\mathbf{u}}(\mathbf{x}) \in [a, b]) .$$

It would be possible that for some \mathbf{x} , $g_{\mathbf{u}}(\mathbf{x}) \not\subseteq [a, b]$ and $r_{\mathbf{u}}(\mathbf{x}) \in [a, b]$. But for any \mathbf{x} , if $g_{\mathbf{u}}(\mathbf{x}) \subseteq [a, b]$, we would have $r_{\mathbf{u}}(\mathbf{x}) \in [a, b]$. In other words, we would only have false negatives if we used $g_{\mathbf{u}}(\mathbf{x})$ for yield computation. Such a case is shown in Figure 3.2a, where we have a consistent result at \mathbf{x}_1 and a false positive at \mathbf{x}_2 .

However, $g_{\mathbf{u}}(\mathbf{x})$ is a statistical over-approximation of $r_{\mathbf{u}}(\mathbf{x})$. Although we can guarantee that $r_{\mathbf{u}}(\mathbf{x}) \in g_{\mathbf{u}}(\mathbf{x})$ for a large proportion of the stochastic parameter space, there may exists \mathbf{x} such that $r_{\mathbf{u}}(\mathbf{x}) \notin g_{\mathbf{u}}(\mathbf{x})$. This means that it is also possible to obtain a false positive result if we use $g_{\mathbf{u}}(\mathbf{x})$ for yield computation. Figure 3.2b shows the case for a statistically sound model. Apparently, we have a false positive at \mathbf{x}_1 and a false negative at \mathbf{x}_2 .

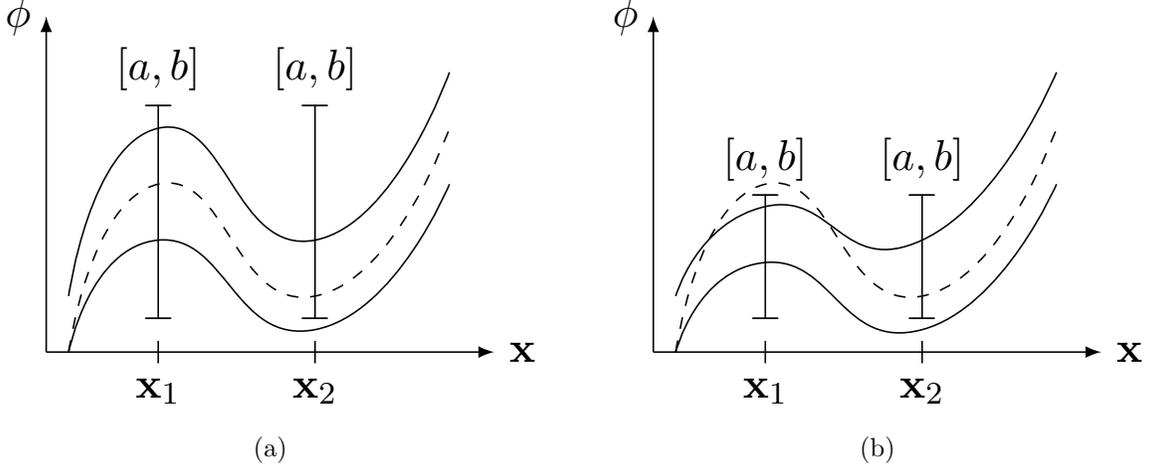


Figure 3.2: Response specifications in sound models (a) and in statistically sound models (b).

The following theorem shows a lower bound of the true yield in terms of the yield with respect to a given θ_0 statistically sound model $g_{\mathbf{u}}(\mathbf{x})$.

Theorem 3.3.1 (Lower Bound of Yield). Consider a response surface $r(\mathbf{u}, \mathbf{x})$ in a black-box system \mathcal{M} and a model $g(\mathbf{u}, \mathbf{x})$ such that

$$r(\mathbf{u}, \mathbf{x}) \prec_{\theta_0}^{\{\mathbf{u}\}} g(\mathbf{u}, \mathbf{x}).$$

Suppose that $S_\phi : \phi \in [a, b]$ is a response specification. Let Y_r be the yield of \mathcal{M}_u and Y_g be the yield of \mathcal{M}_u computed using g ,

$$Y_r = \Pr_{F_{\mathbf{x}}(\mathbf{x})} (r_{\mathbf{u}}(\mathbf{x}) \in [a, b]), \quad Y_g = \Pr_{F_{\mathbf{x}}(\mathbf{x})} (g_{\mathbf{u}}(\mathbf{x}) \subseteq [a, b]). \quad (3.5)$$

The two yields Y_r and Y_g satisfy the following inequality:

$$Y_r \geq \min(\theta_0, Y_g - (1 - \theta_0)). \quad (3.6)$$

Proof. Suppose that we evaluate whether a point \mathbf{x} is safe by checking whether $g_{\mathbf{u}}(\mathbf{x}) \subseteq [a, b]$. Clearly, we may have both false negatives, in which cases $g_{\mathbf{u}}(\mathbf{x}) \not\subseteq [a, b]$ and $r_{\mathbf{u}}(\mathbf{x}) \in [a, b]$, and false positives, in which cases $g_{\mathbf{u}}(\mathbf{x}) \subseteq [a, b]$ and $r_{\mathbf{u}}(\mathbf{x}) \notin [a, b]$.

To understand when we have false negatives and false positives, and when we have consistent conclusions from $g_{\mathbf{u}}(\mathbf{x})$ and $r_{\mathbf{u}}(\mathbf{x})$, let us divide the stochastic parameter space \mathbb{X} into four regions:

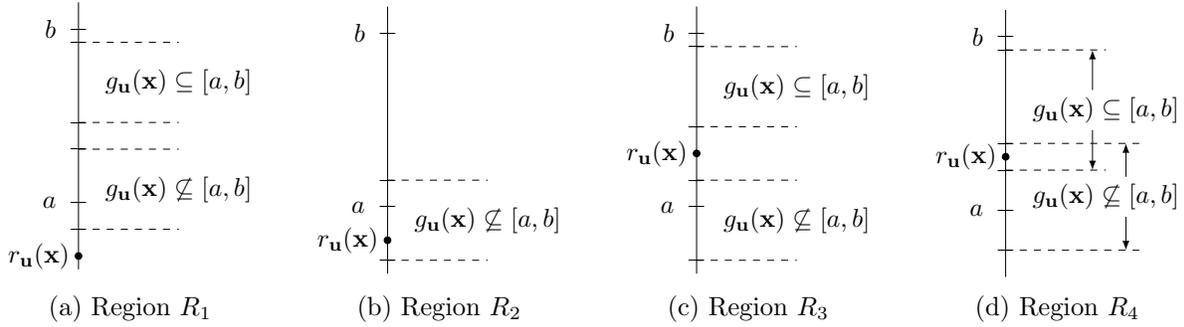


Figure 3.3: Relationship among $g_{\mathbf{u}}(\mathbf{x})$, $r_{\mathbf{u}}(\mathbf{x})$ and the interval $[a, b]$ in the four regions R_1 , R_2 , R_3 and R_4 in the proof of Theorem 3.3.1.

- $R_1 \equiv \{\mathbf{x} \mid r_{\mathbf{u}}(\mathbf{x}) \notin [a, b], r_{\mathbf{u}}(\mathbf{x}) \notin g_{\mathbf{u}}(\mathbf{x})\}$,
- $R_2 \equiv \{\mathbf{x} \mid r_{\mathbf{u}}(\mathbf{x}) \notin [a, b], r_{\mathbf{u}}(\mathbf{x}) \in g_{\mathbf{u}}(\mathbf{x})\}$,
- $R_3 \equiv \{\mathbf{x} \mid r_{\mathbf{u}}(\mathbf{x}) \in [a, b], r_{\mathbf{u}}(\mathbf{x}) \notin g_{\mathbf{u}}(\mathbf{x})\}$,
- $R_4 \equiv \{\mathbf{x} \mid r_{\mathbf{u}}(\mathbf{x}) \in [a, b], r_{\mathbf{u}}(\mathbf{x}) \in g_{\mathbf{u}}(\mathbf{x})\}$,

and investigate them one by one.

- For some $\mathbf{x} \in R_1$, if \mathbf{x} is shown to be safe, i.e., $g_{\mathbf{u}}(\mathbf{x}) \subseteq [a, b]$, \mathbf{x} is a false positive since $r_{\mathbf{u}}(\mathbf{x}) \notin [a, b]$ in R_1 . Otherwise, both $g_{\mathbf{u}}(\mathbf{x})$ and $r_{\mathbf{u}}(\mathbf{x})$ conclude that \mathbf{x} is unsafe.
- For some $\mathbf{x} \in R_2$, $g_{\mathbf{u}}(\mathbf{x}) \subseteq [a, b]$ is impossible since it leads to a contradiction with the assumptions $r_{\mathbf{u}}(\mathbf{x}) \notin [a, b]$ and $r_{\mathbf{u}}(\mathbf{x}) \in g_{\mathbf{u}}(\mathbf{x})$. If $g_{\mathbf{u}}(\mathbf{x}) \not\subseteq [a, b]$, \mathbf{x} is indeed unsafe since $r_{\mathbf{u}}(\mathbf{x}) \notin [a, b]$. Therefore, we have neither false positives nor false negatives in R_2 .
- In R_3 , we cannot have false positives since $r_{\mathbf{u}}(\mathbf{x}) \in [a, b]$. We have a consistent conclusion that \mathbf{x} is safe if $g_{\mathbf{u}}(\mathbf{x}) \subseteq [a, b]$, or a false negative if $g_{\mathbf{u}}(\mathbf{x}) \not\subseteq [a, b]$.
- R_4 is similar to R_3 . We do not have false positives. We can have a consistent conclusion that \mathbf{x} is safe, or a false negative.

The situation in each of the four regions is shown in Figure 3.3. To summarize, we may have false positives in R_1 , false negatives in R_3 and R_4 and none of them in R_2 .

Let $Y_r^{R_i}$ and $Y_g^{R_i}$ be the proportion of the yields in R_i with respect to $r_{\mathbf{u}}(\mathbf{x})$ and $g_{\mathbf{u}}(\mathbf{x})$, respectively,

$$Y_r^{R_i} = \Pr_{F_{\mathbf{X}}(\mathbf{x})} (\mathbf{x} \in R_i \wedge r_{\mathbf{u}}(\mathbf{x}) \in [a, b]) ,$$

$$Y_g^{R_i} = \Pr_{F_{\mathbf{X}}(\mathbf{x})} (\mathbf{x} \in R_i \wedge g_{\mathbf{u}}(\mathbf{x}) \subseteq [a, b]) .$$

Obviously, we have

$$Y_r = Y_r^{R_1} + Y_r^{R_2} + Y_r^{R_3} + Y_r^{R_4} ,$$

$$Y_g = Y_g^{R_1} + Y_g^{R_2} + Y_g^{R_3} + Y_g^{R_4} .$$

Since we may have false positives in R_1 , false negatives in R_3 and R_4 and none of them in R_2 , we have the following relationship between $Y_r^{R_i}$ and $Y_g^{R_i}$:

$$Y_r^{R_1} \leq Y_g^{R_1} ,$$

$$Y_r^{R_2} = Y_g^{R_2} ,$$

$$Y_r^{R_3} \geq Y_g^{R_3} ,$$

$$Y_r^{R_4} \geq Y_g^{R_4} .$$

Hence,

$$Y_r \geq Y_g^{R_2} + Y_g^{R_3} + Y_g^{R_4} = Y_g - Y_g^{R_1} . \quad (3.7)$$

Since $Y_g^{R_1}$ is only concerned with the points in R_1 , it cannot be greater than the probability of $\mathbf{x} \in R_1$,

$$Y_g^{R_1} \leq \Pr_{F_{\mathbf{X}}(\mathbf{x})} (r_{\mathbf{u}}(\mathbf{x}) \notin [a, b] \wedge r_{\mathbf{u}}(\mathbf{x}) \notin g_{\mathbf{u}}(\mathbf{x})) .$$

To proceed, we first restate (3.3) and (3.2) as follows.

$$Y_r = \Pr_{F_{\mathbf{X}}(\mathbf{x})} (r_{\mathbf{u}}(\mathbf{x}) \in [a, b]) ,$$

$$\theta_0 \leq \Pr_{F_{\mathbf{X}}(\mathbf{x})} (r_{\mathbf{u}}(\mathbf{x}) \in g_{\mathbf{u}}(\mathbf{x})) .$$

According to Fréchet inequalities, $\Pr(A \wedge B) \leq \min(\Pr(A), \Pr(B))$, we have

$$\begin{aligned} Y_g^{R_1} &\leq \Pr_{F_{\mathbf{X}}(\mathbf{x})} (r_{\mathbf{u}}(\mathbf{x}) \notin [a, b] \wedge r_{\mathbf{u}}(\mathbf{x}) \notin g_{\mathbf{u}}(\mathbf{x})) \\ &\leq \min \left(\Pr_{F_{\mathbf{X}}(\mathbf{x})} (r_{\mathbf{u}}(\mathbf{x}) \notin [a, b]), \Pr_{F_{\mathbf{X}}(\mathbf{x})} (r_{\mathbf{u}}(\mathbf{x}) \notin g_{\mathbf{u}}(\mathbf{x})) \right) \\ &\leq \min(1 - Y_r, 1 - \theta_0) \end{aligned} \quad (3.8)$$

Substitute (3.8) into (3.7), we have

$$Y_r \geq Y_g - \min(1 - Y_r, 1 - \theta_0) = \begin{cases} Y_g - (1 - Y_r) & \text{if } Y_r \geq \theta_0, \\ Y_g - (1 - \theta_0) & \text{if } Y_r < \theta_0. \end{cases}$$

Unifying these two cases, we have

$$Y_r \geq \min(\theta_0, Y_g - (1 - \theta_0)). \quad \square$$

Theorem 3.3.1 provides a lower bound on the yield with respect to $r_{\mathbf{u}}(\mathbf{x})$ in terms of θ_0 and the yield with respect to a statistically sound model $g_{\mathbf{u}}(\mathbf{x})$. It is well known that Fréchet inequalities are often quite pessimistic in practice. Empirical evidence shows that the bound can often be relaxed into

$$Y_r \geq Y_g. \quad (3.9)$$

It is important to realize that the yields discussed above are not estimations of any kind, such as Monte-Carlo sampling. They are the exact probabilities as in the equations (3.5). In practice, it is prohibitive to compute Y_r , if not impossible. On the other hand, it may also be difficult to compute Y_g exactly if $g_{\mathbf{u}}(\mathbf{x})$ is not in a simple form. Computing the set of values that satisfies inequality constraints is an interesting and challenging problem by itself, which is beyond the scope of this thesis. In this, we use Monte-Carlo sampling to estimate Y_g . The details are presented in Chapter 4.

Remark 5. Theorem 3.3.1 and its proof assumes that $g(\mathbf{u}, \mathbf{x})$ is a statistically sound model of $r(\mathbf{u}, \mathbf{x})$. Strictly speaking, we can never guarantee that with 100% confidence. This is because, as we see in Chapter 4, statistical soundness is achieved through a generalization technique based on sequential hypothesis testing. We have learned that any sequential hypothesis testing technique has an associated Type I and Type II error. Thus, the theorem is only valid when we do not commit either type of error. Fortunately, as shown in Chapter 2, the probabilities of these errors can be bounded to a reasonably low level. In reality, our method is usually not concerned with those cases that either type of error is committed. \diamond

Example 3.3.1 (A Two-Mass-Spring System - Yield). We revisit the two-mass-spring system in Example 3.2.1. Suppose that we have two specifications $r_o \leq 15\%$ and $r_o \leq 20\%$. Let us compare the true yield Y_r and the yield Y_g computed with respect to the model $g_1(\mathbf{x})$ in Example 3.2.1, a 95% statistically sound model of the overshoot r_o . Both of the yields are estimated through 10000 Monte-Carlo simulations.

For the first specification $r_o \leq 15\%$, $Y_r = 46.3\%$ and $Y_g = 39.1\%$. For the second specification $r_o \leq 20\%$, $Y_r = 100\%$ and $Y_g = 98.3\%$. Notice that in both cases Y_r and Y_g satisfy the empirical bound in the inequality (3.9). ||

3.4 Comparison with Statistical Model Checking

Statistical soundness is closely related to SMC. As we show in Chapter 4, statistical soundness is achieved through a generalization technique based on SMC, or more precisely, sequential Bayesian test. In this section, we discuss the similarity and difference between statistical soundness and SMC.

It is not hard to see that both techniques can be applied to check a probabilistic property of the form

$$\Pr(\phi \in [a, b]) \geq \theta_0,$$

where ϕ is a response in a black-box system \mathcal{M} . Recall that SMC solves such a problem through sequential hypothesis testing (see Section 2.2 for a short review). Essentially, it treats the probability $\Pr(\phi \in [a, b])$ as a population parameter θ in a properly defined population, and collects simulation results to accept either $\mathcal{H}_1 : \theta \geq \theta_0$ or $\mathcal{H}_2 : \theta < \theta_0$. On the contrast, using a technique based on statistical soundness, we need to construct a statistically sound relational model g that encloses the response surface of ϕ , and compute (or estimate) the probability $\Pr(g \subseteq [a, b])$. As stated in Theorem 3.3.1, $\Pr(g \subseteq [a, b])$, together with θ_0 , provides a lower bound on θ . Thus, if this lower bound is greater or equal to θ_0 , the property is satisfied. In practice, we often use the empirical bound in (3.9) to check whether a property holds.

These two techniques have their own strengths and weaknesses. SMC can be implemented

easily and applied to any black-box system as long as the system is computable. However, it has been shown that in SMC, the closer θ_0 is to the population parameter θ , the more experiments we are expected to perform in order to accept either hypothesis (see Younes [94] and Jha [42] for details, and Remark 1 in Chapter 2 for an intuitive discussion). On the other hand, statistical soundness based techniques do not suffer from this aspect. But their performance and usefulness depend on the quality of the constructed relational models, which should be a carefully chosen statistical over-approximation of the response surface. As a counterexample, consider a trivial statistically sound model that maps any parameter values to the physical limits of the response. Such a model is indeed statistically sound but is useless in practice. The following example demonstrates these issues in both SMC and statistical soundness.

Example 3.4.1 (Statistical Soundness versus SMC). Let us continue with the two-mass-spring system in Example 3.2.1 and Example 3.3.1. Consider the following property:

$$\Pr(r_o \leq 15\%) \geq 0.45,$$

where r_o is the overshoot of the step response $y(t)$. As mentioned in Example 3.3.1, the yield of the specification $r_o \leq 15\%$ is 46.3%. Therefore, the property should be satisfied.

First, we use Bayesian SMC to verify it. Recall that this involves updating the Bayes factor for each new observation. The Bayes factor is computed as follows,

$$B = \frac{\int_{\theta_0}^1 \theta^{m_1} (1 - \theta)^{m_2} d\theta}{\int_0^{\theta_0} \theta^{m_1} (1 - \theta)^{m_2} d\theta},$$

where m_1 is the number of observations for which $r_o \leq 15\%$ and m_2 is the number that $r_o > 15\%$. Assume that the threshold T of Bayes factor is 100. Several runs of Bayesian SMC show that it typically needs more than 1000 simulations to conclude that the property holds.

Next, we handle the property with a technique, **statistically sound model inference (SSMI)**, that is introduced in Chapter 4. Using ordinary least squares, this technique fits a polynomial \hat{g} to the response surface of r_o . We then generalize \hat{g} into a statistically sound model

g . The property is verified by checking whether the yield computed with respect to g is greater than 0.45. Apparently, the quality of the statistically sound model depends on the accuracy of the polynomial. Compared to more than 1000 simulations in SMC, we use a total of 400 simulations to construct the polynomial \hat{g} and transform it into the statistically sound model g . In Example 3.2.1, we have already seen the statistically sound model based on a first-order polynomial, which is shown below for convenience.

$$\begin{aligned} g_\ell(k, m_1, m_2) &= 0.146 - 0.026k + 0.021m_1 + 0.021m_2, \\ g_u(k, m_1, m_2) &= 0.155 - 0.026k + 0.021m_1 + 0.021m_2, \\ g(k, m_1, m_2) &= [g_\ell(k, m_1, m_2), g_u(k, m_1, m_2)]. \end{aligned}$$

From Example 3.3.1 we know that the yield with respect to g is only 39.1%. Thus, we fail to show that the property is true. An immediate thought is that the accuracy of the first-order polynomials g_ℓ and g_u may not be enough. Hence, we try to construct a cubic polynomial. This leads to the following statistically sound model:

$$\begin{aligned} g_\ell(k, m_1, m_2) &= 0.146 - 0.022k + 0.003k^2 - 0.004k^3 + 0.017m_1 - 0.003km_1 + \\ &\quad 0.004k^2m_1 - 0.005m_1^2 - 0.016km_1^2 + 0.006m_1^3 + 0.019m_2 - \\ &\quad 0.001km_2 - 0.001k^2m_2 - 0.008m_1m_2 - 0.018km_1m_2 + \\ &\quad 0.018m_1^2m_2 - 0.005m_2^2 - 0.017km_2^2 + 0.021m_1m_2^2 + 0.005m_2^3, \\ g_u(k, m_1, m_2) &= 0.154 - 0.022k + 0.003k^2 - 0.004k^3 + 0.017m_1 - 0.003km_1 + \\ &\quad 0.004k^2m_1 - 0.005m_1^2 - 0.016km_1^2 + 0.006m_1^3 + 0.019m_2 - \\ &\quad 0.001km_2 - 0.001k^2m_2 - 0.008m_1m_2 - 0.018km_1m_2 + \\ &\quad 0.018m_1^2m_2 - 0.005m_2^2 - 0.017km_2^2 + 0.021m_1m_2^2 + 0.005m_2^3, \\ g(k, m_1, m_2) &= [g_\ell(k, m_1, m_2), g_u(k, m_1, m_2)]. \end{aligned}$$

Using this model, the estimated yield is 45.1%. Thus we show that the property is true. ||

A few observations from Example 3.4.1 should be mentioned. First, we confirm that SMC does have a performance degradation when the hypothesized population parameter (θ_0) is close to

the true population parameter (θ). Compared with SMC, SSMI requires much fewer simulations in those cases. But it may fail to verify properties without a careful choice of relational models.

In the cases that θ_0 is distant from θ , SMC usually needs fewer simulations than SSMI. This is because SSMI, which constructs polynomial approximations and generalizes them, performs more functionality than SMC, which provides a “likely yes/no” answer. This is in particular useful if we need to tune design parameters in a black-box system to meet certain specifications. The following example shows an application of SSMI that is presented in Chapter 6.

Example 3.4.2 (Optimization in A Two-Mass-Spring System). From Example 3.3.1, we know that in the two-mass-spring system, the yield of the response specification $r_o \leq 15\%$ is only 46.3% at the nominal design point. Recall that the system has two tunable design parameters, pole location $p = -1000$ and zero location $z = -1$. Let us try and see if we can find a design point in $p \in [-1200, -800]$ and $z \in [-1.2, -0.8]$, such that the specification has a better yield.

We do this by constructing a statistically sound model that is different than the one in Example 3.4.1. The model, in terms of the design parameters p and z , is shown as follows.

$$g_\ell(p, z) = 0.121 + 0.006p + 0.078z,$$

$$g_u(p, z) = 0.198 + 0.017p - 0.086z,$$

$$g(p, z) = [g_\ell(p, z), g_u(p, z)].$$

It is 95% statistically sound with respect to the nominal design point. Next, we search for a new design point that satisfy the specification with respect to the model g . The detailed approach, which is introduced in Chapter 6, is skipped here. Intuitively, we pick up the point that is most likely to satisfy the specification according to g , and try to verify in the concrete system that it is indeed safe. In this example, a new design point $p = -1200$ and $z = -0.928$ is found, which leads to a 100% yield of the specification $r_o \leq 15\%$. ||

In principle, it is possible to apply SMC to design parameter tuning (see, e.g., Palaniappan et al. [68] and Jha et al. [44]). Such an approach often involves searching individual design parameter values and running SMC for each of them. It can, however, result in prohibitively large

number of simulations. Compared to those approaches, the technique that we introduce is more straightforward and requires less computation.

A final comparison between SMC and statistical soundness is on the ability to verify a sequence of properties. SMC requires a new run of simulations whenever a new property comes. For instance, for a specification $\phi \in [a, b]$, we may fail to verify that $\Pr(\phi \in [a, b]) \geq 90\%$ and decide to try with $\Pr(\phi \in [a, b]) \geq 80\%$. In this case, SMC needs to generate new simulation traces in order to guarantee that the observations are truly random (see Clarke et al. [19]). On the contrast, SSMI only involves evaluations of the constructed model. For large systems in which each individual simulation takes a long time, SSMI can significantly reduce the computational cost.

Chapter 4

Statistically Sound Model Inference

As discussed in Chapter 3, the behavior of a black-box system can be modeled by a set of real-valued responses. For example, in a ring oscillator, oscillation frequency, phase noise and power consumption are all important responses that define how the circuit behaves. The correctness and performance properties of a system are expressed as ranges over the responses. For a system to work well, each response usually has an acceptable range. This is called a response specification (see Definition 3.1.2). If the system satisfies all the specifications, we say that it is safe. In practice, it can be difficult to design safe systems due to process variations, external perturbations and many other factors. Hence, for an unsafe system, we are interested in its probability of being safe. This is known as the yield of a system. Obviously, a safe system has a 100% yield.

Recall that there are two types of parameters which affect the behavior of a black-box system \mathcal{M} : design parameters \mathbf{u} and stochastic parameters \mathbf{x} . Design parameters, as the name suggests, are controlled by designers. They are used to tune the system so that it operates as expected. This chapter does not consider these types of parameters. A technique that deals with design parameters is discussed in Chapter 6. On the other hand, stochastic parameters are considered uncontrollable, arising primarily due to the randomness in the environment, the manufacturing process, and a lack of understanding of the physics involved in system design. They are usually assumed to follow certain distributions, such as a (truncated) normal distribution or an exponential distribution. Variations of stochastic parameters can result in the responses of the system deviating from the ideal behavior. In the worst case, they may lead to low yield and expensive failures.

This chapter introduces a simulation-based technique called **statistically sound model inference (SSMI)**. For a black-box system under stochastic parameter variations, SSMI constructs a statistically sound model of each response in the system and computes the yield of a specification with respect to the model. To achieve statistical soundness, it combines ordinary least squares (OLS) regression and a generalization technique that is developed using statistical model checking (SMC). The models produced by SSMI can be useful to designers. For instance, besides yield estimation, they can also be used to learn the distribution of a response, plot safe regions of the parameter space and identify the sensitivity of the response in the stochastic parameters. The content of this chapter is originally published by [97].

We organize this chapter as follows. First, we present an overview of SSMI. The technical details are introduced in Section 4.2 and Section 4.3. In particular, Section 4.2 discusses how to apply OLS regression to construct a basis functional model, and Section 4.3 shows a generalization technique that transforms the basis model into a statistically sound model. Finally, we demonstrate the capability of SSMI with several applications.

4.1 Overview

Consider a black-box system $\mathcal{M} = (\mathbf{u}, \mathbf{x}, \phi, F_{\mathbf{X}}, r)$ as in Definition 3.1.1. This chapter and Chapter 5 assume that the design parameters \mathbf{u} are fixed to the nominal values, and are only concerned with the stochastic parameters \mathbf{x} . Hence, we omit \mathbf{u} in both \mathcal{M} and r and simply write $\mathcal{M} = (\mathbf{x}, \phi, F_{\mathbf{X}}, r)$ and $r(\mathbf{x})$. With this assumption, the response surface $r(\mathbf{x})$ depends only on the stochastic parameters \mathbf{x} . Before we proceed, let us introduce a running example that is used throughout the discussion of SSMI.

Example 4.1.1 (A Basic Buck Converter). Figure 4.1 shows the circuit diagram of a buck converter. A buck converter is a DC-DC converter that converts higher-level DC input voltages to lower-level DC output voltages. It is an important analog circuit and is widely used in portable electronic devices such as cellular phones and laptop computers. The circuit in Figure 4.1 represents a basic buck converter. The transistors S_p and S_n are regarded as ideal switches. The voltage

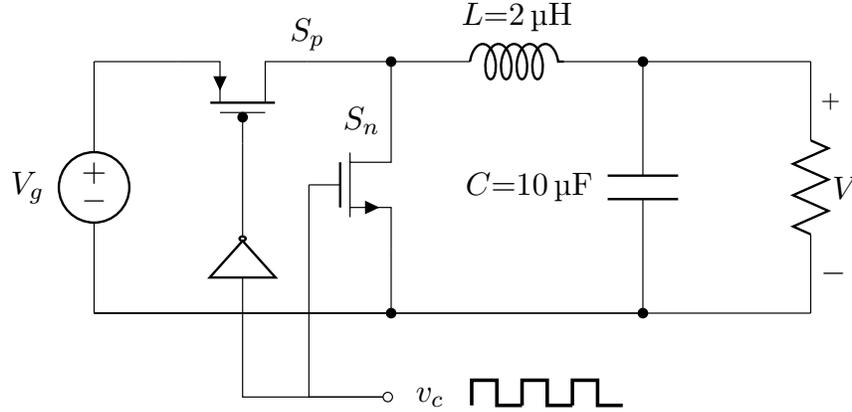


Figure 4.1: A basic buck converter in which L and C are considered as uniform random variables.

v_c is a square wave. It controls the switches S_p and S_n so that only one of them is on at any time. The switching between S_p and S_n results in the charging and discharging of the capacitor C . Effectively, the inductor L and the capacitor C form a low-pass filter which filters out the high frequency component of the signal. Thus, the output V is a small changing wave consisting of a large DC component and some low frequency oscillations which are called voltage ripples.

An important performance metric for buck converters is the amplitude of the voltage ripple, denoted as Δv . Usually, a specification of the form $\Delta v \leq v_0$ is required to ensure the functionality of a buck converter. This example assumes that $v_0 = 30$ mV. When we design this circuit, we choose nominal value for the inductor L and the capacitor C to be $L = 2$ μ H and $C = 10$ μ F. However, for a manufactured circuit, it is quite unlikely that L and C are exactly the nominal values. Due to stochastic parameter variations, it is possible that the nominal design satisfies the specification $\Delta v \leq 30$ mV whereas the implemented circuit has $\Delta v > 30$ mV.

To analyze the correctness of the circuit in the presence of parameter variations, we treat L and C as stochastic parameters, $\mathbf{x} = (L, C)$, with the following uniform distributions:

$$L \sim U(1.8, 2.2)\mu\text{H}, C \sim U(9, 11)\mu\text{F}. \quad (4.1)$$

Since the system is simple, the response surface of the voltage ripple Δv can be derived in terms

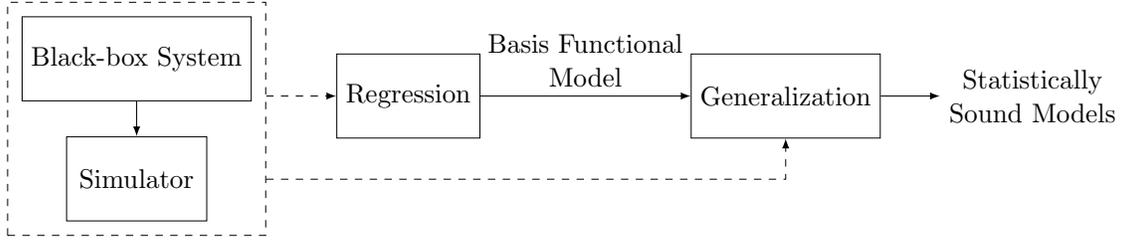


Figure 4.2: A high-level flow of SSML.

of L and C , assuming that the transistors act as ideal switches.

$$\Delta v = r(L, C) = \frac{V_g - V}{16LC} DT_s^2, \quad (4.2)$$

where V is the DC component of the output voltage, D is the duty cycle and T_s is the time period of the control voltage v_c . Let $V = 3\text{ V}$, $D = 0.25$ and $T_s = 2\text{ }\mu\text{s}$. From (4.2), we can show that the system meets the specification when

$$LC \geq 18.75\text{ }\mu\text{H}\mu\text{F},$$

which is not always the case given the distributions of L and C in (4.1). Note that unless the system is as simple as this buck converter, it is usually impossible to derive a closed-form representation of the response surface. ||

For a black-box system $\mathcal{M} = (\mathbf{x}, \phi, F_{\mathbf{X}}, r)$, SSML aims to construct a θ_0 statistically sound relational model $g(\mathbf{x})$ of the response surface $r(\mathbf{x})$ such that

$$\Pr_{F_{\mathbf{X}}(\mathbf{x})} (r(\mathbf{x}) \in g(\mathbf{x})) \geq \theta_0.$$

To achieve this, the key idea is to combine the strengths of regression and SMC. SSML first builds an accurate functional approximation $\hat{g}(\mathbf{x})$ of the response surface $r(\mathbf{x})$ and then generalizes $\hat{g}(\mathbf{x})$ into a statistically sound model $g(\mathbf{x})$. Figure 4.2 shows a high-level flow of SSML, which consists of two steps: **regression** and **generalization**.

4.1.1 Regression

In this step, SSMI applies ordinary least squares (OLS) to a set of simulation data to compute an approximation $\hat{g}(\mathbf{x})$ of the response surface $r(\mathbf{x})$. The model $\hat{g}(\mathbf{x})$ is called the **basis functional model**. Simulation data are collected by performing random sampling of the stochastic parameters \mathbf{x} following the joint distribution $F_{\mathbf{X}}(\mathbf{x})$, and simulating the system \mathcal{M} for each of the sampled data points. We denote the data for i th simulation as $(\mathbf{x}^{(i)}, \phi^{(i)})$.

For curve fitting, we use polynomials as target functions. Hence $\hat{g}(\mathbf{x})$ has the following form:

$$\hat{g}(x_1, \dots, x_n) = \sum_{d_1 + \dots + d_n \leq d} c_{d_1, \dots, d_n} \cdot x_1^{d_1} \cdots x_n^{d_n},$$

where $d_i \geq 0$, d is the degree of the polynomial and c_{d_1, \dots, d_n} are the unknown coefficients. OLS computes the coefficients such that the \mathcal{L}_2 error between $\hat{g}(\mathbf{x})$ and $r(\mathbf{x})$ is minimized with respect to the simulation data,

$$\min_{c_{d_1, \dots, d_n}} \sum_{i=1}^N \left(\phi^{(i)} - \hat{g}(\mathbf{x}^{(i)}) \right)^2.$$

Example 4.1.2 (A Basic Buck Converter - Regression). Let us continue with the buck converter in Example 4.1.1. To reason about the behavior of the voltage ripple Δv under stochastic parameter variations, we compute a basis functional model $\hat{g}(L, C)$ using OLS regression. We choose a quadratic polynomial as the target function. Using 20 simulations, we obtain the following function:

$$\hat{g}(L, C) = 28.1 - 2.82L + 0.31L^2 - 2.82C + 0.28LC + 0.30C^2.$$

Note that L and C are normalized to $[-1, 1]$ and the unit of $\hat{g}(L, C)$ is mV. ||

As we have learned, as a functional approximation of the response surface $r(\mathbf{x})$, $\hat{g}(\mathbf{x})$ provides little guarantee on the behavior of the system \mathcal{M} . A statistical soundness guarantee is achieved through the next step, generalization.

4.1.2 Generalization

In this step, SSMI derives a **tolerance interval** $I = [\ell, u]$ that generalizes the basis functional model $\hat{g}(\mathbf{x})$ into a θ_0 statistically sound relational model $g(\mathbf{x})$ defined as

$$g(\mathbf{x}) \equiv [\hat{g}(\mathbf{x}) + \ell, \hat{g}(\mathbf{x}) + u],$$

Tolerance intervals are derived using Bayesian SMC. Let $\theta_0 \in (0, 1)$ be a given probability. SSMI formulates a pair of hypotheses

$$\mathcal{H}_1 : \Pr_{F_{\mathbf{x}(\mathbf{x})}}(r(\mathbf{x}) \in [\hat{g}(\mathbf{x}) + \ell, \hat{g}(\mathbf{x}) + u]) \geq \theta_0,$$

$$\mathcal{H}_2 : \Pr_{F_{\mathbf{x}(\mathbf{x})}}(r(\mathbf{x}) \in [\hat{g}(\mathbf{x}) + \ell, \hat{g}(\mathbf{x}) + u]) < \theta_0.$$

For an interval $I = [\ell, u]$, it simulates the system with sequentially sampled data points and checks whether \mathcal{H}_1 can be accepted. If \mathcal{H}_1 is accepted, it indicates that the model $g(\mathbf{x})$ with the current I is a θ_0 statistically sound model of $r(\mathbf{x})$. Otherwise, the interval I is updated and the test is performed with the new interval.

Example 4.1.3 (A Basic Buck Converter - Generalization). Continued from Example 4.1.2, we generalize the basis functional model $\hat{g}(L, C)$. This takes 102 simulations, yields a tolerance interval $I = [-75, 73] \mu\text{V}$ with $\theta_0 = 0.95$ and $T = 100$. Hence, we have a 95% statistically sound model

$$g_\ell(L, C) = 27.9 - 2.82L + 0.31L^2 - 2.82C + 0.28LC + 0.30C^2,$$

$$g_u(L, C) = 28.2 - 2.82L + 0.31L^2 - 2.82C + 0.28LC + 0.30C^2,$$

$$g(L, C) = [g_\ell(L, C), g_u(L, C)].$$

Now we use the model $g(L, C)$ to verify the specification $\Delta v \leq 30 \text{ mV}$. The yield with respect to $g(L, C)$ is 77.3%, compared to the true yield 77.6% according to (4.1) and (4.2). A statistically safe region in the parameter space is implicitly defined by the model $g(L, C)$ and the specification. In Figure 4.3, the shaded region is the safe parameter values predict by SSMI. On the contrast, the solid line shows the analytic boundary

$$LC = 18.75 \mu\text{H} \mu\text{F}.$$

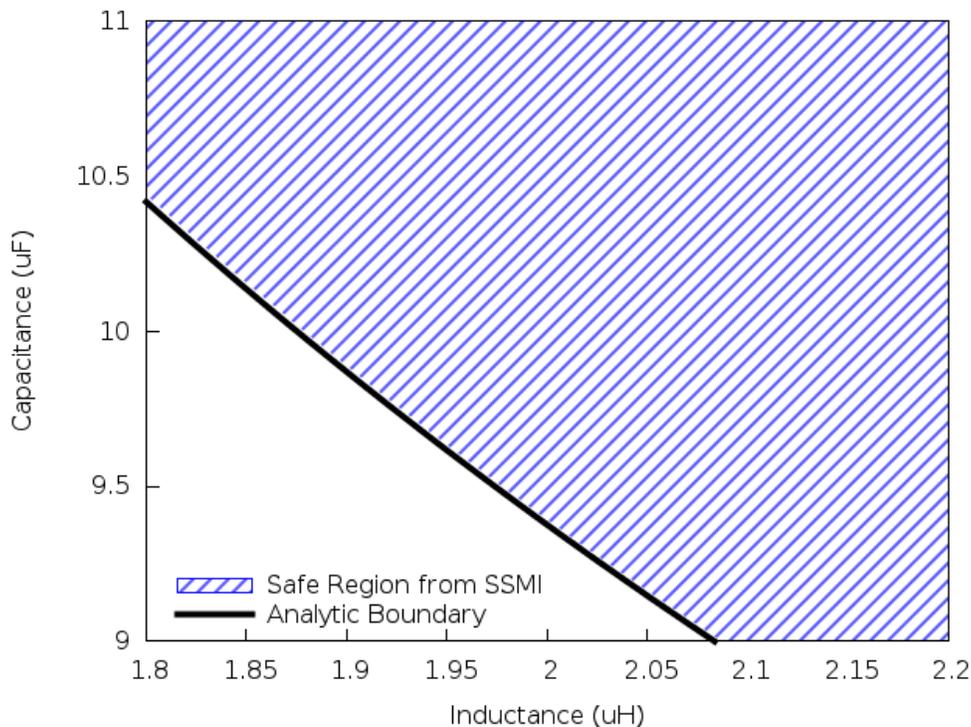


Figure 4.3: A comparison between the safe regions of the basic buck converter predicted by SSMI and derived from Equation (4.2).

The region above the solid line is the true safe region. Note that we can barely observe the difference between the two regions. ||

The next section presents the technical details in OLS regression and generalization.

4.2 Ordinary Least Squares Regression

It is well known that any continuous function over a bounded domain can be approximated “arbitrarily closely” by a polynomial. In practice, the degree of a polynomial is often fixed in advance, leaving the coefficients unknown. Such a polynomial is called the target function. To find a good approximation, we need to compute the coefficients so that the error between the target function and the function to be approximated is minimized. The following part of this section shows a simple scheme based on OLS regression. A more scalable approach is introduced in Chapter 5.

We define a degree vector $\mathbf{d} = (d_1, \dots, d_n)$ to be a vector of positive integers. The vector can

be compared with a degree $d \geq 0$ such that

$$\mathbf{d} \leq d \Leftrightarrow \sum_{i=1}^n d_i \leq d.$$

With a degree vector $\mathbf{d} = (d_1, \dots, d_n)$, the product $x_1^{d_1} \cdots x_n^{d_n}$ can be compactly written as $\mathbf{x}^{\mathbf{d}}$.

Hence a polynomial $\hat{g}(\mathbf{x})$ of degree d has the following form:

$$\hat{g}(\mathbf{x}) = \sum_{\mathbf{d} \leq d} c_{\mathbf{d}} \mathbf{x}^{\mathbf{d}}.$$

Let $\mathbf{c} = (c_{\mathbf{d}_1}, c_{\mathbf{d}_2}, \dots)$ be the vector of unknown coefficients. Assume that the response surface $r(\mathbf{x})$ is a continuous function. With a random sample $(\mathbf{x}^{(i)}, \phi^{(i)})$ of size N , the coefficients in the target function are determined by solving

$$\min_{\mathbf{c}} \sum_{i=1}^N \left(\phi^{(i)} - \hat{g}(\mathbf{x}^{(i)}) \right)^2.$$

4.2.1 A Resampling Heuristic

For OLS regression, there is a well-known lower bound on the sample size N in order to avoid over-fitting. Let n be the number of stochastic parameters and d be the degree of the target polynomial. The minimum sample size follows the inequality

$$N \geq \binom{n+d}{d} = \frac{(n+d)!}{n! \cdot d!}. \quad (4.3)$$

In practice, such a lower bound is often not enough to obtain a good fit. This is especially a problem when there are many stochastic parameters. In these cases, smaller sample sizes are more likely to result in partial explorations of the parameter space, resulting in a polynomial approximation that represents an artifact of the simulation data rather than the behavior of the system. On the other hand, a large sample size incurs unnecessary overheads in both simulation and regression.

From a statistical point of view, OLS regression is a kind of **point estimation** [17]. The classic approach to determine the sample size for a point estimation is to specify a desired tolerance of error and compute a sample size that is large enough so that with a high statistical confidence, the distance between the estimated point and the true point is smaller than the error. This approach is

Input: Black-box System \mathcal{M} , Simulation Data $D = ((\mathbf{x}^{(1)}, \phi^{(1)}), \dots, (\mathbf{x}^{(N)}, \phi^{(N)}))$,
Distance ϵ , Sampling Factor s , Number of Folds k

Output: Polynomial $\hat{g}(\mathbf{x})$

```

1 while true do
2    $D_1, \dots, D_k =$  split data into  $k$  folds such that  $D_i \cap D_j = \emptyset$  and  $\bigcup D_i = D$  ;
3   for  $i \leftarrow 1$  to  $k$  do
4      $\hat{g}_i(\mathbf{x}) =$  apply OLS to the set of data  $\bigcup_{j \neq i} D_j$  ;
5   end
6    $\epsilon_{ij} =$  compute Euclidean distance between the coefficients of  $\hat{g}_i(\mathbf{x})$  and those of  $\hat{g}_j(\mathbf{x})$  ;
7   if for any  $i, j, \epsilon_{ij} \geq \epsilon$  then
8      $D' =$  run  $s \cdot N$  random simulations ;
9      $D = D \cup D'$  ;
10  else
11     $\hat{g}(\mathbf{x}) = \frac{1}{k} \sum_{i=1}^k \hat{g}_i(\mathbf{x})$  ;
12    break ;
13  end
14 end

```

Algorithm 1: A resampling heuristic to construct good polynomial approximations.

effective when there are only a few population parameters to estimate. However, for OLS regression, there are often tens or hundreds of coefficients that need to be determined.

We introduce a heuristic called **resampling**. This heuristic determines whether a given set of simulation data are large enough to produce a good approximation. If not, it runs more simulations until a good approximation can be obtained. The heuristic is illustrated in Algorithm 1. Consider a random sample of size N . the heuristic first partitions it into k folds (typically k ranges from 4 to 10) of $\frac{N}{k}$ data points. Then it constructs k different sets, each with $N \cdot \frac{k-1}{k}$ data points, such that each fold is left out exactly once. Using each set of data, k polynomials $\hat{g}_i, i = 1, \dots, k$ are computed through OLS regression. Due to the randomized sampling, if the sample size N is large enough, each set should contain data that are spread in the stochastic parameter space, thus leading to k polynomials with similar coefficients. Hence, the sample size N is considered to be large enough if the Euclidean distance between the coefficients of different polynomials are smaller than a given distance ϵ , i.e.,

$$\|\mathbf{c}_i - \mathbf{c}_j\| < \epsilon, i \neq j, i, j = 1, \dots, k, \quad (4.4)$$

If (4.4) is satisfied, we construct $\hat{g}(\mathbf{x})$ by taking average of the k polynomials. On the other hand,

if (4.4) does not hold, we run another $s \cdot N$ random simulations and collect the data to form a larger data set. The resampling heuristic provides a systematic way to determine the sample size and reduces the bias of the resulting approximation.

Example 4.2.1 (A Basic Buck Converter - Resampling). With the discussion in this section, let us reveal more details in Example 4.1.2. Recall that we use 20 simulations to build the basis functional model $\hat{g}(L, C)$, which is a quadratic polynomial. Given the number of stochastic parameters $n = 2$ and the polynomial degree $d = 2$, the minimum number of data points in order to avoid over-fitting is 6. For resampling, we would like to use $k = 5$ folds. Hence the minimum number of data points is $6 \times \frac{5}{4} \approx 8$. We choose to use 10 data points initially. However, this data set does not yield consistent polynomials for $\epsilon = 0.05$. The coefficients in the following two polynomials

$$\hat{g}_1(L, C) = 28.1 - 2.81L + 0.31L^2 - 2.81C + 0.28LC + 0.27C^2,$$

$$\hat{g}_4(L, C) = 28.1 - 2.83L + 0.30L^2 - 2.77C + 0.28LC + 0.30C^2,$$

have a Euclidean distance of 0.055. Using a sampling factor $s = 1$, we run another 10 simulations. With a total of 20 data points, we build the model $\hat{g}(L, C)$ as shown in Example 4.1.2. ||

Remark 6. In practice, the idea of resampling can be applied in more general cases. For example, there are cases that we are limited to a given set of simulation data, either because the system is not available or the simulation cost is too high. To improve the approximation accuracy with a limited amount of data, we can follow the resampling procedure to compute k polynomials and construct a single approximation by taking the average. ◇

4.2.2 Complexity

The computational complexity of the regression step is dominated by OLS. Assume that the sample size N is large enough to avoid over-fitting (i.e., satisfies (4.3)). It can be shown that OLS has a time complexity of $O(N \cdot |\mathbf{c}|^2)$ where $|\mathbf{c}| = \binom{n+d}{d}$ is the number of unknown coefficients in a target polynomial of degree d [34]. Fixing the degree d , we have

$$|\mathbf{c}| = \prod_{i=1}^d (n - i + 1).$$

The time complexity becomes $O(N \cdot n^{2d})$. On the other hand, OLS has a space complexity of $O(N \cdot n^d)$ wherein the matrices that represent the polynomial terms are fully dense. Apparently, as the number of stochastic parameters grows, OLS quickly becomes inefficient in both time and memory. As a solution, Chapter 5 shows a more sophisticated regression technique to handle the cases of many stochastic parameters.

4.3 Generalization

4.3.1 Tolerance Interval

Recall that a polynomial approximation $\hat{g}(\mathbf{x})$ of the response surface $r(\mathbf{x})$ provides little guarantee on the behavior of the system. This section introduces a generalization technique that aims to derive a tolerance interval I that generalizes $\hat{g}(\mathbf{x})$ into a statistically sound model $g(\mathbf{x})$. First, we formally define the meaning of a tolerance interval.

Definition 4.3.1 (Tolerance Interval). For a statistical population \mathcal{P} , a real-valued interval $I = [a, b]$, $a, b \in \mathbb{R}$ is a $(\theta_0, 1 - \alpha)$ tolerance interval if with a level $1 - \alpha$ of confidence, for any individual $p \in \mathcal{P}$,

$$\Pr(p \in I) \geq \theta_0.$$

In SSMI, a $(\theta_0, 1 - \alpha)$ tolerance interval $I = [\ell, u]$ is associated with a basis functional model $\hat{g}(\mathbf{x})$ of the response surface $r(\mathbf{x})$. The population, with respect to which the tolerance interval is defined, is the difference between $r(\mathbf{x})$ and $\hat{g}(\mathbf{x})$ for any $\mathbf{x} \in \mathbb{X}$, i.e.,

$$\mathcal{P} \equiv \{r(\mathbf{x}) - \hat{g}(\mathbf{x}) \mid \mathbf{x} \in \mathbb{X}\}. \quad (4.5)$$

Thus, an individual in the population is the difference between $r(\mathbf{x})$ and $\hat{g}(\mathbf{x})$ for some \mathbf{x} . The basis functional model $\hat{g}(\mathbf{x})$ and the tolerance interval I form a relational model $g(\mathbf{x})$,

$$g(\mathbf{x}) \equiv [\hat{g}(\mathbf{x}) + \ell, \hat{g}(\mathbf{x}) + u]. \quad (4.6)$$

With the formulation in (4.5) and (4.6), we can show that $g(\mathbf{x})$ is θ_0 statistically sound model of $r(\mathbf{x})$ and such a guarantee is provided with a $(1 - \alpha)$ confidence.

Theorem 4.3.1 (Statistical Soundness with Tolerance Interval). Consider a basis functional model $\hat{g}(\mathbf{x})$ of a response surface $r(\mathbf{x})$ and a $(\theta_0, 1 - \alpha)$ tolerance interval $I = [\ell, u]$ whose population is defined by (4.5). With a $(1 - \alpha)$ confidence, a relational model

$$g(\mathbf{x}) \equiv [\hat{g}(\mathbf{x}) + \ell, \hat{g}(\mathbf{x}) + u]$$

is a θ_0 statistically sound model of $r(\mathbf{x})$.

Proof. By the definition of statistical soundness (Definition 3.2.3), $g(\mathbf{x})$ is a θ_0 statistically sound model of $r(\mathbf{x})$, or equivalently $r(\mathbf{x}) \prec_{\theta_0} g(\mathbf{x})$, if

$$\Pr_{F_{\mathbf{x}}(\mathbf{x})} (r(\mathbf{x}) \in g(\mathbf{x})) \geq \theta_0.$$

Substituting the definition of $g(\mathbf{x})$ in (4.6) into the above inequality, we need to show that

$$\Pr_{F_{\mathbf{x}}(\mathbf{x})} (r(\mathbf{x}) - \hat{g}(\mathbf{x}) \in I) \geq \theta_0. \quad (4.7)$$

This is immediate by the definition of tolerance interval with a population defined in (4.5). Hence, we have shown that $g(\mathbf{x})$ is a θ_0 statistically sound model of $r(\mathbf{x})$.

Note that since I is a $(\theta_0, 1 - \alpha)$ tolerance interval, we have $(1 - \alpha)$ confidence that (4.7) is true. Thus the level of confidence that $r(\mathbf{x}) \prec_{\theta_0} g(\mathbf{x})$ is also $(1 - \alpha)$. \square

4.3.2 Algorithms for the Derivation of Tolerance Intervals

Given a black-box system $\mathcal{M} = (\mathbf{x}, \phi, F_{\mathbf{x}}, r)$ and a basis functional model $\hat{g}(\mathbf{x})$ of the response surface $r(\mathbf{x})$, we are interested in deriving a tolerance interval. To understand this, let us first consider a simpler problem. Suppose that we have an interval I . For given θ_0 and α , how do we show whether I is a $(\theta_0, 1 - \alpha)$ tolerance interval with respect to $\hat{g}(\mathbf{x})$ and $r(\mathbf{x})$? It is not hard to see that this is equivalent to the problem of checking whether $g(\mathbf{x})$, as defined in (4.6), is a θ_0 statistically sound model of $r(\mathbf{x})$ with a confidence of $(1 - \alpha)$. From Chapter 3, we learn that statistical soundness can be guaranteed by sequential hypothesis testing. Given an interval

$I = [\ell, u]$, the problem is formulated as a pair of hypotheses,

$$\mathcal{H}_1 : \Pr_{F_{\mathbf{x}(\mathbf{x})}} (r(\mathbf{x}) \in [\hat{g}(\mathbf{x}) + \ell, \hat{g}(\mathbf{x}) + u]) \geq \theta_0,$$

$$\mathcal{H}_2 : \Pr_{F_{\mathbf{x}(\mathbf{x})}} (r(\mathbf{x}) \in [\hat{g}(\mathbf{x}) + \ell, \hat{g}(\mathbf{x}) + u]) < \theta_0.$$

This can be solved by sequential Bayesian test introduced in Section 2.1.2. Recall that we need to repeatedly compute the Bayes factor

$$B = \frac{\Pr(D | \mathcal{H}_1)}{\Pr(D | \mathcal{H}_2)},$$

where $D = (z_1, z_2, \dots)$ is a collection of random variates of a Bernoulli random variable denoting the outcome of the relation

$$r(\mathbf{x}^{(i)}) \in [\hat{g}(\mathbf{x}^{(i)}) + \ell, \hat{g}(\mathbf{x}^{(i)}) + u] \quad (4.8)$$

such that $z_i = 1$ if (4.8) is true for the data point $\mathbf{x}^{(i)}$ and $z_i = 0$ otherwise. The relation in (4.8) is called an **inclusion test**. If $\mathbf{x}^{(i)}$ passes the inclusion test, it is said to be in favor of \mathcal{H}_1 . Otherwise, it is in favor of \mathcal{H}_2 .

If the Bayes factor B grows larger than a pre-defined threshold T , we accept \mathcal{H}_1 and conclude that $g(\mathbf{x}) \equiv [\hat{g}(\mathbf{x}) + \ell, \hat{g}(\mathbf{x}) + u]$ is a θ_0 statistically sound model of $r(\mathbf{x})$. The level of confidence is indicated by the threshold T . As shown in Section 2.1.2.3, the Type I error α of sequential Bayesian test is bounded by the following inequality:

$$\alpha \leq \frac{1}{T + 1}.$$

Hence, the level of confidence $(1 - \alpha)$ that \mathcal{H}_1 is accepted correctly satisfies

$$1 - \alpha \geq \frac{T}{T + 1}.$$

The interval $I = [\ell, u]$ is thus a $\left(\theta_0, \frac{T}{T + 1}\right)$ tolerance interval with respect to the basis functional model $\hat{g}(\mathbf{x})$ and the response surface $r(\mathbf{x})$.

However, the goal is to derive a tolerance interval rather than checking whether a given interval is a tolerance interval. To solve this problem, we introduce a generalization procedure

Input: Black-box System \mathcal{M} , Basis Functional Model $\hat{g}(\mathbf{x})$, Probability θ_0 , Threshold T

Output: Tolerance Interval $I = [\ell, u]$

```

1  $I = [0, 0]$  ;
2  $B = 0$  ;
3 while  $B < T$  do
4    $\mathbf{x}^{(i)}$  = draw a point following the distribution of the stochastic parameters ;
5    $\phi^{(i)}$  = simulate  $\mathcal{M}$  with  $\mathbf{x} = \mathbf{x}^{(i)}$  ;
6   if  $\phi^{(i)} \notin [\hat{g}(\mathbf{x}^{(i)}) + \ell, \hat{g}(\mathbf{x}^{(i)}) + u]$  then
7      $B = 0$  ;
8      $\ell = \min(\ell, \phi^{(i)} - \hat{g}(\mathbf{x}^{(i)}))$  ;
9      $u = \max(u, \phi^{(i)} - \hat{g}(\mathbf{x}^{(i)}))$  ;
10  else
11     $B =$  recompute Bayes factor taking  $(\mathbf{x}^{(i)}, \phi^{(i)})$  into account ;
12  end
13 end

```

Algorithm 2: A generalization procedure that derives tolerance intervals.

based on sequential Bayesian test in Algorithm 2. The algorithm starts with a zero interval $I = [\ell, u], \ell = u = 0$. It repeatedly performs the inclusion test $\phi^{(i)} \in [\hat{g}(\mathbf{x}^{(i)}) + \ell, \hat{g}(\mathbf{x}^{(i)}) + u]$ with data points $(\mathbf{x}^{(i)}, \phi^{(i)})$ drawn following the cdf $F_{\mathbf{X}}(\mathbf{x})$ of the stochastic parameters. We say that the inclusion test is a success if the point $(\mathbf{x}^{(i)}, \phi^{(i)})$ passes it, and a failure otherwise. Upon each success, the algorithm updates the Bayes factor and continues until a failure occurs. In this case, it updates ℓ and u of the interval I to enclose the data point $(\mathbf{x}^{(i)}, \phi^{(i)})$ that causes the failure. Also, the Bayes factor B is reset to zero to indicate that we start with a new interval. The algorithm terminates when the Bayes factor B grows larger than the threshold T .

Now let us take a deeper look at Algorithm 2. The algorithm sequentially draws random data points until the Bayes factor B grows large enough. Suppose that the sequence $D = (z_1, z_2, \dots)$ indicates the outcome of each inclusion test, where $z_i = 1$ refers to a success and $z_i = 0$ a failure. From the description of the algorithm, we know that only those data points that pass the inclusion test contribute to the growth of the Bayes factor. The others are instead used to update the interval I . Furthermore, for the algorithm to terminate, there must be enough consecutive successful inclusion tests at the end of the sequence D . Let K be the number of **consecutive** observations

Input: Black-box System \mathcal{M} , Basis Functional Model $\hat{g}(\mathbf{x})$, Probability θ_0 , Threshold T

Output: Tolerance Interval $I = [\ell, u]$

```

1  $K = -\frac{\log(T+1)}{\log \theta_0} - 1$  ;
2  $I = [0, 0]$  ;
3 count = 0 ;
4 while count <  $K$  do
5    $\mathbf{x}^{(i)}$  = draw a point following the distribution of the stochastic parameters ;
6    $\phi^{(i)}$  = simulate  $\mathcal{M}$  with  $\mathbf{x} = \mathbf{x}^{(i)}$  ;
7   if  $\phi^{(i)} \notin [\hat{g}(\mathbf{x}^{(i)}) + \ell, \hat{g}(\mathbf{x}^{(i)}) + u]$  then
8     count = 0 ;
9      $\ell = \min(\ell, \phi^{(i)} - \hat{g}(\mathbf{x}^{(i)}))$  ;
10     $u = \max(u, \phi^{(i)} - \hat{g}(\mathbf{x}^{(i)}))$  ;
11  else
12    count = count + 1 ;
13  end
14 end

```

Algorithm 3: A simplified generalization procedure.

that support \mathcal{H}_1 at the end of D . When the algorithm terminates, the Bayes factor is

$$B = \frac{\int_{\theta_0}^1 \theta^{m-1} (1-\theta)^{m_2} d\theta}{\int_0^{\theta_0} \theta^{m-1} (1-\theta)^{m_2} d\theta} = \frac{\int_{\theta_0}^1 \theta^K d\theta}{\int_0^{\theta_0} \theta^K d\theta} = \frac{1 - \theta_0^{K+1}}{\theta_0^{K+1}}.$$

Since $B \geq T$, we have

$$\frac{1 - \theta_0^{K+1}}{\theta_0^{K+1}} \geq T.$$

Let us rearrange the inequality into

$$\frac{1}{\theta_0^{K+1}} \geq T + 1.$$

Taking logarithm on both sides of the inequality, we have

$$-(K+1) \cdot \log \theta_0 \geq \log(T+1).$$

Since $\theta_0 \in (0, 1)$, $-\log \theta_0$ is positive. Hence,

$$K \geq -\frac{\log(T+1)}{\log \theta_0} - 1. \quad (4.9)$$

We say that $K = -\frac{\log(T+1)}{\log \theta_0} - 1$ is the **run length** of the algorithm. The inequality (4.9) provides a lower bound of K such that when we collect K consecutive observations that support

\mathcal{H}_1 , we can terminate the algorithm and conclude that the resulting interval is a $\left(\theta_0, \frac{T}{T+1}\right)$ tolerance interval. As a consequence, we do not need to compute the Bayes factor repeatedly. With this inequality, we introduce a new generalization procedure, shown in Algorithm 3, that simplifies Algorithm 2. As before, we start with a zero interval and repeatedly draw data points. But instead of computing Bayes factors, we use a variable `count` to record the number of successes in the inclusion test. Once we observe a run of K consecutive successes, we terminate the algorithm.

Theorem 4.3.2 (Equivalence of Two Generalization Algorithms). For a black-box system \mathcal{M} , the tolerance intervals produced by Algorithm 2 and Algorithm 3 are the same, given that $\hat{g}(\mathbf{x})$, θ_0 and T are the same for the two algorithms.

Proof. The equivalence is proved by the construction of Algorithm 3 discussed above. \square

Table 4.1 shows some values of the run length K for the given probability θ_0 and threshold T . Increasing θ_0 and T yields a larger K , which in turn results in a statistically sound model with better statistical guarantee and higher level of confidence. From (4.9), it can be easily shown that the growth of K is more sensitive to the growth of θ_0 . In practice, we find that $\theta_0 = 0.95$ and $T = 100$ provide a good trade-off between statistical guarantee and computational cost.

Table 4.1: Run length K for common values of θ_0 and T .

	$T = 10$	30	100	500	1000
$\theta_0 = 0.9$	22	32	43	59	65
0.95	46	66	89	121	134
0.99	238	341	459	618	687
0.999	2396	3432	4612	6213	6905

The following theorem is concerned with the termination of the generalization procedure.

Theorem 4.3.3 (Termination of Generalization). The generalization procedure shown in Algorithm 2 and Algorithm 3 terminates with probability one.

Proof. Since the two algorithm are equivalent as shown by Theorem 4.3.2, we prove the termination of Algorithm 3. For a black-box system $\mathcal{M} = (\mathbf{x}, \phi, F_{\mathbf{X}}, r)$ and a basis functional model $\hat{g}(\mathbf{x})$ of the

response surface $r(\mathbf{x})$, let $I_h = [\ell_h, u_h]$ be the minimum hypothetical tolerance interval such that for all $\mathbf{x} \in \mathbb{X}$, and for any $\ell \leq \ell_h$ and $u \geq u_h$,

$$\hat{g}(\mathbf{x}) + \ell \leq r(\mathbf{x}) \leq \hat{g}(\mathbf{x}) + u.$$

Suppose that at the i th step of the loop of Algorithm 3, the interval is $I_i = [\ell_i, u_i]$. We construct two sequences (p_1, p_2, \dots) and (q_1, q_2, \dots) such that

$$p_i = \ell_i - \ell_h,$$

$$q_i = u_h - u_i.$$

Since ℓ_i are non-increasing and u_i are non-decreasing, both (p_1, p_2, \dots) and (q_1, q_2, \dots) are non-increasing. From the description of the algorithm, we can see that it terminates if and only if there exists some $i > 0$ such that all the observations $(z_{i+1}, \dots, z_{i+K})$ equal to 1, where K is the run length of the algorithm and z_j is defined with respect to (4.8). Since when $z_j = 1$, ℓ_j and u_j are the same as ℓ_{j-1} and u_{j-1} respectively, it is equivalent to say that both the sequence (p_i, \dots, p_{i+K}) and (q_i, \dots, q_{i+K}) are constant.

Without loss of generality, we assume that if the sequence (p_i, \dots, p_{i+K}) is constant for some i , then (q_i, \dots, q_{i+K}) is also constant. Given that the sequence (p_1, p_2, \dots) is non-increasing, we consider the following two cases:

- For some i , the sequence (p_i, p_{i+1}, \dots) remains constant forever. In this case, the algorithm terminates at p_{i+K} by construction;
- For some i , there exists $j > i$ such that $p_i > p_j$. In this case, we assume that

$$p_i - p_j \geq \epsilon_{fp},$$

where ϵ_{fp} is the tolerance for floating point errors. In other words, if $p_i - p_j < \epsilon_{fp}$, we consider that $p_i = p_j$. With a non-increasing sequence, p_i eventually becomes non-positive for some i , i.e., $\ell_i \leq \ell_h$. Once that happens, p_i can no longer change since

$$\hat{g}(\mathbf{x}) + \ell_i \leq \hat{g}(\mathbf{x}) + \ell_h \leq r(\mathbf{x}).$$

Thus we conclude to the first case.

Hence, we have proved that the algorithm terminates with probability one. \square

Example 4.3.1 (A Basic Buck Converter - Generalization with Details). Example 4.1.3 shows that generalization takes 102 simulations to find a tolerance interval $I = [-75, 73] \mu\text{V}$ with $\theta_0 = 0.95$ and $T = 100$. Figure 4.4 shows a trace of how the interval I changes during generalization. Initially, we have a zero interval $I = [0, 0]$. After four failed inclusion tests, I is generalized into $[-75, 73] \mu\text{V}$ and stays unchanged until the algorithm terminates. \parallel

$$\begin{array}{cccccccc}
 I = & [0, 0] & [-52, 0] \mu\text{V} & \cdots & [-52, 67] \mu\text{V} & \cdots & [-52, 73] \mu\text{V} & \cdots & [-75, 73] \mu\text{V} \\
 i = & 0 & 1 & \cdots & 4 & \cdots & 7 & \cdots & 13
 \end{array}$$

Figure 4.4: Snapshots of the interval I during generalization in Example 4.3.1 of the basic buck converter.

4.3.3 Complexity

It is not hard to see that the space complexity of the algorithm is $O(1)$ since it does not store any data structure. On the other hand, the time complexity of generalization is $O(N)$, where N is the required number of simulations for the algorithm to terminate. N depends on many factors, including the quality of the basis functional model $\hat{g}(\mathbf{x})$, the probability θ_0 and the threshold T of the Bayesian test. With a model $\hat{g}(\mathbf{x})$ that has reasonable accuracy, and proper choices of θ_0 and T , we often observe that $N = O(K)$, where K is the run length of the algorithm. Hence, the empirical time complexity of the algorithm is $O(K)$.

4.4 Applications

This section demonstrates SSMI on a few benchmark examples, including a motor controller, a low-pass filter and a buck converter with realistic switches and control logic. The experiments are run on a machine with an AMD Athlon II quad-core 2.8 GHz CPU and 4 G RAM. The implementation is done in Python 2.7.

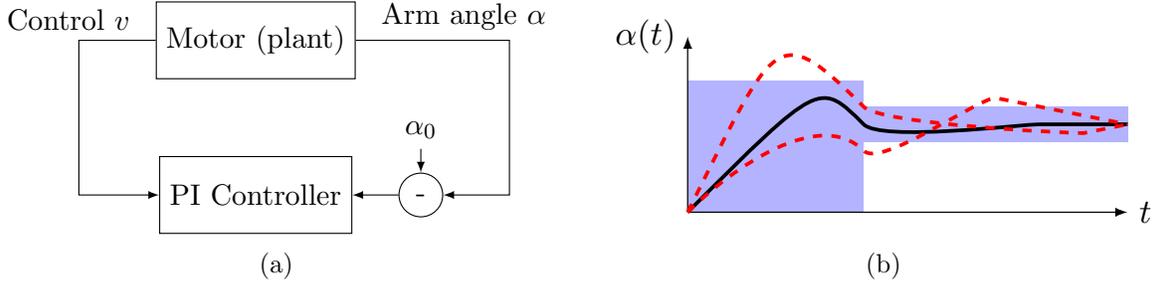


Figure 4.5: A motor with a PI controller (a) and its response specifications (b). The solid line is a trajectory that satisfies the specifications and the red ones violate the specifications.

4.4.1 Motor Controller

Figure 4.5a 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 system has three state variables, the angle of the arm α , the angular velocity ω and the armature current i . It is governed by the following ODEs:

$$\begin{aligned}\frac{d\alpha}{dt} &= \omega, \\ \frac{d\omega}{dt} &= \frac{1}{J} \cdot (-b\omega + Ki + mgL \sin(\alpha)), \\ \frac{di}{dt} &= \frac{1}{L} \cdot (-K\omega - Ri + V).\end{aligned}$$

There are 5 stochastic parameters in the system, which are listed in Table 4.2. Each parameter is assumed to have 10% variation (uniformly) around the nominal value. It is desired that the step response $\alpha(t)$ satisfies the following response specifications.

- Over $t \in [0, 2]$, $\alpha(t) \leq 1.5$. The specification is $\phi_1 \geq 0$ where

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

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

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

Table 4.2: Stochastic parameters in the motor plant.

	Meaning	Nominal Value	Range
J	Moment of Inertia	0.01 kg m ²	[0.009, 0.011]kg m ²
b	Length of the arm	0.1 m	[0.09, 0.11]m
K	Motor torque constant	0.01 N m A ⁻¹	[0.009, 0.011]N m A ⁻¹
R	Resistance	0.1 Ω	[0.09, 0.11] Ω
L	Inductance	0.5 H	[0.45, 0.55]H

The system is designed in Matlab[®] with Simulink[®]. We treat the system as a black-box system with stochastic parameters $\mathbf{x} = (J, b, K, R, L)$ and responses $\phi = (\phi_1, \phi_2, \phi_3)$. Table 4.3 shows the results of verifying the system with SSMI. The “Spec” column shows the index of the specifications, where “all” refers to the conjunction of all the specifications. The second column Y_r shows the Monte-Carlo yield estimation of the original system using 1000 random simulations. The results for SSMI are shown under the “SSMI” column. We illustrate with two cases, regression with quadratic polynomials ($d = 2$) and with cubic polynomials ($d = 3$). For each case, Sim_R and Sim_G are the number of simulations in regression and generalization. T_R and T_G are the time spent in these steps. Y_g refers to the Monte-Carlo yield estimation with respect to the corresponding statistically sound models using 1000 random simulations. In both cases, generalization is done with $\theta_0 = 0.95$ and $T = 100$.

Note that the column Sim_R represents the number of simulations after applying the resampling heuristic. Although we show the number of simulations for each specification, we do not run the simulations separately. For instance, for $d = 2$, we run a total of 150 simulations in regression. Similarly, in generalization we run a total of 279 simulations.

Now let us compare the yields with respect to different models. Observe that the yields with respect to statistically sound models (Y_g under $d = 2$ and $d = 3$) are lower than that with respect to the response surface. This confirms the empirical bound $Y_r \geq Y_g$ (see Section 3.3). On the other hand, the yields of each specification, as well as their conjunction, for $d = 3$ are consistently closer to the true yield than those for $d = 2$. It indicates that the cubic basis functional model is a better approximation than the quadratic one.

Table 4.3: Verification results of the motor controller ($\theta_0 = 0.95$ and $T = 100$).

Spec	Y_r	SSMI									
		$d = 2, \mathbf{c} = 21$					$d = 3, \mathbf{c} = 56$				
		Sim_R	T_R	Sim_G	T_G	Y_g	Sim_R	T_R	Sim_G	T_G	Y_g
1	93.1%	100		279		85.8%	150		288		88.3%
2	95.8%	150	71 s	128	98 s	70.9%	250	103 s	178	101 s	83.2%
3	95.5%	100		117		89.2%	150		149		94.3%
all	92.1%	-	-	-	-	69.5%	-	-	-	-	81.4%

To further illustrate the usage of statistically sound models, we show a plot of the safe regions predicted by the cubic statistically sound models for each specification in Figure 4.6. The regions are drawn in terms of the stochastic parameters b and J , with K , L and R fixed to their nominal values. According to the statistically sound models of the response ϕ_1 , ϕ_2 and ϕ_3 , specification (3) is satisfied for all b and J , and of specification (1) and (2) are satisfied in the two shaded regions. The intersection of these two regions represents the safe region for all the specifications. The dots in the figure show some safe values of b and J from simulations. Notice that only one point is slightly off the intersected region, indicating a good coverage ($\geq 95\%$ of the stochastic parameter space) of the statistically sound models.

4.4.2 Low-Pass Filter

A low-pass filter aims to retain the low frequency components of its input signals and attenuates the components whose frequencies are higher than the cutoff frequency of the filter. Figure 4.7 shows an analog low-pass filter which is built with analog devices. The principle of analyzing and designing such a circuit can be found in many elementary analog circuit design books, such as Millman and Halkias [63].

The circuit consists of an operational amplifier (opamp), three resistors R_1 , R_2 and R_3 and a capacitor C_1 . The opamp is designed with 9 CMOS transistors M_1, \dots, M_9 and a compensation capacitor C_c . Due to process variations, the parameters in these devices in a real circuit are likely to be different from those in a transistor-level design. We assume that each transistor has 4 stochastic parameters, the gate-oxide thickness t_{ox} , the zero-biased threshold voltage v_t , the channel width

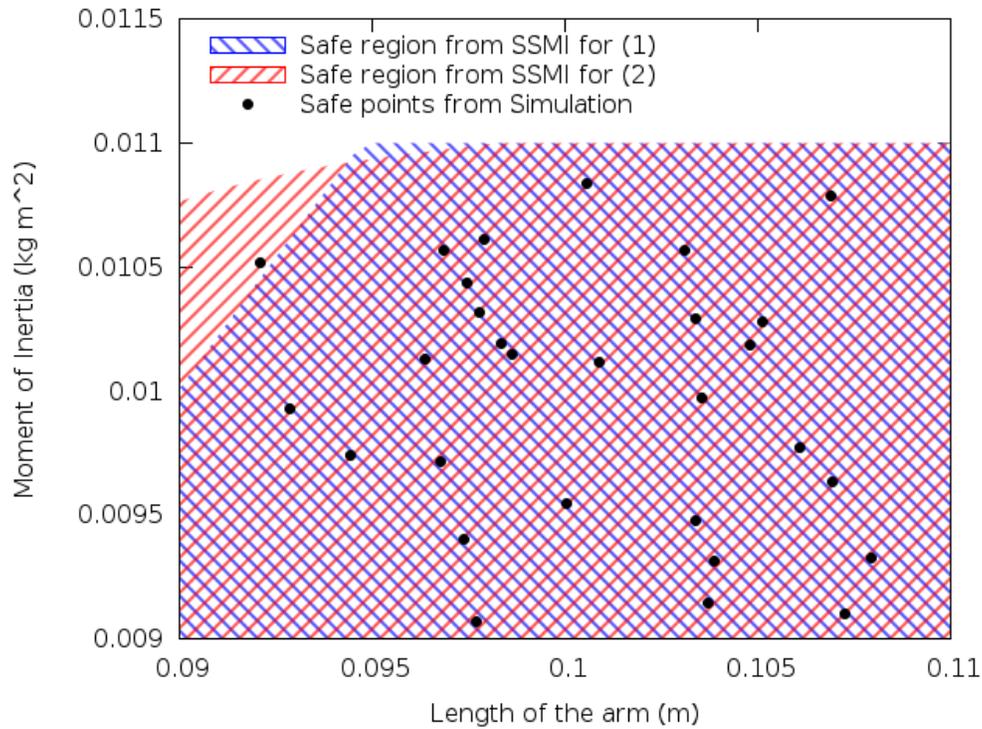


Figure 4.6: Safe regions for specification (1) and (2) of the motor controller in terms of b and J .

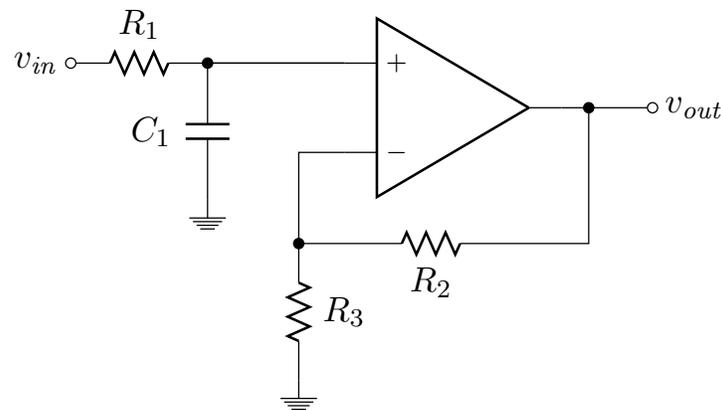


Figure 4.7: An analog low-pass filter.

w and the channel length l . They are assumed to follow normal distributions. Also, the resistors and capacitors in the circuit are considered to follow normal distributions. In total, we have 41 stochastic parameters.

Three important responses of this circuit are considered in this example. They are passband

Table 4.4: Verification results of the low-pass filter ($d = 1$, $|\mathbf{c}| = 42$).

Spec	Y_r	SSMI										
		Sim_R	T_R	$\theta_0 = 0.95, T = 100$			$\theta_0 = 0.95, T = 1000$			$\theta_0 = 0.99, T = 100$		
				Sim_G	T_G	Y_g	Sim_G	T_G	Y_g	Sim_G	T_G	Y_g
1	100%	150		242		99.6%	379		99.5%	986		99.6%
2	99.7%	200	30 s	211	30 s	98.8%	382	45 s	98.8%	941	112 s	98.8%
3	85.6%	200		232		82.0%	301		81.8%	968		82.0%
all	85.4%	-	-	-	-	81.3%	-	-	81.2%	-	-	81.3%

frequency f_p , cutoff frequency f_c , and stopband frequency f_s . Passband frequency f_p is the frequency at which the output signal is 1 dB below the input. Cutoff frequency f_c is the frequency at which the output signal is 3 dB below the input. Stopband frequency f_s is the frequency at which the output signal is 20 dB below the input. Given the variations of the stochastic parameters, it is desired that the responses satisfy the following specifications:

$$(1) f_p \geq 7 \text{ KHz}, (2) f_c \geq 14 \text{ KHz}, (3) f_s \geq 0.15 \text{ MHz}.$$

The circuit is designed and simulated in LTSpice[®] [1], a freely available SPICE simulator. Table 4.4 shows the verification results. The columns have similar meanings as in Table 4.3. In this example, however, we compare the outcomes of using different θ_0 and T in generalization. We fix the degree of the basis functional model to 1, and run the experiments with the following settings of generalization

- $\theta_0 = 0.95$ and $T = 100$, which corresponds to a run length $K = 89$;
- $\theta_0 = 0.95$ and $T = 1000$, which corresponds to a run length $K = 134$;
- $\theta_0 = 0.99$ and $T = 100$, which corresponds to a run length $K = 459$;

The three settings rely on the same basis functional models. Observe that the yields under the three settings are almost the same, but the required number of simulations and time are quite different. This indicates that although in theory, larger θ_0 and T lead to a statistically sound model with a better statistical guarantee and a higher confidence, in practice we do not need them to be very large

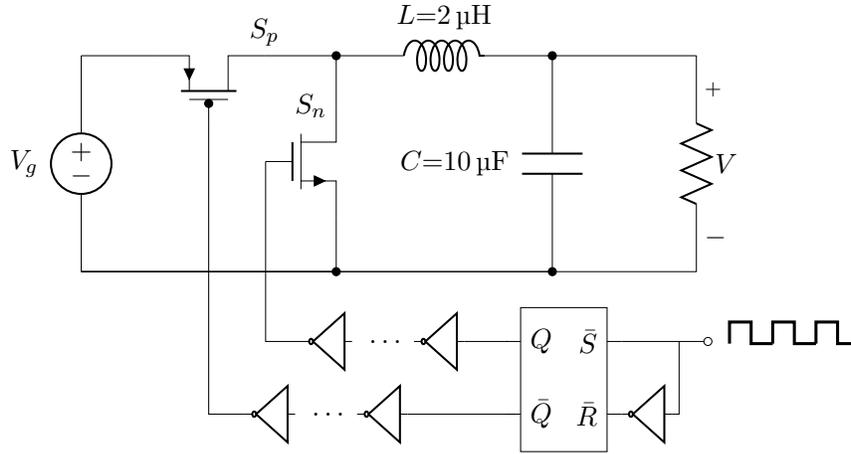


Figure 4.8: A buck converter with realistic switches and control logic.

in order to obtain a reasonably good model. Furthermore, in many cases including this example, little can be benefited from larger θ_0 and T but simulation cost can increase significantly.

4.4.3 Buck Converter

Figure 4.8 shows a buck converter with realistic switches and control logic. It has the same functionality as the basic buck converter in Example 4.1.1. The switches in this circuit is implemented by PMOS and NMOS transistors. Also, the control voltage for the switches are generated through an \overline{SR} NAND latch and two inverter chains. The output Q and \overline{Q} of the latch has opposite parities. The inverter chain connected to Q has an odd number of stages, and the other one has an even number of stages. Consequently, the voltages applied to the gate of S_p and S_n are at the same logic level, i.e., either both digital 1 or both digital 0. This guarantees that the switches S_p and S_n are not turned on at the same time.

Besides L and C in Example 4.1.1, we assume that the process parameters in the two transistors are also stochastic. Table 4.5 summaries the process parameters that we are concerned with in the NMOS transistor S_n . The PMOS transistor S_p has the same parameters but with different nominal values. We assume that each parameter, including L and C , follows a normal distribution with the nominal value μ_0 as the mean and $0.05\mu_0$ as the standard deviation. In total, we have 24

Table 4.5: Stochastic parameters of the transistor S_n in the buck converter. S_p has the same parameters but with different nominal values.

	Meaning	Nominal Value	Distribution
w	Channel width	1000 μm	$N(\mu_0, 0.05\mu_0)$ $\mu_0 = \text{nominal values}$
l	Channel length	35 nm	
epsrox	Gate dielectric constant relative to vacuum	3.9	
toxe	Electrical gate equivalent oxide thickness	1.15 nm	
toxp	Physical gate equivalent oxide thickness	0.9 nm	
xj	S/D junction depth	10 nm	
ndep	Channel doping concentration	$4.12 \times 10^{18} \text{ cm}^{-3}$	
ngate	Poly Si gate doping concentration	$1 \times 10^{23} \text{ cm}^{-3}$	
nsd	Source/drain doping concentration	$2 \times 10^{20} \text{ cm}^{-3}$	
rsh	Source/drain sheet resistance	5 Ω/\square	
rshg	Gate electrode sheet resistance	0.4 Ω/\square	

stochastic parameters.

Recall that the voltage ripple Δv , the amplitude of the oscillation upon the DC output voltage, is an important response in a buck converter (see Example 4.1.1). Besides Δv , we are also interested in the power consumption w of the circuit. Suppose that it is desired to verify that¹

$$(1) \Delta v \leq 5 \text{ mV}, (2) w \leq 50 \text{ mW}.$$

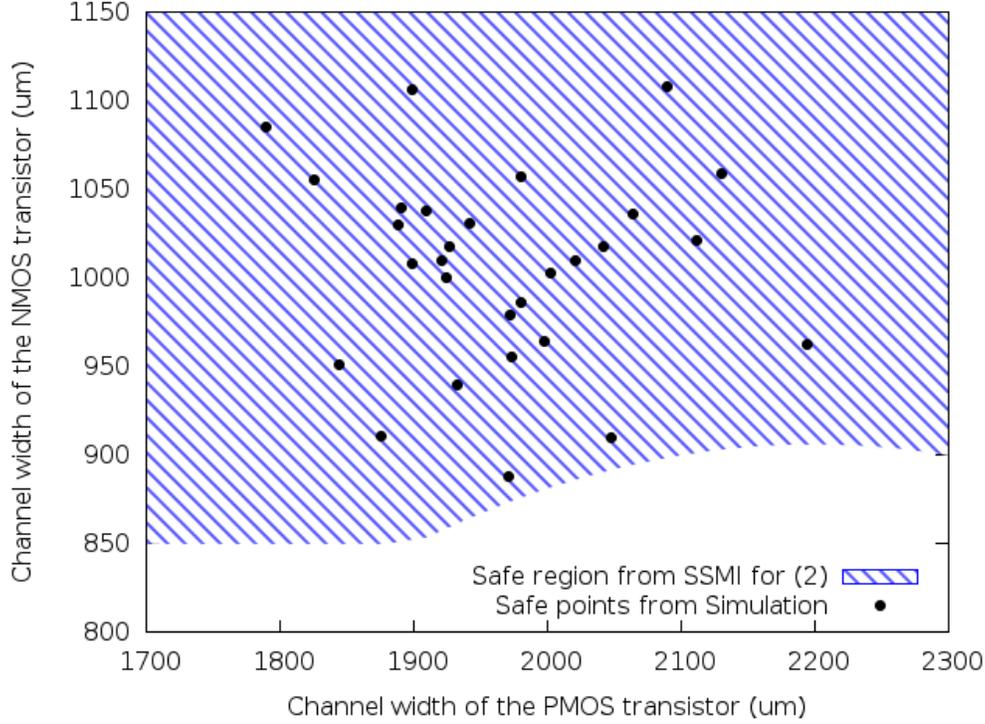
The circuit is designed and simulated in LTSpice[®] [1], a freely available SPICE simulator. Table 4.6 shows the verification results. We present the case for $d = 1$ and $d = 2$, i.e., linear and quadratic basis functional models. Observe that for specification (1), both $d = 1$ and $d = 2$ lead to 100% yields with respect to the corresponding statistically sound models. For specification (2), however, the yield in the $d = 1$ case is only 40.7%. Although still satisfying the inequality $Y_r \geq Y_g$, it under-estimates the true yield excessively. On the other hand, a quadratic basis functional model approximates the response surface more accurately. The estimated yield with respect to the resulting statistically sound models is boosted to 69.9%.

Figure 4.9 shows the safe region for specification (2) predicted by the quadratic statistically sound model in terms of the channel width of S_p , w_p and the channel width of S_n , w_n . The other

¹ Although the circuit in this example looks similar to the one in Example 4.1.1, it uses real devices and a different input voltage $V_g = 0.9 \text{ V}$. Hence, the specification about Δv is also different.

Table 4.6: Verification results of the buck converter ($\theta_0 = 0.95$ and $T = 100$).

Spec	Y_r	SSMI									
		$d = 1, \mathbf{c} = 25$					$d = 2, \mathbf{c} = 325$				
		Sim_R	T_R	Sim_G	T_G	Y_g	Sim_R	T_R	Sim_G	T_G	Y_g
1	100%	250		279		100%	800		213		100%
2	72.3%	250	1.3 h	192	1.5 h	40.7%	900	5.5 h	166	1.3 h	69.9%
all	72.3%	-	-	-	-	40.7%	-	-	-	-	69.9%

Figure 4.9: Safe region for specification (2) of the buck converter in terms of the channel width of S_p and the channel width of S_n .

stochastic parameters are fixed to their nominal values. The dots represents some safe values of w_p and w_n from simulations. Apparently, all these points fall in the safe region predicted by SSMI.

4.5 Summary

This chapter introduces SSMI, a statistical verification approach. SSMI combines ideas from regression and statistical model checking, and introduces a response surface modeling approach that provides statistical guarantees for black-box systems. It consists of two components, regression and

generalization. The regression technique used in this chapter is ordinary least squares, which is simple but not powerful enough to deal with the cases for many stochastic parameters. The next chapter presents a sparse regression algorithm that can handle hundreds of stochastic parameters.

Chapter 5

A Sparse Approximation Method

As seen in Chapter 4, ordinary least squares (OLS) is not capable to handle systems with a large number of stochastic parameters, since it requires a number of simulations that grows exponentially in the number of stochastic parameters and the degree of the target polynomial. With enough data, OLS computes the coefficient of every term in the polynomial. However, not all these terms are equally important. In many practical applications, most of them even have coefficients close to 0, so that dropping them from the polynomial leads to little loss of accuracy. Such a feature is known as sparsity. A regression algorithm that exploits the sparse structure of the target function is a sparse approximation algorithm.

Sparse approximation is closely related to two important categories of techniques: compressed sensing and uncertainty quantification. Compressed sensing originates from the area of signal processing and aims to reconstruct signals accurately using a small number of random samples [58, 80, 18, 24, 16, 15, 20, 14]. Compressed sensing relies on techniques such as matching pursuit [58], LASSO [80] and basis pursuit [18]. The key problem is to solve under-determined linear systems $\mathbf{X}\boldsymbol{\beta} = \mathbf{y}$ where \mathbf{X} is a $N \times n$ matrix with $N < n$. In other words, there are more unknowns than equations. Such a problem usually has infinitely many solutions. Hence, extra constraints, such as smoothness [81] or sparsity [80], are required in order to get an unique solution.

Uncertainty quantification is an emerging area that studies how to characterize uncertainties in a system and their effects on the responses of the system. Conventionally, Monte-Carlo techniques have been the main approach for uncertainty quantification. These methods do not suffer from the

curse of dimensionality, but it is well known that they have a slow rate of convergence [71]. In recent years, alternative approaches, such as stochastic Galerkin schemes based on polynomial chaos expansion [23, 4, 60, 90] and stochastic collocation schemes [5, 67, 89, 72, 59], have been proposed. Compared to Monte-Carlo simulation, these approaches are more effective in modeling and propagating uncertainties in a system. However, they suffer from the curse of dimensionality such that their computational costs grow rapidly as the number of stochastic parameters in the system increases.

There have been many interesting approaches for sparse approximation [55, 75, 26, 64, 54, 25]. This chapter presents a sparse approximation technique that combines generalized polynomial chaos (gPC) [88] and LASSO [80] (see Section 2.3 for an introduction). This technique considers black-box systems with stochastic parameters and discovers low-degree polynomial approximations of the response surface as a function of the stochastic parameters. It includes a heuristic that discovers relevant terms in the polynomial approximation, and a regression algorithm based on LASSO to construct polynomial approximations. The heuristic efficiently discards basis functions that contribute little to the response surface. Then the coefficients of the remaining basis functions are computed under \mathcal{L}_1 regularization. The content of this chapter is originally published by Zhang et al. [99].

This chapter is organized as follows. The following section presents a brief overview of gPC. Section 5.2 introduces our sparse approximation approach. Finally, Section 5.4 shows several applications to demonstrate the capability of the proposed approach in the context of SSMI (introduced in Chapter 4).

5.1 Generalized Polynomial Chaos

In its original form, polynomial chaos, introduced by Wiener [87], is a non-sampling-based method to characterize uncertainties and their influence on system responses. It uses Hermite polynomials, a family of orthogonal polynomials, to model stochastic processes with Gaussian random variables. The theory of gPC is developed by Xiu and Karniadakis [90]. It generalizes the

theory of polynomial chaos to model stochastic processes with random variables in various continuous and discrete distributions. Each distribution corresponds to a particular family of orthogonal polynomials taken from the Wiener-Askey scheme (see Koekoek et al. [49] for an introduction to the Askey-scheme). The following presents an elementary level review of gPC. Further details are available from Xiu [88].

5.1.1 Orthogonal Polynomials

Let us first review the basics of orthogonal polynomials. Let $Q_n(x)$ be a general polynomial

$$Q_n(x) = c_0 + c_1x + \cdots + c_nx^n,$$

where n is the degree of the polynomial and c_i are the coefficients of the terms. We consider x as a random variable with a probability density function (pdf) $\omega(x)$. A family of polynomials $\{Q_i(x), i \in \mathbb{N}_0\}$, where \mathbb{N}_0 is the set of non-negative integers, is orthogonal family of polynomials if for some pdf $\omega(x)$ with a domain S ,

$$\int_S Q_n(x)Q_m(x)\omega(x)dx = \gamma_n\delta_{nm}, \quad m, n \in \mathbb{N}_0,$$

where γ_n is a normalization constant, δ_{nm} is the Kronecker delta function such that $\delta_{nm} = 0$ if $n \neq m$ and $\delta_{nm} = 1$ if $n = m$. Clearly,

$$\int_S Q_n(x)Q_n(x)\omega(x)dx = \gamma_n, \quad n \in \mathbb{N}_0.$$

For a family of orthogonal polynomials with respect to a density function $\omega(x)$, we define an inner product $\langle \cdot, \cdot \rangle_{\omega(x)}$ such that

$$\langle Q_n(x), Q_m(x) \rangle_{\omega(x)} = \gamma_n\delta_{nm}, \quad m, n \in \mathbb{N}_0.$$

We show two families of orthogonal polynomials that are commonly used in practice. The first family is known as the **Hermite polynomials** $\{H_i(x), i \in \mathbb{N}_0\}$. These polynomials satisfy the recurrence relation

$$H_{n+1} = xH_n(x) - nH_{n-1}(x), \quad n \in \mathbb{N},$$

where \mathbb{N} is the set of positive integers and

$$\int_{-\infty}^{+\infty} H_n(x)H_m(x)\omega(x)dx = n!\delta_{nm}, \quad \omega(x) = \frac{1}{\sqrt{2\pi}}e^{-\frac{x^2}{2}}.$$

Hermite polynomials are orthogonal with respect to the density function $\omega(x)$. Note that $\omega(x)$ is the pdf of a standard normal random variable. The first few Hermite polynomials are

$$H_0(x) = 1, \quad H_1(x) = x, \quad H_2(x) = x^2 - 1, \quad H_3(x) = x^3 - 3x, \dots$$

Another important family of orthogonal polynomials is **Legendre polynomials** $\{L_i(x), i \in \mathbb{N}_0\}$. They satisfy the recurrence relation

$$L_{n+1}(x) = \frac{2n+1}{n+1}xL_n(x) - \frac{n}{n+1}L_{n-1}(x), \quad n \in \mathbb{N}$$

and

$$\int_{-1}^1 L_n(x)L_m(x)dx = \frac{2}{2n+1}\delta_{nm}.$$

Legendre polynomials are orthogonal with respect to the uniform density function on the interval $[-1, 1]$. The first few Legendre polynomials are

$$L_0(x) = 0, \quad L_1(x) = x, \quad L_2(x) = \frac{3}{2}x^2 - \frac{1}{2}, \quad L_3(x) = \frac{5}{2}x^3 - \frac{3}{2}x, \dots$$

5.1.2 Orthogonal Projection

Let $\{Q_i(x), i \in \mathbb{N}_0\}$ be a family of orthogonal polynomials with respect to a density function $\omega(x)$ with a domain S . According to the classic theorem by Weierstrass, any continuous function $f(x)$ over a bounded interval can be approximated with arbitrarily small error by polynomials.

Suppose that

$$f(x) = c_0 + \sum_{i=1}^{+\infty} c_i Q_i(x), \quad x \in I,$$

where I is some interval. This is known as an **orthogonal projection** of $f(x)$ onto the vector space of polynomials $\{Q_i(x), i \in \mathbb{N}_0\}$. The polynomials Q_i are the **basis functions** of the projection. The coefficients c_i are called the generalized Fourier coefficients. In practice, we usually truncate

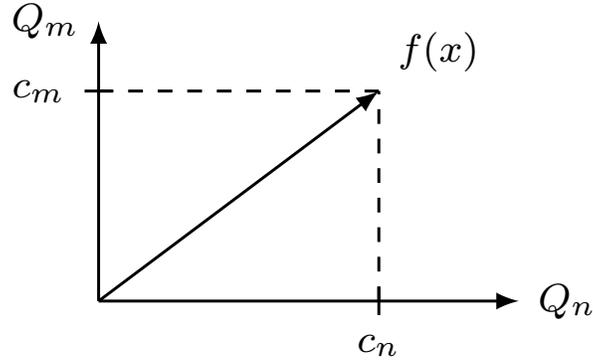


Figure 5.1: An interpretation of orthogonal projection.

the infinite sum up to some degree d and form the following approximation

$$f(x) \approx c_0 + c_1 Q_1(x) + \cdots + c_d Q_d(x).$$

Figure 5.1 shows an interpretation of orthogonal projection. Consider a function $f(x)$ as a vector in the vector space of the orthogonal polynomials $\{Q_i(x), i \in \mathbb{N}_0\}$. The projection maps $f(x)$ onto the direction of $Q_i(x)$. The generalized Fourier coefficient, c_i , represents the length of the projection. Intuitively, the larger c_i is, the more $Q_i(x)$ contributes to $f(x)$.

Consider the inner product between $f(x)$ and $Q_i(x)$ with respect to $\omega(x)$,

$$\begin{aligned} \langle f(x), Q_i(x) \rangle_{\omega(x)} &= \int_S f(x) Q_i(x) \omega(x) dx \\ &= \int_S (c_0 + c_1 Q_1(x) + c_2 Q_2(x) \cdots) Q_i(x) \omega(x) dx. \end{aligned}$$

By orthogonality of the basis functions $Q_i(x)$, we have

$$\langle f(x), Q_i(x) \rangle_{\omega(x)} = c_i \int_S Q_i(x) Q_i(x) \omega(x) dx = c_i \gamma_i.$$

For a fixed density function $\omega(x)$, the normalization constants γ_i can be pre-computed. Hence, if we can compute the inner product, the coefficients c_i are simply

$$c_i = \frac{1}{\gamma_i} \langle f(x), Q_i(x) \rangle_{\omega(x)}. \quad (5.1)$$

Table 5.1: Correspondence between the distribution of a random variable and the family of orthogonal polynomials as its gPC basis [88].

	Distribution	Orthogonal Family	Support
Continuous	Gaussian	Hermite	$(-\infty, \infty)$
	Uniform	Legendre	$[a, b]$
	Beta	Jacobi	$[a, b]$
	Gamma	Laguerre	$[0, \infty)$
Discrete	Poisson	Charlier	$\{0, 1, 2, \dots\}$
	Binomial	Krawtchouk	$\{0, 1, \dots, N\}$
	Negative binomial	Meixner	$\{0, 1, 2, \dots\}$
	Hypergeometric	Hahn	$\{0, 1, \dots, N\}$

5.1.3 Generalized Polynomial Chaos

Let Z be a random variable with the distribution $F_Z(z)$. The **gPC basis functions** with respect to Z are the orthogonal polynomials $\{Q_i(z), i \in \mathbb{N}_0\}$ satisfying

$$\mathbb{E}[Q_n(Z)Q_m(Z)] = \int_{\text{dom}(Z)} Q_n(z)Q_m(z)dF_Z(z) = \gamma_n\delta_{nm}, \quad m, n \in \mathbb{N}_0. \quad (5.2)$$

where \mathbb{E} computes the expectation over the distribution $F_Z(z)$ and

$$\gamma_n = \mathbb{E}[Q_n^2(Z)], \quad n \in \mathbb{N}_0 \quad (5.3)$$

are the normalization constants. From (5.2) and (5.3), we establish a correspondence between the distribution of the random variable Z and the family of orthogonal polynomials as its gPC basis functions. Table 5.1 shows the correspondence for some common distributions.

We have introduced the gPC basis functions for single random variable. Let us consider the case for multiple random variables. Let $\mathbf{Z} = (Z_1, Z_2, \dots, Z_k)$ be a random vector of k mutually independent random variables with the joint distribution

$$F_{\mathbf{Z}}(\mathbf{z}) = F_{Z_1}(z_1)F_{Z_2}(z_2) \dots F_{Z_k}(z_k),$$

where $\mathbf{z} = (z_1, z_2, \dots, z_k)$ and $F_{Z_i}(z_i)$ are the marginal distributions of the random variable Z_i . Let $\mathbf{d} = (d_1, \dots, d_k)$ be a vector of non-negative integers, and $|\mathbf{d}| = \sum_{i=1}^k d_i$. The gPC basis function of degree $|\mathbf{d}|$ with respect to \mathbf{Z} is

$$Q_{\mathbf{d}}(\mathbf{z}) = Q_{d_1}(z_1) \dots Q_{d_k}(z_k),$$

where $Q_{d_i}(z_i)$ is the gPC basis function of degree d_i for the random variable Z_i . For $\mathbf{d}_n = (d_{n1}, \dots, d_{nk})$ and $\mathbf{d}_m = (d_{m1}, \dots, d_{mk})$, the basis functions $Q_{\mathbf{d}_n}(\mathbf{z})$ and $Q_{\mathbf{d}_m}(\mathbf{z})$ satisfy

$$\langle Q_{\mathbf{d}_n}(\mathbf{z}), Q_{\mathbf{d}_m}(\mathbf{z}) \rangle = \gamma_{\mathbf{d}_n} \delta_{\mathbf{d}_n, \mathbf{d}_m},$$

where

$$\gamma_{\mathbf{d}_n} = \gamma_{d_{n1}} \cdots \gamma_{d_{nk}}$$

and

$$\delta_{\mathbf{d}_n, \mathbf{d}_m} = \begin{cases} 1 & d_{ni} = d_{mi}, i = 1, \dots, k, \\ 0 & \text{otherwise.} \end{cases}$$

Note that mutual independence of the the random variables Z_i are necessary in the above formulation. We refer the interested readers to Xiu and Karniadakis [90] for a treatment of non-independent random variables.

5.2 A Low-Degree Approximation Algorithm

5.2.1 Overview

Consider a black-box system $\mathcal{M} = (\mathbf{x}, \phi, F_{\mathbf{X}}, r)$ with n stochastic parameters $\mathbf{x} = (x_1, \dots, x_n)$ following the distribution $F_{\mathbf{X}}(\mathbf{x})$, a response ϕ with a response surface $\phi = r(\mathbf{x})$. Recall that with a set of simulation data $\{\mathbf{x}^{(i)}, \phi^{(i)}\}$, we need to find a polynomial $\hat{g}(\mathbf{x})$ such that

$$\min_{\mathbf{c}} \sum_{i=1}^N \left\| \phi^{(i)} - \hat{g}(\mathbf{x}^{(i)}) \right\|_2^2, \quad (5.4)$$

where \mathbf{c} is the vector of unknown coefficients and N is the size of the simulation data. The number of unknown coefficients $|\mathbf{c}|$ in the polynomial grows exponentially in n , the number of stochastic parameters and d , the degree of the polynomial. For large systems, it is computational prohibitive to collect data from a large number of simulation runs since a single simulation may take hours or even days. In these cases, OLS fails since it requires more data than the number of unknowns. To solve this problem, this section introduces a sparse approximation approach that combines gPC and LASSO (see Section 2.3), a \mathcal{L}_1 regularization technique.

Given a set of simulation data $\{\mathbf{x}^{(i)}, \phi^{(i)}\}$ with N data points, we construct an $N \times n$ matrix \mathbf{X} and an N vector \mathbf{y} such that

$$\mathbf{X} = \begin{pmatrix} x_1^{(1)} & \cdots & x_n^{(1)} \\ \vdots & \ddots & \vdots \\ x_1^{(N)} & \cdots & x_n^{(N)} \end{pmatrix}, \mathbf{y} = \begin{pmatrix} \phi^{(1)} \\ \vdots \\ \phi^{(N)} \end{pmatrix}. \quad (5.5)$$

We assume that the target polynomial $\hat{g}_d(\mathbf{x})$ of degree d has the following form:

$$\hat{g}_d(\mathbf{x}) = \sum_{|\mathbf{d}_i| \leq d} c_{\mathbf{d}_i} Q_{\mathbf{d}_i}(\mathbf{x}),$$

where $c_{\mathbf{d}_i}$ are the unknown coefficients and $\mathbf{x}^{\mathbf{d}_i}$ stands for $x_1^{d_{i1}} \cdots x_n^{d_{in}}$. $\{Q_{\mathbf{d}_i}(\mathbf{x})\}$ is a set of gPC basis functions with respect to the distribution $F_{\mathbf{X}}(\mathbf{x})$. We write S_k to denote the set of basis functions of degree k . An $N \times |S_k|$ matrix \mathbf{X}_k is constructed from \mathbf{X} and S_k ,

$$\mathbf{X}_k = \begin{pmatrix} Q_{\mathbf{d}_1}(\mathbf{x}^{(1)}) & \cdots & Q_{\mathbf{d}_{|S_k|}}(\mathbf{x}^{(1)}) \\ \vdots & \ddots & \vdots \\ Q_{\mathbf{d}_1}(\mathbf{x}^{(N)}) & \cdots & Q_{\mathbf{d}_{|S_k|}}(\mathbf{x}^{(N)}) \end{pmatrix}, |\mathbf{d}_i| = k, i = 1, \dots, |S_k|, \quad (5.6)$$

where $|\mathbf{d}_i|$ is the sum of the vector \mathbf{d}_i and $|S_k|$ represents the cardinality of S_k . The coefficients of the basis functions in S_k is denoted by a vector $\boldsymbol{\beta}_k$,

$$\boldsymbol{\beta}_k = \begin{pmatrix} c_{\mathbf{d}_1} \\ \vdots \\ c_{\mathbf{d}_{|S_k|}} \end{pmatrix}, |\mathbf{d}_i| = k, i = 1, \dots, |S_k|.$$

The problem in (5.4) can be written in the following matrix form:

$$\min_{(\boldsymbol{\beta}_0, \dots, \boldsymbol{\beta}_d)} \left\| \mathbf{y} - \sum_{k=0}^d \mathbf{X}_k \boldsymbol{\beta}_k \right\|_2^2, \quad (5.7)$$

The proposed sparse approximation algorithm, shown in Algorithm 4, aims to find a polynomial approximation of a degree as low as possible. Suppose that we have an $N \times n$ matrix \mathbf{X} and an N vector \mathbf{y} as shown in (5.5), such that \mathbf{X} represents the stochastic parameters \mathbf{y} represents the response ϕ . The algorithm works iteratively as follows. It discovers a polynomial approximation

Input: Matrices \mathbf{X} , Vector \mathbf{y} , Target Degree d , Distribution $F_{\mathbf{X}}(\mathbf{x})$, Dropping Threshold η , Termination Threshold ϵ

Output: Sparse Approximation $\hat{g}(\mathbf{x})$

```

1 res =  $\mathbf{y}$  ;
2  $S = \emptyset$  ;
3 for  $k \leftarrow 1$  to  $d$  do
4    $S_k, \mathbf{X}_k =$  construct gPC basis functions of degree  $k$  ;
5    $S'_k, \mathbf{X}'_k =$  choose the basis functions from  $S_k$  with a dropping threshold  $\eta$  ;
6    $\beta'_k =$  compute the unknown coefficients using  $\mathbf{X}'_k$  and res ;
7    $S''_k =$  collect the basis functions with non-zero coefficients ;
8    $S = S \cup S''_k$  ;
9   res = res -  $\mathbf{X}'_k \beta'_k$  ;
10  if  $\|\mathbf{res}\|_2 < \epsilon$  then
11    | break ;
12  end
13 end
14  $\beta_S =$  recompute the coefficients of the basis functions in  $S$  ;
15  $\hat{g}(\mathbf{x}) =$  construct a polynomial with  $\beta_S$  and  $S$  ;

```

Algorithm 4: A sparse approximation algorithm.

of a certain degree k using the matrix \mathbf{X}_k , which is constructed from \mathbf{X} and S_k , the set of basis functions of degree k , and the residual vector **res**, where **res** equals to \mathbf{y} initially.

At the k th iteration, we choose a subset of the basis functions S'_k that are considered “important” to the approximation from S_k , and construct the corresponding matrix \mathbf{X}'_k (see Section 5.2.2 for details). Intuitively, we estimate the coefficients of the basis functions and prune those with coefficients that are close to 0. Using the basis functions in S'_k , we compute the unknown coefficients β'_k (see Section 5.2.3 for details) and collect the corresponding basis functions with non-zero coefficients into a set S . The residual vector **res** is updated such that the contribution from the degree- k approximation to the response ϕ , represented by $\mathbf{X}'_k \beta'_k$, are subtracted from **res**. The iteration terminates if either the \mathcal{L}_2 norm of **res** becomes smaller than a pre-defined termination threshold ϵ , or k reaches the given target degree d . Finally, we recompute the coefficients of the basis functions in S and construct the approximation $\hat{g}(\mathbf{x})$.

5.2.2 Choosing a Subset of Basis Functions via gPC

Consider a set of basis functions S_k of degree k . Since we assume that the response surface $r(\mathbf{x})$ permits a sparse approximation, it is desirable to choose a subset of basis functions from S_k rather than using all of them. Given that the basis functions have the same degree, we regard those with larger coefficients more important for that they contribute more to the response surface $r(\mathbf{x})$ compared to those with smaller coefficients.

From (5.1), we know that coefficients of the basis functions can be computed by orthogonal projection, i.e.,

$$c_{\mathbf{d}_i} = \frac{1}{\gamma_{\mathbf{d}_i}} \langle r(\mathbf{x}), Q_{\mathbf{d}_i}(\mathbf{x}) \rangle .$$

However, it can be difficult, if not impossible, to compute $c_{\mathbf{d}_i}$ exactly because

- The response surface $r(\mathbf{x})$ can be evaluated but does not have a closed form;
- The number of stochastic parameters \mathbf{x} is usually large in practice.

As a possible solution, sparse grid quadrature methods work well when the dimension of \mathbf{x} is relatively small (see, e.g., Gerstner and Griebel [33]). But they cannot handle a large number of stochastic parameters. Also, these methods rely on the ability to sample at certain points, which is not always realizable in practice.

In our case, the coefficients are used to prune unimportant basis functions. Hence, they do not have to be precise as long as they can reflect the relative importance of the basis functions. Since the inner product $\langle r(\mathbf{x}), Q_{\mathbf{d}_i}(\mathbf{x}) \rangle$ is essentially an expectation over $F_{\mathbf{X}}(\mathbf{x})$,

$$\langle r(\mathbf{x}), Q_{\mathbf{d}_i}(\mathbf{x}) \rangle = \int_{\mathbb{X}} r(\mathbf{x}) Q_{\mathbf{d}_i}(\mathbf{x}) dF_{\mathbf{X}}(\mathbf{x}) = \mathbb{E} [r(\mathbf{x}) Q_{\mathbf{d}_i}(\mathbf{x})] ,$$

where \mathbb{X} is the domain of $F_{\mathbf{X}}(\mathbf{x})$, we use Monte-Carlo simulation to compute an estimation of $c_{\mathbf{d}_i}$,

$$\hat{c}_{\mathbf{d}_i} = \frac{1}{\gamma_{\mathbf{d}_i} N} \sum_{j=1}^N \phi^{(j)} \cdot Q_{\mathbf{d}_i}(\mathbf{x}^{(j)}) , \quad (5.8)$$

Due to the slow convergence of Monte-Carlo methods, $\hat{c}_{\mathbf{d}_i}$ are usually not accurate estimates of $c_{\mathbf{d}_i}$, which can sometimes be far away from $c_{\mathbf{d}_i}$ when N is small. To improve the accuracy of $\hat{c}_{\mathbf{d}_i}$ such

that they can at least reflect the relative importance of the basis functions, we compute a k -fold average instead of a single estimation. The computation is similar to the resampling heuristic in Section 4.2.1. We divide the set of simulation data into k folds, each with $\frac{N}{k}$ data. Then we evaluate (5.8) k times such that for the i th evaluation, the i th fold of data are excluded. The k estimates are averaged, producing a single estimate $\hat{c}_{\mathbf{d}_i}$.

Note that $\hat{c}_{\mathbf{d}_i}$ can at best capture the trend of $c_{\mathbf{d}_i}$. They should not be used as coefficients of the polynomial approximation of the response surface, which usually leads to a poor quality. However, the evaluation of (5.8) is efficient even with a large number of stochastic parameters. Furthermore, it can be easily parallelized. Consequently, we build a **filtering stage** with the estimates $\hat{c}_{\mathbf{d}_i}$ such that basis functions, which have estimated coefficients $\hat{c}_{\mathbf{d}_i}$ that are smaller than a specified threshold η , are dropped from the set S_k . This leads to a subset S'_k of the basis functions of degree k .

5.2.3 Computing Unknown Coefficients

For the basis functions in S'_k , we can construct an $N \times |S'_k|$ matrix \mathbf{X}'_k as shown in (5.6). We need to compute the coefficients β'_k such that the error between **res**, which initially equals to \mathbf{y} , and $\mathbf{X}'_k \beta'_k$ are minimized, i.e.,

$$\min_{\beta'_k} \|\mathbf{res} - \mathbf{X}'_k \beta'_k\|_2^2. \quad (5.9)$$

Obviously, if $N \geq |S'_k|$, (5.9) is over-determined and can be solved by OLS without over-fitting. If $N < |S'_k|$, we use LASSO to solve (5.9). LASSO adds a regularization constraint on the coefficients β'_k and solves the following problem:

$$\min_{\beta'_k} \|\mathbf{res} - \mathbf{X}'_k \beta'_k\|_2^2 + \lambda \|\beta'_k\|_1. \quad (5.10)$$

The extra term forces the coefficients $c_{\mathbf{d}_i}$ to behave “regularly” so that they cannot range over many orders of magnitude. Furthermore, due to the nature of \mathcal{L}_1 norm, proper choices of λ result in sparse solutions (see Section 2.3 for an introduction). In general, a larger λ leads to a sparser solution. When λ approaches 0, LASSO reduces to OLS fitting. In practice, λ is often determined

by cross validation, i.e., choosing a series of values for λ and cross-validating each of them to find the one with the smallest error.

The vector $\mathbf{X}'_k \boldsymbol{\beta}'_k$ represents the quantities of the response ϕ approximated by the basis functions S'_k of degree k . Once a solution of $\boldsymbol{\beta}_k$ is found, the residual vector \mathbf{res} is updated by subtracting $\mathbf{X}'_k \boldsymbol{\beta}'_k$ so that the contributions from S'_k are removed. The new residual vector serves as the “response” values with respect to which the approximation of degree $k + 1$ is built.

The iteration in Algorithm 4 terminates if either the \mathcal{L}_2 norm of the residual vector \mathbf{res} becomes smaller than a pre-defined termination threshold ϵ , or k reaches the target degree d . In the former case (and $k < d$), we have a polynomial approximation of a degree lower than the target degree. We call this **early termination**. It is preferable since a low-degree approximation is always considered better than a high-degree one when they have similar accuracy. At this point, we have collected a set of basis functions S . Although coefficients are available for each basis function in S , they are computed with respect to the residual vector at each iteration and may not be accurate when the basis functions are combined. Hence, we recompute all the coefficients by solving the following problem:

$$\min_{\boldsymbol{\beta}_S} \|\mathbf{y} - \mathbf{X}_S \boldsymbol{\beta}_S\|_2^2, \quad (5.11)$$

where \mathbf{X}_S is an $N \times |S|$ matrix constructed according to (5.6) for the basis functions in S , and $\boldsymbol{\beta}_S$ are the unknown coefficients. As the problem in (5.9), (5.11) is solved by OLS if $N \geq |S|$, or LASSO otherwise. Finally, with $\boldsymbol{\beta}_S = (c_{\mathbf{d}_1}, \dots, c_{\mathbf{d}_{|S|}})$, we have the following polynomial approximation:

$$\hat{g}(\mathbf{x}) = \sum_{i=1}^{|S|} c_{\mathbf{d}_i} Q_{\mathbf{d}_i}(\mathbf{x}).$$

5.3 Discussion of the Algorithm

The proposed sparse approximation algorithm has two salient features:

- It combines gPC and LASSO in a way that the efficiency of the powerful \mathcal{L}_1 technique LASSO is enhanced.

- Regardless to the target degree, it produces polynomial approximations with degrees that are as low as possible;

The first feature is affected by the parameter η , which controls how “aggressive” the algorithm is in dropping unimportant basis functions. If η approaches 0, then the filtering stage considers almost every basis functions to be important. This is equivalent to performing LASSO alone. On the other hand, a reasonable choice of η can prune many basis functions that indeed have small coefficients, and thus result in a smaller problem that is solved subsequently by either OLS or LASSO. For systems with a response surface that permits sparse representations, the result problem can be much smaller than the original problem. In practice, η is often set to a small number times the maximum coefficients of the basis functions of a certain degree, e.g., $0.01 \cdot \max(\hat{c}_{\mathbf{d}_1}, \hat{c}_{\mathbf{d}_2}, \dots)$, where $\hat{c}_{\mathbf{d}_i}$ are computed as in (5.8). Intuitively, for basis functions of the same degree, if the coefficient is very small compared to others, the corresponding basis function does not have much contribution to the response surface and thus can be dropped.

The second feature, which is affected by the parameter ϵ , is convenient for designer. Basically, the target degree reflects how complex we believe a response surface is, and how complex model we would like to tolerate. In general, simpler models are preferable since they provides cleaner explanations on the relationship between stochastic parameters and responses. The value of ϵ represents a trade-off between accuracy and model simplicity. By choosing a large ϵ , we may find a model with a degree lower than the target degree. This is not achievable by applying LASSO directly. The parameter ϵ is usually set according to the response vector \mathbf{y} , e.g., $0.01 \cdot \|\mathbf{y}\|_2$.

5.4 Applications

First, the proposed sparse approximation algorithm is demonstrated using a set of randomly generated sparse polynomials. Then it is applied to three benchmark examples, including a three-stage ring oscillator, an eight-bit digital-analog converter (DAC) and a low-pass filter. The experiments are run on a machine with an AMD Athlon II quad-core 2.8 GHz CPU and 4G RAM. The

implementation is done in Python 2.7.

5.4.1 Randomly Generated Sparse Polynomials

The first application is to compare the performance of the proposed sparse approximation algorithm with LASSO, using a set of randomly generated sparse polynomials. These polynomials have a fixed degree of $k = 2$ and contain stochastic parameters ranging from $n = 20$ to $n = 100$. In particular, we choose $n = 20, 50, 80, 100$. At each n , we generate 50 random polynomials. Each polynomial has a 20% sparsity level, which means that only 20% of the basis functions have non-zero coefficients. The coefficients are also randomly generated.

We use three criteria, running time, number of remaining basis functions, and averaged percentage error, to judge the performance of the two approaches. The average percentage error is computed as the sum of the absolute error between the exact values and the approximated values, divided by the number of data points. The results are shown as scatter plots in Figure 5.2. For each criterion, the figures show the comparisons of the two approaches for the randomly generated polynomials at each n .

First, in Figure 5.2a, observe that in most cases, our approach is faster than LASSO. Figure 5.2b shows that our approach usually produces approximations that have larger numbers of basis functions than LASSO does. Finally, Figure 5.2c indicates that the approximations from the two approaches have similar accuracy, with those from LASSO slightly more accurate. From these figures, it can be seen that compared to LASSO, the proposed approach trades the size of the approximations for a lower computational cost. In practice, the accuracy of approximations from our approach is usually comparable with that from LASSO.

5.4.2 Ring Oscillator

Consider a three-stage ring oscillator shown in Figure 5.3. At the steady state, the circuit outputs an oscillating signal with a fixed frequency f , which is determined by the propagation delay of the NOT gate formed by a PMOS and an NMOS transistor. The oscillation frequency f is

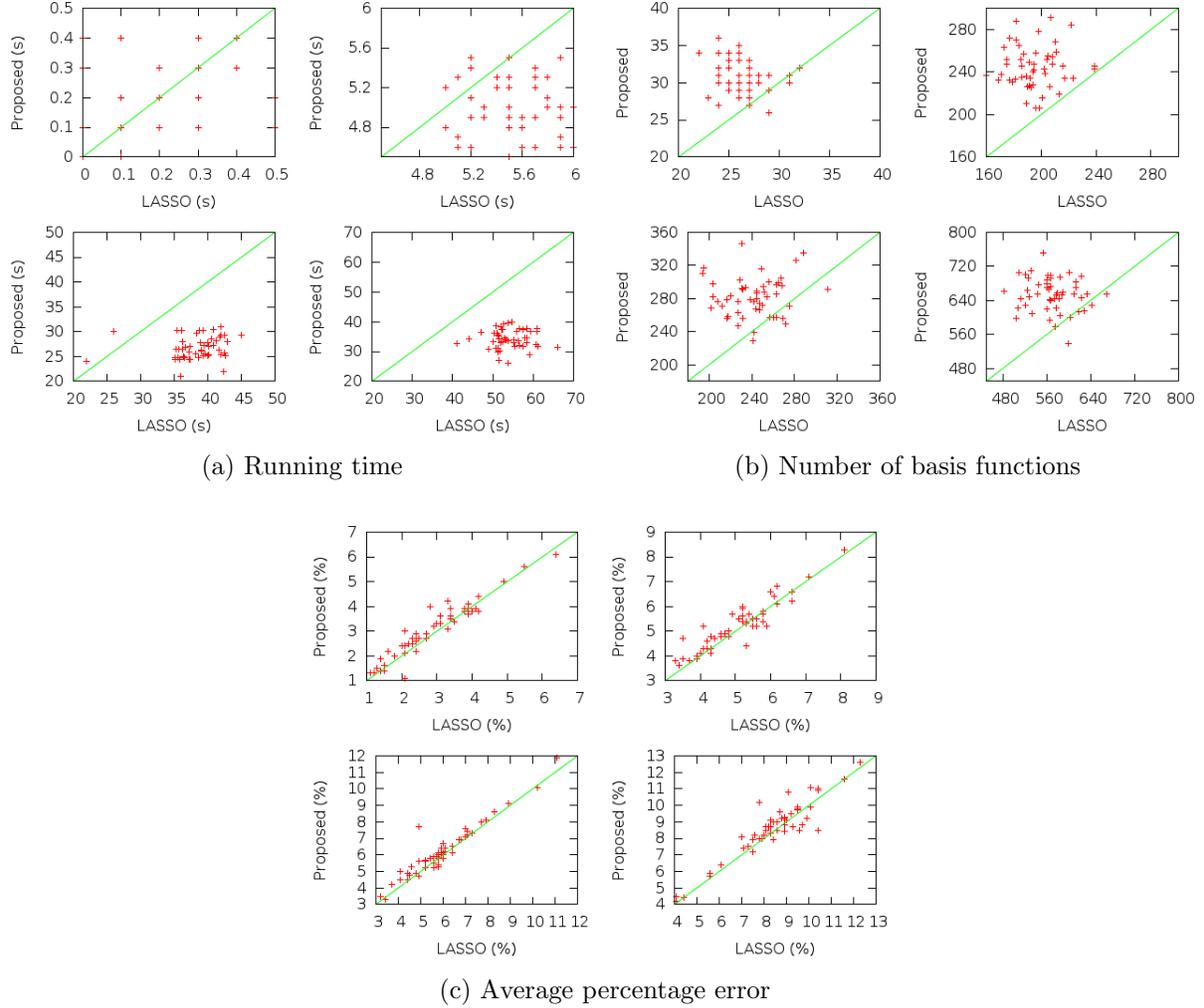


Figure 5.2: Comparisons between LASSO and the proposed approach in terms of time (a), number of basis functions (b), and average percentage error (c), using a set of randomly generated polynomials. The four figures in each group correspond to $n = 20$ (top left), $n = 50$ (top right), $n = 80$ (bottom left), and $n = 100$ (bottom right).

affected by the process parameters in each transistor. Table 5.2 shows the stochastic parameters in the NMOS transistors. The PMOS transistors have the same parameters but with different nominal values. In total, we have 66 stochastic parameters. We are interested to verify the following response specification:

$$(1) 1.8 \text{ GHz} \leq f \leq 2.2 \text{ GHz} .$$

The circuit is designed and simulated in LTSpice[®] [1]. We apply the proposed sparse approx-

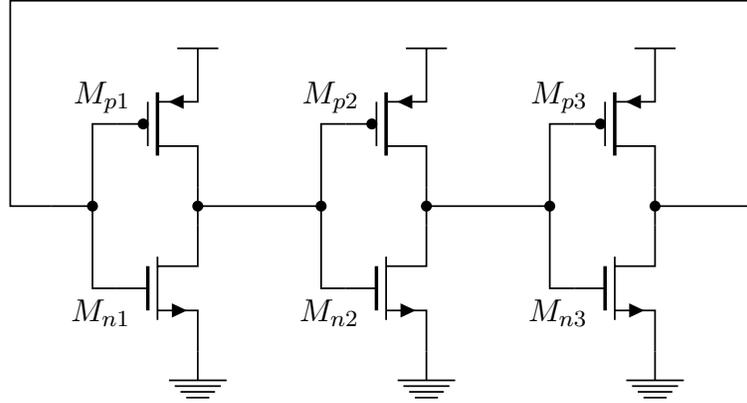


Figure 5.3: A three-stage ring oscillator.

imation algorithm in the context of SSMI (see Chapter 4), i.e., use the algorithm in the regression step of SSMI. The verification results are compared to SSMI with OLS and SSMI with LASSO. Table 5.3 shows the results of the three approaches. To model the response surface of the oscillation frequency f , we use quadratic polynomials as target functions, where the terms in the polynomial are gPC basis functions constructed according to the distribution of the stochastic parameters. The column “Spec” is the index of the specification. Y_r shows the Monte-Carlo yield estimation from 1000 random simulations. Under the “Method” column, “Sparse” represents the proposed sparse approximation algorithm, and “OLS” and “LASSO” represents ordinary least squares and LASSO, respectively. Sim_R and Sim_G show the number of simulations used in regression and generalization of SSMI. T_R and T_G show the time of the form $A + B$ spent in the two steps, where A refers to the simulation time and B is the computation time of SSMI. The columns $|S|$ and d are the number of basis functions in the final approximation (i.e., in the basis functional model introduced in Section 4.1), and the degree of the approximation, respectively. Finally, Y_g under “SSMI” shows the Monte-Carlo yield estimation using 1000 random simulations with respect to statistically sound models.

Observe that all the three methods lead to similar estimated yields. Since the generalization procedures are identical for the three methods, it indicates that the three basis functional models have similar accuracy with respect to the response surface. However, the column $|S|$ shows that

Table 5.2: Stochastic parameters of an NMOS transistor in the ring oscillator. The PMOS transistors have the same parameters but with different nominal values.

	Meaning	Nominal Value	Distribution
w	Channel width	10 μm	$N(\mu_0, 0.05\mu_0)$ $\mu_0 = \text{nominal values}$
l	Channel length	35 nm	
epsrox	Gate dielectric constant relative to vacuum	3.9	
toxe	Electrical gate equivalent oxide thickness	1.15 nm	
toxp	Physical gate equivalent oxide thickness	0.9 nm	
xj	S/D junction depth	10 nm	
ndep	Channel doping concentration	$4.12 \times 10^{18} \text{ cm}^{-3}$	
ngate	Poly Si gate doping concentration	$1 \times 10^{23} \text{ cm}^{-3}$	
nsd	Source/drain doping concentration	$2 \times 10^{20} \text{ cm}^{-3}$	
rsh	Source/drain sheet resistance	5 Ω/\square	
rshg	Gate electrode sheet resistance	0.4 Ω/\square	

the three basis functional models consist of different numbers of basis functions. The numbers of “Sparse” and “LASSO” are close to each other, but are significantly smaller than that of “OLS”. Given that “OLS” takes into account all the basis functions of a degree up to 2, it shows that the response surface of f admits a sparse approximation using only a small fraction of these basis functions. As a consequence, the generalization time (i.e., the second time of T_G) of “OLS” is much larger than that of “Sparse” and “LASSO” since the evaluation time of a polynomial with more than 2000 terms is much longer than that of a polynomial with less than 400 terms. The column d indicates that all the three models have a degree of 2.

To construct the basis functional models, the three methods use 800, 3000 and 800 simulations, respectively. Given that the target function has $\binom{66+2}{2} = 2278$ terms, the problem is under-determined for “Sparse” and “LASSO” and over-determined for “OLS”.¹ Comparing the regression time of “Sparse” and “LASSO”, we find that “Sparse” is more efficient. This is because “Sparse”, which combines gPC and LASSO, employs a filtering stage before formulating the approximation into an \mathcal{L}_1 regularized minimization problem. This stage effectively removes those basis functions that are deemed to be unimportant, leaving a smaller problem to the LASSO solver. In this example, the filtering stage prunes 1042 out of 2211 basis functions of degree 2, resulting in

¹ Due to the computational cost, we do not use the resampling heuristic introduced in Section 4.2.1 to determine the sample size for “OLS”. Instead, we simply take a k -fold average with each fold leaving out exactly once.

Table 5.3: Verification results of the ring oscillator using the proposed sparse approximation algorithm, OLS and LASSO ($\theta_0 = 0.95$ and $T = 100$), with quadratic polynomials as target functions.

Spec	Y_r	SSMI							
		Method	Sim_R	T_R	$ S $	d	Sim_G	T_G	Y_g
1	67.6%	Sparse	800	11 min + 36 s	376	2	265	3 min + 4 s	58.1%
		OLS	3000	0.6 h + 1.2 h	2278		301	4 min + 31 s	59.2%
		LASSO	800	11 min + 58 s	354		336	5 min + 5 s	58.3%

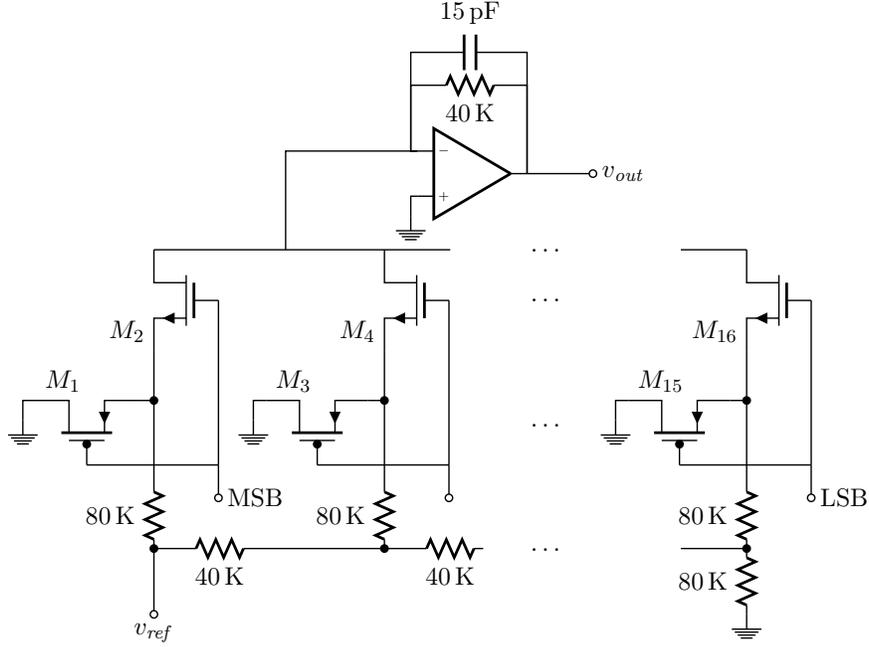


Figure 5.4: An eight-bit digital-to-analog converter.

a LASSO problem with only 1169 unknowns. By contrast, if we apply LASSO directly to compute the coefficients of the degree-2 basis functions, we need to solve a problem with 2211 unknowns.

5.4.3 Digital-to-Analog Converter

Figure 5.4 shows an eight-bit digital-to-analog converter (DAC) [45]. The circuit consists of an operational amplifier, which consists of 16 CMOS transistors and a compensation capacitor, and 8 conversion stages, three of which are shown and the rests are omitted for clarity. In total, the circuit has 32 transistor, 16 resistors and 2 capacitors. This circuit aims to convert the eight-bit digital inputs, denoted by the terminal MSB (most significant bit) through LSB (least significant

Table 5.4: Stochastic parameters of an NMOS transistor in the DAC. The PMOS transistors have the same parameters but with different nominal values.

	Meaning	Nominal Value	Distribution
w	Channel width	-	$N(\mu_0, 0.05\mu_0)$ $\mu_0 = \text{nominal values}$
l	Channel length	$2 \mu\text{m}$	
tox	Gate oxide thickness	38.2 nm	
xj	S/D junction depth	$0.2 \mu\text{m}$	
nsub	Substrate doping concentration	$6.8 \times 10^{15} \text{ cm}^{-3}$	
vto	Zero-biased threshold voltage	0.77 V	
rsh	Source/drain sheet resistance	$0.1 \Omega/\square$	

bit), into an analog signal v_{out} . The output signal v_{out} has a voltage that is in proportion to the digital inputs as a binary number. For instance, given a supply voltage of 5 V, an input of 00000000 leads to an output of 0 V, and 10000000 leads to 2.5 V.

We are interested in two responses of this circuit, zero-code error e_z and gain error e_g . Zero-code error e_z is measured by the value of the output v_{out} when the input signals are all 0. It shows the basis offset of the circuit with respect to an ideal DAC, which has a 0 zero-code error. Gain error e_g indicates how well the slope of the transfer function in a DAC matches the slope of the ideal transfer function. It is measured by the difference between the full-scale range and the actual range as a percent of the full-scale range. In this example, we have a supply voltage of 5 V and assume the following response specifications:

$$(1) e_z \leq 0.3 \text{ V}, (2) e_g \leq 10\%.$$

The stochastic parameters for the NMOS transistors are shown in Table 5.4. The PMOS transistors have the same parameters but with different nominal values. We assume a total of 242 stochastic parameters, 7 for each transistor, 1 for each resistor and 1 for each capacitor.

The circuit is designed and simulated in LTSpice[®] [1], a freely available SPICE simulator. The verification results are shown in Table 5.5. The columns have the same meaning as in Table 5.3. We use quadratic polynomials as target functions for the three methods, the proposed approach, OLS and LASSO. Since there are 242 stochastic parameters, the quadratic target function consists of $\binom{242+2}{2} = 29646$ basis functions. Hence, we spend 38000 simulations in ‘‘OLS’’ with 5-fold

Table 5.5: Verification results of the DAC using the proposed sparse approximation algorithm, OLS and LASSO ($\theta_0 = 0.95$ and $T = 100$), with quadratic polynomials as target functions.

Spec	Y_r	SSMI							
		Method	Sim_R	T_R	$ S $	d	Sim_G	T_G	Y_g
1	94.0%	Sparse	1000	1.1 h + 3 s	234	1	101	7 min + 2 s	85.1%
		OLS	38000	37 h + M/O	29646	2	-	-	-
		LASSO	1000	1.1 h + 211 s	1103	2	175	12 min + 10 s	85.7%
2	99.5%	Sparse	1000	1.1 h + 5 s	227	1	159	9 min + 3 s	98.5%
		OLS	38000	35 h + M/O	29646	2	-	-	-
		LASSO	1000	1.1 h + 199 s	1007	2	231	15 min + 12 s	99.1%

average. Effectively, 30400 simulations are used for each OLS regression. As shown in the table, the regression step for “OLS” runs out of memory (indicated by “M/O”). As a consequence, the subsequent steps cannot be performed.

Now let us focus on comparisons between “Sparse” and “LASSO”. First, notice that for both specifications, they generate basis functional models with similar accuracy, which is indicated by the estimated yields (the Y_g column under “SSMI”). However, the model from “Sparse” has much fewer terms than the model from “LASSO”. In addition, the former has a degree of 1. It is because “Sparse” encounters an early termination, resulting in a model with a degree lower than the degree of the target function. It means that an affine function provides a reasonably good explanation on the relationship between the stochastic parameters and the response. “Sparse” can detect this and terminate without trying basis functions of degree 2. On the other hand, “LASSO” takes all the basis functions into account and constructs a slightly more accurate model. But compared to “Sparse”, the regression time (the second time under T_R) increases from less than 10s to 200s, and the generalization time (the second time under T_G) are also longer. Hence, the overall benefit from applying “LASSO” directly is minimal.

Table 5.6: Verification results of the low-pass filter with different η and ϵ of the proposed algorithm and OLS ($\theta_0 = 0.95$ and $T = 100$), using quadratic polynomials as target functions.

Spec	Y_r	SSMI							
		Method	Sim_R	T_R	$ S $	d	Sim_G	T_G	Y_g
1	99.7%	Sparse-1	300	42 s + 1 s	14	1	161	23 s + 1 s	97.2%
		Sparse-2	300	42 s + 3 s	40	2	134	21 s + 2 s	99.1%
		OLS	1200	3 min + 1 min	903	2	222	32 s + 6 s	98.9%

5.4.4 Low-Pass Filter

The last application is on the low-pass filter that is previously introduced in Section 4.4.2. The circuit has 41 stochastic parameters. Let us consider the response specification

$$(1) f_c \geq 14 \text{ KHz},$$

where f_c is the cutoff frequency of the filter.

We consider the effects of the tuning parameters η and ϵ in the proposed algorithm. We use quadratic polynomials as target functions, which has $\binom{41+2}{2} = 903$ unknown coefficients. Table 5.6 shows the comparison between two different settings of the proposed algorithm, with OLS as the reference case. “Sparse-1” sets η to be 1% of the maximum gPC coefficients for all the basis functions of a certain degree, computed as in (5.8), and ϵ to be $0.01 \cdot \|\mathbf{y}\|_2$. “Sparse-2” sets η to be 0.1% of the maximum gPC coefficients for all the basis functions of a certain degree, and ϵ to be $0.001 \cdot \|\mathbf{y}\|_2$. Clearly, the setting for “Sparse-2” is more conservative in dropping basis functions and early termination.

Observe that the setting for “Sparse-1” leads to an early termination of the algorithm, yielding a degree-1 approximation with 14 basis functions. On the other hand, “Sparse-2” constructs a degree-2 approximation employing 40 basis functions. This demonstrates the effects of η and ϵ . The model from “Sparse-2” is more accurate (indicated by the Y_g column under “SSMI”) than that from “Sparse-1”. However, in practice, we generally prefer the latter since it has reasonable accuracy and more importantly, provides a cleaner explanation on the relationship between the stochastic parameters and the response.

5.5 Summary

This chapter introduces a sparse approximation algorithm that combined gPC and LASSO. The algorithm has two salient features. First, it improves the efficiency of LASSO. Second, it can produce polynomial approximations of degrees lower than the target degree. This chapter also presents several applications of the algorithm in the context of SSML. Compared to OLS and LASSO, the algorithm shows good performance in both accuracy and computational cost.

Chapter 6

Statistically Sound Optimization

Chapter 4 and Chapter 5 discuss the problem of modeling the effects of stochastic parameter variations and statistically verifying response specifications. For a system that fails to satisfy all the specifications, the design has to be optimized so that in the new design, the stochastic parameter variations can be tolerated. Such optimization can happen at two different levels:

- Changing the values of design parameters in the system;
- Redesigning the architecture or the topology of the system.

For designers, the first type of optimization is preferable since it is less expensive than redesigning the whole system. However, there are cases in which no matter how we change the design parameter values, the specifications cannot be satisfied. In those cases, we have to resort the second type of optimization.

This chapter focuses on the first type of optimization. We address the problem of exploring values of design parameters of a black-box system that are “robust” with respect to stochastic parameter variations. For instance, a control system designer often faces the problem of selecting gain values in the controller so that resulting design is correct for the stochastic variations in the plant. Similarly, the problem of designing “robust” analog circuits that can function correctly under stochastic process variations is also well known. Thus, a common theme involves a black-box system whose output responses depend on a few design parameters that are controllable, and many uncontrollable stochastic parameters with known probability distributions. We seek to adjust the

design parameters so that the system satisfies the specifications with a given probability bound.

We present a simulation-based approach, **SSMI-opt**, that combines quantile regression [50] and SSMI introduced in Chapter 4. SSMI-opt aims to verify whether a black-box system is safe, i.e. satisfies all the specifications, at the nominal design point (i.e., the nominal values of design parameters), and if not, search for a new design point at which the system is safe. Compared with SSMI, SSMI-opt uses a different scheme for the verification, which enables the optimization towards the design parameters. SSMI-opt iterates the search for a safe design point using three steps:

- (1) Using quantile regression, construct a relational model that models the response in terms of the design parameters. The effects of stochastic parameter variations are “marginalized”;
- (2) Search for a new design point such that it satisfies all the specifications with respect to the relational model;
- (3) Using SSMI, verify whether in the actual design, the new design point satisfies all the specifications. If not, continue from step (2).

This chapter is organized as follows. The following section presents an overview of SSMI-opt. Section 6.2 reviews quantile regression and shows how it is applied to our problem. Section 6.3 introduces an algorithm which generalizes the model from quantile regression into a statistically sound model at a certain design point. The resulting model is used to verify whether the design point is safe and if not, find a safe design point. Finally, SSMI-opt is demonstrated with several benchmark examples.

6.1 Overview

Consider a black-box system $\mathcal{M} = (\mathbf{u}, \mathbf{x}, \phi, F_{\mathbf{X}}, r)$ 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.¹ Assume that the design parameters are controllable, i.e., we can assign arbitrary values to them, and the stochastic pa-

¹ Notice that unlike in the previous two chapters, in this chapter, we bring back the design parameters in the tuple of black-box systems.

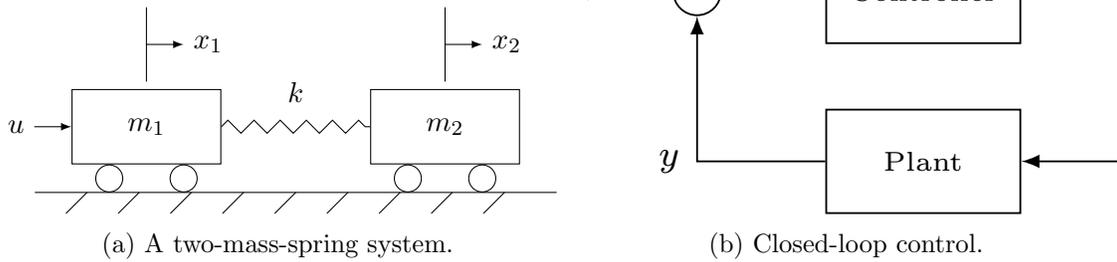


Figure 6.1: A two-mass-spring system and the closed-loop system with a controller.

rameters, which follow the joint distribution $F_{\mathbf{x}}(\mathbf{x})$, are uncontrollable. The response ϕ is defined by the response surface $r(\mathbf{u}, \mathbf{x})$ which has an unknown analytic form. We assume that $r(\mathbf{u}, \mathbf{x})$ is computable. A response specification of a black-box system has the form $\phi \in [a, b]$, which shows acceptable values of ϕ .

Given a black-box system, we aim to find a design point that satisfies all the specifications. The problem is solved as follows. First, we statistically verify whether the system is safe with its nominal design parameters \mathbf{u}_0 and variational stochastic parameters. Formally, it checks whether

$$\Pr_{F_{\mathbf{x}}(\mathbf{x})} (r(\mathbf{u}_0, \mathbf{x}) \in [a, b]) \geq \theta_0 \quad (6.1)$$

is true, where θ_0 is a specified probability. If not, we search for a new design point $\mathbf{u}_{new} \in \mathbb{U}$ that satisfies (6.1). Note that although SSMI can also verify (6.1) as shown in Chapter 4, it is not able to search for new design points if the system at the nominal point is unsafe.

In the following, we introduce a running example that illustrates SSMI-opt. The example is first shown in Section 3.2. For the convenience of reading, it is presented anew.

Example 6.1.1 (A Two-Mass-Spring System). A two-mass-spring system [86] is shown in Figure 6.1a. 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 6.1b, a controller is used to track y with r , the reference position.

A lead compensator controls the plant, which has two tunable parameters, the pole location $p \in [-1200, -800]$ and the zero location $z \in [-1.2, -0.8]$. The design parameters p and z have the following nominal values: $p_0 = -1000$ and $z_0 = -1$. The goal is to design the controller so that the step response $y(t)$ of the closed-loop system satisfies

- The settling time t_s is less than 2.5 s,

$$(1) t_s \leq 2.5 \text{ s};$$

- The overshoot of the step response, r_o , as a percentage of the steady state value, is less than 15%,

$$(2) r_o \leq 15\%. \quad \parallel$$

The key idea of SSMI-opt is to construct an empirical model that is statistically sound with respect to certain design points for the response surface $r(\mathbf{u}, \mathbf{x})$. Such a model is constructed by quantile regression (detailed in Section 6.2) and the generalization technique from SSMI (detailed in Section 6.3). It provides statistical soundness guarantee at the design points of interest.

Now let us first recall the meaning of statistical soundness (defined in Chapter 3, Definition 3.2.3). For a black-box system \mathcal{M} , a θ_0 statistically sound model $g(\mathbf{u}, \mathbf{x})$ of the response surface $r(\mathbf{u}, \mathbf{x})$ satisfies

$$\Pr_{F_{\mathbf{x}}(\mathbf{x})} (r(\mathbf{u}, \mathbf{x}) \in g(\mathbf{u}, \mathbf{x})) \geq \theta_0, \quad \mathbf{u} \in \{\mathbf{u}_1, \dots, \mathbf{u}_n\},$$

where $\{\mathbf{u}_1, \dots, \mathbf{u}_n\}$ is a set of design points with respect to which the model $g(\mathbf{u}, \mathbf{x})$ is statistically sound. In this chapter, the model $g(\mathbf{u}, \mathbf{x})$ is derived from a relational model $\hat{g}(\mathbf{u})$, which is only in terms of the design parameters \mathbf{u} . In the following, we write $g(\mathbf{u})$ instead of $g(\mathbf{u}, \mathbf{x})$ to emphasize that the model is independent of the stochastic parameters \mathbf{x} .

Figure 6.2 shows a high-level flow of SSMI-opt. First, using quantile regression, we compute a relational model

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

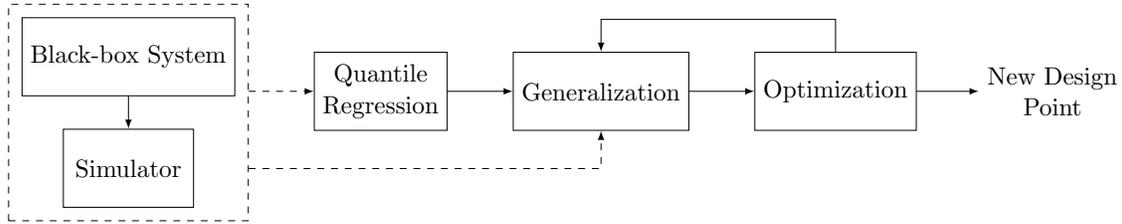


Figure 6.2: A high-level flow of SSMI-opt.

where $\hat{g}_\ell(\mathbf{u})$ and $\hat{g}_u(\mathbf{u})$ are affine functions. The model $\hat{g}(\mathbf{u})$ approximates the response surface $r(\mathbf{u}, \mathbf{x})$ with $\mathbf{u} \in \mathbb{U}$ and $\mathbf{x} \in \mathbb{X}$. The simulation data used in quantile regression consist of a random sample on the design and the stochastic parameters and the values of the response. Note that \hat{g} is not guaranteed to be statistically sound.

Next, we check whether the nominal design point \mathbf{u}_0 satisfies the specifications under stochastic parameter variations. This is achieved by applying the generalization procedure of SSMI, which derives a relational model $g(\mathbf{u})$ from $\hat{g}(\mathbf{u})$ that is statistically sound at \mathbf{u}_0 . Intuitively, the procedure fixes the design parameters to \mathbf{u}_0 and samples the stochastic parameters sequentially. A tolerance interval $[\ell, u]$ is computed so that a long enough sequence of the observed responses fall in the interval

$$[\hat{g}_\ell(\mathbf{u}_0) + \ell, \hat{g}_u(\mathbf{u}_0) + u]. \quad (6.2)$$

The interval (6.2) is statistically sound with respect to the possible response values at \mathbf{u}_0 . Hence,

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

is statistically sound at \mathbf{u}_0 . For a response specification $\phi \in [a, b]$, if (6.2) is contained in $[a, b]$, we conclude that with a high probability (which depends on θ_0), the system is safe at \mathbf{u}_0 . Otherwise, we search for a new design point that yields a safe system.

The search is performed with respect to the relational model $g(\mathbf{u})$. We aim to find a design point $\mathbf{u}_{new} \in \mathbb{U}$ that has the largest margin from violating the specifications. Since $g(\mathbf{u})$ is statistically sound only at \mathbf{u}_0 , the point \mathbf{u}_{new} is not guaranteed to satisfy the specifications in the actual system. Hence, to claim that \mathbf{u}_{new} is a safe design point, we apply generalization again to

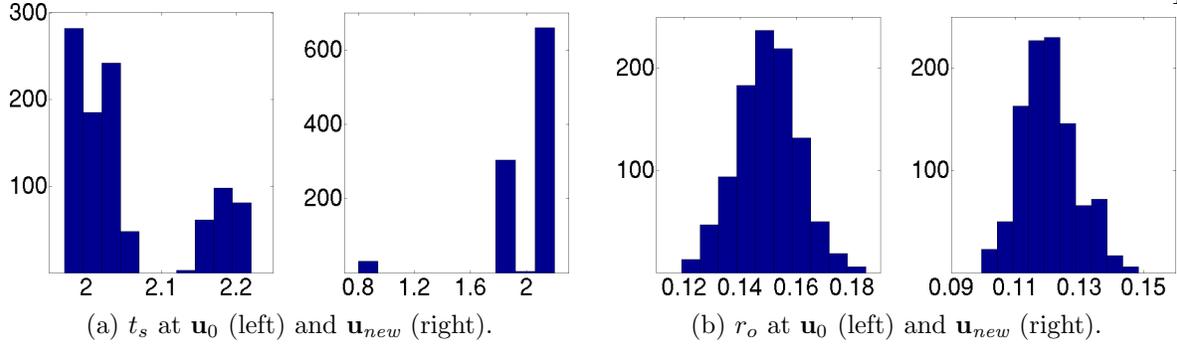


Figure 6.3: Histogram of the settling time t_s (left, in seconds) and the overshoot percentage r_o (right, as percentage) in the two-mass-spring system.

transform $g(\mathbf{u})$ into a statistically sound model at \mathbf{u}_{new} , and check whether the specifications hold. The procedure continues until either we show that the system is statistically safe at some \mathbf{u}_{new} , or no new point can be found. In the later case, it is still possible that there exists design points that satisfy the specifications since the search is done with respect to a statistical over-approximation of the response surface. In SSMI-opt, we simply report that we cannot find a safe design point for $\mathbf{u} \in \mathbb{U}$ and $\mathbf{x} \in \mathbb{X}$.

Example 6.1.2 (An Optimized Two-Mass-Spring System). Let us continue with Example 6.1.1. We simulate the system with randomly sampled design parameters $p \in [-1200, -800]$ and $z \in [-1.2, -0.8]$, and stochastic parameters $m_1 \in [0.8, 1.2]$, $m_2 \in [0.8, 1.2]$ and $k \in [0.8, 1.2]$. Using SSMI-opt, we show that the nominal design point $\mathbf{u}_0 = (p_0, z_0)$, where $p_0 = -1000$ and $z_0 = -1$, is unsafe. In addition, we find a safe design point $\mathbf{u}_{new} = (p_{new}, z_{new})$, where $p_{new} = -1200$ and $z_{new} = -0.928$. Figure 6.3 shows the histograms of the settling time t_s and the overshoot percentage r_o at \mathbf{u}_0 and \mathbf{u}_{new} . Apparently, the system violates the specification $r_o \leq 15\%$ at \mathbf{u}_0 . After optimization, the histograms confirm that both specifications are satisfied. \parallel

The following sections present the technical details of quantile regression, generalization, and optimization in the context of SSMI-opt.

6.2 Quantile Regression

This section briefly reviews quantile regression and shows how to compute lower and upper bound functions using quantile regression. First, we recall the meaning of quantile. For a real-valued random variable X with a distribution $F_X(x) = \Pr(X \leq x)$, 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 $\Pr(X \geq x)$ is at most $1 - \tau$.

Consider a black-box system $\mathcal{M} = (\mathbf{u}, \mathbf{x}, \phi, F_{\mathbf{X}}, r)$ with design parameters \mathbf{u} , stochastic parameters \mathbf{x} and a response $\phi = r(\mathbf{u}, \mathbf{x})$. We write $r(\mathbf{u})$ to denote the marginalized response surface, which is a relational model such that for all $\mathbf{u} \in \mathbb{U}$ and $\mathbf{x} \in \mathbb{X}$,

$$r(\mathbf{u}) = [\min(r(\mathbf{u}, \mathbf{x})), \max(r(\mathbf{u}, \mathbf{x}))].$$

For a fixed \mathbf{u} , $r(\mathbf{u})$ can be regarded as a random variable. A τ th quantile function $g_\tau(\mathbf{u}) = Q_X(\tau)$, where $X = r(\mathbf{u})$, maps the design parameters onto the τ th quantile of the marginalized response surface $r(\mathbf{u})$. In SSMI-opt, the goal of quantile regression is to approximate the quantile function $g_\tau(\mathbf{u})$ with an affine function of the form

$$\hat{g}_\tau(\mathbf{u}; \mathbf{c}) = c_0 + \sum_{i=1}^m c_i u_i,$$

where $\mathbf{c} = (c_0, c_1, \dots, c_m)$ 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 $\hat{g}_\tau(\mathbf{u})$,

$$\min_{\mathbf{c}} \|g_\tau(\mathbf{u}) - \hat{g}_\tau(\mathbf{u}; \mathbf{c})\|. \quad (6.3)$$

Since $g_\tau(\mathbf{u})$ is often not available, (6.3) 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 $\{\mathbf{u}^{(i)}, \mathbf{x}^{(i)}, \phi^{(i)}\}$ with N data points, quantile regression relies on the following penalty function,

$$\rho_\tau(\mathbf{e}) = \sum_{\substack{i=1 \\ e_i \geq 0}}^N \tau e_i + \sum_{\substack{i=1 \\ e_i \leq 0}}^N (\tau - 1) e_i, \quad (6.4)$$

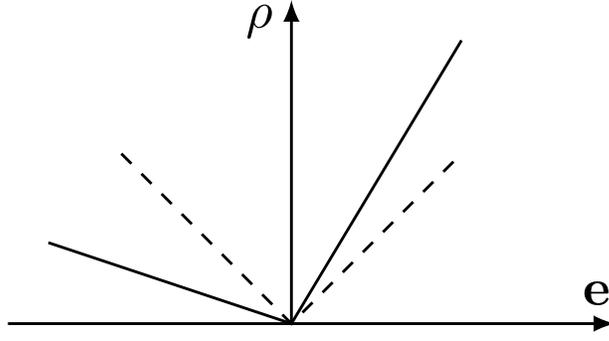


Figure 6.4: An example of (6.4) with $\tau > 0.5$ (solid) and a \mathcal{L}_1 penalty function (dashed), a special case of (6.4) with $\tau = 0.5$.

where $e_i = \phi^{(i)} - \hat{g}_\tau(\mathbf{u}^{(i)})$ are the residuals between the response 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), (6.4) 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 (6.4) leads to the following optimization problem.

$$\min_{\mathbf{c}} \rho_\tau(r(\mathbf{u}, \mathbf{x}) - \hat{g}_\tau(\mathbf{u}; \mathbf{c})) . \quad (6.5)$$

Since (6.4) is piecewise linear, it has a unique minimum. Figure 6.4 shows a comparison between (6.4) and the \mathcal{L}_1 penalty function, i.e., $\tau = 0.5$.

The problem (6.5) is solved as a linear program [50]. The penalty function (6.4) is encoded by adding auxiliary variables $\mathbf{s} = (s_1, \dots, s_N)$ and $\mathbf{t} = (t_1, \dots, t_N)$. 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}_τ , respectively. With \mathbf{s} and \mathbf{t} , we write (6.5) as

$$\begin{aligned} \min_{\mathbf{c}} \quad & \sum_{i=1}^N \tau s_i + \sum_{i=1}^N (1 - \tau) t_i \\ \text{subject to} \quad & \\ & \phi^{(i)} - \hat{g}_\tau(\mathbf{u}^{(i)}; \mathbf{c}) = s_i - t_i, \quad i = 1, 2, \dots, N, \\ & \mathbf{s} \geq 0, \quad \mathbf{t} \geq 0. \end{aligned} \quad (6.6)$$

The first constraint forces \mathbf{s} and \mathbf{t} to be complementary. To minimize the objective function, at most one of s_i and t_i should be non-zero. The last two constraints ensures \mathbf{s} and \mathbf{t} to be non-negative.²

Example 6.2.1 (A Two-Mass-Spring System - Quantile Regression). Let us continue from Example 6.1.2 and elaborate the process of quantile regression. With a set of simulation data, we compute a lower bound function $\hat{g}_\ell(\mathbf{u})$ of the settling time t_s and the overshoot percentage r_o using $\tau = 0.01$, and an upper bound function $\hat{g}_u(\mathbf{u})$ using $\tau = 0.99$. For t_s , we have

$$\begin{aligned}\hat{g}_\ell(p', z') &= 1.157 + 0.040p' + 0.707z', \\ \hat{g}_u(p', z') &= 2.220 + 0.001p' - 0.051z' .\end{aligned}\tag{6.7}$$

For r_o , we have

$$\begin{aligned}\hat{g}_\ell(p', z') &= 0.129 - 0.006p' + 0.078z', \\ \hat{g}_u(p', z') &= 0.198 - 0.017p' + 0.086z' .\end{aligned}\tag{6.8}$$

Note that in these functions, p' and z' are the parameters p and z normalized to the interval $[-1, 1]$.

Hence the nominal design point \mathbf{u}_0 corresponds to $p' = 0$ and $z' = 0$. ||

It is important to understand that the formulation in (6.6) only solves for $\tau \in (0, 1)$. For $\tau = 0$ and $\tau = 1$, (6.6) fails to find the maximum lower bound and the minimum upper bound. This is because in the two cases, (6.4) 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_s in Example 6.2.1 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 $\hat{g}_\tau(\mathbf{u})$ is not necessarily close to the true lower (upper) bound. In the case that there are outliers in the simulation data, $\hat{g}_\tau(\mathbf{u})$ can be distant from the true bound. On the contrast, $\hat{g}_\tau(\mathbf{u})$ 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 $\hat{g}_\ell(\mathbf{u})$ and $\hat{g}_u(\mathbf{u})$ to indicate the estimated lower and the upper bound, respectively. By default, we assume that $\hat{g}_\ell(\mathbf{u})$ is computed with $\tau = 0.01$ and $\hat{g}_u(\mathbf{u})$ with $\tau = 0.99$.

² Notice the sign change in the second sum of the objective function in (6.4) and (6.6).

6.3 An Iterative Optimization Algorithm

As mentioned in Section 6.1, $\hat{g}_\ell(\mathbf{u})$ and $\hat{g}_u(\mathbf{u})$ form a relational model $\hat{g}(\mathbf{u}) \equiv [\hat{g}_\ell(\mathbf{u}), \hat{g}_u(\mathbf{u})]$. Clearly, $\hat{g}(\mathbf{u})$ is not necessarily statistically sound and thus does not provide guarantees on the behavior of the actual system. This section shows how to apply the generalization technique from SSMI to transform the relational model $\hat{g}(\mathbf{u})$ into a statistically sound model $g(\mathbf{u})$ with respect to some design point. The resulting model $g(\mathbf{u})$ is used to check whether the design point is safe and if not, search for a new point that satisfies the specifications.

6.3.1 Generalization of Relational Models

Recall that statistical soundness is defined with respect to a finite set of design points $\{\mathbf{u}_1, \dots, \mathbf{u}_n\}$. Since our goal is to learn whether the specifications hold at the nominal design point \mathbf{u}_0 and if not, find a new point \mathbf{u}_{new} that satisfies them, we are only concerned with statistical soundness at \mathbf{u}_0 and some \mathbf{u}_{new} .

For some fixed design point, $\hat{g}(\mathbf{u})$ becomes an interval. We aim to derive a tolerance interval $[\ell, u]$ such that

$$\Pr_{F_{\mathbf{X}}(\mathbf{x})} (r(\mathbf{u}, \mathbf{x}) \in [\hat{g}_\ell(\mathbf{u}) + \ell, \hat{g}_u(\mathbf{u}) + u]) \geq \theta_0. \quad (6.9)$$

The interval $[\hat{g}_\ell(\mathbf{u}) + \ell, \hat{g}_u(\mathbf{u}) + u]$ is a statistically sound bound for the response ϕ at the fixed design point \mathbf{u} under the stochastic parameter variations. The problem (6.9) can be solved by the generalization procedure introduced in Section 4.3. Recall that for generalization to work, we need to specify a probability θ_0 and a threshold T for the sequential Bayesian test. The parameter θ_0 represents the coverage of the statistically sound model with respect to distribution of the stochastic parameters, and T specifies the confidence level of drawing a correct conclusion based on a finite set of observations. The two parameters are used to compute a run length K ,

$$K = -\frac{\log(T + 1)}{\log \theta_0} - 1 \quad (6.10)$$

Input: Black-box System \mathcal{M} , Relational Model $\hat{g}(\mathbf{u}) = [\hat{g}_\ell(\mathbf{u}), \hat{g}_u(\mathbf{u})]$, Design Point \mathbf{u} , Probability θ_0 , Threshold T

Output: Statistically Sound Model $g(\mathbf{u})$ at \mathbf{u}

```

1  $K = -\frac{\log(T+1)}{\log \theta_0} - 1$  ;
2  $\ell, u, \text{count} = 0$  ;
3 while  $\text{count} < K$  do
4    $\mathbf{x}^{(i)}$  = draw a point following the distribution of the stochastic parameters ;
5    $\phi^{(i)}$  = Simulate  $\mathcal{M}$  with  $\mathbf{x} = \mathbf{x}^{(i)}$  ;
6   if  $\phi^{(i)} \notin [\hat{g}_\ell(\mathbf{u}) + \ell, \hat{g}_u(\mathbf{u}) + u]$  then
7      $\text{count} = 0$  ;
8      $\ell = \min(\phi - \hat{g}_\ell(\mathbf{u}), \ell)$  ;
9      $u = \max(\phi - \hat{g}_u(\mathbf{u}), u)$  ;
10  else
11     $\text{count} = \text{count} + 1$  ;
12  end
13 end
14  $g(\mathbf{u}) = [\hat{g}_\ell(\mathbf{u}) + \ell, \hat{g}_u(\mathbf{u}) + u]$  ;

```

Algorithm 5: An generalization algorithm that provides statistical soundness at fixed \mathbf{u} .

such that once we collect K consecutive observations that satisfy

$$\phi^{(i)} \in [\hat{g}_\ell(\mathbf{u}) + \ell, \hat{g}_u(\mathbf{u}) + u], \quad (6.11)$$

we terminate the generalization procedure and report $[\ell, u]$ as the tolerance interval.

Algorithm 5 shows the algorithm that generalizes the relational model $\hat{g}(\mathbf{u})$ into a statistically sound model $g(\mathbf{u})$ at some given design point \mathbf{u} . The inputs of the algorithm are the black-box system \mathcal{M} , the relational model $\hat{g}(\mathbf{u}) = [\hat{g}_\ell(\mathbf{u}), \hat{g}_u(\mathbf{u})]$, a fixed design point \mathbf{u} , a probability θ_0 , and a threshold T . The algorithm first computes a run length K according to (6.10) (see Section 4.3 for details), and initialize the interval $[\ell, u]$ and a `count` variable to 0. The `count` variable records the number of consecutive supportive observations. Next, we sample the stochastic parameters and simulate the system. If (6.11) holds for some $\phi^{(i)}$, `count` is incremented by 1. Otherwise, it is reset to 0 and the interval $[\ell, u]$ is updated so that (6.11) becomes valid. The algorithm terminates if `count` reaches K . In this case, the model

$$g(\mathbf{u}) = [\hat{g}_\ell(\mathbf{u}) + \ell, \hat{g}_u(\mathbf{u}) + u] \quad (6.12)$$

is θ_0 statistically sound at the design point \mathbf{u} .

Notice that Algorithm 5 is similar to Algorithm 3 in Section 4.3. The only difference between the two algorithms lies in that the tolerance interval in Algorithm 3 is computed with respect to the basis functional model, a function of the stochastic parameters, whereas the tolerance interval in Algorithm 5 is with respect to a relational model $\hat{g}(\mathbf{u})$ that models the possible values of the response in terms of the design parameters. As we see in the following, the use of the relational model $\hat{g}(\mathbf{u})$ enables the optimization of the system.

Theorem 6.3.1. Algorithm 5 terminates with probability one.

Proof. The proof of this theorem is almost identical to the proof of Theorem 4.3.3 in Section 4.3 and is left to interested readers. \square

Algorithm 5 yields a θ_0 statistically sound model $g(\mathbf{u})$ at a given design point \mathbf{u} . We claim that for the design point \mathbf{u} , we have a high level of confidence that the response ϕ has a probability of at least θ_0 falling in the interval indicated by (6.12). Section 2.1.2.3 and Section 4.3 show that the level of confidence is at least $1 - \frac{1}{T+1}$. Hence with large θ_0 and T , the interval (6.12) is close to a true over-approximation of the possible values of the response ϕ under the stochastic parameter variations. Hence, to verify whether specifications $\phi \in [a, b]$ hold at the design point \mathbf{u} , we simply check whether (6.12) 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 a new design point.

Example 6.3.1 (A Two-Mass-Spring System - Generalization). Continued from Example 6.2.1, we show how the relational models (6.7) for t_s and (6.8) for r_o are generalized. At the nominal design point \mathbf{u}_0 , (6.7) becomes an interval $[1.157, 2.220]$ and (6.8) becomes $[0.129, 0.198]$. With $\theta_0 = 0.95$ and $T = 100$, the interval for t_s is shown to be a statistically sound bound for t_s and the interval for r_o is generalized into $[0.121, 0.198]$. Hence, the statistically sound model for t_s at \mathbf{u}_0 is

$$\begin{aligned}\hat{g}_\ell(p', z') &= 1.157 + 0.040p' + 0.707z', \\ \hat{g}_u(p', z') &= 2.220 + 0.001p' - 0.051z',\end{aligned}\tag{6.13}$$

Input: Statistically Sound Model $g_0(\mathbf{u})$ at the nominal design point \mathbf{u}_0 , Response Specification $\phi \in [a, b]$

Output: New Design Point \mathbf{u}_{new}

```

1  $i = 0$  ;
2 while true do
3    $i = i + 1$  ;
4    $\mathbf{u}_i =$  pick up a candidate design point that satisfies  $\phi \in [a, b]$  according to  $g_{i-1}(\mathbf{u})$  ;
5   if  $\mathbf{u}_i$  is not available then
6     Report that a new design point cannot be found ;
7     break ;
8   end
9    $g_i(\mathbf{u}) =$  generalize  $g_{i-1}(\mathbf{u})$  into a statistically sound model at  $\mathbf{u}_i$  ;
10  if  $\mathbf{u}_i$  satisfies  $\phi \in [a, b]$  according to  $g_i(\mathbf{u})$  then
11     $\mathbf{u}_{new} = \mathbf{u}_i$  ;
12    break ;
13  end
14 end

```

Algorithm 6: An iterative algorithm that finds a safe design point.

which is the same as the relational model (6.7), and that for r_o is

$$\begin{aligned}\hat{g}_\ell(p', z') &= 0.121 - 0.006p' + 0.078z', \\ \hat{g}_u(p', z') &= 0.198 - 0.017p' + 0.086z'.\end{aligned}\tag{6.14}$$

Therefore, at the nominal design point \mathbf{u}_0 , t_s satisfies the response specification $t_s \leq 2.5$ s and r_o violates the specification $r_o \leq 15\%$. ||

6.3.2 Optimization

Suppose that for the black-box system \mathcal{M} , the nominal design point \mathbf{u}_0 is not safe, i.e., does not satisfy all the specifications. We denote the model in (6.12) as $g_0(\mathbf{u})$, indicating that it is statistically sound at \mathbf{u}_0 . To find a new design point, we introduce an iterative procedure shown in Algorithm 6. At the i th iteration, we try to find a candidate \mathbf{u}_i that is safe **with respect to $g_{i-1}(\mathbf{u})$** . 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 that satisfies all the specifications.

Assume that \mathbf{u}_i is found. Since $g_{i-1}(\mathbf{u})$ is not guaranteed to be statistically sound at \mathbf{u}_i , we

apply Algorithm 5 to transform $g_{i-1}(\mathbf{u})$ into a statistically sound model at \mathbf{u}_i , denoted as $g_i(\mathbf{u})$. The new model $g_i(\mathbf{u})$ is used to check whether \mathbf{u}_i is a safe design point. If yes, we report $\mathbf{u}_{new} = \mathbf{u}_i$ and conclude that with a high probability, the system \mathcal{M} with the design parameter values \mathbf{u}_{new} satisfies the specifications. Otherwise, we try to find another design point using the new model $g_i(\mathbf{u})$.

Given that the model $g_i(\mathbf{u})$ consists of affine functions as the lower and the upper bound,

$$g_i(\mathbf{u}) = [\hat{g}_{i\ell}(\mathbf{u}) + \ell, \hat{g}_{iu}(\mathbf{u}) + u] ,$$

it is easy to pick up a candidate point 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. For a specification $\phi \in [a, b]$, the solution is to search for the point that has the largest margin from violating the specification using the following linear program:

$$\begin{aligned} & \max_{\mathbf{u}_i \in \mathbb{U}} (b - \hat{g}_{iu}(\mathbf{u}_i) - u) + (\hat{g}_{i\ell}(\mathbf{u}_i) + \ell - a) \\ & \text{subject to} \\ & a \leq \hat{g}_{i\ell}(\mathbf{u}_i) + \ell \leq \hat{g}_{iu}(\mathbf{u}_i) + u \leq b. \end{aligned} \tag{6.15}$$

Obviously, if (6.15) is infeasible, then we cannot find any candidate design point. It is also immediate to extend the linear program (6.15) to handle multiple specifications.

6.4 Applications

We present three applications: (1) a ring oscillator circuit modeled at the transistor-level, (2) an insulin pump that controls the blood glucose level of diabetic patients, and (3) an aircraft flight control model. All models have stochastic parameter variations. We use SSMI-opt to search for safe design points of these systems. The experiments are performed on a AMD Athlon II quad-core 2.8 GHz CPU with 4 G RAM. SSMI-opt is implemented in Python-2.7.

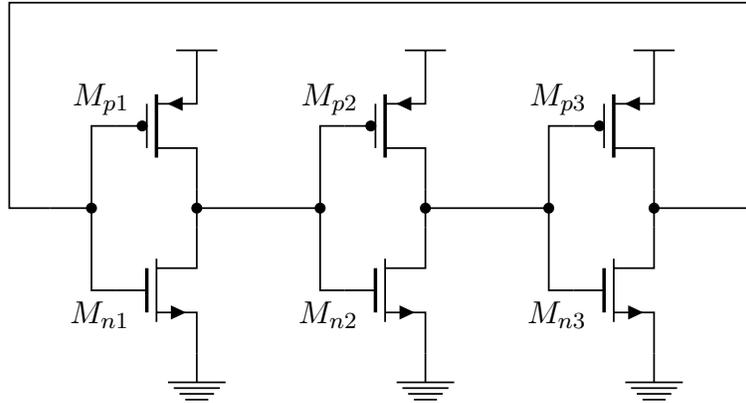


Figure 6.5: A three-stage ring oscillator.

6.4.1 Ring Oscillator

Figure 6.5 shows a three-stage ring oscillator. It is designed to oscillate at a frequency $f = 2.1$ GHz with a power consumption $w = 5$ 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. For this circuit, the response specifications are

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

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 satisfied under the nominal design point and if not, choose new values for the width and length of each transistor. For the original design, the channel width of each NMOS transistor, W_n , is $10 \mu\text{m}$ and that of each PMOS transistor, W_p , is $20 \mu\text{m}$. The channel lengths of NMOS and PMOS transistors, L_n and L_p , are 35 nm . These values are chosen based on manually tuning.

For a ring oscillator, the transistors of the same type usually have the same channel width and length. In order to avoid choosing meaningless design point, we add this as an additional constraints in the search of a candidate point (i.e., in (6.15)). We use LTSpice[®] [1], a freely available SPICE simulator, to simulate the circuit. The results are shown in Table 6.1. The column Y_7 shows the

Table 6.1: Optimization results for the three-stage ring oscillator ($\theta = 0.95$ and $T = 100$). The unit of I_o and I_{new} for specification (1) is GHz, and that for specification (2) is mW.

Spec	Y_r		SSMI-opt							
	\mathbf{u}_0	\mathbf{u}_{new}	I_0	Sim_R	T_R	Iters	Sim_G	T_G	T_O	I_{new}
1	95.8%	98.9%	[2.05, 2.23]				309			[2.04, 2.19]
2	60.1%	100%	[5.18, 5.85]	500	307 s	1	332	233 s	1 s	[4.75, 5.41]
all	60.0%	98.9%					-			

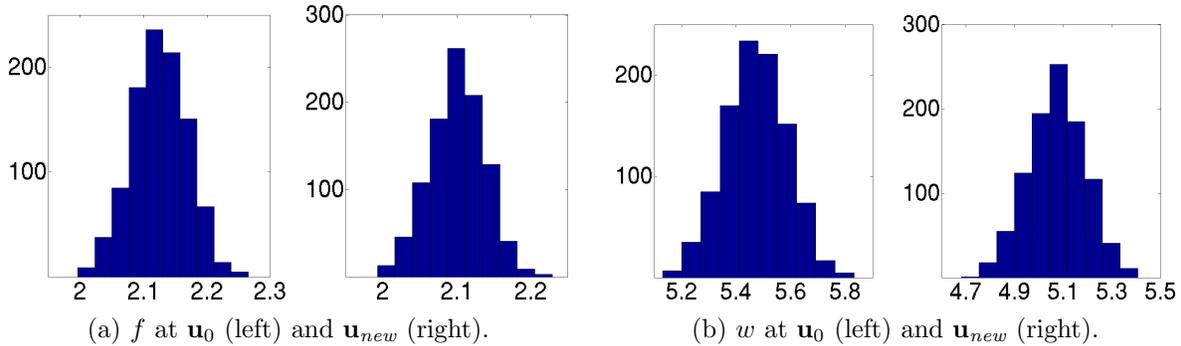


Figure 6.6: Histograms of f (left, GHz) and w (right, mW) at in the ring oscillator.

yields of each specification at \mathbf{u}_0 and \mathbf{u}_{new} estimated through 1000 Monte-Carlo simulations. Sim_R and Sim_G are the number of simulations used in quantile regression and generalization, respectively. Sim_G represents the total number of simulations for all the iterations. T_R , T_G and T_O are the time spent in quantile regression, generalization, and optimization. T_R and T_G include the time for simulation. “Iters” shows the number of iterations used to find the new design point. Finally, I_0 and I_{new} are the statistically sound bound for the responses at \mathbf{u}_0 and \mathbf{u}_{new} .

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_0 violates the specification (2) excessively. The new design point found by our approach is $W_n = 9 \mu\text{m}$, $W_p = 16 \mu\text{m}$ and $L_n = L_p = 35 \text{ nm}$. This design point yields performance bounds that satisfies both specifications, which is confirmed by the Monte-Carlo yield estimation. The yield is boosted from 60% to almost 100%. Figure 6.6 shows the histograms of the two responses, f and w , at \mathbf{u}_0 and \mathbf{u}_{new} . Obviously, we have a significant performance improvement.

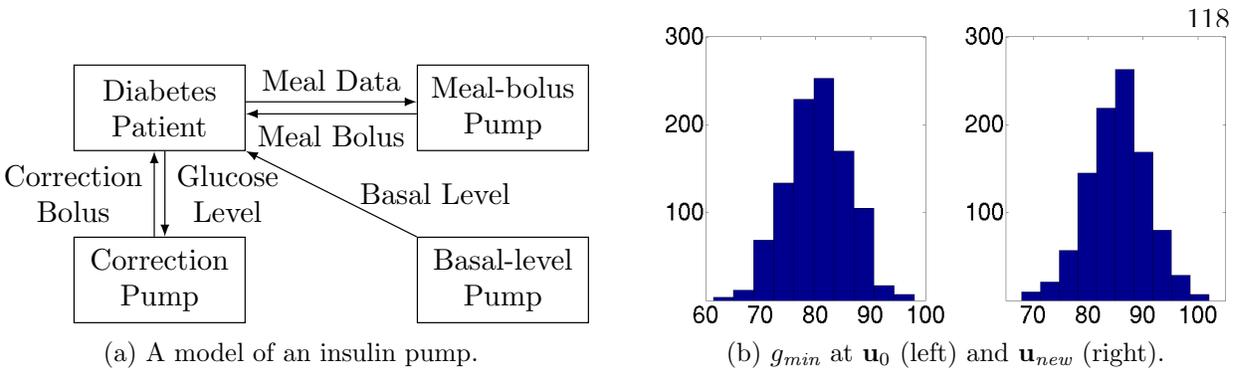


Figure 6.7: A model of an insulin pump (left) and the histograms of $\min(g(t))$, the minimum glucose level during simulation (right).

6.4.2 Insulin Pump

We study a previously published model of an insulin pump used by type-1 diabetic patients [73, 21]. Our model incorporates a physiological model of the human insulin-glucose response from Dalla Man et al. [21], models of sensor errors and a typical pump usage by type-1 diabetic patients [73]. 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, 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 [73]. 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. [21]. Our study here focuses on a single model patient. The total simulation time is 1400 min.

There are many important correctness properties. Ideally, the human blood glucose level

should be between 70 mg/dl and 180 mg/dl. A level lower than 70 mg/dl is called **hypoglycemia**, and a level higher than 180 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 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 mg/dl and 240 mg/dl over $t \in [0, T]$ where T is the total simulation time.

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

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

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

Table 6.2 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 $\text{basal} = 0.3$, $\text{icRatio} = 0.06$ and $\text{cor} = 0.06$. Observe that the pump works well except that it has a 3.8% chance of dangerous hypoglycemia. **SSMI-opt 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 $\text{basal} = 0.225$, $\text{icRatio} = 0.080$ and $\text{cor} = 0.049$. Histograms of $\min(g(t))$ at \mathbf{u}_0 and \mathbf{u}_{new} are shown in Figure 6.7b.

6.4.3 Aircraft Flight Control System

Figure 6.8 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

Table 6.2: Optimization results for the insulin pump model ($\theta = 0.95$ and $T = 100$). The units of I_0 and I_{new} for specification (1) and (2) are mg/dl, and that for specification (3) is min.

Spec	Y_r		SSMI-opt							
	\mathbf{u}_0	\mathbf{u}_{new}	I_0	Sim_R	T_R	Iters	Sim_G	T_G	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]				423			[48.6, 213.3]
4	100%	100%	[6.0%, 18.8%]				420			[6.2%, 20.0%]
all	96.2%	99.6%	-							

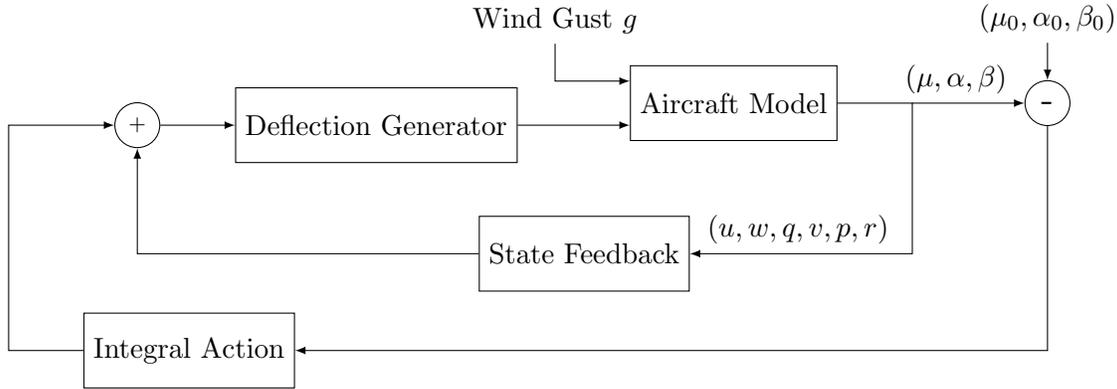


Figure 6.8: An aircraft flight control model.

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 matrices³ along with the stochastic wind disturbance. In all, we have 73 stochastic parameters. The following specifications concern the step response of $\mu(t)$, $\alpha(t)$

³ Originally, this system concerns with fault-tolerant control of an aircraft. It discusses how to find design parameters so that the aircraft can maintain its performance when there are actuator failures. We consider modeling uncertainties rather than actuator failures.

Table 6.3: Optimization results for the aircraft flight control model ($\theta = 0.95$ and $T = 100$).

Spec	Y_r		SSMI-opt											
	\mathbf{u}_0	\mathbf{u}_{new}	I_0	Sim_R	T_R	Iters	Sim_G	T_G	T_O	I_{new}				
1	100%	100%	[1.40, 6.47]s				326			[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	500	307s	1	479	341s	2s	[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%	-											

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 6.3 presents the results of applying our approach. Observe that the specification (2) and (5) are not satisfied at \mathbf{u}_0 , confirmed by both the Monte-Carlo simulations and the performance bounds I_0 . 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 6.9 shows the histograms of t_α and r_α at \mathbf{u}_0 and \mathbf{u}_{new} , which clearly shows the performance improvement.

Now let us compare I_0 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.

6.5 Summary

This chapter introduces SSMI-opt, a design parameter optimization technique for black-box systems under stochastic parameter variations. SSMI-opt combines quantile regression and the generalization technique of SSMI. Given a black-box system, a relational model is first computed

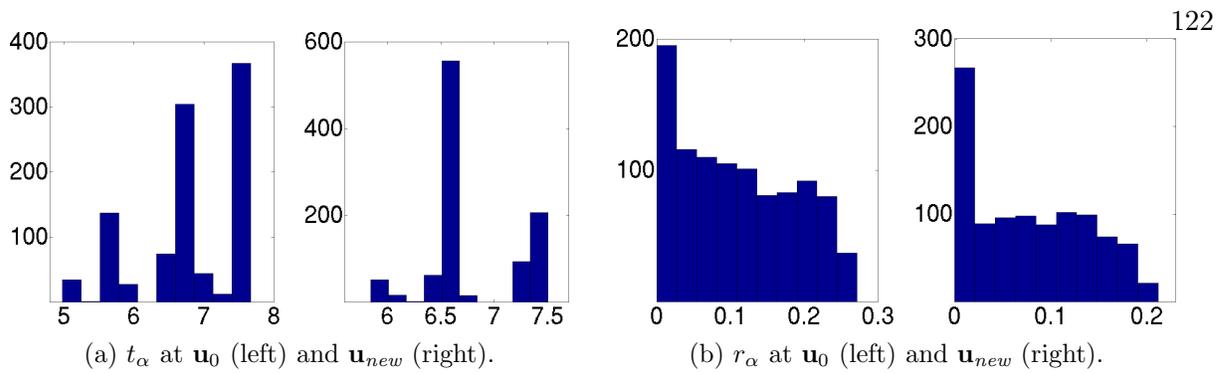


Figure 6.9: Histograms of t_α (left, in seconds) and r_α (right, as percentage) in the aircraft flight control model.

via quantile regression to approximate the marginalized response surface. Then the relational model is generalized into a statistically sound model at the nominal design point, which is used to verify the specifications. If the nominal point is shown to be unsafe, we search for a new design point. Several benchmark examples are also presented to demonstrate the capability of SSMI-opt.

Chapter 7

Conclusion

7.1 Summary of this Thesis

As the complexity of practical systems grows, conventional analysis approaches, including Monte-Carlo simulation and symbolic reasoning, gradually become inefficient. Recognizing this, this thesis proposes a set of techniques that can be applied to the statistical reasoning of these systems. These techniques target on the verification and optimization problems of black-box systems, for which only a computable function explaining the input-output relation is retained and the knowledge of the internal workings is not required. This final chapter summarizes the proposed techniques and point out a few future directions.

Chapter 4 introduces a statistical verification technique, statistically sound model inference (SSMI). The idea of SSMI is to build basis functional models between the stochastic parameters in a black-box system and the responses of the system, and generalize these models to achieve statistical soundness. The statistically sound models are shown to over-approximates the response surface in a large proportion of the stochastic parameter space. From a verification point of view, these models can be used to under-approximate, in a statistical sense, the safe regions of the stochastic parameter space, i.e., stochastic parameter values that satisfy the specifications of the system. Also, the yields computed with respect to these models serve as the lower bound of the true yields.

Chapter 5 presents a sparse approximation algorithm that aims to extend the ability of SSMI to handle systems with many stochastic parameters. The algorithm combines generalized polynomial chaos (gPC) and LASSO into a stepwise procedure. At each step, gPC forms a filtering

stage, which removes some unimportant basis functions, and benefiting from this stage, LASSO solves a smaller problem. The algorithm is also featured with early termination. It terminates as soon as the polynomial approximation has enough accuracy. This often lead to an approximation of a degree lower than the target degree.

Chapter 6 discusses a statistically sound optimization technique, SSMI-opt. It is applied to tune the design parameters in a black-box system when the system violates the specifications in the face of stochastic parameter variations. SSMI-opt relies on quantile regression and a modified generalization procedure of SSMI. It constructs statistically sound relational models in terms of the design parameters, which “marginalize” the effects of the stochastic parameters. An iterative optimization algorithm is developed to find candidate design points from these models and verify that these points are indeed safe in the actual system. The outcome of SSMI-opt is a new design point, or a conclusion that such a point may not be available due to stringent specifications.

7.2 Future Work

7.2.1 Statistically Sound Model Inference

A challenge arises in the generalization procedure of SSMI. Recall that generalization transforms a basis functional model into a statistically sound relational model through the derivation of a tolerance interval. The derivation is “strict” in the sense that all the observed data points are covered by the generalized model. This behavior is not always desirable. Consider the case that the simulation data contain outliers. The current generalization procedure is not able to “skip” those observations and can result in an excessively large tolerance interval. Hence, one future direction is to develop a more flexible generalization technique that filters out the outliers without sacrificing the provided statistical guarantee. It is also interesting to explore relational models with asymmetric lower and upper functionals.

7.2.2 Sparse Approximation

In this thesis, the proposed sparse approximation algorithm relies on two tunable parameters, η which controls how “aggressive” the algorithm is in dropping unimportant basis functions, and ϵ which affects the termination of the algorithm. Currently, these parameters need to be tuned manually. A bad choice can lead to approximations with poor accuracy. In the future, it is worthy investigating approaches to automatically determine η and ϵ .

7.2.3 Statistically Sound Optimization

SSMI-opt uses affine functions to construct the relational models. The primary reason for this is that affine functions lead to linear constraints in the iterative optimization algorithm, which can be easily solved by linear programming. A valuable topic for future research is to employ higher-degree polynomials into the relational models and as a consequence, how to solve for the optimization problem induced by the non-affine constraints.

7.3 Combining Statistical and Symbolic Techniques

In the final part of this thesis, let us consider again the strengths and weaknesses of symbolic and statistical techniques. Although existing symbolic techniques have many drawbacks, including the requirement for detailed system models and the poor scalability when dealing with continuous/hybrid systems, the dominant advantage over any other techniques is that they provide formally guaranteed conclusions. Hence, it is unlikely that symbolic techniques will be useless in the verification and optimization of large systems. The key is, however, to **find situations that are suitable for these techniques**. On the other hand, statistical techniques are fast but less accurate. It should be realized that even a conclusion from some statistical technique is shown to be true with a probability of 99.9%, it still has a chance to be wrong. Such a chance could be detrimental in certain situations.

At this point, it is worth taking some time to think about what the future direction of

verification and optimization will be. A question would be: is it possible to combine symbolic and statistical techniques? With such a combination, it is desirable to have a technique that is fast, scalable and can provide formal guarantees at least to some extent. A possible strategy may involve using statistical techniques to construct abstract models and then applying symbolic techniques to reason the abstracted behaviors. Alternatively, one may also run statistical techniques when the behavior of a system is “regular”, and switch to symbolic reasoning when it becomes “questionable”. There are many interesting approaches to be explored. It is hoped that the techniques discussed in this thesis can open a new paradigm for statistical verification and optimization, and contribute to the future research in this area.

Bibliography

- [1] LTSpice: A high performance SPICE simulator, schematic capture and waveform viewer. URL <http://www.linear.com/designtools/software/>.
- [2] Seyed Nematollah Ahmadyan, Jayanand Asok Kumar, and Shobha Vasudevan. Goal-oriented stimulus generation for analog circuits. In DAC, pages 1018–1023, 2012.
- [3] Matthias Althoff, Soner Yaldiz, Akshay Rajhans, Xin Li, Bruce H. Krogh, and Larry T. Pileggi. Formal verification of phase-locked loops using reachability analysis and continuization. In ICCAD, pages 659–666, 2011.
- [4] Ivo Babuska, Raúl Tempone, and Georgios E Zouraris. Galerkin finite element approximations of stochastic elliptic partial differential equations. SIAM Journal on Numerical Analysis, 42(2):800–825, 2004.
- [5] Ivo Babuška, Fabio Nobile, and Raul Tempone. A stochastic collocation method for elliptic partial differential equations with random input data. SIAM review, 52(2):317–355, 2010.
- [6] Satish Batchu. Automatic Extraction of Behavioral Models from Simulations of Analog/Mixed-Signal (AMS) Circuits. PhD thesis, University of Utah, Salt Lake City, Utah, 2011.
- [7] James O. Berger. Statistical decision theory and Bayesian analysis. Springer series in statistics. Springer, New York, NY, 2nd edition, 1985.
- [8] James O. Berger. Could fisher, jeffreys and neyman have agreed on testing? Statistical Science, 18(1):1–32, 2003.
- [9] James O. Berger. The case for objective bayesian analysis. Bayesian Analysis, 1(3):385–402, 2006.
- [10] James O Berger, Lawrence D Brown, and Robert L Wolpert. A unified conditional frequentist and bayesian test for fixed and sequential simple hypothesis testing. The Annals of Statistics, pages 1787–1807, 1994.
- [11] F. De Bernardinis, M. I. Jordan, and A. Sangiovanni-Vincentelli. Support vector machines for analog circuit performance representation. In DAC, pages 964–969, 2003.
- [12] Armin Biere, Alessandro Cimatti, Edmund Clarke, and Yunshan Zhu. Symbolic model checking without BDDs. Springer, 1999.

- [13] Armin Biere, Alessandro Cimatti, Edmund M Clarke, Ofer Strichman, and Yunshan Zhu. Bounded model checking. Advances in Computers, 58:117–148, 2003.
- [14] Alfred M Bruckstein, David L Donoho, and Michael Elad. From sparse solutions of systems of equations to sparse modeling of signals and images. SIAM review, 51(1):34–81, 2009.
- [15] Emmanuel Candes and Justin Romberg. Sparsity and incoherence in compressive sampling. Inverse problems, 23(3):969, 2007.
- [16] Emmanuel Candes, Justin Romberg, and Terence Tao. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. Information Theory, IEEE Transactions on, 52(2):489–509, 2006.
- [17] George Casella and Roger L. Berger. Statistical Inference. Cengage Learning, 2nd edition, 2001.
- [18] Scott Shaobing Chen, David L Donoho, and Michael A Saunders. Atomic decomposition by basis pursuit. SIAM journal on scientific computing, 20(1):33–61, 1998.
- [19] Edmund Clarke, Alexandre Donzé, and Axel Legay. On simulation-based probabilistic model checking of mixed-analog circuits. Formal Methods in System Design, 36(2):97–113, 2010.
- [20] Albert Cohen, Wolfgang Dahmen, and Ronald DeVore. Compressed sensing and best -term approximation. Journal of the American mathematical society, 22(1):211–231, 2009.
- [21] C. Dalla Man, R.A. Rizza, and C. Cobelli. Meal simulation model of the glucose-insulin system. Biomedical Engineering, IEEE Transactions on, 54(10):1740–1749, 2007.
- [22] Thao Dang and Tarik Nahhal. Coverage-guided test generation for continuous and hybrid systems. Formal Methods in System Design, 34(2):183–213, 2009.
- [23] Manas K Deb, Ivo M Babuška, and J Tinsley Oden. Solution of stochastic partial differential equations using galerkin finite element techniques. Computer Methods in Applied Mechanics and Engineering, 190(48):6359–6372, 2001.
- [24] David L Donoho. Compressed sensing. Information Theory, IEEE Transactions on, 52(4):1289–1306, 2006.
- [25] Alireza Doostan and Gianluca Iaccarino. A least-squares approximation of partial differential equations with high-dimensional random inputs. J. Comput. Phys., 228(12):4332–4345, 2009.
- [26] Alireza Doostan and Houman Owhadi. A non-adapted sparse approximation of pdes with stochastic inputs. Journal of Computational Physics, 230(8):3015–3034, 2011.
- [27] Bradley Efron, Trevor Hastie, Iain Johnstone, Robert Tibshirani, et al. Least angle regression. The Annals of statistics, 32(2):407–499, 2004.
- [28] Christian Ellen, Sebastian Gerwinn, and Martin Fränzle. Statistical model checking for stochastic hybrid systems involving nondeterminism over continuous domains. Journal of Software Tools for Technology Transfer, 2014. To Appear in Special issue on Statistical Model Checking.

- [29] Georgios E Fainekos and George J Pappas. Robustness of temporal logic specifications for continuous-time signals. Theoretical Computer Science, 410(42):4262–4291, 2009.
- [30] Zhuo Feng and Peng Li. Performance-oriented statistical parameter reduction of parameterized systems via reduced rank regression. In ICCAD, pages 868–875, 2006.
- [31] G.S. Fishman. A First Course In Monte Carlo. Duxbury advanced series. Duxbury, Thomson Brooks/Cole, 2006.
- [32] Goran Frehse, Bruce H. Krogh, and Rob A. Rutenbar. Verifying analog oscillator circuits using forward/backward abstraction refinement. In DATE, pages 257–262, 2006.
- [33] Thomas Gerstner and Michael Griebel. Numerical integration using sparse grids. Numerical algorithms, 18(3-4):209–232, 1998.
- [34] Gene H Golub and Charles F Van Loan. Matrix computations, volume 3. JHU Press, 2012.
- [35] Gene H Golub, Michael Heath, and Grace Wahba. Generalized cross-validation as a method for choosing a good ridge parameter. Technometrics, 21(2):215–223, 1979.
- [36] Douglas M Hawkins. The problem of overfitting. Journal of chemical information and computer sciences, 44(1):1–12, 2004.
- [37] Lars Hedrich and Erich Barke. A formal approach to nonlinear analog circuit verification. In ICCAD, pages 123–127, 1995.
- [38] D. Henriques, J. Martins, P. Zuliani, A. Platzer, and E.M. Clarke. Statistical model checking for markov decision processes. In QEST’12, 2012.
- [39] Thomas Héroult, Richard Lassaigne, Frédéric Magniette, and Sylvain Peyronnet. Approximate probabilistic model checking. In Verification, Model Checking, and Abstract Interpretation, pages 73–84, 2004.
- [40] Harold Jeffreys. Some tests of significance, treated by the theory of probability. Proc. Cambridge Philosophy Society, (31):203–222, 1935.
- [41] Harold Jeffreys. Theory of Probability. Oxford Univ. Press, Oxford, 3rd edition, 1961.
- [42] Sumit Kumar Jha. Model Validation and Discovery for Complex Stochastic Systems. PhD thesis, Computer Science Department, Carnegie Mellon University, Pittsburgh, Pennsylvania, 2010.
- [43] Sumit Kumar Jha, Edmund M. Clarke, Christopher James Langmead, Axel Legay, André Platzer, and Paolo Zuliani. A Bayesian approach to model checking biological systems. In CMSB, pages 218–234, 2009.
- [44] Sumit Kumar Jha, Raj Gautam Dutta, Christopher J Langmead, Susmit Jha, and Emily Sassano. Synthesis of insulin pump controllers from safety specifications using bayesian model validation. International journal of bioinformatics research and applications, 8(3):263–285, 2012.

- [45] Bozena Kaminska, Karim Arabi, I Bell, P Goteti, JL Huertas, B Kim, A Rueda, and M Soma. Analog and mixed-signal benchmark circuits-first release. In Proceedings of International Test Conference, pages 183–190, 1997.
- [46] Hans W Kamp. Tense Logic and the Theory of Linear Order. PhD thesis, UCLA, Los Angeles, CA, USA, 1968.
- [47] Robert E. Kass and Adrian E. Raftery. Bayes factors. J. Amer. Stat. Assoc., 90(430):774–795, 1995.
- [48] David Kleinbaum, Lawrence Kupper, Azhar Nizam, and Eli Rosenberg. Applied regression analysis and other multivariable methods. Cengage Learning, 2013.
- [49] Roelof Koekoek, Tom H Koornwinder, Peter A Lesky, and René F Swarttouw. Hypergeometric orthogonal polynomials and their q-analogues. Springer, 2010.
- [50] Roger Koenker. Quantile regression. Number 38. Cambridge university press, 2005.
- [51] Dhanashree Kulkarni. Improved model generation and property specification for analog/mixed-signal circuits. PhD thesis, University of Utah, Salt Lake City, Utah, 2013.
- [52] Robert P. Kurshan and Kenneth L. McMillan. Analysis of digital circuits through symbolic reduction. IEEE Trans. on CAD of Integrated Circuits and Systems, 10(11):1356–1371, 1991.
- [53] Constantino M Lagoa, Fabrizio Dabbene, and Roberto Tempo. Hard bounds on the probability of performance with application to circuit analysis. Circuits and Systems, IEEE Transactions on, 55(10):3178–3187, 2008.
- [54] Xin Li. Finding deterministic solution from underdetermined equation: large-scale performance variability modeling of analog/RF circuits. IEEE Trans. CAD, 29(11):1661–1668, 2010.
- [55] Xin Li, Jiayong Le, Lawrence T Pileggi, and Andrzej Strojwas. Projection-based performance modeling for inter/intra-die variations. In Proceedings of the 2005 IEEE/ACM International conference on Computer-aided design, pages 721–727, 2005.
- [56] Scott Little, David Walter, Kevin Jones, and Chris Myers. Analog/mixed-signal circuit verification using models generated from simulation traces. In Automated Technology for Verification and Analysis, pages 114–128. 2007.
- [57] Scott Little, David Walter, Chris J. Myers, Robert A. Thacker, Satish Batchu, and Tomohiro Yoneda. Verification of analog/mixed-signal circuits using labeled hybrid Petri nets. IEEE Trans. on CAD of Integrated Circuits and Systems, 30(4):617–630, 2011.
- [58] Stéphane G Mallat and Zhifeng Zhang. Matching pursuits with time-frequency dictionaries. Signal Processing, IEEE Transactions on, 41(12):3397–3415, 1993.
- [59] Lionel Mathelin and M Yousuff Hussaini. A stochastic collocation algorithm for uncertainty analysis. Citeseer, 2003.
- [60] Hermann G Matthies and Andreas Keese. Galerkin methods for linear and nonlinear elliptic stochastic partial differential equations. Computer Methods in Applied Mechanics and Engineering, 194(12):1295–1331, 2005.

- [61] Kenneth L McMillan. Symbolic model checking. Springer, 1993.
- [62] AJ MCNEIL. Estimating the tails of loss severity distributions using extreme value theory. ASTIN Bulletin, 27(1):117–138, 1997.
- [63] Jacob Millman and Christos Constantine Halkias. Integrated Electronics. McGraw-Hill electrical and electronic engineering series. Tata McGraw-Hill Publishing Company, 1972.
- [64] Alex Mitev, Michael Marefat, Dongsheng Ma, and Janet Meiling Wang. Principle Hessian direction-based parameter reduction for interconnect networks with process variation. IEEE Trans. VLSI Systems, 18(9):1337–1347, 2010.
- [65] Chris J Myers, Nathan Barker, Kevin Jones, Hiroyuki Kuwahara, Curtis Madsen, and Nam-Phuong D Nguyen. ibiosim: a tool for the analysis and design of genetic circuits. Bioinformatics, 25(21):2848–2849, 2009.
- [66] J. Neyman and E. S. Pearson. On the problem of the most efficient tests of statistical hypotheses. Royal Society of London Philosophical Transactions Series A, 231:289–337, 1933.
- [67] Fabio Nobile, Raúl Tempone, and Clayton G Webster. A sparse grid stochastic collocation method for partial differential equations with random input data. SIAM Journal on Numerical Analysis, 46(5):2309–2345, 2008.
- [68] SuheendraK. Palaniappan, BenjaminM. Gyori, Bing Liu, David Hsu, and P.S. Thiagarajan. Statistical model checking based calibration and analysis of bio-pathway models. In CMSB, pages 120–134, 2013.
- [69] Amir Pnueli. The temporal logic of programs. In Foundations of Computer Science, 18th Annual Symposium on, pages 46–57, 1977.
- [70] Murray Rosenblatt. Remarks on a multivariate transformation. The annals of mathematical statistics, pages 470–472, 1952.
- [71] Reuven Y Rubinstein and Dirk P Kroese. Simulation and the Monte Carlo method, volume 707. John Wiley & Sons, 2011.
- [72] Sethuraman Sankaran and Alison L Marsden. A stochastic collocation method for uncertainty quantification and propagation in cardiovascular simulations. Journal of biomechanical engineering, 133(3):031001, 2011.
- [73] S. Sankaranarayanan, C. Miller, R. Raghunathan, H. Ravanbakhsh, and G. Fainekos. 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, pages 1610–1617, 2012.
- [74] Koushik Sen, Mahesh Viswanathan, and Gul Agha. Statistical model checking of black-box probabilistic systems. In CAV, pages 202–215, 2004.
- [75] Amith Singhee and Rob A. Rutenbar. Beyond low-order statistical response surfaces: latent variable regression for efficient, highly nonlinear fitting. In DAC, pages 256–261, 2007.

- [76] Amith Singhee and Rob A Rutenbar. From finance to flip flops: A study of fast quasi-monte carlo methods from computational finance applied to statistical circuit analysis. In ISQED, pages 685–692, 2007.
- [77] Amith Singhee and Rob A Rutenbar. Statistical blockade: a novel method for very fast monte carlo simulation of rare circuit events, and its application. In DATE, pages 235–251, 2008.
- [78] Ilya M Sobol. On the distribution of points in a cube and the approximate evaluation of integrals. USSR Computational Mathematics and Mathematical Physics, 7(4):86–112, 1967.
- [79] Kai Strunz and Qianli Su. Stochastic formulation of spice-type electronic circuit simulation with polynomial chaos. ACM Trans. Model. Comput. Simul., 18(4), 2008.
- [80] Robert Tibshirani. Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society. Series B (Methodological), pages 267–288, 1996.
- [81] Andrey Nikolayevich Tikhonov. Numerical methods for the solution of ill-posed problems, volume 328. Springer, 1995.
- [82] Saurabh K. Tiwary, Anubhav Gupta, Joel R. Phillips, Claudio Pinello, and Radu Zlatanovici. First steps towards SAT-based formal analog verification. In ICCAD, pages 1–8, 2009.
- [83] A. Wald. Sequential tests of statistical hypotheses. The Annals of Mathematical Statistics, 16(2):117–186, 1945.
- [84] David Walter, Scott Little, Chris J. Myers, Nicholas Seegmiller, and Tomohiro Yoneda. Verification of analog/mixed-signal circuits using symbolic methods. IEEE Trans. on CAD of Integrated Circuits and Systems, 27(12):2223–2235, 2008.
- [85] Ying-Chih Wang, Anvesh Komuravelli, Paolo Zuliani, and Edmund M. Clarke. Analog circuit verification by statistical model checking. In ASP-DAC, pages 1–6, 2011.
- [86] Bong Wie and Dennis S. Bernstein. A benchmark problem for robust control design. In American Control Conference, pages 961–962, May 1990.
- [87] Norbert Wiener. The homogeneous chaos. American Journal of Mathematics, 60(4):897–936, 1938.
- [88] Dongbin Xiu. Numerical methods for stochastic computations: a spectral method approach. Princeton University Press, 2010.
- [89] Dongbin Xiu and Jan S Hesthaven. High-order collocation methods for differential equations with random inputs. SIAM Journal on Scientific Computing, 27(3):1118–1139, 2005.
- [90] Dongbin Xiu and George Em Karniadakis. The wiener–askey polynomial chaos for stochastic differential equations. SIAM Journal on Scientific Computing, 24(2):619–644, 2002.
- [91] Chao Yan and Mark Greenstreet. Oscillator verification with probability one. In FMCAD, pages 165–172, 2012.
- [92] Leyi Yin, Yue Deng, and Peng Li. Verifying dynamic properties of nonlinear mixed-signal circuits via efficient SMT-based techniques. In ICCAD, pages 436–442, 2012.

- [93] Heebyung Yoon, P Variyam, A Chatterjee, and N Nagi. Hierarchical statistical inference model for specification based testing of analog circuits. In VLSI Test Symposium, pages 145–150, 1998.
- [94] Håkan L. S. Younes. Verification and Planning for Stochastic Processes with Asynchronous Events. PhD thesis, Computer Science Department, Carnegie Mellon University, Pittsburgh, Pennsylvania, 2005.
- [95] Håkan L. S. Younes and Reid G. Simmons. Probabilistic verification of discrete event systems using acceptance sampling. In CAV, pages 223–235, 2002.
- [96] Yan Zhang, Sriram Sankaranarayanan, and Fabio Somenzi. Piecewise linear modeling of nonlinear devices for formal verification of analog circuits. In FMCAD, pages 196–203, 2012.
- [97] Yan Zhang, Sriram Sankaranarayanan, Fabio Somenzi, Xin Chen, and Erika Ábraham. From statistical model checking to statistical model inference: Characterizing the effect of process variations in analog circuits. In ICCAD, 2013.
- [98] Yan Zhang, Sriram Sankaranarayanan, and Fabio Somenzi. Statistically sound verification and optimization for complex systems. In ATVA, 2014. To Appear.
- [99] Yan Zhang, Sriram Sankaranarayanan, Fabio Somenzi, Xin Chen, and Erika Ábraham. Sparse statistical model inference for analog circuits under process variations. In ASP-DAC, pages 449–454, 2014.
- [100] Paolo Zuliani, André Platzer, and Edmund M. Clarke. Bayesian statistical model checking with application to Simulink/Stateflow verification. In HSCC, pages 243–252, 2010.